



# **RENT2GO**

## **CAR RENTAL PROJECT**

# PRESENTATION

# Araç Kiralama Pazar Büyüklüğü ve Trendler

Küresel araç kiralama pazarı, 2023 yılında yaklaşık 86 milyar USD olarak tahmin edilmiştir ve 2028 yılına kadar yıllık ortalama %7 büyümeye ile 120 milyar USD'ye ulaşması beklenmektedir. Elektrikli araçların yaygınlaşması ve çevre bilincinin artmasıyla sürdürülebilir araç kiralama hizmetlerine olan talep artmaktadır. Mobil platformlar üzerinden araç kiralama hizmetlerine olan ilgi giderek artmaktadır. Kullanıcılar, kolay erişim ve hızlı rezervasyon avantajları nedeniyle online platformları tercih etmektedir.

# Hedef Kitle

**Bireysel Kullanıcılar:** Kısa süreli tatilciler, iş seyahati yapanlar, şehir içi ulaşım için araç kiralayanlar.

**Kurumsal Kullanıcılar: Şirketlerin çalışanlarına sunduğu araç kiralama hizmetleri, filo kiralama.**

Turistler: Özellikle havaalanı transferleri ve şehir turu hizmetleri için araç kiralayan yabancı turistler.

# Rekabet Analizi

**Global Oyuncular: Avis, Hertz, Enterprise gibi büyük uluslararası firmalar**

## Yerel Oyuncular: Türkiye'de Garenta, Central Rent a Car gibi güçlü yerel markalar

Dijital Platformlar: Getaround, Turo gibi P2P araç kiralama platformları

# Teknolojik Trendler

Yapay Zeka ve Makine Öğrenimi: Kiralama süreçlerinin otomasyonu, müşteri davranış analizi ve kişiselleştirilmiş tekliflerin sunulması.

Nesnelerin İnterneti (IoT): Araçların uzaktan izlenmesi, bakım ve performans analizi

**Blockchain: Kiralama sözleşmeleri ve ödemelerin güvenli ve şeffaf bir şekilde yönetilmesi**

# BENCHMARKING

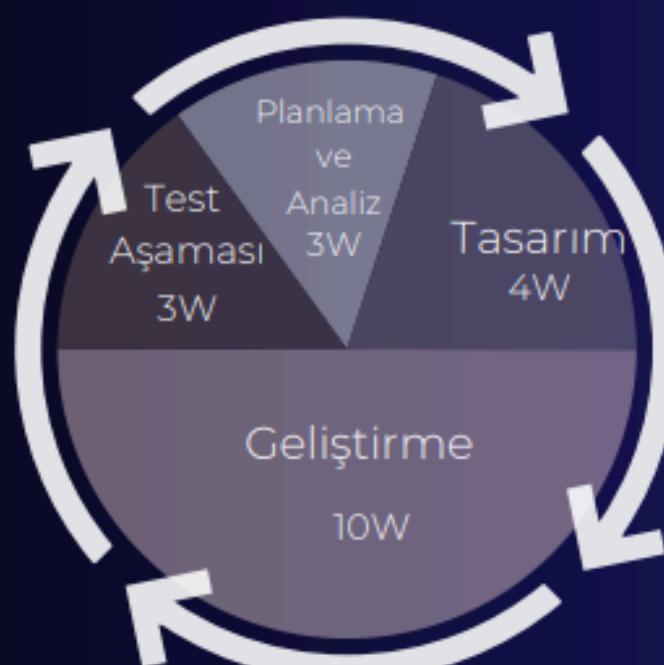
## PROJE AMAÇ VE KAPSAMI

Bu proje, kullanıcıların ihtiyaçlarına uygun araç kiralama hizmetlerini kolay ve hızlı bir şekilde bulmalarını sağlayacak mobil ve web tabanlı bir uygulama geliştirmeyi hedeflemektedir.

## EKİP VE PROJE SÜRESİ



- Proje Yöneticisi
- İş Analisti
- UI/UX Tasarımcı
- Mobil Uygulama Geliştirici
- Frontend Geliştirici
- Backend Geliştirici
- QA Mühendisi
- DevOps Mühendisi
- Müşteri Destek Uzmanı



## KULLANILACAK TEKNOLOJİLER



Kotlin-Swift  
React.js , Vue.js  
HTML5, CSS3  
TypeScript  
Node.js Express.js  
PostgreSQL  
TensorFlow

## MALİYET VE YATIRIM

Geliştirme: \$120,000  
Pazarlama: \$50,000  
Diğer Maliyetler: \$30,000  
Toplam Maliyet: \$200,000

ROI (Yatırım Getiri)  
İlk Yıl Gelir: \$300,000  
2 Yıl İçinde Karlılık: %50  
Yıllık Büyüme: %20



Hayalleriniz  
yol haritalarımızdır.



# PROJE KAPSAMI

## Temel Özellikler (Core Features)

Aşağıda listelenen özellikler, uygulamanın ilk sürümünde bulunması gereken temel fonksiyonlardır:

### 1. Kullanıcı Kayıt ve Giriş

- Kullanıcıların hesap oluşturma ve mevcut hesaplarla giriş yapabilme özelliği.

### 2. Araç Arama

- Kullanıcıların araçları arama ve filtreleme seçenekleri.

### 3. Araç Rezervasyonu

- Seçilen araç için rezervasyon yapabilme imkanı.

### 4. Ödeme Sistemleri

- Güvenli ödeme işlemleri ve çeşitli ödeme seçenekleri.

### 5. Kullanıcı Profili Yönetimi

- Kullanıcıların kişisel bilgilerini güncelleyebilmesi ve yönetebilmesi.

### 6. Araç Teslim ve İade Süreci

- Araçların teslim alınması ve iade edilmesi sürecinin yönetimi.

### 7. Müşteri Desteği

- Canlı destek veya sıkça sorulan sorular bölümü.

### 8. Araç Sigortası

- Araç sigorta seçeneklerinin sunulması.

### 9. Bildirimler

- Rezervasyon hatırlatmaları ve promosyon bildirimleri.

### 10. Konum Tabanlı Hizmetler

- GPS ile araçların konumunun takibi.

# PROJE KAPSAMI

## Ek Geliştirmeler (Additional Features)

İlk sürümden sonra uygulanabilecek ek geliştirmeler:

- Çoklu Dil Desteği ve Çoklu Para Birimi Desteği
  - Uygulamanın birden fazla dili ve para birimini desteklemesi.
- Kişiselleştirilmiş Araç Önerileri
  - Kullanıcı deneyimine göre araçların kişilere önerilmesi
- İleri Tarihli Rezervasyon
  - Gelecekteki tarihler için rezervasyon yapabilme.
- 7/24 Chatbot
  - 7/24 müşteri desteği sağlanması.
- Karbon Ayak İzi
  - Araçların doğada bıraktığı karbon izini hesaplayan tool.

# PROJE KAPSAMI

Önceliklendirme

## Yüksek Öncelikli Özellikler:

- Kullanıcı Kayıt ve Giriş
- Araç Arama
- Araç Rezervasyonu
- Ödeme Sistemleri

## Orta Öncelikli Özellikler:

- Kullanıcı Profili Yönetimi
- Araç Teslim ve İade Süreci
- Müşteri Desteği
- Araç Sigortası

## Düşük Öncelikli Özellikler:

- Bildirimler
- Konum Tabanlı Hizmetler

## Ek Geliştirmeler (İlerleyen Süreçlerde):

- Çoklu Dil Desteği ve Para Birimi Desteği
- Kişiselleştirilmiş Araç Önerileri
- İleri Tarihli Rezervasyon
- 7/24 Chatbot Desteği
- Karbon Ayak İzi

PROJE KAPSAMI için detaylı dosyamızı Rent2Go  
klasörümüzde KapsamDökümanı.pdf dosyasında bulabilirsiniz

# BACKLOG MANAGEMENT

Belirlenen ve önceliklendirilen özellikleri yapacağımız için aşağıda 3 MVP sunulmuştur, bu MVP'ler için hazırlanan backlog items ve onlara bağlı tasks listesini Ren2Go klasöründeki BacklogItems.pdf dosyasında bulabilirsiniz.



01

MVP 1: Kullanıcı Kayıt, Giriş ve Araç Arama

02

MVP 2: Araç Rezervasyonu ve Ödeme Sistemleri

03

MVP 3: Kullanıcı Profili ve Araç Teslim/İade Süreci

BACKLOG MANAGEMENT için detaylı dosyamızı Rent2Go klosörümüzde BacklogItems.pdf dosyasında bulabilirsiniz

# TIME MANAGEMENT



## JIRA Üzerinde Projemizin Yönetimi ve Görev Takibi

### Product Backlog

Tasks ▾

Entegre et Otomatikleştir / 1 Davet et / 1 ...

All Sprints Yeni görev Aramal... Kişi Filtre Sırala Gizle / 2 Grupla: / 1 ...

Backlog

	Görev	Owner	Status	+
<input type="checkbox"/>	Kullanıcı kayıt formu ve form doğrulama kuralları		Done	
<input type="checkbox"/>	Kimlik doğrulama API'lerinin entegrasyonu ve doğrulama işlemleri		Done	
<input type="checkbox"/>	Giriş formu, kimlik doğrulama ve oturum kapatma fonksiyonları		In Progress	
<input type="checkbox"/>	Ödeme bilgileri formu ve ödeme gateway entegrasyonu		In Progress	
<input type="checkbox"/>	Kullanıcı arayüzü tasarımı ve hatalı giriş bildirimleri		In Progress	
<input type="checkbox"/>	Şifre sıfırlama işlemleri		In Progress	
<input type="checkbox"/>	Dil dosyaları, çeviri sistemi ve kullanıcı dil menüsü		In Progress	
<input type="checkbox"/>	Para birimi API entegrasyonu ve profil tercihlerinin saklanması		In Progress	
<input type="checkbox"/>	Karbon ayak izi hesaplama ve görsel sunum		Windows'u İncelemeye Başladım	
<input type="checkbox"/>	Testler ve ayrıntılı dokümantasyon		In Progress	<b>Yardım</b>

# SPRINTS (EYLÜL 2024-HAZİRAN 2025)

Sprint 1- Proje Başlangıcı ve Planlama					1 - 30 Eyl	Yıpranmışlık	Başlat	...
Görev	Owner	Pozisyonu	Status	+ + görev ekle				
<input type="checkbox"/> Proje gereksinimlerinin toplanması ve Jira'da backlog itemlarının oluşturulması.		Business Analyst	Done					
<input type="checkbox"/> Proje yapısının belirlenmesi ve teknoloji seçimi		Software Developer	Done					
<input type="checkbox"/> Temel UI/UX tasarımlarının başlangıcı		UX/UI Engineer	Done					
<input type="checkbox"/> Başlangıç test stratejisi oluşturulması		Test Analyst	Done					
+ görev ekle								
Sprint 2- UI/UX Tasarımı					1 - 31 Eki	Yıpranmışlık	Başlat	...
Görev	Owner	Pozisyonu	Status	+ + görev ekle				
<input type="checkbox"/> Kimlik doğrulama ve ödeme sistemleri için testlerin		Test Analyst	Done					
<input type="checkbox"/> Kimlik doğrulama ve ödeme sistemi gereksinimlerinin doğrulanması		Business Analyst	Done					
+ görev ekle								
Sprint 2- UI/UX Tasarımı					1 - 31 Eki	Yıpranmışlık	Başlat	...
Görev	Owner	Pozisyonu	Status	+ + görev ekle				

<input type="checkbox"/>	Gelişmiş özelliklerin (gerçek zamanlı araç takibi, kişiselleştirilmiş öneriler) geliştirilmesi	<a href="#">+</a>		Software Developer	Done	
<input type="checkbox"/>	Gelişmiş özellikler için entegrasyon testlerinin yapılması	<a href="#">+</a>		Test Analyst	Done	
<input type="checkbox"/>	+ görev ekle					

#### Sprint 4- Temel Özelliklerin Geliştirilmesi

1 - 31 Ara

Yıpranmışlık

Başlat

...

Görev	Owner	Pozisyonu	Status	
<input type="checkbox"/> Kullanıcı profili, dil ve para birimi desteği gibi temel özelliklerin geliştirilmesi		Software Developer	Done	
<input type="checkbox"/> Gelişmiş özelliklerin (gerçek zamanlı araç takibi, kişiselleştirilmiş öneriler) geliştirilmesi		Software Developer	Done	
<input type="checkbox"/> Gelişmiş özellikler için entegrasyon testlerinin yapılması		Test Analyst	Done	
<input type="checkbox"/> + görev ekle				

#### Sprint 4- Temel Özelliklerin Geliştirilmesi

1 - 31 Ara

Yıpranmışlık

Başlat

...

Görev	Owner	Pozisyonu	Status	
<input type="checkbox"/> Kullanıcı profili, dil ve para birimi desteği gibi temel özelliklerin geliştirilmesi		Software Developer	Done	
<input type="checkbox"/> Güvenlik ve performans testlerinin yapılması		Test Analyst	Done	
<input type="checkbox"/> Test sonuçlarına göre gerekli düzeltmeleri yapmak		Software Developer	Done	
<input type="checkbox"/> + görev ekle				

#### Sprint 6- Sistem Entegrasyonu ve Testler

Şub 1, '25 - Şub 28, '25

Yıpranmışlık

Başlat

...

Görev	Owner	Pozisyonu	Status	
<input type="checkbox"/> + görev ekle		Software Developer	Done	

<input type="checkbox"/>	Güvenlik ve performans testlerinin yapılması		Test Analyst	Done
<input type="checkbox"/>	Test sonuçlarına göre gerekli düzeltmeleri yapmak		Software Developer	Done
<input type="checkbox"/>	+ görev ekle			<div style="width: 100%; background-color: #2e7131; height: 10px;"></div>

▼ Sprint 6- Sistem Entegrasyonu ve Testler

		Şub 1, '25 - Şub 28, '25			Yıpranmışlık	Başlat	...
	Görev	Owner	Pozisyonu	Status			
<input type="checkbox"/>	Canlı izleme ve ilk geri bildirimlerin toplanması		Test Analyst	Done			
<input type="checkbox"/>	Proje kapanış raporunun hazırlanması ve müşteriye sunulması		Business Analyst	Done			
<input type="checkbox"/>	+ görev ekle			<div style="width: 100%; background-color: #2e7131; height: 10px;"></div>			

▼ Sprint 8- Son Rötuşlar ve Yayın Öncesi Hazırlıklar

		Nis 1, '25 - Nis 30, '25			Yıpranmışlık	Başlat	...
	Görev	Owner	Pozisyonu	Status			
<input type="checkbox"/>	Tüm geliştirmelerin son kontrollerini yapma		Software Developer	Done			
<input type="checkbox"/>	Canlı izleme ve ilk geri bildirimlerin toplanması		Test Analyst	Done			
<input type="checkbox"/>	Proje kapanış raporunun hazırlanması ve müşteriye sunulması		Business Analyst	Done			
<input type="checkbox"/>	+ görev ekle			<div style="width: 100%; background-color: #2e7131; height: 10px;"></div>			

▼ Sprint 8- Son Rötuşlar ve Yayın Öncesi Hazırlıklar

		Nis 1, '25 - Nis 30, '25			Yıpranmışlık	Başlat	...
	Görev	Owner	Pozisyonu	Status			
<input type="checkbox"/>	Canlı izleme ve ilk geri bildirimlerin toplanması		Test Analyst	Done			
<input type="checkbox"/>	Proje kapanış raporunun hazırlanması ve müşteriye sunulması		Business Analyst	Done			
<input type="checkbox"/>	+ görev ekle			<div style="width: 100%; background-color: #2e7131; height: 10px;"></div>			

▼ Sprint 8- Son Rötuşlar ve Yayın Öncesi Hazırlıklar

		Nis 1, '25 - Nis 30, '25			Yıpranmışlık	Başlat	...
	Görev	Owner	Pozisyonu	Status			
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							

TIME MANAGEMENT için detaylı dosyamızı Rent2Go klasörümüzde TimeManagementJira.pdf dosyasında bulabilirsiniz

# REQUIREMENTS

İstenilen proje için yapılan ön görüşmelerde ilk sürüm için, ekip arkadaşlarımızın hazırlamış olduğu, Functional Requirements içeren dosyayı, Ren2Go klasöründe *FunctionalRequirements.pdf* içerisinde, Nonfunctional Requirements içeren dosyayı ise, Ren2Go klasöründe *NonfunctionalRequirements.pdf* içerisinde bulabilirsiniz.

Proje detaylandırıldıktan sonra hazırlanan 50 requirement Use Case Methodology'e göre, Who-Why-What sorularına cevaplayacak şekilde hazırlanmış, use case tablosu haline getirilmiş ve adım adım tanımlamaları yapılmıştır. Bu caseler aşağıdaki formatta hazırlanmıştır. Detaylı dosyası ise Ren2Go klasöründeki Requirements.pdf dosyası içerisindeindedir.

## Use Case 1: Kullanıcı Kayıt İşlemi

- Who: Kullanıcı
- Why: Uygulamayı kullanabilmek için bir hesap oluşturması gerekiyor.
- What: Kullanıcı, kişisel bilgilerini girerek bir hesap oluşturur.

### Adımlar:

1. Kullanıcı, "Kayıt Ol" butonuna tıklar.
2. Kullanıcı, isim, e-posta ve şifre gibi bilgilerini girer.
3. Kullanıcı, "Kayıt Ol" butonuna tıklar.
4. Sistem, girilen bilgileri doğrular.
5. Sistem, kullanıcının bilgilerini veritabanına kaydeder.
6. Sistem, kayıt işleminin başarılı olduğunu gösterir.

# REQUIREMENTS

Proje detaylarındaki 50 requirements için hazırlanan use case başlıklarını aşağıdaki şekildedir, detaylı dosyası ise Ren2Go klasöründeki Requirements.pdf dosyası içerisindeindedir.

- Use Case 1: Kullanıcı Kayıt İşlemi
- Use Case 2: Kullanıcı Giriş İşlemi
- Use Case 3: Şifre Kurtarma İşlemi
- Use Case 4: Araç Arama
- Use Case 5: Araç Filtreleme
- Use Case 6: Araç Rezervasyonu
- Use Case 7: Ödeme İşlemi
- Use Case 8: Rezervasyon Geçmişini Görüntüleme
- Use Case 9: Araç Teslim Alma
- Use Case 10: Araç İade Etme
- Use Case 11: Kullanıcı Profili Güncelleme
- Use Case 12: Kullanıcı Bildirim Ayarları
- Use Case 13: Kullanıcı Parola Değiştirme
- Use Case 14: Araç İnceleme
- Use Case 15: Favori Araçları Kaydetme
- Use Case 16: Araç Sigortası Satın Alma
- Use Case 17: Araç Sigortası Yönetimi
- Use Case 18: Kullanıcı Yorumları ve Puanlama
- Use Case 19: Canlı Destek Kullanımı
- Use Case 20: Araç Teslim Yerini Değiştirme
- Use Case 21: Araç Kiralama Sözleşmesi Görüntüleme

# REQUIREMENTS

Use Case 22: Kiralama Süresi Uzatma

Use Case 23: Araç Durumu Kontrolü

Use Case 24: Araç Teslimat Takibi

Use Case 25: Araç İade Randevusu

Use Case 26: Kullanıcı Erişim Yetkilerini Yönetme

Use Case 27: Araç Sigortası Güncelleme

Use Case 28: Çoklu Dil Desteği Ayarları

Use Case 29: Çoklu Para Birimi Desteği

Use Case 30: Promosyon ve Kampanya Kullanımı

Use Case 31: Kullanıcı Şikayet ve Geri Bildirim Gönderme

Use Case 32: Kiralama İptali ve Ücret İadesi

Use Case 33: Kullanıcı Geçmişine Dayalı Araç Önerileri

Use Case 34: Yorum ve Değerlendirme Yapma

Use Case 35: Araç Temizlik ve Bakım Talebi

Use Case 36: Farklı Sürücü Ekleyebilme

Use Case 37: Sigorta Poliçesi Satın Alma

Use Case 38: Bildirim Ayarlarını Düzenleme

Use Case 39: Kullanıcının Profil Bilgilerine Göre Araç Önerileri

Use Case 40: Araç Hasar Bildirimi

Use Case 41: Araç Güvenlik Bildirimi

Use Case 42: Müşteri Destek Talebi Oluşturma

Use Case 43: Fatura Görüntüleme ve İndirme

# REQUIREMENTS

Use Case 44: Kullanıcı Profili Güncelleme

Use Case 45: Geçmiş Rezervasyonları Görüntüleme

Use Case 46: Kiralama Ücretleri Hesaplama

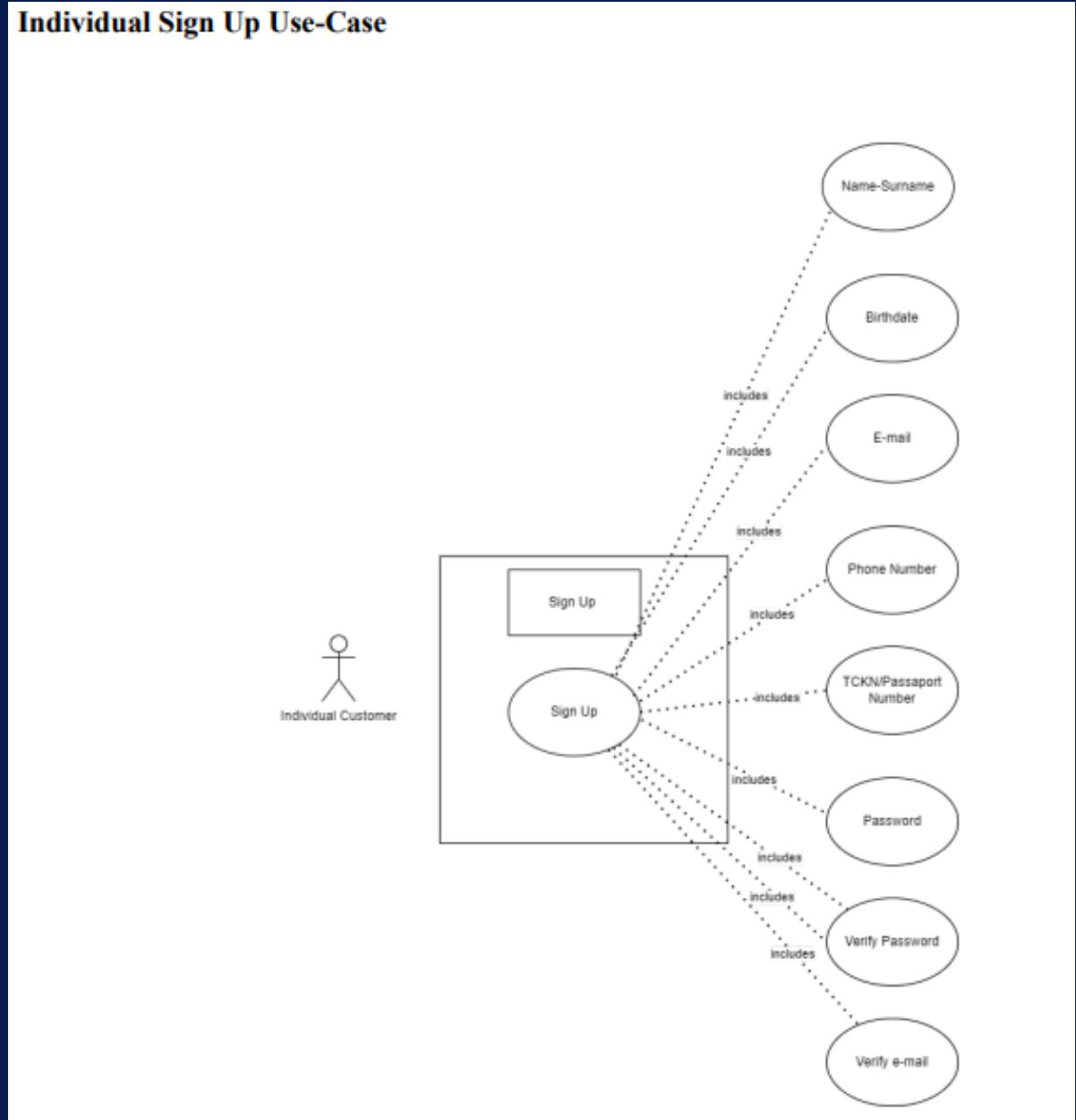
Use Case 47: Kullanıcı Adres Yönetimi

Use Case 48: Mobil Uygulama Güncellemelerini Kontrol Etme

Use Case 49: Lokasyon Tabanlı Araç Önerileri

Use Case 50: Kullanıcı Destek Sohbeti Başlatma

# UML DIAGRAMS



Use Case Diyagramı, bir sistemin kullanıcılar (aktörler) ile nasıl etkileşimde bulunduğu gösteren bir davranışsal UML diyagramıdır. Bu diyagram, sistemin sunduğu işlevlerin genel bir özeti sunar ve sistemin ne yapması gerekiğine odaklanır.

Class Diyagramı, sistemin nesne yönelimli yapısını gösteren bir yapısal UML diyagramıdır. Bu diyagram, sistemdeki sınıfları, bu sınıfların özelliklerini, metodlarını ve aralarındaki ilişkileri gösterir. Sınıf diyagramı, yazılım tasarımının ve mimarisinin en temel bileşenlerinden biridir.

UML diyagramları için detaylı dosyamızı Rent2Go klosörümüzde UMLDiagrams.pdf dosyasında bulabilirsiniz



# DATA MINING

- 1 Verilerden anlamlı bilgiler çıkararak daha iyi kararlar almak.
- 2 Müşteri eğilimlerini anlamak ve stratejileri bu verilere dayandırmak
- 3 Hedef kitleyi anlamak.
- 4 Pazarlama kampanyalarını ve müşteri bağlılığı stratejilerini optimize etmek.



Data Mining detaylı dosyamızı Rent2Go klosörümüzde yer alan database tasarıımı için [SignupAnalysis&DatabaseDesign.pdf](#) dosyamızda; data mining ve sql ile ilgili detayları da [DataMiningSQLUsing.pdf](#) dosyasında bulabilirsiniz

Haziran ayında  
20-22 yaş arası sisteme kayıt olan bireysel  
kullanıcılar

Properties SQL rent2go-fmss/postgres@PostgreSQL 16\*

rent2go-fmss/postgres@PostgreSQL 16 No limit

Query History

```
1 SELECT *
2 FROM individual_customers
3 WHERE EXTRACT(YEAR FROM AGE(birth_date)) BETWEEN 20 AND 22
4 AND EXTRACT(MONTH FROM created_date) = 6;
5
6
```

Data Output Messages Notifications

id [PK] integer created\_date date updated\_date date birth\_date date first\_name character varying(255) last\_name character varying(255) national\_id\_no character varying(255) user\_id integer phone\_number character varying(255)

	id [PK] integer	created_date date	updated_date date	birth_date date	first_name character varying(255)	last_name character varying(255)	national_id_no character varying(255)	user_id integer	phone_number character varying(255)
1	110	2024-06-26	2024-03-31	2004-01-14	Matelda	Smitoune	94730517439	127	451 411 5861

Haziran ayında  
20-22 yaş arası sisteme kayıt olan bireysel  
kullanıcıların sayısı

Properties SQL rent2go-fmss/postgres@PostgreSQL 16\*

rent2go-fmss/postgres@PostgreSQL 16 No limit

Query History

```
1 SELECT COUNT(*)
2 FROM individual_customers
3 WHERE EXTRACT(YEAR FROM AGE(birth_date)) BETWEEN 20 AND 22
4 AND EXTRACT(MONTH FROM created_date) = 6;
5
6
```

Data Output Messages Notifications

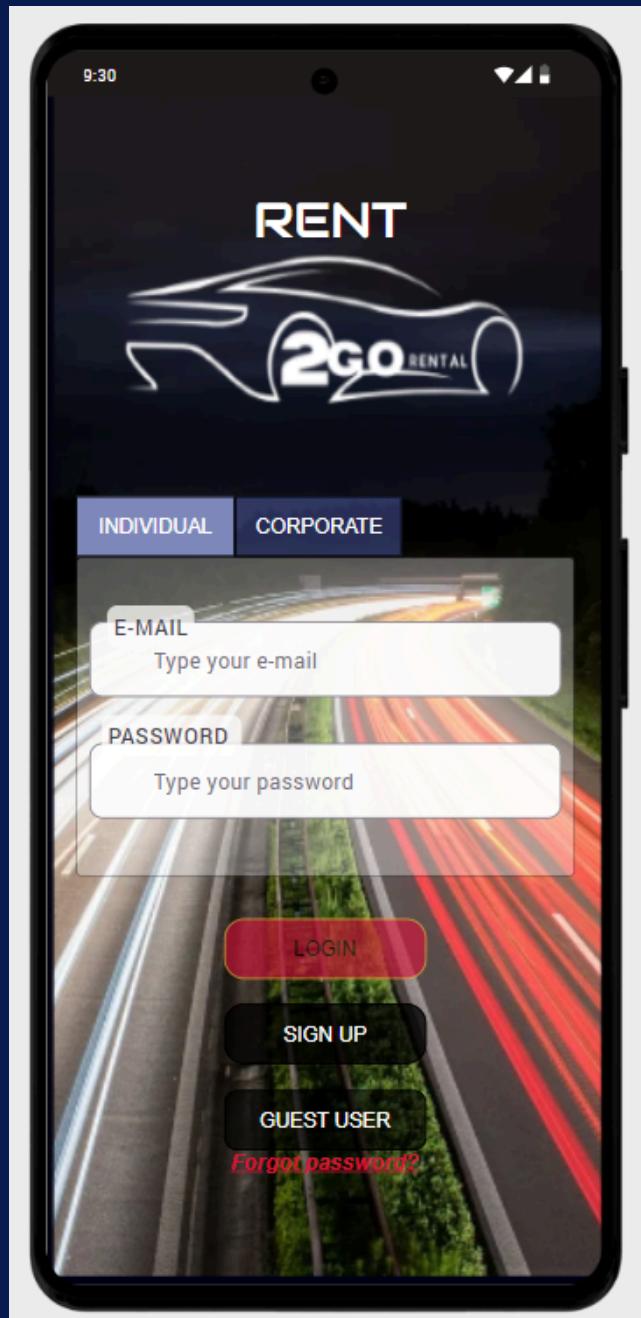
count bigint

	count bigint
1	1

# MOCK UP-UI/UX

## LOG IN EKRANLARI

\*Uygulama ui/ux tasarımı ‘justinmind’ ile tamamlanmıştır.

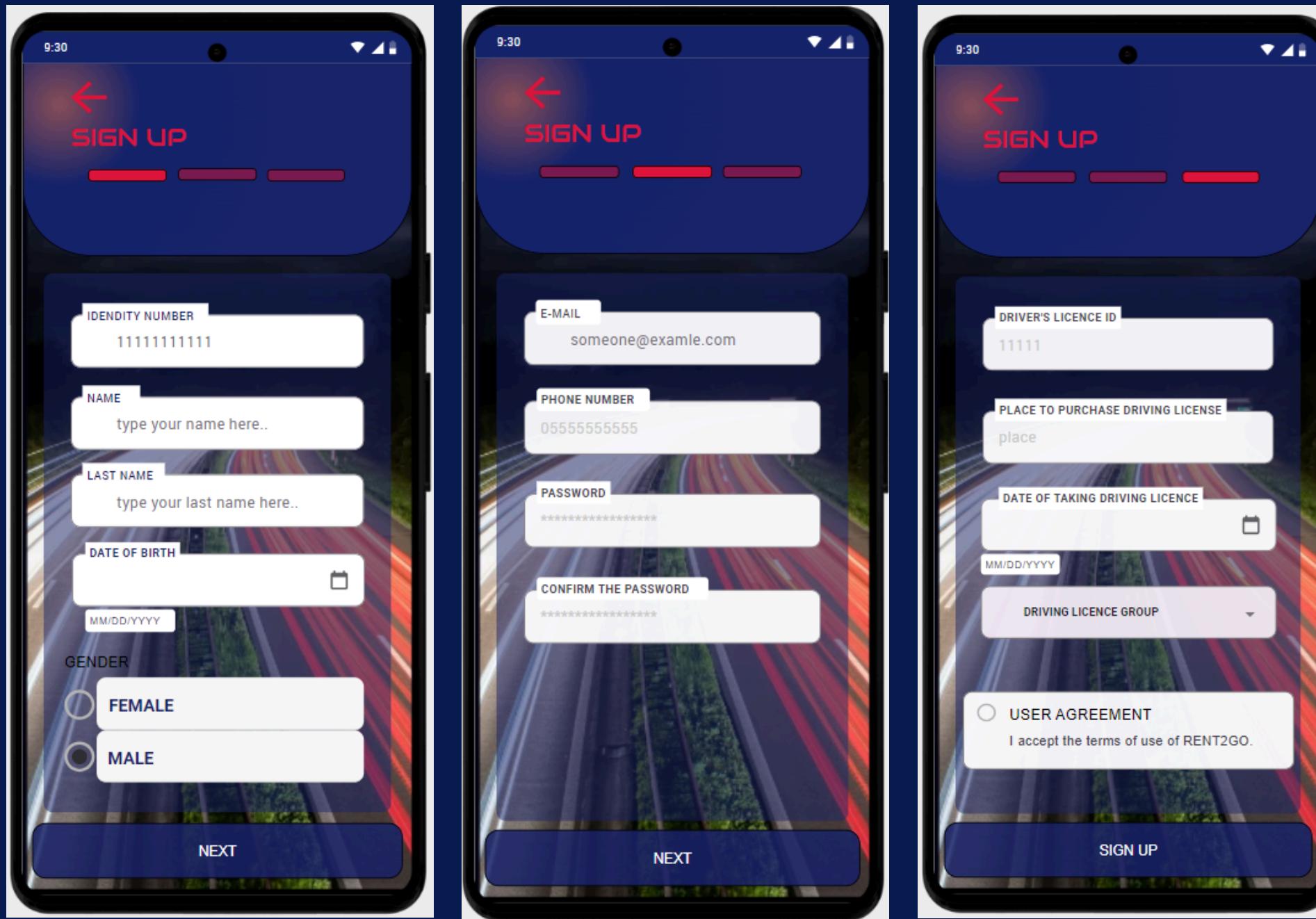


Uygulamanın Login ekranları kurumsal ve bireysel seçenekleri barındırmakta ve bu menülerden kurumsal veya bireysel kullanıcılar için giriş işlemleri sağlanabilecektir. kayıtlı olmayan kullanıcılar için misafir giriş ve kayıt olma seçenekleri yer alacak ve butonlarla geçiş sağlanacaktır.

Mock up UI/UX tasarımımızın kısa bir videosu linkte yer almaktadır : <https://youtu.be/Kd6E59kE0xw>  
Ayrıca proje dosyaları arasında Rent2GoAppDemo.mp4 video dosyasında mobil, Rent2Go.mp4 video dosyasında ise web tasarım videoarını bulabilirsiniz.

# MOCK UP-UI/UX

## SIGN UP EKRANLARI

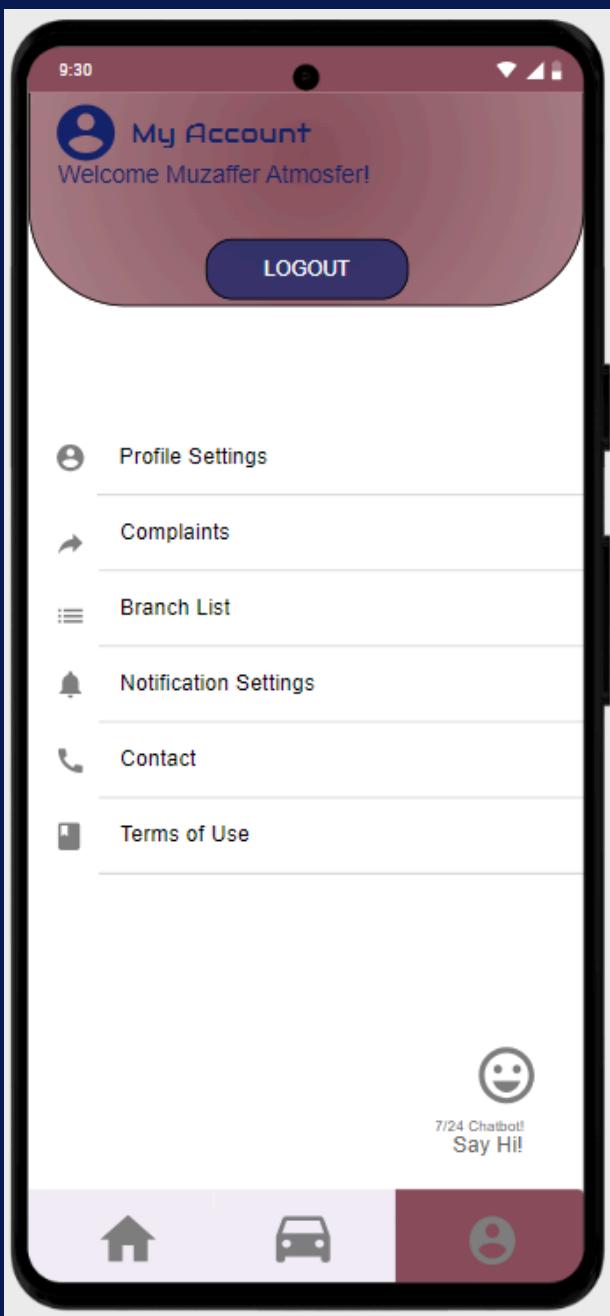
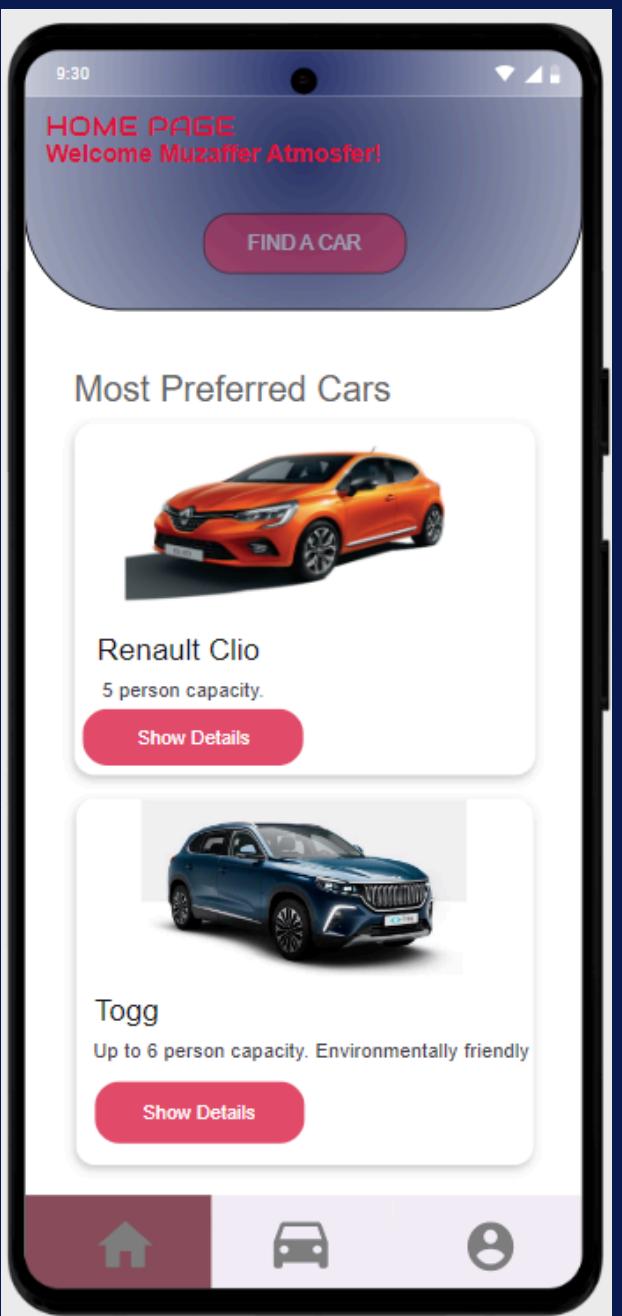


Uygulamanın sign up ekranlarında gerekli bilgiler kullanıcıdan alınacaktır. Bilgiler alınırken gruplarına göre ekranlara ayrılarak göz yormadan kullanıcılarından bilgiler alınmış olacaktır.

Mock up UI/UX tasarımımızın kısa bir videosu linkte yer almaktadır :  
<https://youtu.be/Kd6E59kE0xw>  
Ayrıca proje dosyaları arasında Rent2GoAppDemo.mp4 video dosyasında mobil, Rent2Go.mp4 video dosyasında ise web tasarım videoarını bulabilirsiniz.

# MOCK UP-UI/UX

## UYGULAMA EKRANLARI



Uygulamanın Home Page ve Account sayfa görüntülerini yandaki gibi planlanmıştır. Bottom nav bar sayesinde uygulamda ekranlar arası kolayca sağlanmıştır. Ana ekranandan profil sayfasına geçişin ardından uygulamadan çıkış işlemi kolayca gerçekleştirilebilmektedir. aynı zamanda profil düzlemeleri ve kullanıcı ile ilgili düzlemeler, bilgiler ve benzeri bilgiler burada yer almaktır ve düzenlenlenebilmektedir.

Mock up UI/UX tasarımımızın kısa bir videosu linkte yer almaktadır : <https://youtu.be/Kd6E59kE0xw>  
Ayrıca proje dosyaları arasında Rent2GoAppDemo.mp4 video dosyasında mobil, Rent2Go.mp4 video dosyasında ise web tasarım videoarını bulabilirsiniz.

# MIGRATION & API APPLICATION

## Müşteri Verilerinin Migrate Edilmesi

Müşteri verilerini migrate edilmesi, mevcut bir uygulamada bulunan müşteri verilerinin uygulamamıza taşınması işlemidir. Bu çalışma olan 50bin müşterinin yeni uygulamamıza aktarılması için yapılmıştır. Migrate işlemi sırasında, verilerin yapısı, formatı ve ilişkileri yeni sistemle uyumlu hale getirilir. Aynı zamanda, verilerin doğruluğunu ve bütünlüğünü korumak için çeşitli testler ve dönüşümler yapılır.

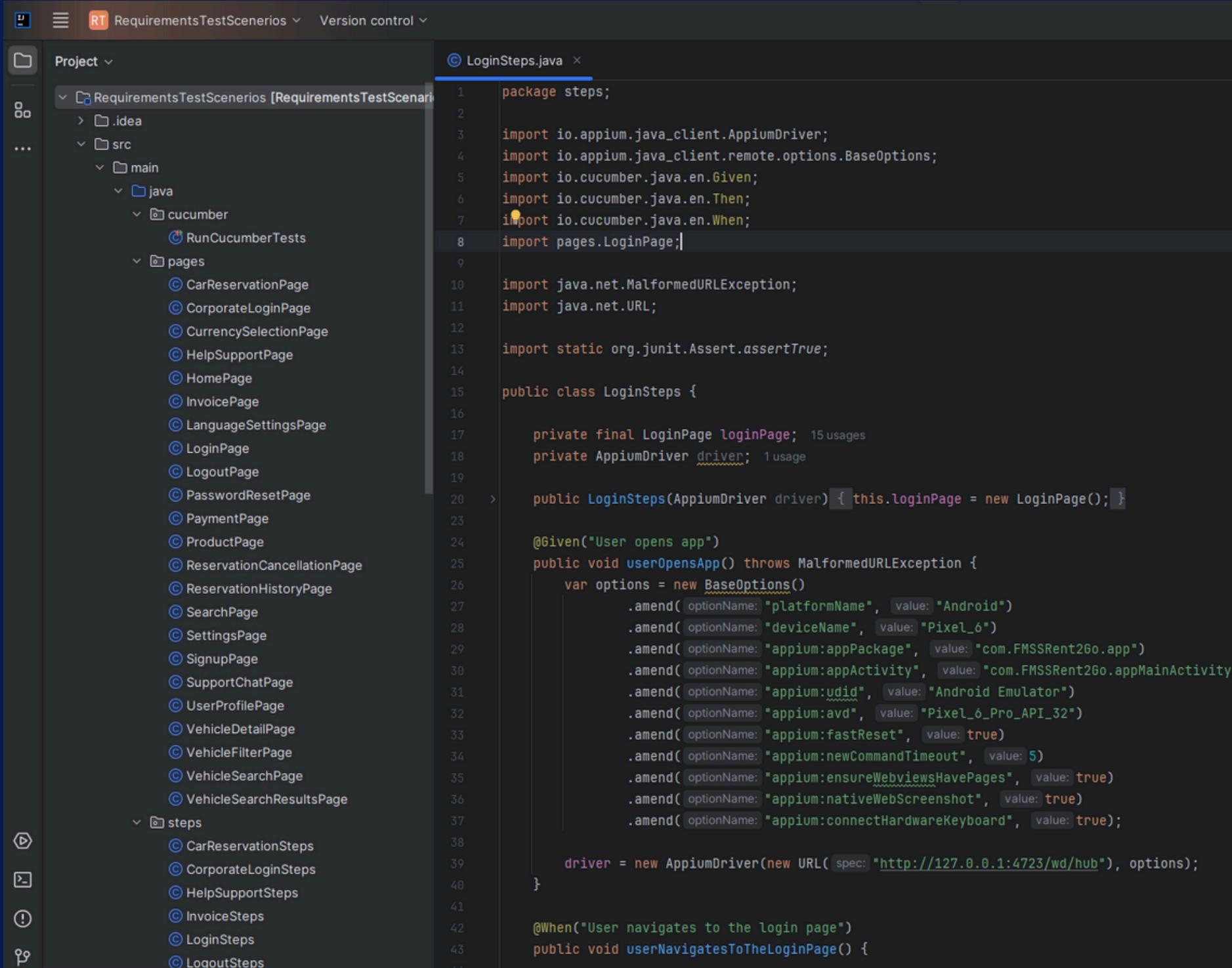
Müşteri Verilerinin Migrate Edilmesi detaylı dosyamızı Rent2Go klosörümüzde yer alan Migration & Api Application.pdf dosyasında bulabilirsiniz

# SECURITY AND ALTERNATIVES SEARCHING



SECURITY AND ALTERNATIVES SEARCHING detaylı dosyamızı Rent2Go klosörümüzde yer alan AlternativesSecuritySearching.pdf dosyasında bulabilirsiniz

# TEST SENARYOLARI VE TEST OTOMASYON



The screenshot shows an IDE interface with a project named "RequirementsTestScenarios". The code editor displays a file named "LoginSteps.java". The code uses the Cucumber framework for test scenarios and Appium for mobile testing. It includes imports for AppiumDriver, BaseOptions, Given, Then, When, and LoginPage. The class "LoginSteps" contains a constructor that initializes an AppiumDriver and a LoginPage. It also includes a @Given annotation for "User opens app" and a @When annotation for "User navigates to the login page". The project structure on the left shows various pages and steps files.

```
package steps;

import io.appium.java_client.AppiumDriver;
import io.appium.java_client.remote.options.BaseOptions;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
import pages.LoginPage;

import java.net.MalformedURLException;
import java.net.URL;

import static org.junit.Assert.assertTrue;

public class LoginSteps {

    private final LoginPage loginPage; 15 usages
    private AppiumDriver driver; 1 usage

    public LoginSteps(AppiumDriver driver) { this.loginPage = new LoginPage(); }

    @Given("User opens app")
    public void userOpensApp() throws MalformedURLException {
        var options = new BaseOptions()
            .amend(optionName: "platformName", value: "Android")
            .amend(optionName: "deviceName", value: "Pixel_6")
            .amend(optionName: "appium:appPackage", value: "com.FMSSRent2Go.app")
            .amend(optionName: "appium:appActivity", value: "com.FMSSRent2Go.appMainActivity")
            .amend(optionName: "appium:udid", value: "Android Emulator")
            .amend(optionName: "appium:avd", value: "Pixel_6_Pro_API_32")
            .amend(optionName: "appium:fastReset", value: true)
            .amend(optionName: "appium:newCommandTimeout", value: 5)
            .amend(optionName: "appium:ensureWebviewsHavePages", value: true)
            .amend(optionName: "appium:nativeWebScreenshot", value: true)
            .amend(optionName: "appium:connectHardwareKeyboard", value: true);

        driver = new AppiumDriver(new URL(spec: "http://127.0.0.1:4723/wd/hub"), options);
    }

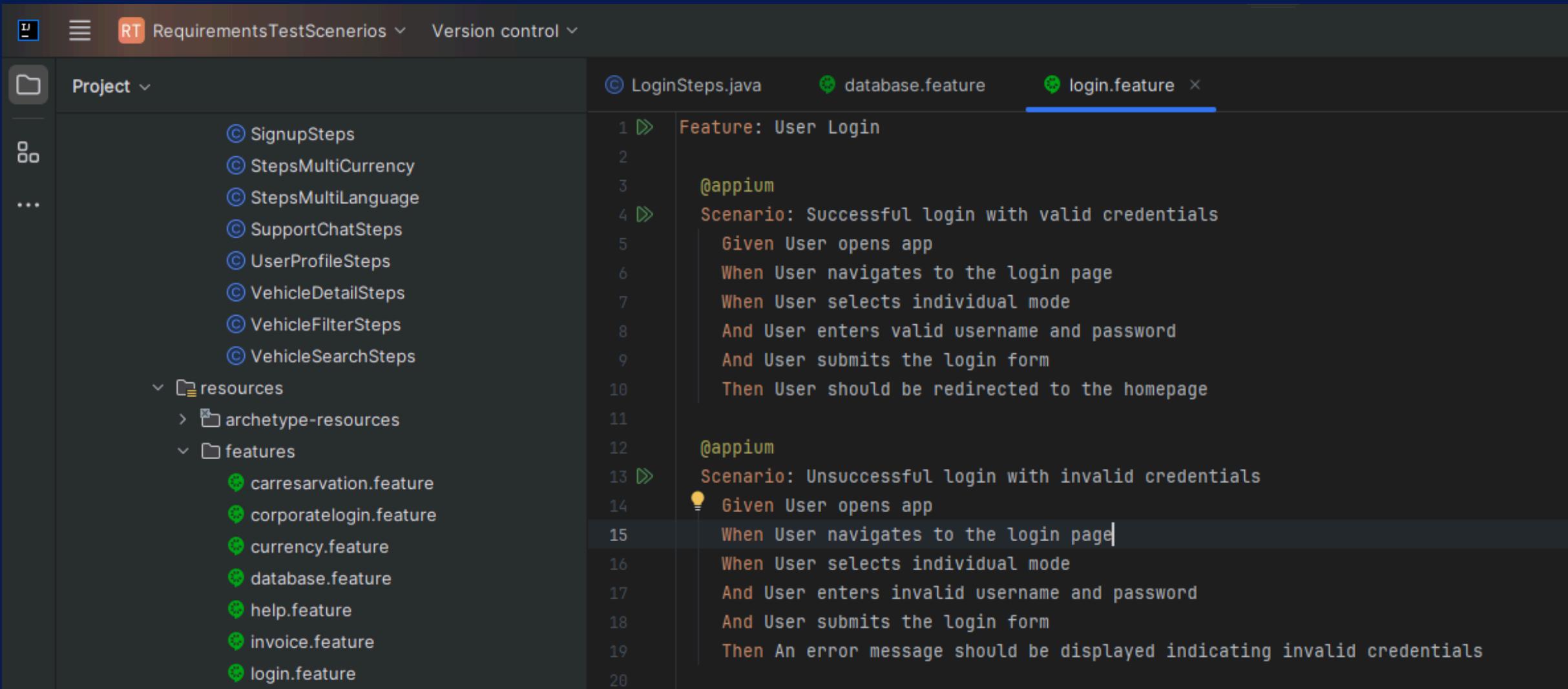
    @When("User navigates to the login page")
    public void userNavigatesToTheLoginPage() {
    }
}
```

**UI/UX tasarımını tamamladığımız demo uygulamamız için test senaryoları ve test otomasyon projelerini de hazırladık. Test otomasyon ve senaryolarını hazırlarken intelliJ, appium, cucumber gibi teknolojileri kullandık ve otomasyonları JAVA programlama dilinde yazdık.**

**Test otomasyon ve senaryoları projemizde yer alan UML diyagrlardan yola çıkılarak yapılmıştır. UML diyagrlar detaylı dosyamızı Rent2Go klosörümüzde yer alan UMLDiagrams.pdf dosyasında bulabilirsiniz**

# TEST SENARYOLARI VE TEST OTOMASYON

Hazırlanan bu projede database test senaryolarına da yer verdik. PostgreSQL ile oluşturduğumuz database için veritabanında tutulması gereken veri başlıklarının, veri tiplerinin, foreign key'lerin doğruluğunu test edecek senaryoları hazırladık.



The screenshot shows a code editor interface with a dark theme. On the left, there's a project tree titled "RequirementsTestScenarios" containing various step definitions and feature files. The main area displays two files: "LoginSteps.java" and "login.feature". The "login.feature" file is currently active and shows Gherkin test steps for user login, both successful and unsuccessful cases, annotated with @appium tags. The Java file "LoginSteps.java" contains the implementation of these steps using the Cucumber API.

```
Feature: User Login
  @appium
  Scenario: Successful login with valid credentials
    Given User opens app
    When User navigates to the login page
    When User selects individual mode
    And User enters valid username and password
    And User submits the login form
    Then User should be redirected to the homepage

  @appium
  Scenario: Unsuccessful login with invalid credentials
    Given User opens app
    When User navigates to the login page
    When User selects individual mode
    And User enters invalid username and password
    And User submits the login form
    Then An error message should be displayed indicating invalid credentials
```

Database senaryoları projemizde yer alan database tasarımları ve tablolardan yola çıkılarak yapılmıştır. Database detaylı dosyamızı Rent2Go klosörümüzde yer alan SignupAnalysis&DatabaseDesign.pdf dosyasında bulabilirsiniz