



# **FMSS SIGN UP EKRANI İŞ ANALİZİ RAPORU**

# İçindekiler

1. Giriş .....	3
2. Gereksinim Analizi .....	4
2.1 Araç Kiralama Uygulaması Kayıt Ol Ekranı Gereksinim Analizi .....	4
2.1.1 İşlevsel Gereksinimler (Functional Requirements).....	4
2.1.2 Teknik Gereksinimler (Non-Functional Requirements).....	5
2.1.3 Paydaş Gereksinimleri .....	5
2.1.4 Arayüz Tasarımı ve Veritabanı Entegrasyonu .....	7
3. Veritabanı Değişiklik Yönetimi .....	19
3.1 Update Yöntemi: .....	19
3.2 Flag Kullanımı: .....	19
3.3 Status Label Kullanımı: .....	19

## Şekiller Tablosu

Şekil 1 Arayüz Ekran Tasarımı .....	7
Şekil 2 Sign Up Ekranı Arayüz Tasarımı .....	8
Şekil 3 Hesap Aktivasyon İşlemi .....	9
Şekil 4 Login Ekranı.....	10
Şekil 5 Customers Tablosu .....	16
Şekil 6 Veritabanı Tablo Yönetimi .....	17

# 1.Giriş

İş analizi, bir organizasyonun iş süreçlerini, sistemlerini ve gereksinimlerini anlama, değerlendirme ve iyileştirme amacıyla gerçekleştirilen sistematik bir çalışmadır. İş analistleri, mevcut süreçleri inceleyerek iyileştirme fırsatlarını belirler, ilgili paydaşlarla iş birliği yaparak gereksinimleri toplar ve bu gereksinimlerin hayata geçirilmesi için çözümler geliştirir. İş analizi süreci, bir yazılım veya sistem geliştirme projesinde temel bir rol oynar, çünkü doğru gereksinimlerin tanımlanması, projenin başarısı için kritik öneme sahiptir. Ayrıca, iş analizi süreci, projedeki belirsizlikleri azaltarak, geliştirme ekibinin daha etkili ve verimli çalışmasına olanak tanır.

Bu döküman, mobil bir araç kiralama uygulamasının kayıt olma sürecine odaklanan iş analizi çalışmasının sonuçlarını içermektedir. Dökümanda, kullanıcı kayıt olma ekranının işlevleri, gereksinimleri, kullanıcı deneyimi (UX) ile ilgili öne çıkan unsurlar ve iyileştirme önerileri ele alınmıştır.

## Kayıt Olma Ekranının Önemi

Mobil bir uygulamanın kayıt olma ekranı, kullanıcının uygulama ile ilk etkileşime geçtiği noktadır ve kullanıcı deneyimi açısından büyük bir öneme sahiptir. Kullanıcıların uygulamaya kaydolma süreci, onların uygulamayı kullanmaya devam edip etmeyeceklerine dair ilk kararlarını etkiler. Bu nedenle, kayıt olma süreci basit, anlaşılır ve kullanıcı dostu olmalıdır. İyi tasarlanmış bir kayıt ol ekranı, kullanıcıların güvenini kazanmaya yardımcı olurken, karmaşık ve uzun bir süreç kullanıcıların uygulamayı terk etmesine neden olabilir.

Bu dökümanda, mobil araç kiralama uygulamasının kayıt olma ekranının iş analizi gerçekleştirilmiş, mevcut süreçlerin etkinliği değerlendirilmiş ve kullanıcı deneyimini iyileştirmek için öneriler sunulmuştur. Kullanıcı bilgilerini toplama, veri güvenliği ve kullanıcı arayüzü (UI) tasarımı gibi kritik unsurlar üzerinde durularak, kayıt olma sürecinin hem kullanıcılar hem de işletme açısından en verimli şekilde yapılandırılması hedeflenmiştir.

## 2. Gereksinim Analizi

Gereksinim analizi, bir sistemin ya da yazılımın geliştirilmesi sırasında, projenin başarılı olması için ne tür işlevler ve özellikler gerektiğini belirleme sürecidir. Bu analiz, hem teknik gereksinimleri (sistem mimarisi, veri yapıları, algoritmalar) hem de işlevsel gereksinimleri (kullanıcı arayüzü, kullanıcı davranışı, iş akışları) içerir. Amaç, tüm paydaşların ihtiyaçlarını netleştirerek, proje boyunca bu gereksinimlerin karşılanmasını sağlamaktır. Gereksinim analizi, yazılım projelerinde belirsizlikleri en aza indirir ve ekiplerin ortak bir hedef doğrultusunda çalışmasına olanak tanır.

özellikleri belirlemek. Kullanıcıların sıkça karşılaştığı sorunları veya eksiklikleri tespit ederek, bu hatalardan kaçınabilirsiniz.

### 2.1 Araç Kiralama Uygulaması Kayıt Ol Ekranı Gereksinim Analizi

#### 2.1.1 İşlevsel Gereksinimler (Functional Requirements)

- **Kullanıcı Bilgileri Girişi:**
  - Kullanıcı, ad, soyad, e-posta adresi, telefon numarası, doğum tarihi, TC kimlik numarası ve şifre gibi temel bilgilerini girmelidir.
  - Kullanıcı, geçerli bir e-posta adresi ve telefon numarası girmelidir.
  - Doğum tarihi için uygun bir tarih seçici (date picker) kullanılmalıdır.
  - Girilen şifre, belirlenen güvenlik gereksinimlerini karşılamalıdır.
  - TC kimlik numarası, belirlenen formatta olmalı ve geçerliliği kontrol edilmelidir.
  - Doğum tarihi, doğru formatta ve kullanıcı yaşı için uygun bir aralıkta olmalıdır (örn. 18 yaşından büyük olma zorunluluğu).
- **Kullanıcı Sözleşmesi ve Gizlilik Politikası:**

- Kullanıcı, kayıt olmadan önce kullanım şartları ve gizlilik politikasını kabul etmelidir.
- Sözleşme metinlerine yönlendiren bağlantılar bulunmalıdır.
- **Hata ve Uyarı Mesajları:**
  - Geçersiz ya da eksik bilgi girildiğinde, kullanıcıyı bilgilendiren hata mesajları gösterilmelidir.
  - E-posta adresi, telefon numarası veya TC kimlik numarası zaten kullanılıyorsa, kullanıcıya uygun bir uyarı verilmelidir.
- **Gizlilik ve Güvenlik:**
  - Kullanıcı verileri güvenli bir şekilde saklanmalı ve şifrelenmelidir.
  - TC kimlik numarası ve diğer hassas bilgiler, özel olarak korunmalıdır.
- **Kullanıcı Doğrulama:**
  - Kayıt sonrası, kullanıcının e-posta adresine doğrulama e-postası gönderilmelidir.
  - Kullanıcı, doğrulama işlemini tamamladıktan sonra sisteme giriş yapabilmelidir.

## 2.1.2 Teknik Gereksinimler (Non-Functional Requirements)

- **Performans**

Aynı anda yüzlerce kullanıcının kayıt yapabileceği şekilde ölçeklenebilir olmalıdır.
- **Kullanılabilirlik**

Kullanıcı arayüzü, basit ve kullanıcı dostu olacak şekilde tasarlanmalıdır.
- **Uyumluluk**

Çeşitli tarayıcılar ve cihazlar arasında uyumluluk sağlanmalıdır.

## 2.1.3 Paydaş Gereksinimleri

- **İşletme Sahipleri**

Kullanıcı verilerinin toplanması ve analiz edilmesi için uygun altyapının sağlanması gerekmektedir.

Kayıt süreci, kullanıcıların hızlı bir şekilde üye olmasını teşvik etmelidir.

- **Kullanıcılar**

Kayıt süreci, kullanıcıları karmaşık işlemlerle zorlamadan basit ve hızlı olmalıdır.

- **Geliştiriciler**

Uygulamanın kolayca bakım yapılabilir ve genişletilebilir olması sağlanmalıdır.

Tüm işlevler, iyi belgelenmiş ve modüler bir yapıda olmalıdır.

## 2.1.4 Arayüz Tasarımı ve Veritabanı Entegrasyonu

### Arayüz Ekranı Tasarım Analizi

Sign Up

Name

Surname

E-mail

Phone Number

Password

Confirm Password

[Kişisel verilerimin işlenmesine ilişkin bilgilendirme metnini okudum ve kabul ediyorum.](#) ☐

Uygulamanın oluşturduğu kampanyalar ve fırsatlar hakkında bilgi alabilmek için uygulamanın e-mail ve SMS göndermesini istiyorum. ☐

Sign Up

Şekil 1 Arayüz Ekran Tasarımı

Sign Up

Name

Surname

E-mail

Phone Number

Password

Confirm Password

Kişisel verilerimin işlenmesine ilişkin bilgilendirme metnini [okudum ve kabul ediyorum.](#) ☒

Uygulamanın oluşturduğu kampanyalar ve fırsatlar hakkında bilgi alabilmek için uygulamanın e-mail ve SMS göndermesini istiyorum. ☒

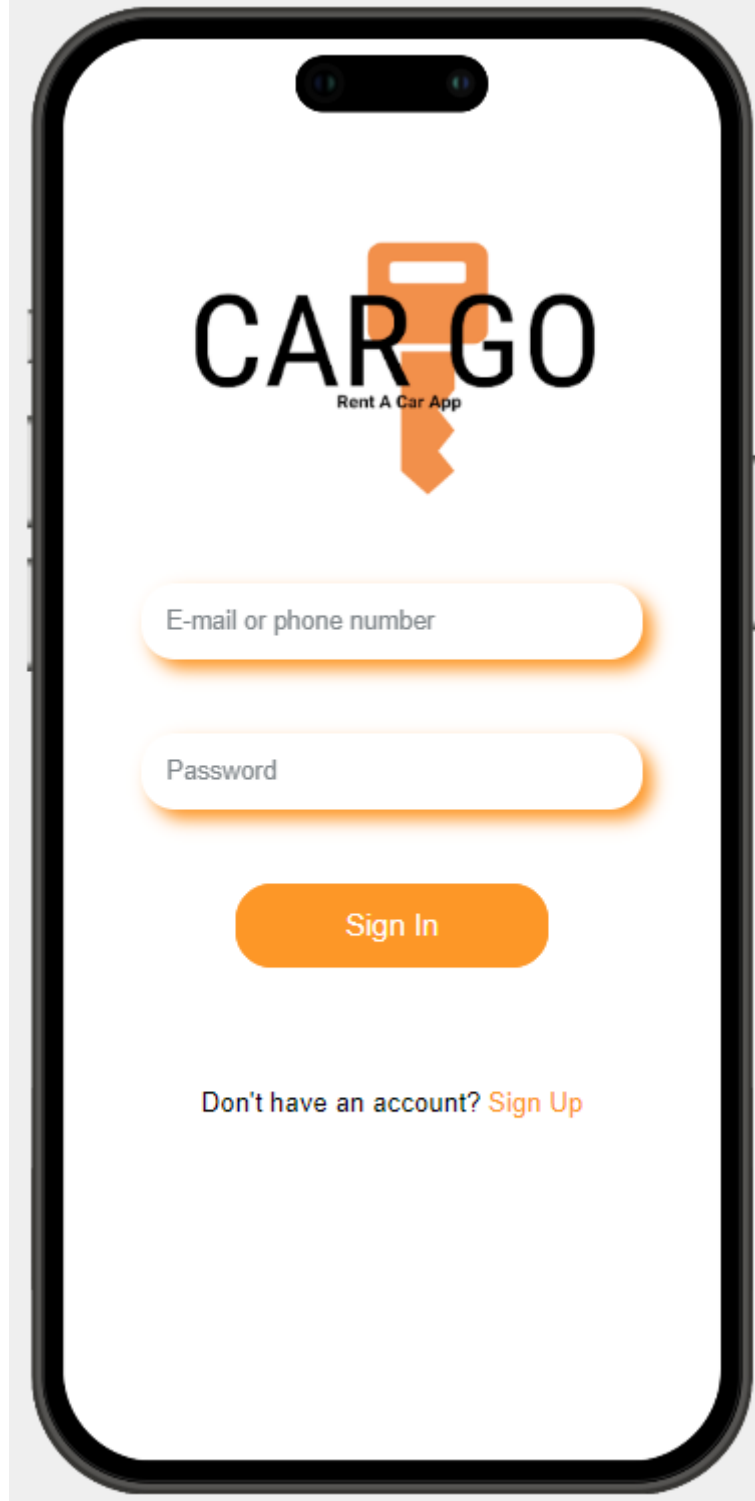
Sign Up

Şekil 2 Sign Up Ekranı Arayüz Tasarımı





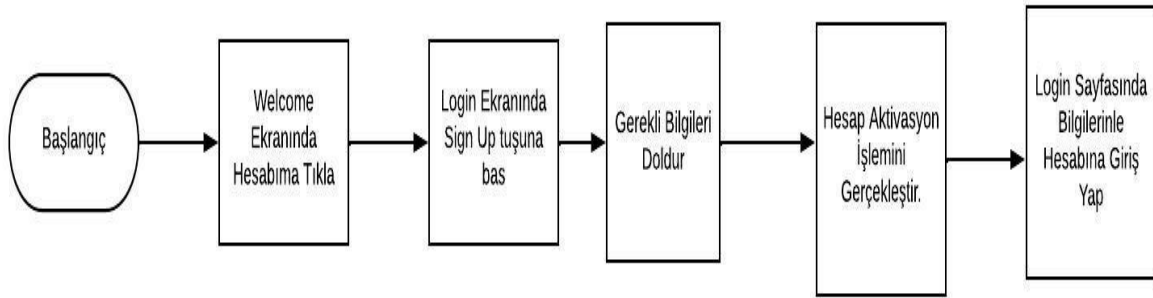
řekil 3 Hesap Aktivasyon İşlemi



Şekil 4 Login Ekranı

Bu kayıt olma ekranı, mobil bir uygulama için tasarlanmış olup oldukça sade ve kullanıcı dostu bir arayüze sahiptir. Form alanları, kullanıcıların adım adım bilgi girişi yapmasını kolaylaştıracak şekilde düzenli bir şekilde dikey olarak hizalanmıştır. Beyaz arka plan, form alanlarının etrafındaki turuncu gölgeleri öne çıkararak, kullanıcıların ilgili alanları kolayca fark etmesini sağlar. “Sign Up” butonu, turuncu rengiyle dikkat çekici bir aksiyon butonu olarak öne çıkar. Ekranın alt kısmındaki onay kutuları, veri işleme izni ve promosyon iletişimi için gerekli izinleri almak amacıyla yerleştirilmiştir. Genel olarak, bu tasarım kullanıcıların formu rahatça doldurmasını sağlayacak şekilde düzenlenmiştir, ancak kontrast ve metin okunabilirliği gibi küçük iyileştirmelerle daha modern ve çekici bir hale getirilebilir. Kullanıcı bu ekranı tamamladıktan sonra, SMS doğrulama ekranına yönlendirilir.

SMS doğrulama ekranında, kullanıcı telefon numarasına gönderilen 6 haneli SMS kodunu girmek zorundadır. Bu ekran, doğrulama işlemini kolay ve anlaşılır bir şekilde sunmakta. Ekranın üst kısmında, gizlenmiş şekilde gösterilen telefon numarası ve kullanıcıdan SMS kodunu girmesini isteyen bir talimat bulunur. Kodun girileceği alanlar, kutucuklar şeklinde sunulmuş ve her bir kutucuk tek bir rakamı kabul edecek şekilde tasarlanmıştır. Turuncu renkte “Onayla” butonu, kullanıcıyı işlem tamamlamaya yönlendirir. Doğru kod girildiğinde, ekranın alt kısmında kullanıcıya hesabının başarıyla oluşturulduğu ve artık e-posta ile şifresiyle giriş yapabileceği bilgisi verilir. Bu bilgilendirme de, başarılı bir işlemi simgeleyen yeşil bir tik işareti ile birlikte sunulmuştur. Sonrasında, “Giriş Yap” butonuna basıldığında, kullanıcı giriş ekranına geri yönlendirilir ve hesap bilgileriyle uygulamaya giriş yapabilir. Bu akış, kullanıcıya net ve adım adım ilerleyen bir süreç sunarak, kayıt işlemini sorunsuz bir şekilde tamamlamasını sağlar. Aşağıdaki şemada SignUp süreci akışı gösterilmektedir.



## **Arayüzde Girilen Bilgilerin Database'de Tutulma Şeklinin Analizi**

Kayıt olma ekranında kullanıcının girdiği bilgiler veritabanında belirli sütunlarda saklanır. Bu bilgiler, genellikle kullanıcı arayüzünde (UI) form elemanları aracılığıyla alınır ve ardından veritabanına kaydedilir. İşleyişi şu şekilde analiz edebiliriz:

### **1. Kullanıcı Adı ve Şifre:**

- Kullanıcılar genellikle bir kullanıcı adı (username) ve şifre (password) oluştururlar. Kullanıcı adı benzersiz olmalı ve sisteme giriş için kullanılır. Şifrelerin düz metin olarak saklanması güvenlik riskleri taşır. Bu nedenle şifrelerin hash algoritmaları (örneğin, bcrypt, SHA-256) kullanılarak saklanması en iyi uygulamadır.

### **2. Kişisel Bilgiler:**

- Ad (first\_name) ve soyad (last\_name) gibi kişisel bilgiler, form alanlarından alınarak veritabanında saklanır. Bu bilgiler, müşteri hesaplarıyla ilgili işlemler için kullanılır.

### **3. İletişim Bilgileri:**

- E-posta adresi (email) ve telefon numarası (phone\_number), kullanıcının iletişim bilgilerini tutar. E-posta adresi genellikle bir doğrulama sürecinden geçirilir ve benzersiz olması gerekebilir. Telefon numarası da kullanıcı doğrulaması veya çok faktörlü kimlik doğrulama (MFA) gibi işlemler için kullanılabilir.

### **4. Doğum Tarihi ve Ehliyet Numarası:**

- Doğum tarihi (date\_of\_birth) ve ehliyet numarası (license\_number) gibi bilgiler, kullanıcının kimliğini doğrulamak ve belirli hizmetleri (örneğin, araç kiralama) sağlamak için kullanılabilir. Bu tür hassas bilgiler, veri koruma yasalarına uygun olarak saklanmalı ve korunmalıdır.

## 5. Kayıt ve Güncelleme Tarihleri:

- created\_at ve updated\_at sütunları, veritabanında kayıtların ne zaman oluşturulduğunu ve güncellendiğini izlemek için kullanılır. Bu sütunlar, kullanıcı etkinliklerini izlemek ve gerektiğinde kayıtları güncellemek için faydalıdır.

## 6. Hesap Durumu:

- status sütunu, kullanıcının hesabının aktif mi yoksa inaktif mi olduğunu belirlemek için kullanılır. Bu bilgi, sistemin kullanıcı hesaplarını yönetmesine yardımcı olur.

## Customers Tablosundaki Veri Tipleri

**-INT:** Tam sayıları saklar. Genellikle 4 bayt yer kaplar ve -2,147,483,648 ile 2,147,483,647 arasında değerler alabilir.

**-SMALLINT:** Daha küçük tam sayıları saklar. Genellikle 2 bayt yer kaplar ve -32,768 ile 32,767 arasında değerler alabilir.

**-TINYINT:** Çok küçük tam sayıları saklar. Genellikle 1 bayt yer kaplar ve 0 ile 255 arasında değerler alabilir.

**-BIGINT:** Büyük tam sayıları saklar. Genellikle 8 bayt yer kaplar ve -9,223,372,036,854,775,808 ile 9,223,372,036,854,775,807 arasında değerler alabilir.

**-FLOAT:** Ondalıklı sayıları saklar. Genellikle 4 bayt yer kaplar ve yaklaşık 7 basamağa kadar hassasiyet sunar.

**-DOUBLE:** Daha hassas ondalıklı sayıları saklar. Genellikle 8 bayt yer kaplar ve yaklaşık 15 basamağa kadar hassasiyet sunar.

**-DECIMAL:** Kesirli sayıları tam hassasiyetle saklar. İki parametre alır: toplam basamak sayısı ve ondalık basamak sayısı (örneğin, DECIMAL(10,2)).

**-CHAR:** Sabit uzunlukta karakter dizileri saklar. Uzunluk belirtilmelidir (örneğin, CHAR(10)).

**VARCHAR:** Değişken uzunlukta karakter dizileri saklar. Maksimum uzunluk belirtilmelidir (örneğin, VARCHAR(255)).

**-TEXT:** Uzun metin verilerini saklar. TEXT, MEDIUMTEXT, ve LONGTEXT gibi varyasyonları bulunur ve büyük veri miktarlarını depolamak için kullanılır.

**- DATE:** Tarih verilerini saklar. Genellikle YYYY-MM-DD formatında saklanır.

**-TIME:** Saat verilerini saklar. Genellikle HH:MM:SS formatında saklanır.

**-DATETIME:** Tarih ve saat verilerini saklar. Genellikle YYYY-MM-DD HH:MM:SS formatında saklanır.

**-TIMESTAMP:** Tarih ve saat verilerini saklar ve zaman damgalarını temsil eder. Genellikle YYYY-MM-DD HH:MM:SS formatında saklanır ve otomatik olarak güncellenebilir.

**-BOOLEAN:** Boolean değerler saklar. Genellikle TRUE veya FALSE olarak değer alır.

**-BLOB:** Binary Large Objects (BLOBs) yani büyük ikili verileri saklar. Genellikle resimler, sesler veya diğer medya dosyaları için kullanılır.

**-ENUM:** Sınırlı bir değer kümesinden birini seçmek için kullanılır. Belirli bir liste içinden seçilmesi gereken bir değeri saklar.

Şekil1’de yer alan customers tablosu için sütunlar ve veri tipleri şu şekilde sıralanmıştır:

**-customer\_id:** int (Auto Increment, Primary Key)

Bu sütun, her müşteriye atanmış benzersiz bir kimlik numarasını tutar.

**-first\_name:** varchar(50)

Müşterinin adını tutar. Maksimum 50 karakter uzunluğunda olabilir.

**-last\_name:** varchar(50)

Müşterinin soyadını tutar. Maksimum 50 karakter uzunluğunda olabilir.

**-username:** varchar(50)

Müşterinin kullanıcı adını tutar. Maksimum 50 karakter uzunluğunda olabilir.

**-password:** varchar(50)

Müşterinin şifresini tutar. Şifre verisi düz metin olarak saklanıyor gibi görünüyor, bu durum güvenlik açısından çok risklidir ve bunun yerine şifrelerin hashlenerek saklanması önerilir.

**-email:** varchar(255)

Müşterinin e-posta adresini tutar. Maksimum 255 karakter uzunluğunda olabilir.

**-phone\_number:** varchar(20)

Müşterinin telefon numarasını tutar. Maksimum 20 karakter uzunluğunda olabilir, bu da uluslararası telefon numaralarını saklayabilmek için yeterlidir.

**-date\_of\_birth:** date

Müşterinin doğum tarihini tutar. Bu veri türü, tarihin YYYY-MM-DD formatında saklanmasını sağlar.

**-license\_number:** varchar(20)

Müşterinin ehliyet numarasını tutar. Maksimum 20 karakter uzunluğunda olabilir.

**-created\_at:** timestamp

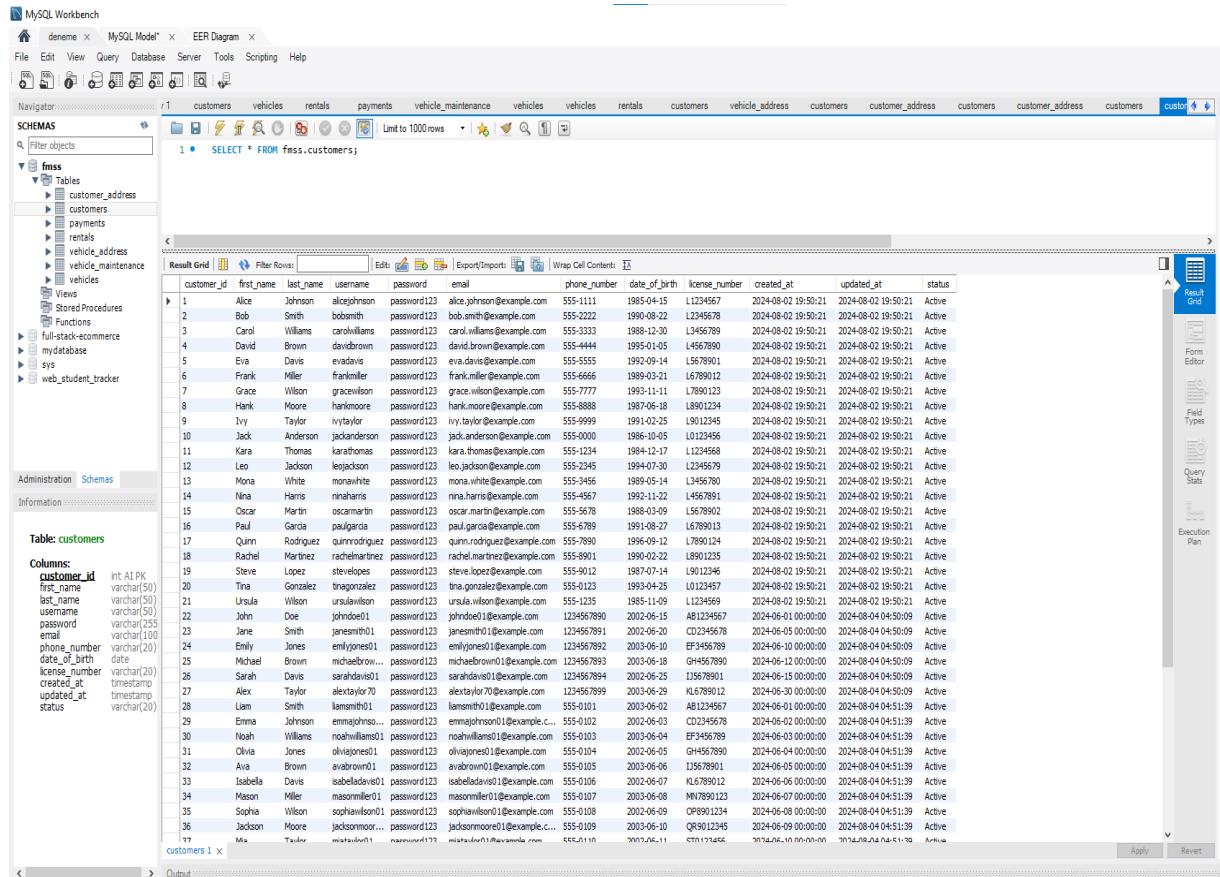
Müşterinin kayıt olma zamanını tutar. timestamp veri türü, tarih ve saati saklar ve otomatik olarak güncellenebilir.

**-updated\_at:** timestamp

Müşteri bilgileri güncellendiğinde bu sütun otomatik olarak güncellenir.

**-status:** varchar(20)

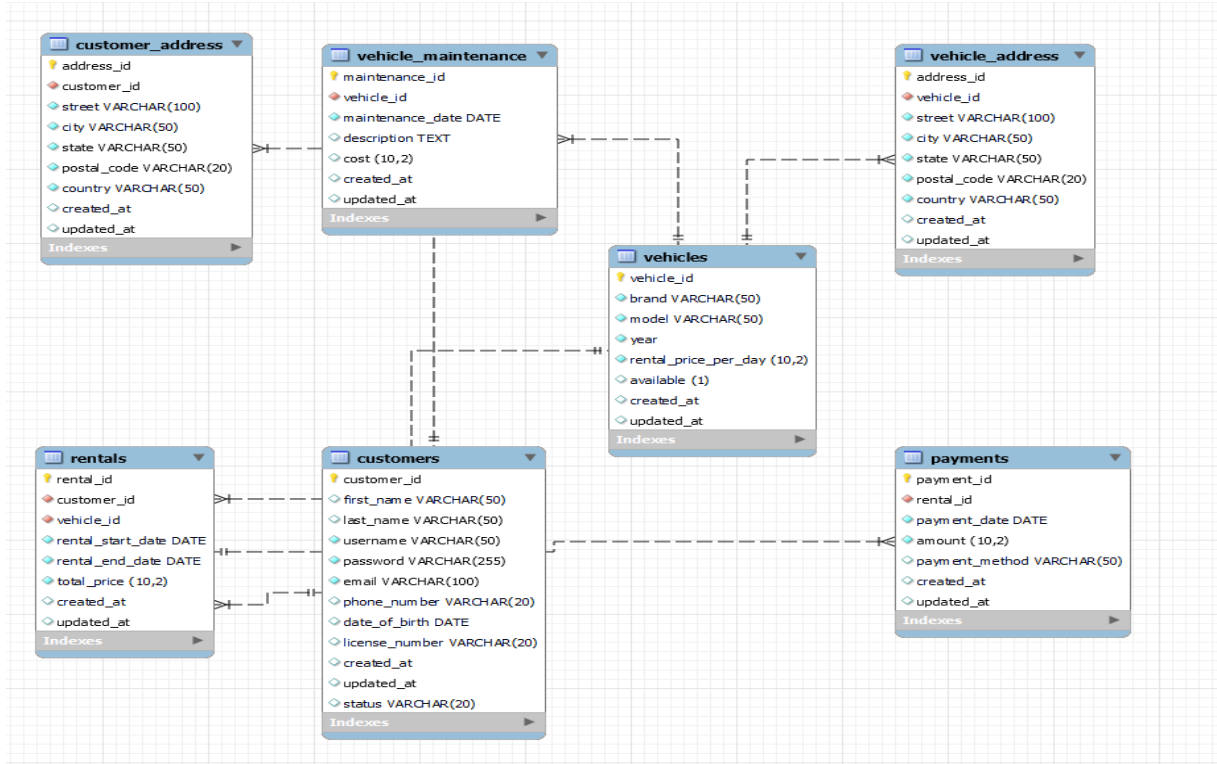
Müşterinin hesabının durumunu (örneğin, "Active" veya "Inactive") tutar. Maksimum 20 karakter uzunluğunda olabilir.



customer_id	first_name	last_name	username	password	email	phone_number	date_of_birth	license_number	created_at	updated_at	status
1	Alice	Johnson	alicejohnson	password123	alice.johnson@example.com	555-1111	1985-04-15	L1234567	2024-08-02 19:50:21	2024-08-02 19:50:21	Active
2	Bob	Smith	bobsmith	password123	bob.smith@example.com	555-2222	1990-08-22	L2345678	2024-08-02 19:50:21	2024-08-02 19:50:21	Active
3	Carol	Williams	carolwilliams	password123	carol.williams@example.com	555-3333	1988-12-30	L3456789	2024-08-02 19:50:21	2024-08-02 19:50:21	Active
4	David	Brown	davidbrown	password123	david.brown@example.com	555-4444	1995-01-05	L4567890	2024-08-02 19:50:21	2024-08-02 19:50:21	Active
5	Eva	Davis	evadavis	password123	eva.davis@example.com	555-5555	1992-09-14	L5678901	2024-08-02 19:50:21	2024-08-02 19:50:21	Active
6	Frank	Miller	frankmiller	password123	frank.miller@example.com	555-6666	1989-03-21	L6789012	2024-08-02 19:50:21	2024-08-02 19:50:21	Active
7	Grace	Wilson	gracewilson	password123	grace.wilson@example.com	555-7777	1993-11-11	L7890123	2024-08-02 19:50:21	2024-08-02 19:50:21	Active
8	Hank	Moore	hankmoore	password123	hank.moore@example.com	555-8888	1987-06-18	L8901234	2024-08-02 19:50:21	2024-08-02 19:50:21	Active
9	Ivy	Taylor	ivy.taylor	password123	ivy.taylor@example.com	555-9999	1991-02-25	L9012345	2024-08-02 19:50:21	2024-08-02 19:50:21	Active
10	Jack	Anderson	jackanderson	password123	jack.anderson@example.com	555-0000	1986-10-05	L0123456	2024-08-02 19:50:21	2024-08-02 19:50:21	Active
11	Kara	Thomas	karathomas	password123	kara.thomas@example.com	555-1234	1984-12-17	L1234567	2024-08-02 19:50:21	2024-08-02 19:50:21	Active
12	Leo	Jackson	leojackson	password123	leo.jackson@example.com	555-2345	1994-07-30	L2345678	2024-08-02 19:50:21	2024-08-02 19:50:21	Active
13	Mona	White	monawhite	password123	mona.white@example.com	555-3456	1989-05-14	L3456789	2024-08-02 19:50:21	2024-08-02 19:50:21	Active
14	Nina	Harris	ninaharris	password123	nina.harris@example.com	555-4567	1992-11-22	L4567890	2024-08-02 19:50:21	2024-08-02 19:50:21	Active
15	Oscar	Martin	oscartmartin	password123	oscar.martin@example.com	555-5678	1988-03-09	L5678902	2024-08-02 19:50:21	2024-08-02 19:50:21	Active
16	Paul	Garcia	paulgarcia	password123	paul.garcia@example.com	555-6789	1991-08-27	L6789013	2024-08-02 19:50:21	2024-08-02 19:50:21	Active
17	Quinn	Rodriguez	quinnrodriguez	password123	quinn.rodriguez@example.com	555-7890	1996-09-12	L7890124	2024-08-02 19:50:21	2024-08-02 19:50:21	Active
18	Rachel	Martinez	rachelmartinez	password123	rachel.martinez@example.com	555-8901	1990-02-22	L8901235	2024-08-02 19:50:21	2024-08-02 19:50:21	Active
19	Steve	Lopez	stevelopez	password123	steve.lopez@example.com	555-9012	1987-07-14	L9012346	2024-08-02 19:50:21	2024-08-02 19:50:21	Active
20	Tina	Gonzalez	tinagonzalez	password123	tina.gonzalez@example.com	555-0123	1993-04-25	L0123457	2024-08-02 19:50:21	2024-08-02 19:50:21	Active
21	Usula	Wilson	usulawilson	password123	usula.wilson@example.com	555-1235	1985-11-09	L1234569	2024-08-02 19:50:21	2024-08-02 19:50:21	Active
22	John	Doe	johndoe01	password123	johndoe01@example.com	1234567890	2002-06-15	A01234567	2024-06-01 00:00:00	2024-08-04 04:50:09	Active
23	Jane	Smith	janesmith01	password123	janesmith01@example.com	1234567891	2002-06-20	C02345678	2024-06-05 00:00:00	2024-08-04 04:50:09	Active
24	Emily	Jones	emilyjones01	password123	emilyjones01@example.com	1234567892	2003-06-10	EF3456789	2024-06-10 00:00:00	2024-08-04 04:50:09	Active
25	Michael	Brown	michaebrown01	password123	michaebrown01@example.com	1234567893	2003-06-18	GH4567890	2024-06-12 00:00:00	2024-08-04 04:50:09	Active
26	Sarah	Davis	sarahdavis01	password123	sarahdavis01@example.com	1234567894	2002-06-25	IJ5678901	2024-06-15 00:00:00	2024-08-04 04:50:09	Active
27	Alex	Taylor	alextaylor70	password123	alextaylor70@example.com	1234567899	2003-06-29	KL6789012	2024-06-30 00:00:00	2024-08-04 04:50:09	Active
28	Liam	Smith	liamsmith01	password123	liamsmith01@example.com	555-0101	2003-06-02	AB1234567	2024-06-01 00:00:00	2024-08-04 04:51:39	Active
29	Emma	Johnson	emmajohnson01	password123	emmajohnson01@example.c...	555-0102	2002-06-03	CD2345678	2024-06-02 00:00:00	2024-08-04 04:51:39	Active
30	Noah	Williams	noahwilliams01	password123	noahwilliams01@example.com	555-0103	2002-06-04	EF3456789	2024-06-03 00:00:00	2024-08-04 04:51:39	Active
31	Olivia	Jones	oliviajones01	password123	oliviajones01@example.com	555-0104	2002-06-05	GH4567890	2024-06-04 00:00:00	2024-08-04 04:51:39	Active
32	Ava	Brown	avabrown01	password123	avabrown01@example.com	555-0105	2003-06-06	IJ5678901	2024-06-05 00:00:00	2024-08-04 04:51:39	Active
33	Isabella	Davis	isabelladavis01	password123	isabelladavis01@example.com	555-0106	2002-06-07	KL6789012	2024-06-06 00:00:00	2024-08-04 04:51:39	Active
34	Heason	Miller	masomiller01	password123	masomiller01@example.com	555-0107	2003-06-08	MN7890123	2024-06-07 00:00:00	2024-08-04 04:51:39	Active
35	Sophia	Wilson	sophiawilson01	password123	sophiawilson01@example.com	555-0108	2002-06-09	OP8901234	2024-06-08 00:00:00	2024-08-04 04:51:39	Active
36	Jackson	Moore	jacksonmoore01	password123	jacksonmoore01@example.c...	555-0109	2003-06-10	QR9012345	2024-06-09 00:00:00	2024-08-04 04:51:39	Active
37	Mia	Taylor	miautaylor01	password123	miautaylor01@example.com	555-0110	2003-06-11	ST0123456	2024-06-10 00:00:00	2024-08-04 04:51:39	Active

Şekil 5 Customers Tablosu





Şekil 6 Veritabanı Tablo Yönetimi

### rentals Tablosu ile İlişki

- **Foreign Key:** customer\_id (rentals tablosunda)
- **İlişki:**

customers tablosundaki customer\_id sütunu, rentals tablosundaki customer\_id sütunu ile ilişkilidir. Bu ilişki, bir müşterinin bir veya birden fazla kiralama işlemi yapabileceğini gösterir. customers tablosundaki bir kayıt silindiğinde, rentals tablosundaki ilgili kiralama kayıtları da silinebilir (eğer "on delete cascade" gibi bir kural tanımlanmışsa).

### customer\_address Tablosu ile İlişki

- **Foreign Key:** customer\_id (customer\_address tablosunda)
- **İlişki:**

customers tablosundaki customer\_id, customer\_address tablosunda customer\_id olarak yer alır. Bu ilişki, bir müşterinin bir veya birden fazla adres bilgisi olabileceğini

gösterir. Müşteriye ait adres bilgileri customer\_address tablosunda saklanır ve bu adresler, customers tablosundaki müşteri kaydı ile ilişkilendirilir.

### **payments Tablosu ile Dolaylı İlişki**

- **Foreign Key (Indirect Relationship):** rental\_id (payments tablosunda)
- **İlişki:**

payments tablosu doğrudan customers tablosuna bağlı değil, ancak dolaylı bir ilişki vardır. payments tablosundaki rental\_id sütunu, rentals tablosundaki rental\_id ile ilişkilidir. rentals tablosu ise customer\_id ile customers tablosuna bağlıdır. Bu dolaylı ilişki, bir müşterinin yaptığı her kiralama için bir ödeme kaydı tutulduğunu gösterir.

### **Genel Analiz**

- **Veritabanı Tasarımı ve İlişkiler:**

customers tablosu, kullanıcı (müşteri) bilgilerini tutan merkez tablodur ve diğer tablolarla güçlü ilişkiler kurarak veritabanı bütünlüğünü sağlar. rentals tablosu, müşterinin kiralama işlemlerini takip ederken, customer\_address tablosu, müşterinin birden fazla adresini saklayabilir. payments tablosu, yapılan ödemelerin kaydını tutarken dolaylı olarak müşteri bilgileri ile ilişkilidir.

Bu ilişkiler, bir müşterinin sisteme kaydolduğu andan itibaren, kiralama ve ödeme işlemlerine kadar olan tüm süreci takip etmeye olanak tanır. Veritabanı tasarımı, müşteri bilgilerini ve bu bilgilerin diğer tablolarla olan ilişkisini etkili bir şekilde yönetir, bu da sistemin işleyişini düzenler.

## 3. Veritabanı Değişiklik Yönetimi

Bu veritabanı modelinde, müşteri, araç, kiralama ve ödeme gibi işlemleri yönetmek için farklı tablolar oluşturulmuştur. Tablolar arasında ilişkiler tanımlanmış ve her tablo, verilerin güncellenmesi gerektiğinde izlenebilir olması için çeşitli sütunlara sahiptir. Şimdi, tabloda yapılacak bir değişikliğin nasıl yönetileceğini ele alalım ve bunu birkaç alternatif çözümle karşılaştıralım:

### 3.1 Update Yöntemi:

Veritabanındaki mevcut bir kaydın güncellenmesi gerektiğinde, Update yöntemi kullanılır. Bu yöntem, tabloya yeni bir satır eklemeyen, mevcut satırdaki verileri güncelleyerek, değişiklikleri yönetir. Örneğin, rentals tablosunda bir kiralama işlemi süresi uzatıldığında rental\_end\_date sütunu güncellenebilir. Bu yöntem, basit ve doğrudan bir yaklaşım sunar; ancak, değişikliklerin geçmişini tutmak mümkün değildir.

### 3.2 Flag Kullanımı:

Değişikliklerin izlenmesi ve geçmişe yönelik sorguların yapılabilmesi için, tablolar içerisinde bir flag veya durum belirteci kullanılabilir. Örneğin, vehicles tablosunda, bir aracın mevcut durumda kiralanabilir olup olmadığını belirten available sütunu bulunmaktadır. Bu sütun, aracın durumunu 1 (kiralanabilir) veya 0 (kiralanamaz) olarak işaretleyebilir. Böylece, mevcut durum korunur ancak geçmişe yönelik durumlar kaydedilmez.

### 3.3 Status Label Kullanımı:

Status label veya durum etiketleri, bir kaydın farklı durumlarını tanımlamak için kullanılabilir. Bu yöntem, customers tablosunda status sütunu gibi örneklerde görülebilir. Bu sütun, müşteri durumunu aktif, pasif, yasaklı vb. durumlarla ifade edebilir. Bu yaklaşım, sistemdeki değişiklikleri yönetmek için esnek bir yöntem sunar ve daha detaylı durum izleme imkanı sağlar.

### Developer Perspektifinden Değerlendirme:

Bir geliştirici açısından, veritabanı tablolarında yapılacak değişikliklerin yönetiminde hangi yöntemin kullanılacağı, sistemin ihtiyaçlarına ve veri bütünlüğüne göre belirlenmelidir. Update yöntemi hızlı ve düşük maliyetlidir, ancak değişiklik geçmişi tutulmaz. Flag kullanımı, belirli durumları izlemek için basit bir yol sağlar, ancak geçmişi korumaz. Status label kullanımı ise, daha esnek ve detaylı izleme imkanı sunar, ancak daha karmaşık ve yönetimi zor olabilir. Sonuç olarak, geliştiricinin sistemin gereksinimlerini dikkatlice analiz ederek, en uygun yöntemi seçmesi gerekir.

Örneğin, bir müşteri durumu değişikliğini izlemek için status label kullanmak, müşterinin farklı zamanlardaki durumlarını sorgulama ihtiyacını karşılayabilirken, available flag, araçların kiralanabilir olup olmadığını basitçe izlemek için yeterli olabilir. Geliştirici, performans ve veri yönetimi arasındaki dengeyi sağlayarak en uygun çözümü tercih etmelidir.