

Otobüs Bileti Rezervasyon Sistemi Raporu

Proje Ekibindeki Kişiler

- **Ammar Al Jasem**
Öğrenci No: 220260603
- **Ammar Kenj**
Öğrenci No: 210260601
- **Veysel Karani Kılıçerkan**
Öğrenci No: 230260603

Projenin Amacı

Bu proje, bir otobüs taşımacılığı şirketinin bilet rezervasyon ve yönetim süreçlerini dijitalleştirerek, hızlı, düzenli ve hatasız bir şekilde işlemlerini gerçekleştirmesini sağlamayı amaçlamaktadır. Proje kapsamında şunlar gerçekleştirilmiştir:

- Merkezi bir veri tabanı tasarımı: Tüm kullanıcı, otobüs, otogar, sefer ve rezervasyon bilgilerini tek bir sistemde toplayarak verilerin tutarlı ve erişilebilir olmasını sağlar.
- Bilet rezervasyon ve yönetimi: Otobüs seferlerinin planlanması, koltuk rezervasyonu ve bilet satış işlemleri detaylı bir şekilde kaydedilir ve yönetilir.
- Şikayet ve bildirim modülleri: Yolcu geri bildirimlerinin (şikayetlerin) kolayca alınması ve sistem üzerinden yolculara bilgilendirme yapılması.
- Kullanıcılar ve yetkilendirme işlevlerinin entegrasyonu: Farklı kullanıcı tipleri (yolcular, operatörler ve sistem yöneticileri) için erişim kontrolü ve görev yetkilendirmesi sağlanır.
- Ödeme ve iptal işlemleri: Rezervasyon ödeme işlemleri ve iptaller için entegre bir işlem altyapısı sunulur.
- Geçiş noktaları ve durak yönetimi: Uzun mesafeli seferlerde ara durak ve geçiş noktalarının düzenlenmesi, güzergah planlamasına katkı sağlar.

Gereksinimler

1. Veri Tabanı:

SQL Server kullanılmış ve aşağıdaki tablolar oluşturulmuş:
(Yolcular,Otobusler,Sehirler,Otogarlar,Seferler,Rezervasyonlar,Odemeler,Bildirimler,Sikayetler,Soforler,Operatorler,GecisNoktalari)

2. Stored Procedures ve Triggers:

- a- Veri manipölasyonu ve otomasyon için kullanılmıştır.
- b-Commit ve rollback mekanizmalarıyla hata yönetimi.

3. ER Diyagramı:

Sistem tabloları ve ilişkilerini gösteren bir diyagram oluşturulmuştur.

4. Uygulama Testleri:

- a. Gerçek senaryolara uygun olarak prosedürler ve trigger'ler test edilmiştir.

Proje Aşamaları

1. Veri Tabanı Tasarımı:

Tablolar:

- **Yolcular:** Yolcu bilgilerini tutar.
- **Otobüsler:** Otobüs bilgilerini tutar.
- **Şehirler:** Şehir adlarını ve şehir bilgilerini tutar.
- **Otogarlar:** Şehirlerdeki otogar bilgilerini tutar.
- **Seferler:** Otobüs seferlerini kaydeder.
- **Rezervasyonlar:** Yolcuların otobüs seferlerine yaptığı rezervasyonları kaydeder.
- **Ödemeler:** Rezervasyonlara ait ödeme bilgilerini tutar.
- **Bildirimler:** Yolculara gönderilen bildirimleri tutar.
- **Şikayetler:** Yolcular tarafından iletilen şikayetleri kaydeder.
- **Şoförler:** Şoför bilgilerini tutar.
- **Operatörler:** Otogar operatörlerini tanımlar.
- **Geçiş Noktaları:** Sefer güzergahlarındaki ara durakları tanımlar.

Tablo Örnekleri:

Yolcular tablosu:

```
CREATE TABLE Yolcular (  
YolcuID INT IDENTITY(1,1) PRIMARY KEY ,  
Ad NVARCHAR(50) NOT NULL,  
Soyad NVARCHAR(50) NOT NULL,  
Eposta NVARCHAR(100) UNIQUE NOT NULL,  
Sifre NVARCHAR(255) NOT NULL,  
Telefon NVARCHAR(10),  
Adres NVARCHAR(255),  
Cinsiyet NVARCHAR(10)
```

```
);  
insert into Yolcular(Ad,Soyad,Eposta,Sifre,Telefon,Adres,Cinsiyet) values  
('Muhammed','Karaca','muhammed12@gmail.com','Muhammed12a','5314285466','Istan  
bul','Erkek');
```

Otobusler tablosu:

```
CREATE TABLE Otobusler (  
    OtobusID INT PRIMARY KEY IDENTITY(1,1),  
    PlakaNo NVARCHAR(9) UNIQUE NOT NULL,  
    Kapasite INT NOT NULL,  
    OtobusTuru NVARCHAR(10),  
    Model smallint,  
    Ozellikler NVARCHAR(100)  
);  
  
insert into Otobusler(PlakaNo,Kapasite,OtobusTuru,Model,Ozellikler) values  
('34XYZ123',44,'2+1','2020','Wifi,220V priz,Rahat koltuk');
```

Sehirler tablosu:

```
CREATE TABLE Sehirler (  
    SehirID INT PRIMARY KEY IDENTITY(1,1),  
    SehirAdi NVARCHAR(15) NOT NULL  
);  
  
insert into Sehirler(SehirAdi) values  
('Adana');
```

Otogarlar tablosu:

```
CREATE TABLE Otogarlar (  
    OtogarID INT PRIMARY KEY IDENTITY(1,1),  
    SehirID INT NOT NULL,  
    OtogarAdi NVARCHAR(60) NOT NULL,  
    FOREIGN KEY (SehirID) REFERENCES Sehirler(SehirID)  
);
```

```
INSERT INTO Otogarlar (SehirID, OtogarAdi) VALUES  
(1, 'AdanaOtogari');
```

Seferler tablosu:

```
CREATE TABLE Seferler (  
    SeferID INT PRIMARY KEY IDENTITY(1,1),  
    OtobusID INT NOT NULL,  
    KalkisOtogarID INT NOT NULL,  
    VarisOtogarID INT NOT NULL,  
    Kalkis_Tarihi DATETIME NOT NULL,  
    Varis_Tarihi DATETIME NOT NULL,  
    Fiyat DECIMAL(7, 2) NOT NULL,  
    Durum NVARCHAR(20),  
    FOREIGN KEY (OtobusID) REFERENCES Otobusler(OtobusID),  
    FOREIGN KEY (KalkisOtogarID) REFERENCES Otogarlar(OtogarID),  
    FOREIGN KEY (VarisOtogarID) REFERENCES Otogarlar(OtogarID)  
);
```

```
insert into  
Seferler(OtobusID,KalkisOtogarID,VarisOtogarID,Kalkis_Tarihi,Varis_Tarihi,Fiyat,Durum)  
values  
(1, 10, 20, '2025-01-06 08:00:00', '2025-01-06 12:30:00', 150.00, 'Planlandi');
```

Rezervasyonlar tablosu:

```
CREATE TABLE Rezervasyonlar (  
    Rez_ID INT PRIMARY KEY IDENTITY(1,1),  
    SeferID INT NOT NULL,  
    YolcuID INT NOT NULL,  
    KoltukNo INT NOT NULL,  
    Rez_Tarihi DATETIME NOT NULL,  
    YolculukTuru NVARCHAR(20),  
    Durum NVARCHAR(20),  
    FOREIGN KEY (SeferID) REFERENCES Seferler(SeferID),  
    FOREIGN KEY (YolcuID) REFERENCES Yolcular(YolcuID)
```

);

insert into Rezervasyonlar(SeferID,YolcuID,KoltukNo,Rez_Tarihi,YolculukTuru,Durum)
values

(1, 10, 15, '2025-01-01 12:30:00', 'Tek Yön', 'Onaylandi');

Odemeler tablosu:

```
CREATE TABLE Odemeler (  
    OdemeID INT PRIMARY KEY IDENTITY(1,1),  
    Rez_ID INT NOT NULL UNIQUE,  
    Od_Tarihi DATETIME NOT NULL,  
    Od_Yontemi NVARCHAR(20),  
    Durum NVARCHAR(20),  
    FOREIGN KEY (Rez_ID) REFERENCES Rezervasyonlar(Rez_ID)  
);
```

insert into Odemeler(Rez_ID, Od_Tarihi, Od_Yontemi, Durum) values
(1, '2025-01-01 10:30:00', 'Kredi Kartı', 'Onaylandi');

Bildirimler tablosu:

```
CREATE TABLE Bildirimler (  
    BildirimID INT PRIMARY KEY IDENTITY(1,1),  
    YolcuID INT NOT NULL,  
    SeferID INT NOT NULL,  
    Baslik NVARCHAR(50) NOT NULL,  
    MesajIcerigi NVARCHAR(MAX),  
    Okundu BIT NOT NULL DEFAULT 0,  
    BildirimTuru NVARCHAR(30),  
    FOREIGN KEY (YolcuID) REFERENCES Yolcular(YolcuID),  
    FOREIGN KEY (SeferID) REFERENCES Seferler(SeferID)  
);
```

insert into Bildirimler(YolcuID, SeferID, Baslik, MesajIcerigi, Okundu, BildirimTuru)
values

(1, 1, 'Sefer Bilgisi Güncellendi', 'Sefer saatinde değişiklik olmuştur. Lütfen kontrol ediniz.', 0, 'Saat Değişikliği');

Sikayetler tablosu:

```
CREATE TABLE Sikayetler (  
    SikayetID INT PRIMARY KEY IDENTITY(1,1),  
    YolcuID INT NOT NULL,  
    SeferID INT NOT NULL,  
    Konu NVARCHAR(100) NOT NULL,  
    Aciklama NVARCHAR(MAX),  
    Tarih DATETIME NOT NULL,  
    Durum NVARCHAR(20),  
    FOREIGN KEY (YolcuID) REFERENCES Yolcular(YolcuID),  
    FOREIGN KEY (SeferID) REFERENCES Seferler(SeferID)  
);
```

```
insert into Sikayetler(YolcuID, SeferID, Konu, Aciklama, Tarih, Durum) values  
(51, 51, 'Bavulum Kayboldu', 'Bavulum baska yolcu ile karisilmis en yakin zamanda  
sorunum cozulsun lutfen', '2025-02-21 10:00:00', 'Cozulmus');
```

Soforler tablosu:

```
CREATE TABLE Soforler (  
    SoforID INT PRIMARY KEY IDENTITY(1,1),  
    Ad NVARCHAR(30) NOT NULL,  
    Soyad NVARCHAR(30) NOT NULL,  
    Telefon NVARCHAR(10),  
    EhliyetNo NVARCHAR(15) UNIQUE NOT NULL,  
    CalismaDurumu NVARCHAR(20),  
    DeneyimYili smallint  
);
```

```
INSERT INTO Soforler (Ad, Soyad, Telefon, EhliyetNo, CalismaDurumu, DeneyimYili)  
VALUES  
( 'Ahmet', 'Yilmaz', '5311234567', '12345689012345', 'Aktif', 5),
```

Operatorler tablosu:

```
CREATE TABLE Operatorler (  
    OperatorID INT PRIMARY KEY IDENTITY(1,1),  
    OtogarID INT NOT NULL,  
    Ad NVARCHAR(30) NOT NULL,  
    Soyad NVARCHAR(30) NOT NULL,  
    Telefon NVARCHAR(10),  
    CalismaDurumu NVARCHAR(20),  
    FOREIGN KEY (OtogarID) REFERENCES Otogarlari(OtogarID)  
);
```

```
insert into Operatorler(OtogarID, Ad, Soyad, Telefon, CalismaDurumu) values  
(1, 'Ahmet', 'Yilmaz', '5311234567', 'Aktif');
```

GecisNoktalari tablosu:

```
CREATE TABLE GecisNoktalari (  
    GecisID INT PRIMARY KEY IDENTITY(1,1),  
    SeferID INT NOT NULL,  
    SehirID INT NOT NULL,  
    GecisSirasi INT NOT NULL,  
    Kalkis_Saati DATETIME,  
    Varis_Saati DATETIME,  
    FOREIGN KEY (SeferID) REFERENCES Seferler(SeferID),  
    FOREIGN KEY (SehirID) REFERENCES Sehirler(SehirID)  
);
```

```
INSERT INTO GecisNoktalari (SeferID, SehirID, GecisSirasi, Kalkis_Saati, Varis_Saati)  
VALUES  
(1, 34, 1, '2025-01-01 08:00:00', '2025-01-01 09:00:00');
```

2. Stored Procedures ve Kullanım Senaryoları

Otobüs bilgileri ekleme:

```
CREATE PROCEDURE sp_InsertOtobus
```

```
@PlakaNo NVARCHAR(9),
@Kapasite INT,
@OtobusTuru NVARCHAR(10) = NULL,
@Model SMALLINT = NULL,
@Ozellikler NVARCHAR(100) = NULL
AS
BEGIN
    INSERT INTO Otobusler (PlakaNo, Kapasite, OtobusTuru, Model, Ozellikler)
    VALUES (@PlakaNo, @Kapasite, @OtobusTuru, @Model, @Ozellikler);
END;
```

3. Trigger ve Kullanım Senaryoları

Örnek: Aynı seferde aynı koltuk numarasının birden fazla rezervasyon için kullanılmasını engeller.

```
CREATE TRIGGER trg_RezervasyonKoltukKontrol
ON Rezervasyonlar
AFTER INSERT, UPDATE
AS
BEGIN
    IF EXISTS (
        SELECT SeferID, KoltukNo, COUNT(*)
        FROM Rezervasyonlar
        GROUP BY SeferID, KoltukNo
        HAVING COUNT(*) > 1
    )
    BEGIN
        RAISERROR ('Bu koltuk zaten rezerve edilmiş!', 16, 1);
        ROLLBACK;
    END
END;
```

Test senaryosu:

```
INSERT INTO Seferler (OtobusID, KalkisOtogarID, VarisOtogarID, Kalkis_Tarihi,
Varis_Tarihi, Fiyat, Durum) VALUES (1, 1, 2, '2025-01-10 10:00:00', '2025-01-10
14:00:00', 150.00, 'Aktif');
```

4. Commit ve Rollback Testi:


```
BEGIN TRY
    BEGIN TRANSACTION;

    INSERT INTO Rezervasyonlar (SeferID, YolcuID, KoltukNo, Rez_Tarihi,
YolculukTuru, Durum)
    VALUES (1, 1, 10, GETDATE(), 'Tek Yön', 'Onaylandı');

    DECLARE @RezID INT = SCOPE_IDENTITY();

    COMMIT TRANSACTION;
    PRINT 'Rezervasyon ve ödeme işlemleri başarıyla tamamlandı.';
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION;

    PRINT 'Hata oluştu: ' + ERROR_MESSAGE();
END CATCH;
```

Beklenen Çıktı:

- İşlem sırasında bir hata oluşur (örneğin, aynı koltuk numarasına rezervasyon yapılmaya çalışıldığı için).
- ROLLBACK TRANSACTION işlemi devreye girer ve hiçbir değişiklik yapılmaz.
- Hata mesajı görüntülenir