

Bilet Rezervasyon Sistemi Projesi

*Bilgisayar Mühendisliği
Kocaeli Üniversitesi*

Yaren Aybala KOŞAR
220201023
yarenaybala2004@hotmail.com

Mediha Tuğba KEYİK
220201044
medihatugbakeyik@gmail.com

I. ÖZET

Bir girişimci yolcuların farklı ulaşım firmalarından ya da şirketlerinden rezervasyon yapabildiği bir sistem kurmak istemektedir. Ulaşım şirketleri demiryolu, karayolu ve havayolu seçenekleri ile yolcu taşımaktadır. Bazı firmalar birden fazla seçenekte (havayolu ve karayolu gibi) hizmet verebilir.

Projeye başlarken öncelikle bize verilen ve bizden istenilen verileri analiz etmeye çalıştık. Projeyi yazacağımız programlama dili olan C# için uygun IDE araştırması yaptık. Daha sonra aşamaları kendi içinde parçalayarak adım adım neler yapabileceğimiz üzerinde konuştuk. Kodlama kısmına geçtiğimizde ise ilerlemek fazlasıyla zor oldu çünkü hiç bilmediğimiz bir dilde daha yeni öğrenmeye başladığımız bir yapıda (OOP) kapsamlı ve karmaşık bir projeyi çözümlmek hiç deneyimlemediğimiz şeylerdi. Ama zorlu ve uzun süren uğraşlar sonucu projede epey yol kat ettik. Bu raporda da yaptığımız projeyi her detayıyla paylaşıyor olacağız.

II. GİRİŞ

C# dilinde yazmaya karar verdiğimiz projenin IDE'si için Visual Studio tercih ettik.

Programın ana fonksiyonları şunlardır:

- Admin, company ve customer olarak farklı ekranların ve işlemlerin tanımlanması
- Admin ekranında firma ekleme, çıkarma, görüntüleme fonksiyonları
- Firma panelinde araç ve sefer ekleme, çıkarma; günlük kar hesabının gösterilmesi
- Kullanıcı panelinde bilet arama ve rezerve etme

III. YÖNTEM VE İLERLEYİŞ

Öncelikle yapacağımız işleri adım adım bölüp kendi aramızda bölüştük. Öncelikle birimiz (Yaren Aybala Koşar) formları oluştururken diğeri (Mediha Tuğba Keyik) class yapılarını hazırladı daha sonra ilerlemelerimizi birbirimizle paylaştık ve birbirimizin yaptığı-yapamadığı kısımları paslaşarak ilerlemeye ve çıkan sorunları çözmeye çalıştık bu nedenle kodumuzun tamamına ortak bir çalışmanın eseri diyebiliriz.

İlk olarak giriş ekranını tasarlayarak başladık. Bir yandan da user class'ının yazımını yaptık. Class'ta tanımladığımız login ve CheckCredentials gibi bazı methodları giriş formunda override ettik. Burada giriş yapma özelliği için ILoginable ara yüzünü kullandık. Burada giriş kontrolü için kullanıcı adı ve şifreyi listedeki eşleşmesini kontrol ederek yaptık. Giriş yapanın firma mı yoksa admin mi olduğuna göre ekranların açılmasını sağladık.

Admin ekranında firma ekleme, güncelleme ve silme butonlarının işlevlerini kodladık. Verileri listede tuttuğumuz için firma ve admin ekranı sayfalarına anasayfaya dönüş butonu ekledik. Admin ekranında firmalar ve şifrelerinin görüntülenmesini sağladık.

Firma ekranına giriş yapıldığında giriş yapan firmanın araç ekleme, düzenleme ve silme butonlarını ana sayfaya dönme butonunu ve günlük karı görüntüleme butonu ve labelını tanımladık. Bu ekranda araçları ve seferlerini gösteriyoruz.

Giriş ekranının 3. ve son seçeneği olan bilet satın al butonunda müşterinin bilet arayabilmesi için kullanıcı ekranı formuna yönlendirme sağlıyoruz.

Kullanıcı ekranı formunda müşteriden kalkış ve varış noktalarını, yolculuğu yapacağı tarih ve kişi sayısı bilgilerini girmesini ardından seçtiği filtrelemelere uygun olarak yandaki listviewde görüntülenen seferlerden birini tercih etmesi ve tıklaması istenmektedir.

Sonrasında açılan koltuksecimi ekranında kişi sayısı kadar koltuk seçmelidir. Uygun sayıda koltuk seçtikten sonra açılan BiletSatınAl formuna istenen bilgiler girildikten sonra Ödeme yap butonuna basarak satın alma işlemi tamamlanır.

IV. KULLANDIĞIMIZ SINIF YAPILARI, ARA YÜZLER VE METOTLAR

IProfitable Arayüzü: IProfitable Interface, günlük kar hesaplaması yapabilen sınıflar için bir şablondur ve yalnızca CalculateDailyProfit adlı bir metodu içerir.

ILoginable Arayüzü: Bu arayüz, giriş işlemlerini tanımlar ve **Login** metodunu içerir.

IReservable Arayüz: IReservable, iki yöntem içerir: **IsSeatAvailable** ve **ReserveSeat**. Bu arayüzü uygulayan sınıfların bu yöntemleri uygulaması beklenir.

IsSeatAvailable yöntemi, belirli bir koltuk numarasının kullanılabilir olup olmadığını kontrol eder.

ReserveSeat yöntemi, bir yolcu için bir koltuk rezerve etmekten sorumludur ve yolcu bilgilerini, koltuk numarasını, tarihi (tarih), kalkışı (kalkış) ve varışı (varış) parametrelerini alır.

Company Sınıfı: Company sınıfı, User sınıfından miras alınmış bir sınıftır, bu nedenle kullanıcı adı ve şifre özelliklerine erişebilir. Ayrıca, IProfitable arayüzünü uygular, bu sayede günlük kar hesaplaması yapılabilir. Şirketin adını temsil eden CompanyName özelliği bulunurken, loggedInCompany ise giriş yapan şirketi tutan bir static özelliktir. VehicleFuelPrices, araç yakıt fiyatlarını saklayan bir

sözlüktür ve TransportList şirkete ait taşıma araçlarını içeren bir listeyi temsil eder. Companies, şirket nesnelerini içeren bir listedir ve bu liste örnek şirketlerle başlatılmıştır.

Company sınıfının metodları arasında **CalculateDailyProfit()** günlük karı hesaplamak için kullanılırken, **CalculateTotalPersonnelExpenses()** şirketin toplam personel giderlerini hesaplar. **SetVehicleFuelPrice()** bir aracın yakıt fiyatını ayarlamak için kullanılır, **GetVehicleFuelPrice()** ise belirli bir aracın yakıt fiyatını almak için kullanılır. **Login()** kullanıcı girişi yapar ve giriş başarılı ise ilgili şirketi belirler. **AddTransport(Transport newTransport)** ise yeni bir taşıma aracını şirketin taşıma araçları listesine ekler.

Customer Sınıfı: Customer sınıfı, private List<Reservation> rezervasyonListesi özelliği ile müşterinin yapmış olduğu rezervasyonları tutar. **AddReservation(Reservation reservation)** metodu, müşteriye ait rezervasyon listesine yeni bir rezervasyon eklemek için kullanılır. **RemoveReservation(Reservation reservation)** metodu, müşteriye ait rezervasyon listesinden belirli bir rezervasyonu kaldırmak için kullanılır. **GetReservations()** metodu ise müşteriye ait tüm rezervasyonları içeren bir listeyi döndürür.

Person Sınıfı: Person sınıfı soyut bir sınıftır ve Ad-Soyad özelliklerini içerir. Personel ve Passenger sınıfları bu temel sınıftan miras alır ve kendi kurucu metodlarına sahiptirler.

Personel Sınıfı: Personel sınıfı, Person sınıfından miras alır ve Departman ile Firma özelliklerini içerir. **Personel()** kurucu metodu, ad, soyad, departman ve firma parametreleri ile bir personel nesnesi oluşturur. **GetPersonnelFees(string companyName, string vehicleId)** metodu, personelin maaşını hesaplar.

Passenger Sınıfı: Passenger sınıfı, Person sınıfından miras alır ve Tckn, DogumTarihi, TelefonNo özelliklerini içerir. **Passenger(string ad, string soyad, string tckn, DateTime dogumtar, string telno)** kurucu metodu, ad, soyad, TCKN, doğum tarihi ve telefon numarası parametreleri ile bir yolcu nesnesi oluşturur.

Reservation Sınıfı: Bu sınıf, bir rezervasyonun özelliklerini temsil eder. **Passenger** (Yolcu), **Vehicle** (Araç), koltuk numarası (**SeatCount**), rezervasyon tarihi (**Tarih**), kalkış (**Kalkis**) ve varış (**Varis**) bilgilerini içerir. Ayrıca, **GetReservationDetails** metodunu kullanarak rezervasyonun detaylarını alabilir.

Route Sınıfı: Güzergâh adını (**Name**) ve durak listesini (**Stops**) içerir. **ContainsStops** metodu ile verilen iki durak adının bu güzergâh üzerinde bulunup bulunmadığını kontrol eder. **InitializeRoutes** metodu sabit güzergahları oluştururken, **CalculateTicketPrice** metodu bilet fiyatını hesaplamak için kullanılır.

Transport Sınıfı: Taşıma araçlarını temsil eder ve **VehicleId**, **FuelType**, **SeatCount**, **CompanyName**, **SeferNo**, **Kalkis**, **Varis**, **Tarih**, **isdriver**, **reservedSeats** gibi özelliklere sahiptir. Metotları arasında; toplam fiyatı hesaplamak için **CalculateTotalPrice**, bilet fiyatını hesaplamak için **CalculateTicketPrice**, personel giderlerini hesaplamak için **CalculatePersonnelExpenses**, yakıt fiyatını ayarlamak için **SetFuelPrice**, yakıt fiyatını almak için **GetFuelPrice**, belirli bir koltuğun rezerve edilip edilmediğini kontrol etmek için **IsSeatAvailable**, rastgele rezerve edilmiş koltuk sayısı üretmek için **GenerateRandomReservedSeats**, koltuğu rezerve etmek için **ReserveSeat** gibi metotlar bulunur.

Trip Sınıfı: Şirket adını (**CompanyName**), taşıma aracı kimliğini (**VehicleId**), koltuk sayısını (**SeatCount**) ve sefer numarasını (**SeferNo**) içerir. **CreateTripList** metodu, verilen taşıma araçları listesinden Trip nesnelerini oluşturur.

User Soyut Temel Sınıfı: Kullanıcı bilgilerini temsil eder ve **Username**, **Password** özelliklerine sahiptir. **CheckCredentials** metodu, kullanıcı adı ve şifrenin doğruluğunu kontrol eder. Soyut olan **Login** metodu, türetilen sınıflar tarafından implemente edilmelidir.

Admin Sınıfı: User sınıfından türetilmiş bir sınıftır ve **AdminRole** özelliğini içerir. **CheckAdminLogin** metodu, girilen kullanıcı adı ve şifrenin adminin bilgileriyle eşleşip eşleşmediğini kontrol eder.

Override edilmiş **Login** metodu, giriş bilgilerini kontrol eder ve doğruysa bir AdminEkranı nesnesi oluşturarak admin ekranını gösterir.

Vehicle Soyut Temel Sınıfı: Ulaşım araçlarını temsil eder ve **VehicleName**, **FuelType**, **SeatCount**, **Company**, **FuelPrices** gibi özelliklere sahiptir. **SetFuelPrice** metodu, yakıt türü için fiyatı ayarlar. **CalculateFuelCost** soyut metodu, yakıt maliyetini hesaplamak için alt sınıflar tarafından implemente edilmelidir.

Bus, Train, Airplane Alt Sınıfları: Her biri, Vehicle sınıfından türetilmiş üç farklı ulaşım aracı sınıfını temsil eder. **CalculateFuelCost** metodu ile belirli bir sefer için yakıt maliyetini hesaplar.

Program Sınıfı:

ApplicationConfiguration.Initialize(); Bu metod, ApplicationConfiguration adlı bir sınıfın **Initialize** metodunu çağırarak uygulama yapılandırmasını başlatır. Bu metodun içeriği, uygulamanın yapılandırma ayarlarını başlatmak, okumak veya ayarlamak için gerekli olan işlemleri gerçekleştirir.

Application.Run(new giris()); Bu satırda, Application sınıfının Run metoduna bir giris sınıfının örneği (instance) verilerek uygulamanın çalıştırılması başlatılır.

V. KATKILAR

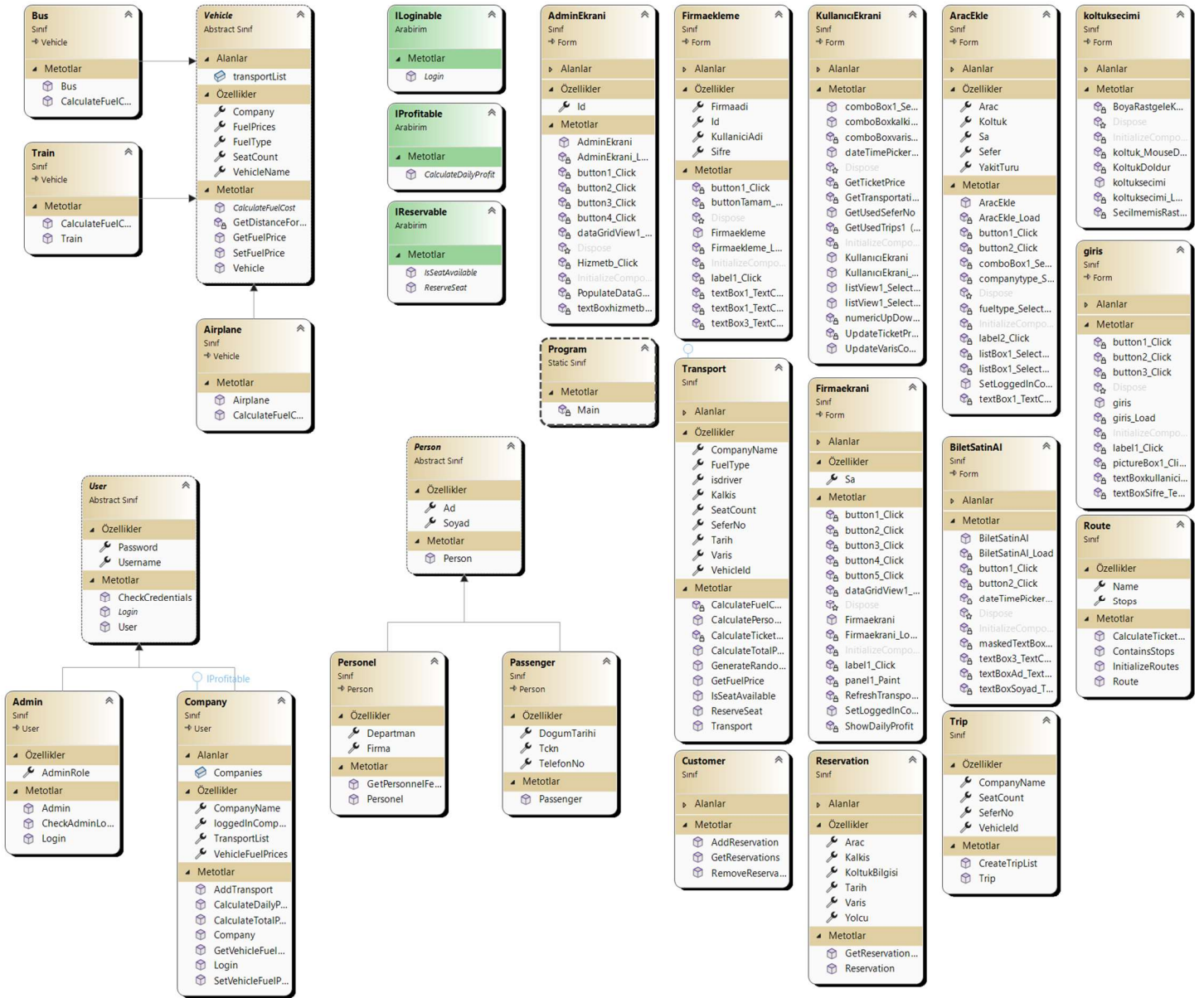
Birbirimizin eksikliğini kapatarak ilerlediğimiz için hangi bölümü hangimizin yaptığını söylemek çok mümkün değil ancak genel hatları itibariyle formlar ile ilgili işlemler (Yaren Aybala Koşar), classlar ile ilgili işlemler (Mediha Tuğba Keyik) tarafından yapıldı.

VI. SONUÇ

Sonuç olarak yazdığımız kodlar ve tasarladığımız formlar ile projenin birçok gerekliliğini sağlayan bir bilet rezervasyon projesi ortaya çıkardık.

KAYNAKLAR

- [1] <https://github.com/uluturkzeynep/Otel-Rezervasyon-Sistemi>
- [2] <https://mustafabukulmez.com/2020/06/20/c-formlar-arasi-veri-aktarimi/#close>
- [3] https://www.youtube.com/watch?v=9LJpavsw_R4
- [4] <https://mcansozeri.wordpress.com/2010/04/22/c-sharp-tur-donusumu-amator-anlatim/>
- [5] <https://mustafabukulmez.com/2020/10/23/c-otobus-bilet-satis-uygulamasi-basit-duzey/>
- [6] <https://github.com/halilozel1903/OtelOtomasyonu>
- [7] <https://github.com/erentrhn07/ucak-rezervasyon-sistemi>
- [8] <https://mustafabukulmez.com/2020/10/23/c-otobus-bilet-satis-uygulamasi-basit-duzey/>
- [9] <https://www.udemy.com/course/otelrezervasyon/>
- [10] https://csharp.hotexamples.com/examples/Otel_Rezervasyon_Sistemi.Formlar/MusteriRezervasyonlar/-/php-musterirezervasyonlar-class-examples.html
- [11] <https://sudedalyan.medium.com/c-ile-restoran-otomasyon-sistemi-c8dfe8b9f589>
- [12] <https://github.com/mrabdullahsahin/Otobus-Bilet-Otomasyonu>
- [13] https://github.com/BinaryLeo/csharp_signup_signup_system
- [14] <https://barisuslucan.com.tr/yazilim/c-dilinde-otobus-otomasyon-uygulamasi/>
- [15] <https://sinandogukanarslan.blogspot.com/2014/08/c-otobus-otomasyonu.html>
- [16] https://www.youtube.com/watch?v=tZIIgMP_nLk
- [17] <http://www.gorselprogramlama.com/otobus-bileti-satis-takip-programi-c-netcnet/>
- [18] <https://www.projevekod.com/2019/03/c-sharp-csharp-basit-otobus-dolmus-rezervasyon-otomasyon-sql-ornek-ornegi.html?m=1>
- [19] <https://www.youtube.com/watch?v=HU6Xrkne4tE>



-UML DİYAGRAMI-