

Uçak, Otobüs, Tren Rezervasyon Sistemi

Programlama Laboratuvarı 2

1. Çağla GÖK
Bilgisayar Mühendisliği
Kocaeli Üniversitesi
220201060

2. Selma YILDIZ
Bilgisayar Mühendisliği
Kocaeli Üniversitesi
220201091

I. ÖZET

Bu projede, bir girişimci için farklı ulaşım firmaları arasında rezervasyon yapılabilmesini sağlayan bir sistem oluşturmayı amaçlıyoruz. Ulaşım şirketleri, demiryolu, karayolu ve havayolu seçenekleriyle yolcu taşıma hizmeti sunmaktadır. Geliştirdiğimiz sistem, admin, firma ve kullanıcı panellerini içermekte olup her bir panelin kendine özgü yetenekleri bulunmaktadır.

II. GİRİŞ

Bu projede, C# programlama dilinin sağladığı avantajları kullanarak, kullanıcılar, firmalar, araçlar, seferler ve rezervasyonları temsil eden sınıfların tasarımını gerçekleştirdik. Ayrıca, Windows Forms teknolojisi kullanarak bir arayüz oluşturduk ve bu arayüz üzerinden rezervasyon işlemlerini gerçekleştirmeyi mümkün kıldık.

Programın temel unsurları şunlardır:

- Admin paneli, mevcut firmaların görüntülenmesini sağlar, yeni firma kayıtları alabilir, firma kayıtlarını silebilir ve hizmet bedelini belirleyebilir.
- Firma paneli, araç ve sefer ekleme/çıkarma gibi işlemleri gerçekleştirebilir, ayrıca günlük kâr hesaplamalarını görebilir.
- Kullanıcı paneli üzerinden bilet arama, rezervasyon yapma ve ödeme işlemleri gibi kullanıcı odaklı işlemler gerçekleştirilebilir.

III. YÖNTEM

Main metodu içerisinde FormBaslangic başlatılır.

FormBaslangic: Burada bulunan iki butondan birisi 'Giriş Yap' butonu, firma ve admin için FormGirisEkrani'ni açarken diğeri ise 'Bilet Bul' butonu, müşteri için FormBiletBul formunu açıyor.

FormGirisEkrani: Burada bulunan GirisKontrol adlı class'ta admin ve firmalar için kullanıcı adı ve şifre bilgileri tutuluyor. Bu formda bulunan 'Giriş Yap' butonuna basıldığında formdaki textbox'lara yazılan kullanıcı adı ve şifre ile GirisKontrol'deki bilgiler karşılaştırılıyor. Bu bilgiler hangi kullanıcıya aitse o kullanıcıya ait form açılıyor. Eğer GirisKontrol'deki bilgilerle eşleşmezse hata mesajı veriliyor.

FormAdmin: Bu formun yapılandırıcısı içerisinde, Company class'ında tanımladığımız ve firmalara ait bilgiler bulunan companyList adlı listedeki bilgiler admin formunda gösteriliyor. 'Firma Ekle' butonuna tıklandığında FormAdminFirmaEkleme formu açılıyor. Bu formda eklenecek firmaya ait bilgilerin girilmesi isteniyor. Bu bilgiler girildikten sonra admin formunda gösteriliyor. 'Firma Sil' butonuna tıklandığında seçili firma siliniyor. 'Firma Güncelle' butonuna tıklandığında seçili firma için güncellenebilecek bilgiler gösteriliyor. 'Hizmet Bedeli Değiştir' butonu ise girilen hizmet bedelini güncelliyor.

FormAFirma: Bu formun yapılandırıcısında bulunan switch-case içerisinde kullanıcı adı bilgisine göre her bir firmaya ait araç ve sefer bilgileri ekranda görüntüleniyor. Ayrıca bu case'lerde Bus, Airplane ve Train class'larında tanımladığımız metodların çağrısı yapılarak her firma için yakıt gideri hesaplanıyor. Sonra Personel class'ında

tanımladığımız bir metodun çağrısı yapılarak her firma için personel gideri hesaplanıyor. Bu iki değer 'zarar' adlı değişkene atanıyor. Aynı şekilde yapılandırıcı içerisinde Transport class'ında tanımladığımız bir metottan her firma için bilet satışından kazanılan miktar 'toplamKazanc' adlı dizinin elemanlarına atanıyor. Her case'de kazançlardan zarar ve hizmet bedeli çıkarılarak 'kar' adını verdiğimiz değişkene atanıyor.

'Araç Ekle' butonuna basıldığında FormAracEkleme formu açılıyor. Bu formda girilen araç türüne göre belirli seferler gösteriliyor. Sonra alınan bu bilgiler diğer formda gösteriliyor. Eklenen yeni araç Transport class'ında bulunan listelere firma türüne bağlı olarak ekleniyor. 'Araç Sil' butonuna basıldığında seçilen araç bilgileri siliniyor. 'Araç Güncelle' butonuna basıldığında ise tekrar FormAracEkleme formu açılarak güncellenecek bilgiler alınıyor. 'Günlük Kârınızı Görmek İçin Tıklayınız' butonuna basıldığında daha önce yapılandırıcı içerisinde hesaplanan 'kar' değeri gösteriliyor.

FormBiletBul: Bu formda müşterinin girdiği bilgiler birer değişkene atanarak Trip class'ında bulunan bir metoda gönderiliyor. Buradan dönen liste FormSeferBilgi formunda müşteriye gösteriliyor.

FormSeferBilgi: Bu formda bir Dictionary tanımlanıyor. Formun yapılandırıcısında bu Dictionary'ye bilgiler atanıyor. Sonrasında kullanıcının seçtiği firma ve araç türüne göre bir switch-case çalışıyor. Her case'de koltuk sayısını parametre alan bir metod çağrısı yapılıyor. Bu metotlar müşteriye gösterilecek aracın içindeki koltukları rastgele atanan değerlere göre seçip maviye boyuyor. Maviye boyanan koltuklar rezerve edilmiş kabul edilip müşterinin seçimine kapatılıyor. Müşteri girdiği kişi sayısından fazla kişi için bilet alamıyor. Müşteri almak istediği koltuğu seçtikten sonra FormKullaniciBilgi formu açılıyor ve bu koltukta seyahat edecek yolcunun bilgileri alınıyor. Bu bilgilerle beraber yapılandırıcıda içerisini doldurduğumuz Dictionary'deki metotlar içerisinde çekilen bilet fiyatı bilgisi müşteriye tabloda gösteriliyor. Sonrasında bu bilgiler Reservation ve Passenger class'larında tanımlanan listelere ekleniyor. 'Ödeme Yap' butonuna basıldığında toplam ücret ekrana yazdırılıp işlem başarıyla bitiriliyor.

ILoginable: KullaniciAdi, KullaniciSifresi değişkenlerini ve Login() metodunu içeren bir Interface'tir.

IProfitable: Sefer adında bir class ve gunlukYakitGideri() adlı metoda sahip bir Interface'tir.

IReservable: reservable() adında bir metoda sahip bir Interface'tir.

User: ILoginable'dan implement edilen Login() metoduna, KullaniciAdi, KullaniciSifresi adlı değişkenlere ve bir yapılandırıcıya sahip Abstract Class'tır.

Vehicle: Yakıt, Firma, SeferNo, KoltukSayisi adlı değişkenlere ve CalculateFuelCost() adlı metoda sahip bir Abstract Class'tır.

Person: Ad, Soyad adlı değişkenlere ve bir yapılandırıcıya sahip bir Abstract Class'tır.

Admin: User Class'ından türetilen, Login() metodunu kullanan ve bir yapılandırıcıya sahip olan bir Class'tır.

Company: IProfitable'den implement edilen, User Class'ından türetilen, companyList, firmaListesi, yakıt listeleri bulunan, çeşitli metod ve değişkenleri bulunan bir Class'tır.

Airplane: Vehicle'den türetilmiş, CalculateFuelCost() metodunu override eden bir Class'tır. CalculateFuelCost() metodunda Company Class'ından çekilen firmaListesi adlı listenin içerisinde dolaşılır. Bu metoda parametre olarak gönderilen değişkenle eşleşen bir liste elemanı bulunduğunda bu elemandan gelen değerlerle, her firma için günlük yakıt maliyeti hesaplanır ve geri döndürülür.

Bus: Vehicle'den türetilmiş, CalculateFuelCost() metodunu override eden bir Class'tır. CalculateFuelCost() metodunda Company Class'ından çekilen firmaListesi adlı listenin içerisinde dolaşılır. Bu metoda parametre olarak gönderilen değişkenle eşleşen bir liste elemanı bulunduğunda bu elemandan gelen değerlerle, her firma için günlük yakıt maliyeti hesaplanır ve geri döndürülür.

Train: Vehicle'den türetilmiş, CalculateFuelCost() metodunu override eden bir Class'tır. CalculateFuelCost() metodunda Company Class'ından çekilen firmaListesi adlı listenin içerisinde dolaşılır. Bu metoda parametre olarak gönderilen değişkenle eşleşen bir liste elemanı

bulduğunda bu elemandan gelen değerlerle, her firma için günlük yakıt maliyeti hesaplanır ve geri döndürülür.

Customer: Nereden, Nereye, BiletTarihi ve Ad isimli değişkenlere sahip bir Class'tır.

Passenger: Person Class'ından türetilmiş, içerisinde çeşitli değişkenler ve passengerlist adında liste içeren bir Class'tır.

Personel: Person Class'ından türetilmiş, içerisinde çeşitli değişkenler, PersonelListesi() ve ToplamPersonelUcreti() adında metotlar bulunan bir Class'tır. PersonelListesi() metodu içerisinde her firmaya ait personel bilgilerini tutan PersonelBilgi adındaki listeyi geri döndürür. ToplamPersonelUcreti() metodu ise bir string parametresi olarak bu parametreyi personelListesi'ndeki değerlerle karşılaştırarak her firmanın toplam personel ücretini geri döndürür.

Reservation: İçerisinde çeşitli değişkenler ve reservation adında liste içeren bir Class'tır.

Route: Çeşitli değişkenler ve rotalistesi() adında metot içeren bir Class'tır. rotalistesi() metodu içerisinde routes adında tanımlanmış listeyi geri döndürür.

Transport: IReservable'dan implement edilen, içerisinde çeşitli değişkenler, her firmaya ait bilgilerin bulunduğu farklı listeler ve ToplamFiyatHesapla() adında metoda sahip bir Class'tır. ToplamFiyatHesapla() metodunda routeList adındaki listede dolaşarak her firmanın bilet satışından kazandığı miktarı bir dizi olarak geri döndürür.

Trip: KullanıcıBilgisi, Sefer, BiletFiyatı adında Class'lara, UygunSeferiBul(), dondur() adlı metotlara sahip bir Class'tır. dondur() metodu içerisinde tanımlanan seferler adlı listeyi geri döndürür. UygunSeferiBul() metodu ise müşterinin girdiği biniş ve iniş şehirlerini içeren seferleri uygunSeferler adlı listeye ekleyip bu listeyi geri döndürür.

IV. KATKILAR

Ekip çalışmasının çok önemli olduğunun farkında olduğumuz için her bir aşamada fikir alışverişinde ve sürekli etkileşimde bulunarak projeyi birlikte tamamladık.

V. SONUÇ

İsterlerden:

Admin paneli başarıyla açılıyor. Bu panelde firma ekleme, çıkarma ve güncelleme yapılarak tabloda gösteriliyor. Hizmet bedeli güncellemesi gerçekleştiriliyor.

Firma paneli başarıyla açılıyor. Bu panelde araç ekleme, çıkarma ve güncelleme yapılarak tabloda gösteriliyor. Her firma günlük kâr miktarını görüntüleyebiliyor.

Kullanıcı paneli herhangi bir üyelik gerektirmeden başarıyla açılıyor. Bilet arama, araç seçme ve ödeme işlemleri başarıyla gerçekleştiriliyor. Rezervasyon bilgileri kullanıcıya gösteriliyor.

VI. KAYNAKÇA

- [1] https://github.com/ahozturk/Backend_Lecture_Materials/blob/main/Lecture_3/Freelancer/Abstract/ICsvConvertible.cs
- [2] <https://www.w3schools.com/cs/index.php>
- [3] https://www.youtube.com/playlist?list=PLVlwRtag6_0s8bdOygxKKZG2ZHXYJRHE
- [4] https://www.youtube.com/playlist?list=PLx7xIXM0VldsDdDgq_tL4B1tdSV7gE78MI
- [5] <https://github.com/saitorhan/OtobusOtomasyon/blob/master/Form1.cs>
- [6] <https://www.youtube.com/watch?v=-gN2cDjxNFs>
- [7] <https://www.youtube.com/watch?v=6PBwilWymIQ>
- [8] <https://github.com/esramazan/HastaneOtomasyonu/blob/main/Hastane/B%C3%B6lEkle.Designer.cs>
- [9] <https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/compiler-messages/>
- [10] <https://www.youtube.com/watch?v=-bD6qUKC8HI>
- [11] <https://www.youtube.com/watch?v=9DD1-lbGVB0>
- [12] <https://www.youtube.com/watch?v=-bD6qUKC8HI&t=1222s>
- [13] <https://www.youtube.com/watch?v=jXiOLIGqc-M&list=PL22TK-6CtDafPHI91N2Swcedg56L9SknO&index=12>
- [14] <https://stackoverflow.com/questions/761194/interface-vs-abstract-class-general-oo>
- [15] <https://stackoverflow.com/questions/572547/what-does-static-mean-in-c>
- [16] <https://stackoverflow.com/questions/5601777/constructor-of-an-abstract-class-in-c-sharp>
- [17] <https://stackoverflow.com/questions/2705163/c-abstract-classes-need-to-implement-interfaces>
- [18] <https://learn.microsoft.com/tr-tr/visualstudio/ide/default-keyboard-shortcuts-in-visual-studio?view=vs-2022>

VII. UML DİYAGRAMI



