

Rezervasyon Sistemi (Türkçe)

1. Ali Can Gündüz
Bilgisayar Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
ac.gunduz7878@gmail.com

Özetçe—Kullanıcıların kolay bir şekilde sefer arayarak rezervasyon yapmasını sağlayan bir sistem geliştirilmiştir. Kullanıcılar seferleri görüntüleyebilir ve tarih kalkış noktası ve hedef noktasını kullanarak görmek istedikleri seferleri listeleyebilirler. Seferlerin tarihleri, aracı, şirketi gibi veriler kullanıcıya gösterilir. Kullanıcı istediği seferin üstüne tıklayarak seçimini yapar ve araçtaki koltuk bilgilerinin gözüktüğü bir ekrana gelir. Bu ekranda sadece boş koltuklar için seçim yapabilir ve yolcu sayısınca koltuk seçimi yapar. Sonrasında ise her seçilen koltuk için yolcu bilgileri istenir. Bu bilgiler isim, soyisim, Tc numarası ve doğum tarihidir. Bu işlemten sonra kullanıcıya rezerve ettiği koltuk bilgileri, sefer bilgileri ve araç bilgileri gösterilir. Ayrıca admin ve şirket sayfaları da vardır ve bu sayfalara giriş için bir login ekranı vardır. Bilgilerin doğru olması sonrasında admin ekranında var olan şirketler görüntülenebilir ve silinebilir, aynı zamanda yeni şirket de eklenilebilir. Şirket ekranında ise şirkete ait olan araçlar görüntülenebilir ve yeni araç ekleme ve araç silme yapılabilir. Aynı zamanda seferler de görüntülenebilir, ekleme ve silme işlemi yapılabilir.

Anahtar Kelimeler — Rezervasyon, araç, yolcu, sefer

I. GİRİŞ

Bu proje kullanıcıların hızlı ve kolay bir şekilde araç bileti aramalarını amaçlamaktadır. Otobüs, uçak ve tren şeklinde üç farklı araç çeşidi vardır. Kullanıcılar yolculuk yapacakları tarihi, kalkış yeri ve varış yeri verilerini girerek istediği seferleri seçebilir. Bu filtrelenen seferlerin birinden ya da daha fazlasından rezervasyon yapılabilir. Filtrelenen seferlere

tıklanıldıktan sonra araçtaki boş ve dolu koltuklar görüntülenir. Bu koltuklardan sadece boş olarak işaretlenenler rezervasyon için seçilebilir ve dolu olan bir koltuğun seçilmesi durumunda program hata verir. Yolculuk yapacak her bir kullanıcı için ayrı bir boş koltuk seçilmesi gereklidir. Her bir kullanıcı için koltuk seçildikten sonra her kullanıcı için gerekli yolcu verilerinin sorulduğu bir pencere açılır. Bu pencerede kullanıcı isim, soy isim, Tc numarası ve doğum tarihi gibi bilgiler alınır ve rezervasyon işlemi tamamlanır ve rezervasyon yapılan koltuk, sefer tarihi, araç ve rezervasyon yapılan aracın şirketi gibi veriler kullanıcıya bir pencere ile verilir. Ayrıca programın ilk giriş ekranında admin ve şirket giriş ekranı vardır. Admin ekranında var olan şirketler görüntülenebilir ve yeni şirket ekleme ve var olan şirketlerin kayıtlarının silinmesi ve hizmet bedeli belirlenmesi yapılabilir. Şirket ekranında ise giriş yapılan şirket için araçların görüntülenmesi, yeni araç ekleme ve var olan araçların kayıtlarının silinmesi ve seferlerin görüntülenmesi, yeni sefer ekleme ve seferlerin silinmesi yapılabilir.

II. YÖNTEM

Proje C++ dilinde Qt Framework kütüphanesi kullanılarak yapılmıştır. İlk pencerede üç buton bulunur ve bunlar sırasıyla admin giriş, şirket giriş ve yolcular için bilet ara butonlarıdır. Admin girişi için gerekli olan kullanıcı adı ve şifre statik olarak kodda tanımlanmıştır. Açılan pencereye veriler girildikten sonra veriler statik olarak tanımlanmış verilerle karşılaştırılır ve aynıysa admin ekranına giriş sağlanır. Admin ekranında bir listWidget tanımlanmıştır. ListWidget içerisine farklı Widget lar ya da text konulabilen bir Widget türüdür. Ekrana girilince programda tanımlanan firmaların isimleri listWidget a yazdırılır. Firmaların görüntülenmesi bu şekilde yapılır. Yeni Firma

eklenmesi için ise verilen lineEdit e firma ismi yazılır ve ekle tuşuna basılır. Bu sayede yazılan text listWidget a eklenir ve aynı zamanda o isme ait bir şirket oluşturulup şirketlerin bulunduğu bir std::vector e eklenir. Firma silme işlemi ise ilk önce firmaların isimlerinin bulunduğu listWidget teki firma ismine tıklanması ve sil butonuna tıklanması ile yapılır. Butona tıklandığında isim listWidget tan silinir ve o isme ait şirket bulunur ve şirketlerin bulunduğu std::vector den çıkartılır.

Şirket giriş ekranında ise şirket ismi ve şifre sorulur. Şirket girişi için gerekli olan bilgiler de statik olarak tanımlanmıştır ve companyCredentials adı verilen bir std::vector de tutulur. Giriş esnasında şirket ismi ve şifre yazıldıktan sonra butona tıklanmasıyla programda şirketlerin giriş bilgilerinin bulunduğu vector üzerinde loop yapılır ve girilen verilerle karşılaştırılır. Eğer girilen bilgiler tutulan hiçbir veriyle uyuşmaz ise ekrana açılan pencerede kullanıcı uyarılır. Eğer bilgiler doğru ise şirket ekranı açılır. Bu ekranda giriş yapan şirkete ait araçlar ve seferler gözüktür. Bu sayfadan aynı zamanda araç ve sefer ekleme de yapılır. Bu işlemler de aynı şekilde ilk önce bir pencere açılması ve bu pencerede gerekli bilgilerin alınması ile olur. Araç ekleme sayfasında aracın türü, programda gösterimde kullanılmak için araç ismi, aracın bağlı olduğu şirketin ismi ve araç koltuk sayısı istenir. Girilen araç türüne bağlı olarak sınıf seçilir ve oluşturulacak araç bu sınıftan girilen verilerle oluşturulur ve program içinde kullanılan araçlar vector üne eklenir. Sefer ekleme sayfasında ise seferde kullanılacak araç ismi, seferin bağlı olduğu şirket ismi, tarih ve sefer ücreti istenir. Girilen verilerle Trip sınıfından bir obje oluşturulup seferlerin saklandığı vektöre eklenir. Aynı zamanda ekrandaki listWidget lar eklemelere ve silinmelere göre güncellenir.

Ana giriş ekranındaki bir diğer buton ise bilet ara butonudur. Bilet ara butonuna tıklandığında büyük bir listWidget ve ara butonu bulunan ayrı bir pencere açılır. ListWidget, seferlerin sıra ile gösterildiği ve seferlerin tarihi, araçları, kalkış noktaları ve hedef noktaları, araçların koltuk sayıları ve şirket adı bulunan başka bir widget ı listeler. Bu listeleme sayesinde kullanıcı istediği seferi seçebilir. Ayrıca ara butonuna tıklayarak kendi isteğine göre kalkış noktası, varış noktası ve ücret seçimi yapabilir, açılan pencerede veriler girildikten sonra

listWidget girilen verilerle filtrelenecek şekilde sadece istenilen seferleri sıralar. Kullanıcı istediği sefere tıklar ve araçtaki koltuk sayılarını gösteren başka bir pencere ile karşılaşır. Bu pencerede koltuklar sıralanır ve boş ve dolu durumları karşılarında yazar. Kullanıcı bu koltuklardan dolu olanları seçtiğinde ekrana seçilemeyeceğini gösteren bir pencere çıkar. Kullanıcı koltuklardan boş olanlardan yolcu sayısı kadar seçebilir. Her bir yolcu için ayrı bir koltuk seçilir ve sonrasında seçilen koltuk sayısı kadar yolcu bilgileri alınır. Açılan pencerede yolcunun ismi, soyismi, Tc numarası ve doğum tarihi alınır. Daha sonra bu veriler Passenger sınıfı ile bir yolcu objesi oluşturulmak için kullanılır. Sonrasında oluşturulan yolcu objesi, sefer bilgilerini ve tarihi taşıyan bir rezervasyon objesi oluşturulur ve rezerve edilen koltuklar dolu olarak değiştirilir ve kullanıcıya rezerv edilen araç, koltuk numarası, araç ismi ve tarih gösterilir.

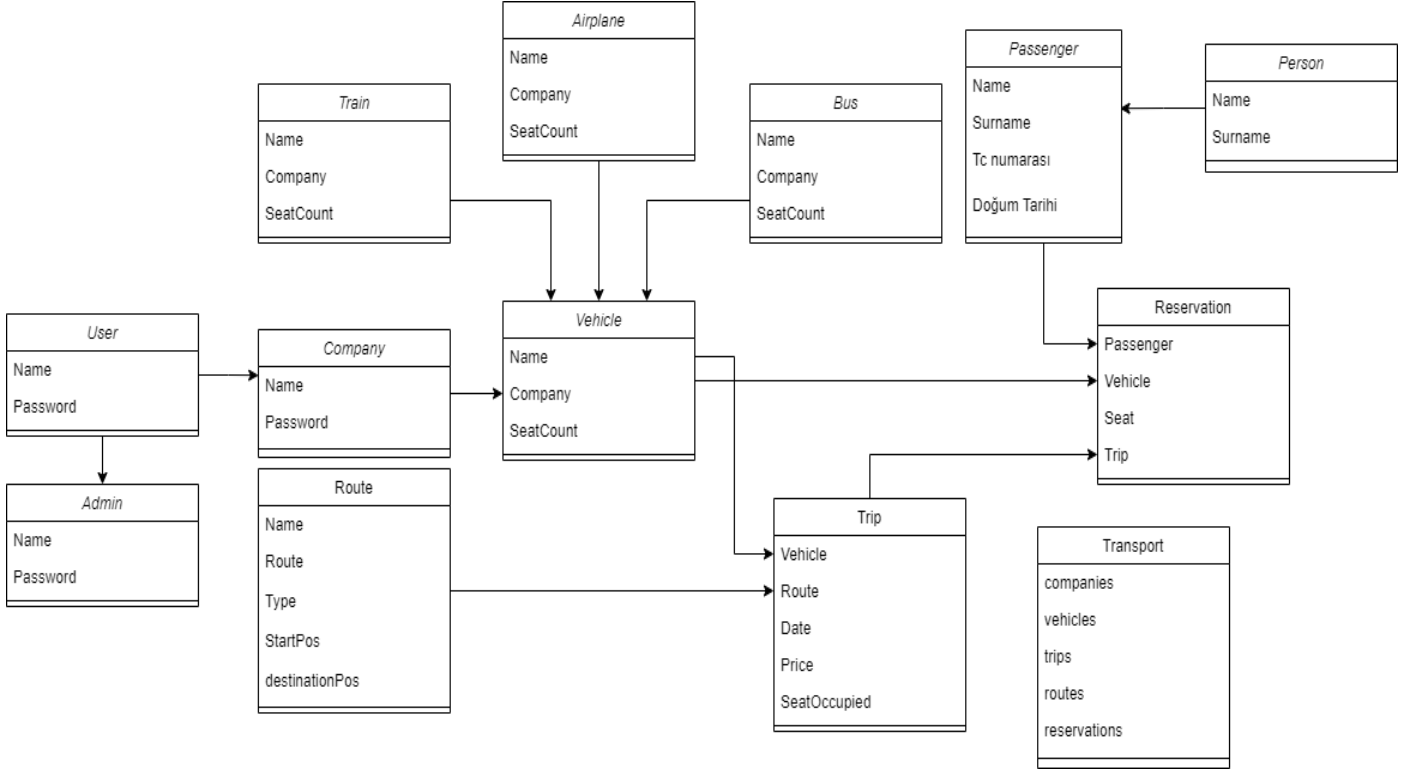
III. KATKILAR

Proje tamamıyla Ali Can Gündüz tarafından yapılmıştır.

IV. SONUÇ

Bu projede kullanıcıların kolay ve hızlı bir şekilde rezervasyon yapması amaçlanmıştır. Arayüz olabildiğince basit tutularak kullanıcı için kullanım kolaylığı amaçlanmıştır. Nesne yönelimli programlama temelleri göz önünde bulundurularak objeler arasında birbirleriyle bağ kurulmuş ve bu bağlar programın farklı yerlerinde kullanılmıştır. Veri tabanı bulunmadığından objeler kendi türleri içinde sınıflarına göre ayrıldıkları vector objelerinde saklanmıştır ve yeni obje oluşturulması veya obje silinmesinde vector güncellenmiştir. Projeye veri tabanı eklenerek daha stabil bir hale getirilebilir.

V. UML DİYAGRAMI



Vehicle: Araçların türetildiği bir abstract sınıftır. Araç isimleri, araçların bağlı olduğu şirket ve araçların koltuk sayıları gibi verileri tutar.

Bus: Vehicle sınıfından extend edilmiş bir sınıftır ve otobüsler oluşturmak için kullanılır. **Airplane:** Vehicle sınıfından extend edilmiş bir sınıftır ve uçak oluşturmak için kullanılır.

Train: Vehicle sınıfından extend edilmiş bir sınıftır ve uçak oluşturmak için kullanılır.

Passenger: Yolcu verilerini tutan bir sınıftır. Yolcuların isim, soyisim, Tc numarası ve doğum tarihi gibi verilerini tutar.

Company: Şirket verilerini tutan bir sınıftır. Şirketlerin isimlerini ve login için kullanacakları verileri tutar.

Route: Güzergah verilerini tutan bir sınıftır. Güzergahı saklayan bir vector, güzergahın başlangıç, bitiş noktalarını, güzergahın ne tür

araçların tarafından kullanılacağını saklayan verileri barındırır.

Trip: Seferlerle ilgili verilerin bulunduğu bir sınıftır. Aldığı veriler içinde Vehicle bulundurur ve bu seferin yapıldığı araçtır. Route seferin yapıldığı güzergah verisidir. Date seferin yapılacağı tarihtir. Price seferin fiyatıdır. SeatOccupied seferdeki aracın koltuklarının dolu veya boş olduğunu gösteren vector verisidir.

Reservation: Yapılan rezervasyonları saklayan sınıftır. Her yeni rezervasyon yapıldığında bu sınıftan bir obje oluşturulur. Bu objenin içerisinde Passenger yolcu objesi bulunur ve bu yolcu verilerini bulunduran objedir. Vehicle rezervasyon yapılan araçtır. Seat hangi koltuk için rezervasyonun yapıldığını belirtir. Koltuğun numarasıdır. Trip ise rezervasyon yapılan seferdir ve güzergah ve tarih gibi diğer değerleri tutar.

User: Admin ve Company sınıflarının extend aldığı, login için gerekecek olan name ve password verilerinin saklandığı sınıftır.

Transport: Şirketlerin, araçların, seferlerin, güzergah ve rezervasyonların saklandığı ve sonradan çağırılması ve eklenip silinmesi durumunda güncellenen verileri tutar.

Person: Yolcu tarafından extend edilen ve isim ve soyisim verilerini tutan sınıf.

KAYNAKLAR

- [1] <https://www.geeksforgeeks.org/vector-in-cpp-stl/>
- [2] <https://www.w3schools.com/cpp/>
- [3] <https://doc.qt.io/qt-6/qtwidgets-index.html>
- [4] <https://doc.qt.io/qt-6/qtgui-index.html>
- [5] <https://doc.qt.io/qt-6/qtcore-index.html>
- [6] https://www.w3schools.com/cpp/cpp_oop.asp
- [7] https://www.w3schools.com/cpp/cpp_classes.asp
- [8] https://www.w3schools.com/cpp/cpp_constructors.asp