



Nesne Yönelimli Programlama
Dönem Ödevi

Yunus Emre Tokmakçı
22011044

Fatih Eren SÖZEN
23011050

Ocak 2026

Video link :

<https://www.youtube.com/watch?v=2rugCxbBVzc>

NESNEYE YÖNELİK PROGRAMLAMA DÖNEM PROJESİ

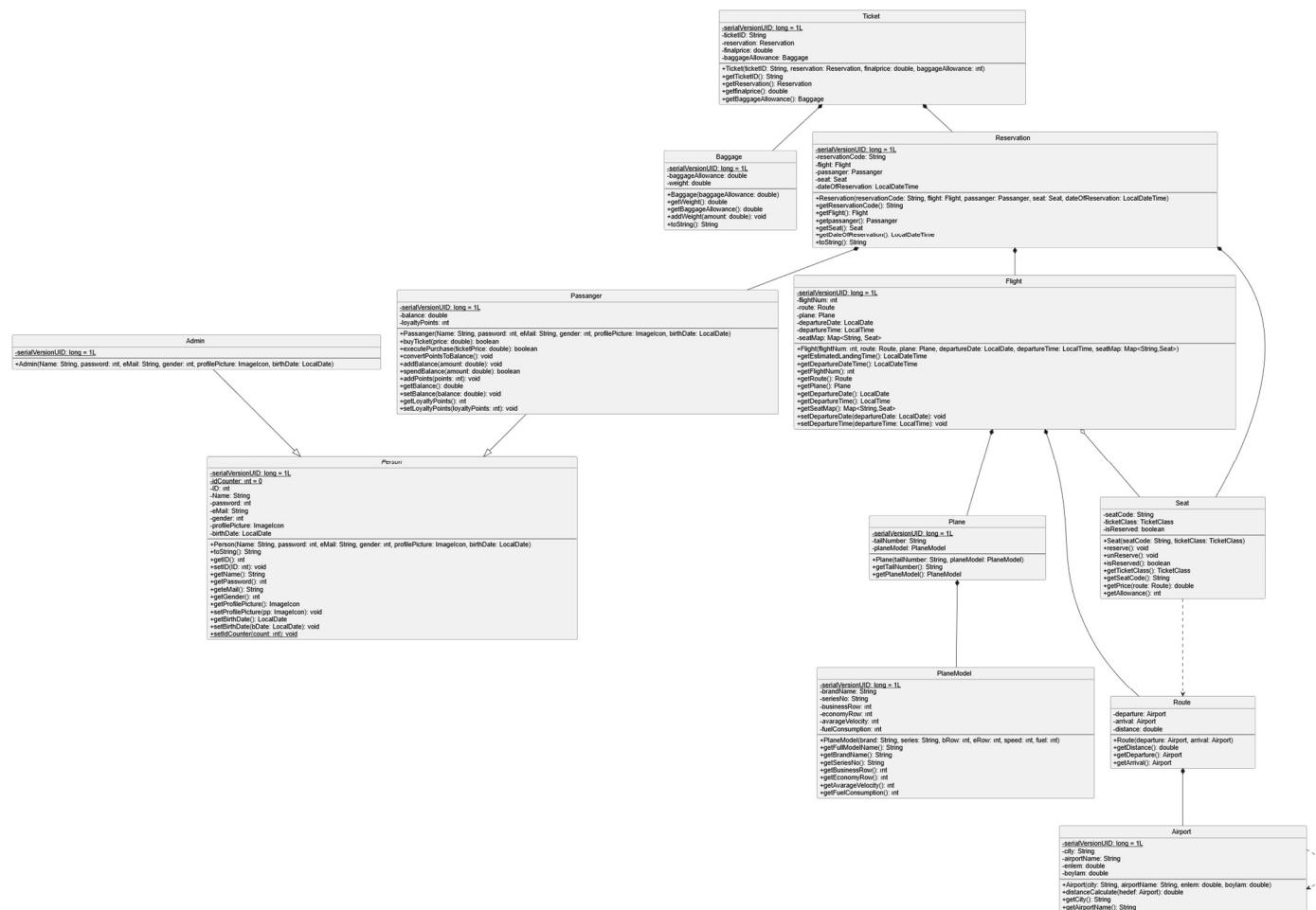


FES-YET AIRLINES

Projemizde 4 temel package bulunmaktadır.

Bunlar: 1- "models": Projenin veri yapılarını ve temel nesnelerini (Uçak, Yolcu, Rezervasyon vb.) barındırır. 2- "managers": Sistemin iş mantığını, hesaplamaları ve veri yönetimini idare eder. 3- "gui": Kullanıcı arayüzüünü, ekran tasarımını ve görsel etkileşimleri oluşturur. 4- "test": Sistemin fonksiyonlarının doğruluğunu doğrulayan birim testlerini (JUnit) içerir.

Models Package UML Diagram:



Projenin veri omurgasını oluşturan models paketi, sistemdeki tüm varlıklar, veri yapılarını ve bu yapıların birbirleriyle olan hiyerarşik ilişkilerini tanımlamaktadır. Bu pakette yer alan sınıflar, sadece veri taşımakla kalmayıp, nesneye özgü temel iş mantıklarını da (örneğin uçuş süresi hesaplama veya puan dönüştürme) kapsar ve Serializable arayüzü sayesinde dosya tabanlı veri saklama sistemine tam uyum sağlar. Kullanıcı yönetimi tarafında, tüm kullanıcı türlerinin ortak atası olan Abstract Person sınıfı; kimlik, iletişim ve güvenlik bilgilerini merkezi bir yapıda tutar. Bu sınıfından türetilen Passanger sınıfı, müşterilere özel căzdan bakiyesi ve sadakat puanı gibi finansal nitelikleri yönetirken; Admin sınıfı, sistem yöneticilerine özgü yetkilendirmeleri temsil eder. Bu kalitim yapısı, kod tekrarını önleyerek kullanıcı yönetimini modüler hale getirir.

Uçuş operasyonlarının temelini oluşturan yapıda ise Airport ve Route sınıfları coğrafi ve mesafesel verileri sağlarken, PlaneModel ve Plane sınıfları filodaki uçakların teknik ve fiziksel özelliklerini tanımlar. Tüm bu bileşenler Flight sınıfında bir araya gelir; bir uçuş nesnesi, belirli bir rotada, belirli bir zaman diliminde ve belirli bir uçakla gerçekleştirilen seferi temsil eder ve kendi içerisindeki koltuk harmasını (`Map<String, Seat>`) yöneterek operasyonel bütünlüğü sağlar.

Satış ve rezervasyon süreçleri ise Seat, Reservation ve Ticket sınıfları üzerinden yürütülür. TicketClass enum yapısı ile kategorize edilen her bir Seat nesnesi, doluluk durumunu takip eder. Bir rezervasyon işlemi, ilgili yolcu, seçilen uçuş ve atanan koltuğu tek bir kayıt altında birleştirir. Bu sürecin son aşaması olan Ticket sınıfı ise, oluşturulan rezervasyonu ve ödeme bilgilerini kapsayarak resmi bilet kaydını oluşturur ve Baggage sınıfı ile yolcunun bagaj haklarını bu biletle ilişkilendirir.

Managers Package UML Diagram:



Sistemin "beyni" olarak adlandırılabilen bu paket, uygulamanın tüm iş mantığını yürütür. Kullanıcı işlemleri, dosya okuma/yazma, fiyat hesaplamaları ve rezervasyon yönetimi gibi operasyonlar burada gerçekleşir. Her bir sınıfın sorumluluğu söylenir:

AssetManager: Sistemin fiziksel varlıklarını yönetir. Havalimanları, uçak filosu ve uçak modellerinin oluşturulmasından, bu verilerin dosyalara (airports.dat, fleet.dat, models.dat) kaydedilip okunmasından sorumludur.

CalculatePrice: Finansal hesaplamaları yapan yardımcı sınıfı. Mesafe bazlı bilet fiyatını, bagaj aşım ücretlerini ve sadakat puanlarının TL karşılığını hesaplar.

FlightManager: Tüm uçuş operasyonlarını idare eder. Yeni uçuş ekleme (çakışma kontrolü yaparak), uçuş silme ve uçuş verilerinin kalıcı olarak saklanması sağlar.

PersonManager: Kullanıcı hesaplarını yönetir. Yönetici ve yolcu kayıtlarını tutar, giriş işlemlerinde şifre doğrulamasını yapar ve kullanıcı verilerini diske kaydeder.

ReservationManager: Rezervasyon ve biletleme sürecini yönetir. Koltuk rezerve etme, bilet oluşturma, rezervasyon iptali ve bu işlemlerin UUID ile benzersiz kodlanması sağlar. Ayrıca geçmiş biletlerin sorgulanması işlevini de görür.

SeatManager: Her uçuş için koltuk planını otomatik olarak oluşturur. Uçağın modeline göre (Business için 4'lü, Economy için 6'lı düzen) koltukları isimlendirir ve hazırlar.

GUI Package UML Diagram:



Kullanıcıların sistemle etkileşime girdiği görsel arayüz katmanıdır. Java Swing kütüphanesi kullanılarak tasarlanmıştır.

- **MainMenu:** Uygulamanın giriş kapısıdır. Kullanıcıların (Admin veya Yolcu) giriş yapmasını sağlayan, marka logosu ve giriş formlarını barındıran ana ekrandır.
- **RegisterPanelPassanger:** Yeni yolcu kaydı oluşturmak için kullanılan kayıt formudur.
- **AdminFrame & PassengerFrame:** Giriş yapan kullanıcının rolüne göre (Yönetici veya Yolcu) açılan ana pencere iskeletleridir. Sol menü ve içerik panelini yönetirler.
- **AssetPanel (Admin):** Yöneticinin uçak filosu, uçak modelleri ve havalimanlarını ekleyip yönettiği paneldir.
- **FlightsPanelAdm (Admin):** Uçuş seferlerinin planlandığı, yeni sefer eklendiği ve listelendiği yönetim ekranıdır.
- **FlightsPanelPassanger (Yolcu):** Yolcuların uçuş araması yapıp kriterlere göre filtreleyebildiği ve bilet alma sürecini başlattığı ekrandır.
- **SeatViewerBasic & SeatViewerPassanger:** Uçak içi koltuk düzenini görselleştiren sınıflardır.

Basic versiyon sadece doluluk oranını gösterirken,

Passanger versiyonu yolcunun interaktif olarak koltuk seçmesine olanak tanır.

- **ConcurrencyDemoPanel:** multithreading yapısını görsel olarak test etmek için kullanılan özel bir simülasyon panelidir. Aynı anda yüzlerce botun koltuk rezervasyonu yapmaya çalışmasını simüle eder.
- **ProfilePanel & PersonInfoPanel:** Kullanıcının profil bilgilerini (isim, şifre vb.) görüntülediği ve düzenleyebildiği ekranlardır.
- **WalletPanel:** Yolcunun cüzdan bakiyesini görüntüleyip para yükleyebildiği ve puanlarını nakde çevirebildiği finansal paneldir.
- **MyTicketsPanel:** Yolcunun geçmiş ve gelecek biletlerini listelediği, detaylarını inceleyip iptal edebildiği "Biletlerim" sayfasıdır.

TEST SONUÇLARI:

```
.  
+-- JUnit Jupiter [OK]  
| +-- SeatManagerTest [OK]  
| | +-- Koltuk kodlari dogru formatta olmali [OK]  
| | +-- Bos koltuk sayisi baslangicta toplam koltuk sayisina esit olmali [OK]  
| | +-- Business ve Economy koltuk sinifi dogru atanmali [OK]  
| | +-- Birden fazla rezervasyon sonrasi koltuk sayisi dogru hesaplanmali [OK]  
| | +-- Zaten rezerve edilmis koltuk isReserved true donmeli [OK]  
| | +-- Olmayan koltuk numarasi icin null doner [OK]  
| | +-- Rezervasyon sonrasi bos koltuk sayisi azalmali [OK]  
| | +-- Toplam koltuk sayisi dogru olmali [OK]  
| '-- Rezervasyon iptali sonrasi koltuk tekrar bos olmali [OK]  
+-- FlightSearchEngineTest [OK]  
| +-- Olmayan sehir icin bos liste doner [OK]  
| +-- Gecmis tarihli ucuslar filtrelenmeli [OK]  
| +-- Flight tarih ve saat bilgisi dogru donmeli [OK]  
| +-- Flight rota bilgisi dogru donmeli [OK]  
| +-- Ucus ekleyince toplam sayi artmali [OK]  
| +-- Flight koltuk haritasi olmali [OK]  
| +-- Kalkis ve varis sehrine gore ucus filtreleme [OK]  
| +-- Sadece kalkis sehrine gore filtreleme [OK]  
| +-- Sadece varis sehrine gore filtreleme [OK]  
| '-- Gecmis ucuslari eleme - tum aktif ucuslar gelecekte [OK]  
-- PriceCalculationTest [OK]  
| +-- Business sinifi icin fiyat hesaplama [OK]  
| +-- Economy sinifi icin fiyat hesaplama [OK]  
| +-- Price multiplier degerleri dogru olmali [OK]  
| +-- Bagaj limiti - Business 40kg, Economy 20kg [OK]  
| +-- Bagaj asimi ucreti hesaplama - limit asimi kg basi 100 TL [OK]  
| '-- Business fiyati Economy fiyatinin 1.5 katı olmali [OK]  
+-- JUnit Vintage [OK]  
'-- JUnit Platform Suite [OK]
```

```
Test run finished after 158 ms
```

```
[       6 containers found      ]  
[       0 containers skipped   ]  
[       6 containers started   ]  
[       0 containers aborted   ]  
[       6 containers successful ]  
[       0 containers failed    ]  
[     25 tests found          ]  
[       0 tests skipped        ]  
[     25 tests started         ]  
[       0 tests aborted        ]  
[     25 tests successful      ]  
[       0 tests failed         ]
```