\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**YMÜ227 Nesne Tabanlı Programlama**

**Proje-2**

**Projede Hedeflenen Konu Başlıkları:** Sınıflar, Nesneler, Kalıtım, Soyut Sınıflar, Arayüzler, Çok-biçimlilik (polymorphism), Java GUI

**Proje Konusu:** Havayolu Bilgi Sistemi Gerçekleştirimi

**Gereksinimler:**

Havayolu şirketlerinin uçuş, personel ve yolcu bilgilerinin takip edilmesini ve yönetilmesini sağlayan bir sitem gerçekleştirimi istenmektedir. Sistemde birden fazla havayolu şirketi, bu şirketlerin farklı tiplerde uçakları ve desteklediği seferler bulunmaktadır. Bir sefere zaman, uçak, personel ve yolcu bilgileri atanarak uçuş bilgisi oluşturulur. Sistemin bu uçuş bilgilerine göre belirli sorguları karşılaması beklenmektedir.

Her havayolunun farklı seferleri bulunmaktadır. Seferde rota ve uçuş süresi gibi her uçuşta değişmeyen bilgiler tutulur. Örneğin; ABC-Air havayolunun İstanbul Atatürk Havaalanı’ndan İzmir Adnan Menderes Havaalanına 45 dakika süreli, 800 km mesafeli ve 142 nolu seferi.

Her uçuş sefer numarası, personel ve yolcu listesi, kalkış zamanı ve uçak bilgilerini içerir. Birden fazla uçuş aynı seferi içerebilir. Örneğin ABC-Air’in aynı sefer numaralı İstanbul İzmir seferi için aynı günde 5 uçuş gerçekleşebilir.

Her havayolu şirketinin farklı tiplerde uçaklardan oluşan bir filosu bulunmaktadır. Uçak tipleri toplam yolcu kapasitesini, farklı sınıflara ait koltuk (ekonomik, business, standart) sayılarını ve bu uçakla gerçekleştirilen uçuşlarda kaç tane personele (pilot ve hostese) gereksinim duyulduğunu belirler.

Her seferin ücreti vardır. Kullanıcı bilet almak istediğinde seçtiği uçuşun sefer ücreti ve kullanıcı tipi göz önüne alınarak bilet fiyatı hesaplanır. VIP ve standart olmak üzere iki adet kullanıcı tipi vardır. Bu tipler yolcu eklenmesi sırasında belirlenir. VIP yolcularda bilet fiyatı hesaplanırken yüzde 20 indirim uygulanır. Fiyat hesaplanırken ayrıca kullanıcının seçtiği koltuk tipi de göz önüne alınır. Fiyat belirlendikten sonra yolcu için seçilen uçuş bilgilerini ve fiyatı içeren bilet hazırlanır.

**Sistemin 3 farklı rolde kullanıcısı olmalıdır;**

**1. Yönetici:** Sefer, uçuş, personel, havayolu gibi bilgilerin sisteme kaydedilmesi, değiştirilmesi, görüntülenmesi ve uçuşlara personel atanması işlemlerini gerçekleştirir.

Sistemin cevap vermesi gereken önemli sorguları şunlardır;

·         Belirli bir personelin uçuşlarının listesi

·         Bir tarih aralığında belirli bir yolcunun uçuşlarının listesi

**2. Personel:** görev aldığı uçuş bilgilerini görüntüler.

·         Kendisine atanmış uçuşların listesi

**3. Yolcu:** istenen kriterlere uygun uçuşların görüntülenip, bu uçuşlardan biri için bilet alınması işlemlerini gerçekleştirir. Bilet alımı, bilet ücreti yolcuya gösterildikten sonra bir düğmeye basıldığında yolcunun uçuş yolcu listesine eklenmesi ve koltuk kapasitesinin düşürülmesi olarak basitleştirilmiştir.

·         Belirli tarih aralığında ve yönde uçuşların listesi

·         Seçilen koltuk tipi için belirli bir tarih aralığında ve yönde yer bulunan uçuşlar.

·         Bir hava yolu şirketinin belirli bir şehirdeki havaalanlarına uçuşlarının listesi

**ÖNEMLİ NOTLAR:**

**1)**Gerçekleştirilen çalışma ile ilgili görsel destekli bir proje raporu hazırlanmalıdır. Raporda, kaynak kodlar yeterli yorum satırları eklenerek çıktı olarak verilmelidir. Ek olarak görsel içerikler barındıran basit bir kullanıcı kılavuzu hazırlanmalıdır.

**2)      Rapor İçeriğinde Dikkat Edilmesi Gereken Noktalar:**

·         Her sayfada sayfa numarası bulunmalıdır.

·         **Proje kapak sayfası**

o   Ders Adı,

o   Proje Adı,

o   Dersin Hocası,

o   Öğrenci Bilgilerini

o   Proje Tarihi bilgilerini içermelidir.

·         **İçindekiler**

o   Giriş

o   Proje Amacı

o   Kullanılan Teknolojiler

o   Proje Geliştirme Süreci

o   Kaynak Kodlar

o   Çıktılar-Sonuç ve Değerlendirmeleri içermelidir.

**3)      Raporun Teslimi:**

·         Proje raporunun teslimi Debsis sistemi üzerinden gerçekleştirilecektir.

**4)**Proje kaynak kodları ve raporlar 19Aralık 2023 Salı gününe kadar OgrenciNo\_Ad\_Soyad isimli bir rar dosyası halinde Debsis’e yüklenmelidir..

**5)**Proje kaynak kodları ve raporlar 20 Aralık 2023 Çarşamba günü Laboratuvar dersinde kontrol edilecektir.

**6)**Proje kullanıcı grafik arayüzleri kullanılarak tasarlanmalıdır (Görsel Programlama).

**7)**Öğrenci sayısının çokluğu nedeni ile proje kontrollerinin verilen zaman diliminde yapılması gerekmektedir.

**8)**Projeler yıl içi notunu etkileyeceği için dersi alttan alan öğrenciler tarafından da yapılmalıdır.

**9)**Proje teslim tarihi, öğrencilerimizin sınavdan önce geri-bildirim alabilmeleri ve hatalarını düzeltebilmeleri amacı ile ertelenmesi mümkün olan en geç tarih (19 Aralık 2023) göz önüne alınarak belirlenmiştir. 19 Aralık 2023 tarihinden sonra ödev teslimi gerçekleştirilemeyecektir.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Bu tür bir sistem geliştirmek için bir dizi bileşen ve fonksiyonel gereksinim bulunmaktadır. İşte bu sistemin temel özellikleri ve gereksinimleri:

1. \*\*Havayolu Şirketleri Yönetimi:\*\*

- Sistem, birden fazla havayolu şirketinin eklenmesini ve yönetilmesini sağlamalıdır.

- Her havayolu şirketi için genel bilgiler (adı, filo bilgileri, desteklenen seferler) kaydedilmelidir.

2. \*\*Uçak Yönetimi:\*\*

- Her havayolu şirketi, farklı tiplerde uçaklara sahiptir. Uçak tipleri, yolcu kapasitesi, koltuk sınıfları ve personel ihtiyaçları gibi bilgilerle birlikte kaydedilmelidir.

3. \*\*Sefer Yönetimi:\*\*

- Her havayolu şirketi için seferler eklenmeli ve yönetilmelidir.

- Seferdeki değişmeyen bilgiler (rota, uçuş süresi) kaydedilmelidir.

- Her sefer için belirli bir uçuş numarası atanmalıdır.

4. \*\*Uçuş Bilgisi Oluşturma:\*\*

- Sisteme uçuş bilgisi eklemek için sefer numarası, personel, yolcu bilgileri, kalkış zamanı ve uçak bilgileri sağlanmalıdır.

- Birden fazla uçuş aynı seferi içerebilmelidir.

5. \*\*Bilet Satışı:\*\*

- Kullanıcıların bilet alabilmesi için sistemde bir bilet satış modülü bulunmalıdır.

- Bilet fiyatı, sefer ücreti, kullanıcı tipi (VIP veya standart), koltuk tipi gibi faktörlere bağlı olarak hesaplanmalıdır.

- VIP kullanıcılara %20 indirim uygulanmalıdır.

6. \*\*Fiyat Hesaplama:\*\*

- Bilet fiyatı, sefer ücreti, kullanıcı tipi ve seçilen koltuk tipine göre hesaplanmalıdır.

7. \*\*Bilet Hazırlama:\*\*

- Kullanıcının seçtiği uçuş bilgilerini ve fiyatını içeren bir bilet hazırlanmalıdır.

8. \*\*Sorgular ve Raporlar:\*\*

- Sistem, belirli sorguları karşılayarak kullanıcılara raporlar sunmalıdır. Örneğin, belirli bir tarih aralığındaki uçuşlar, belirli bir havayolu şirketinin seferleri vb.

Bu gereksinimlerle birlikte, sistemde güvenlik, kullanıcı yönetimi, hata yönetimi gibi non-fonksiyonel gereksinimlere de dikkat edilmelidir. Sistem, kullanıcı dostu bir arayüzle donatılmalı ve veri bütünlüğünü sağlamak için uygun veritabanı yönetim sistemleri kullanmalıdır.

import java.util.ArrayList;

import java.util.List;

class Ucak {

String tip;

int yolcuKapasitesi;

int ekonomikKoltukSayisi;

int businessKoltukSayisi;

int personelSayisi;

public Ucak(String tip, int yolcuKapasitesi, int ekonomikKoltukSayisi, int businessKoltukSayisi, int personelSayisi) {

this.tip = tip;

this.yolcuKapasitesi = yolcuKapasitesi;

this.ekonomikKoltukSayisi = ekonomikKoltukSayisi;

this.businessKoltukSayisi = businessKoltukSayisi;

this.personelSayisi = personelSayisi;

}

}

class Sefer {

int seferNo;

String rota;

int ucusSuresi;

public Sefer(int seferNo, String rota, int ucusSuresi) {

this.seferNo = seferNo;

this.rota = rota;

this.ucusSuresi = ucusSuresi;

}

}

class Ucus {

int seferNo;

String kalkisZamani;

Ucak ucak;

List<String> personelListesi;

List<Yolcu> yolcuListesi;

public Ucus(int seferNo, String kalkisZamani, Ucak ucak, List<String> personelListesi, List<Yolcu> yolcuListesi) {

this.seferNo = seferNo;

this.kalkisZamani = kalkisZamani;

this.ucak = ucak;

this.personelListesi = personelListesi;

this.yolcuListesi = yolcuListesi;

}

}

class Yolcu {

String ad;

String soyad;

String kullaniciTipi; // VIP veya Standart

String koltukTipi; // Ekonomik, Business veya Standart

public Yolcu(String ad, String soyad, String kullaniciTipi, String koltukTipi) {

this.ad = ad;

this.soyad = soyad;

this.kullaniciTipi = kullaniciTipi;

this.koltukTipi = koltukTipi;

}

}

public class HavayoluSistemi {

public static void main(String[] args) {

// Örnek kullanım

Ucak ucak1 = new Ucak("Boeing 737", 150, 100, 20, 8);

Sefer sefer1 = new Sefer(142, "İstanbul - İzmir", 45);

List<String> personelListesi = new ArrayList<>();

personelListesi.add("Pilot - John Doe");

personelListesi.add("Hostes - Jane Doe");

List<Yolcu> yolcuListesi = new ArrayList<>();

yolcuListesi.add(new Yolcu("Alice", "Doe", "VIP", "Business"));

yolcuListesi.add(new Yolcu("Bob", "Doe", "Standart", "Ekonomik"));

Ucus ucus1 = new Ucus(sefer1.seferNo, "2023-01-01 12:00", ucak1, personelListesi, yolcuListesi);

// Sistemdeki bilgileri kullanarak çeşitli işlemler yapılabilir.

// Örneğin, bilet fiyatı hesaplama, sorgular, raporlar, vb.

}

}

import java.util.ArrayList;

import java.util.List;

class HavayoluSistemi {

List<Havayolu> havayoluListesi = new ArrayList<>();

List<Personel> personelListesi = new ArrayList<>();

List<Ucus> ucusListesi = new ArrayList<>();

public void seferEkle(Sefer sefer) {

// Sefer ekleme işlemleri

}

public void ucusEkle(Ucus ucus) {

// Uçuş ekleme işlemleri

}

public void personelEkle(Personel personel) {

// Personel ekleme işlemleri

}

public List<Ucus> personelUcusListesi(String personelAdi) {

// Belirli bir personelin uçuşlarının listesi

// İlgili personelin adına göre ucusListesi'nde arama yapılabilir.

return null;

}

public List<Ucus> yolcuUcusListesi(String yolcuAdi, String baslangicTarihi, String bitisTarihi) {

// Bir tarih aralığında belirli bir yolcunun uçuşlarının listesi

// İlgili tarih aralığına ve yolcunun adına göre ucusListesi'nde arama yapılabilir.

return null;

}

}

class Personel {

String ad;

List<Ucus> atanmisUcuslar = new ArrayList<>();

public Personel(String ad) {

this.ad = ad;

}

public void ucusAta(Ucus ucus) {

// Personel için uçuş ata

atanmisUcuslar.add(ucus);

}

public List<Ucus> atanmisUcusListesi() {

// Kendisine atanmış uçuşların listesi

return atanmisUcuslar;

}

}

class Ucus {

Sefer sefer;

String kalkisZamani;

List<Personel> personelListesi = new ArrayList<>();

List<Yolcu> yolcuListesi = new ArrayList<>();

public Ucus(Sefer sefer, String kalkisZamani) {

this.sefer = sefer;

this.kalkisZamani = kalkisZamani;

}

public void personelEkle(Personel personel) {

// Uçuşa personel ekle

personelListesi.add(personel);

}

public void yolcuEkle(Yolcu yolcu) {

// Uçuşa yolcu ekle

yolcuListesi.add(yolcu);

}

}

class Yolcu {

String ad;

String soyad;

String koltukTipi;

public Yolcu(String ad, String soyad, String koltukTipi) {

this.ad = ad;

this.soyad = soyad;

this.koltukTipi = koltukTipi;

}

}

class Sefer {

int seferNo;

String rota;

int ucusSuresi;

public Sefer(int seferNo, String rota, int ucusSuresi) {

this.seferNo = seferNo;

this.rota = rota;

this.ucusSuresi = ucusSuresi;

}

}

public class Uygulama {

public static void main(String[] args) {

// Örnek kullanım

HavayoluSistemi havayoluSistemi = new HavayoluSistemi();

Sefer sefer1 = new Sefer(142, "İstanbul - İzmir", 45);

havayoluSistemi.seferEkle(sefer1);

Ucus ucus1 = new Ucus(sefer1, "2023-01-01 12:00");

havayoluSistemi.ucusEkle(ucus1);

Personel pilot1 = new Personel("John Doe");

havayoluSistemi.personelEkle(pilot1);

pilot1.ucusAta(ucus1);

Yolcu yolcu1 = new Yolcu("Alice", "Doe", "Business");

ucus1.yolcuEkle(yolcu1);

// Sistemdeki bilgileri kullanarak çeşitli işlemler yapılabilir.

// Örneğin, belirli bir personelin uçuşlarını listeleme, bilet alma, vb.

}

}