



**T.C**  
**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ**  
**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**  
**BİLGİSAYAR/YAZILIM MÜHENDİSLİĞİ**

**GEZGİN GEMİ ŞİRKETİ**

**Hazırlayan**  
**İbrahim Bener Karaca**  
**220501019**  
**<https://github.com/BenerKaraca>**  
**Enes Küçük**  
**220501017**  
**<https://github.com/EnessKucukk>**

**DERS SORUMLUSU:**  
**DR. ÖĞR. ÜYESİ Ercan Ölçer**

**04/05/2024**

# 1 GİRİŞ

## 1.1 Projenin amacı

2 Bu ödevde amaç bir gemi şirketinin operasyonlarını yönetmek için bir arayüz sağlanmasıdır. Ana pencerede gemilerin, seferlerin, limanların, kaptanların ve mürettebatın bilgilerini görüntüleyebilir ve düzenleyebiliriz. Ayrıca her bir değerimiz için ekleme ve silme işlevlerini içeren düğmeler bulunmaktadır. Veritabanı desteği ile gemiler, seferler, limanlar, kaptanlar ve mürettebat bilgileri saklanır ve yönetilir. Burada kullandığımız kütüphaneler yönetimiyle veritabanı ve yönetimi kavramını pratiğe dökmüş olduk.

- Ödevde bizden istenenler şunlardır;

- **Gemilerin Yönetimi:**

- Yolcu gemileri, petrol tankerleri ve konteyner gemileri için ayrı ayrı bilgilerin tutulması.
- Her gemi için seri numarası, adı, ağırlığı ve yapım yılı bilgilerinin saklanması.
- Yolcu gemileri için yolcu kapasitesi, petrol tankerleri için petrol kapasitesi (Litre), konteyner gemileri için konteyner sayısı kapasitesi ve maksimum ağırlık değerlerinin kaydedilmesi.

- **Seferlerin Yönetimi:**

- Her bir sefer için ID, yola çıkış tarihi, dönüş tarihi ve yola çıkış limanı bilgilerinin saklanması.
- Her seferde sadece 1 geminin çıkabileceği ve en az 2 kaptan ile 1 mürettebatın bulunması.
- Geçmişte yapılmış, gelecekte planlanan ve olası seferlerin bilgilerinin tutulması.

- **Limanların Yönetimi:**

- Her liman için liman adı, ülkesi, nüfusu, pasaport istenip istenmediği ve demirleme ücreti verilerinin kaydedilmesi.
- Liman adı ve ülkesinin liman için ayırt edici olması.
- Hiç uğranmamış fakat uğranma ihtimali olan limanların kayıtlarının tutulabilmesi.

- **Kaptanlar ve Mürettebatın Yönetimi:**

- Her kaptan ve mürettebat için ID, ad, soyad, adres, vatandaşlık, doğum tarihi ve işe giriş tarihi bilgilerinin kaydedilmesi.
- Kaptanların lisansları ve mürettebatın çalıştığı görev bilgilerinin saklanması.
- Kaptan ve mürettebatın aynı anda sadece bir seferde bulunabileceği.

- **Yazılımın Gerçekleştirilmesi:**

- Her varlık için ayrı sınıflar yazılması ve ilgili işlemlerin bu sınıflar üzerinden yönetilmesi.
- Veri tabanında gerekli tabloların oluşturulması ve verilerin doğru bir şekilde saklanması.

- Veri tabanı ve form ekranı arasındaki yönetimin nesneler aracılığı ile yapılması.
- Veri tabanı ile ilgili tüm işlemlerin yazılım tarafından gerçekleştirilmesi.

## 3 TASARIM

### 3.1 Kullanılacak teknolojiler

Bu yazılım Python programlama dili kullanılarak geliştirilmiştir. PyQt5 kütüphanesi kullanılarak grafik arayüz oluşturulmuş ve PyQt5.QtSql modülü ile veritabanı işlemleri gerçekleştirilmiştir. PyQt5.QtWidgets modülü, kullanıcı arayüzü öğelerini oluşturmak için kullanılmıştır. PyQt5.QtSql modülü, veritabanı işlemleri için gerekli olan sınıfları ve işlevleri sağlamaktadır. Veritabanı olarak SQLite kullanılmıştır.

AnaPencere sınıfı, PyQt5.QtWidgets modülünün QWidget sınıfından türetilmiştir ve ana pencereyi oluşturur. GemilerEkranı, SeferlerEkranı, LimanlarEkranı, KaptanlarEkranı ve MurettebatEkranı sınıfları da PyQt5.QtWidgets modülünden türetilmiş olup, kullanıcı arayüzünün farklı bölümlerini yönetmektedir.

Bu programda kullanılan PyQt5 kütüphanesi, PyQt5.QtSql modülü ve PyQt5.QtWidgets modülü harici kütüphanelerdir. PyQt5 kütüphanesi, PyQt5.QtWidgets modülü kullanılarak grafik arayüzler oluşturulmasını sağlar ve PyQt5.QtSql modülü ile veritabanı işlemleri gerçekleştirilir

```
import sys
from PyQt5.QtWidgets import (
    QHBoxLayout, QLabel, QLineEdit,
    QWidget, QTableWidgetItem, QDialog, QFormLayout
)
from PyQt5.QtWidgets import QApplication, QWidget, QVBoxLayout, QPushButton, QMessageBox
from PyQt5.QtSql import QSqlDatabase, QSqlQuery
```

### Veri tabanı tasarımı

1. **gemiler:** Gemilerin temel bilgilerini içeren tablo. Her gemi için birincil anahtar olarak seri\_numarasi kullanılır.
  - seri\_numarasi (TEXT): Geminin seri numarası (Birincil Anahtar).
  - ad (TEXT): Geminin adı.
  - agirlik (INTEGER): Geminin ağırlığı.
  - yapim\_yili (INTEGER): Geminin yapım yılı.
  - tip (TEXT): Gemi tipi (yolcu, petrol, konteyner).
  - kapasite (INTEGER): Geminin kapasitesi (yolcu gemileri için yolcu sayısı, petrol tankerleri için litre cinsinden petrol kapasitesi, konteyner gemileri için konteyner sayısı).
  - max\_agirlik (INTEGER): Geminin maksimum taşıma kapasitesi.

2.	<b>seferler:</b> Gemilerin sefer bilgilerini içeren tablo. Her sefer için birincil anahtar olarak sefer_id kullanılır.
	<ul style="list-style-type: none"> <li>sefer_id (INTEGER): Seferin kimlik numarası (Birincil Anahtar, otomatik artan).</li> <li>yola_cikis_tarihi (TEXT): Seferin başlangıç tarihi.</li> <li>donus_tarihi (TEXT): Seferin dönüş tarihi.</li> <li>yola_cikis_limani (TEXT): Seferin başlangıç limanı.</li> </ul>
3.	<b>limanlar:</b> Limanların bilgilerini içeren tablo. Her liman için birincil anahtar olarak liman_adi kullanılır.
	<ul style="list-style-type: none"> <li>liman_adi (TEXT): Limanın adı (Birincil Anahtar).</li> <li>ulke (TEXT): Limanın bulunduğu ülke.</li> <li>nufus (INTEGER): Limanın nüfusu.</li> <li>pasaport_gerekli_mi (TEXT): Limana giriş için pasaport gerekip gerekmediği.</li> <li>demirleme_ucreti (INTEGER): Limanda demirleme ücreti.</li> </ul>
4.	<b>kaptanlar:</b> Kaptanların bilgilerini içeren tablo. Her kaptan için birincil anahtar olarak kaptan_id kullanılır.
	<ul style="list-style-type: none"> <li>kaptan_id (INTEGER): Kaptanın kimlik numarası (Birincil Anahtar, otomatik artan).</li> <li>ad (TEXT): Kaptanın adı.</li> <li>soyad (TEXT): Kaptanın soyadı.</li> <li>adres (TEXT): Kaptanın adresi.</li> <li>vatandaslik (TEXT): Kaptanın vatandaşlık bilgisi.</li> <li>dogum_tarihi (TEXT): Kaptanın doğum tarihi.</li> <li>ise_giris_tarihi (TEXT): Kaptanın işe giriş tarihi.</li> </ul>
5.	<b>murettebat:</b> Mürettebatın bilgilerini içeren tablo. Her mürettebat için birincil anahtar olarak murettebat_id kullanılır.
	<ul style="list-style-type: none"> <li>murettebat_id (INTEGER): Mürettebatın kimlik numarası (Birincil Anahtar, otomatik artan).</li> <li>ad (TEXT): Mürettebatın adı.</li> <li>soyad (TEXT): Mürettebatın soyadı.</li> <li>adres (TEXT): Mürettebatın adresi.</li> <li>vatandaslik (TEXT): Mürettebatın vatandaşlık bilgisi.</li> <li>dogum_tarihi (TEXT): Mürettebatın doğum tarihi.</li> <li>ise_giris_tarihi (TEXT): Mürettebatın işe giriş tarihi.</li> </ul>

## 3.2 Kullanıcı arayüzü tasarımı

Gezgin Gemi Şirketi Ödevi

Seri Numarası	Ad	Ağırlık
Ekle		
Sil		

Petrol Gemileri

Seri Numarası	Ad	Ağırlık
Ekle		
Sil		

Yolcu Gemileri

Seri Numarası	Ad	Ağırlık
Ekle		
Sil		

Konteyner Gemileri

Seri Numarası	Ad	Ağırlık
Ekle		
Sil		

Seferler

ID	Yola Çıkış Tarihi	Dönüş Tarihi	Yola Çıkış Limanı
Ekle			
Sil			

Limanlar

Liman Adı	Ülke	Nüfus	Asaport Gerekli mi	Demirleme Üç
Ekle				
Sil				

Kaptanlar

ID	Ad	Soyad	Adres	Vatandaşlık
Ekle				
Sil				

Mürettebat

ID	Ad	Soyad	Adres	Vatandaşlık	Doğum Tarihi	İşe giriş tarihi
Ekle						
Sil						
Ekle						

Burada arayüz için PyQt5 kütüphanesini kullandık. Arayüzde her bir gemi için farklı tablo, limanlar, kaptan, mürettebat ve seferler için farklı tablolar mevcut. Burada kullanıcı ekleme ve silme işlemlerini gerçekleştirebilir. Düzenleme kısmını da imleci düzenlenmek istenen objeye götürüp tıklanarak yapılabilir.

## 4 UYGULAMA

### 4.1 Kodlanan bileşenlerin açıklamaları

#### 1. DatabaseManager Sınıfı:

- Bu sınıf, veritabanı işlemlerini gerçekleştirir.
- `__init__` metodu, veritabanı bağlantısını oluşturur ve gerekli veritabanı dosyasını açar.
- `create_tables` metodu, veritabanında gerekli tabloları oluşturur.

#### 2. AnaPencere Sınıfı:

- Bu sınıf, uygulamanın ana penceresini oluşturur.
- `__init__` metodu, ana pencerenin başlık ve boyutunu ayarlar, ardından arayüzü oluşturmak için `initUI` metodunu çağırır.
- `initUI` metodu, ana pencerenin arayüzünü oluşturur ve farklı bileşenlerin yerleştirilmesini sağlar.

#### 3. GemilerEkranı Sınıfı:

- Bu sınıf, gemilerin görüntülendiği bölümün arayüzünü oluşturur.
- `__init__` metodu, gemi türünü belirler ve arayüzü oluşturmak için `initUI` metodunu çağırır.
- `initUI` metodu, gemi listesi için bir tablo oluşturur ve eklemek ve silmek için düğmeler ekler.
- `gemi_ekle_dialog_ac` metodu, yeni bir gemi eklemek için bir iletişim kutusu açar.

- `yeni_gemi_kaydet` metodu, iletişim kutusundaki bilgilerle yeni bir gemi oluşturur ve tabloya ekler.
- `gemi_sil` metodu, seçili bir gemiyi tablodan siler.

#### 4. **SeferlerEkranı Sınıfı:**

- Bu sınıf, seferlerin görüntülediği bölümün arayüzünü oluşturur.
- `__init__` metodu, arayüzü oluşturmak için `initUI` metodunu çağırır.
- `initUI` metodu, sefer listesi için bir tablo oluşturur ve eklemek ve silmek için düğmeler ekler.
- `sefer_ekle_dialog_ac` metodu, yeni bir sefer eklemek için bir iletişim kutusu açar.
- `yeni_sefer_kaydet` metodu, iletişim kutusundaki bilgilerle yeni bir sefer oluşturur ve tabloya ekler.
- `sefer_sil` metodu, seçili bir seferi tablodan siler.

#### 5. **LimanlarEkranı Sınıfı:**

- Bu sınıf, limanların görüntülediği bölümün arayüzünü oluşturur.
- `__init__` metodu, arayüzü oluşturmak için `initUI` metodunu çağırır.
- `initUI` metodu, liman listesi için bir tablo oluşturur ve eklemek ve silmek için düğmeler ekler.
- `liman_ekle_dialog_ac` metodu, yeni bir liman eklemek için bir iletişim kutusu açar.
- `yeni_liman_kaydet` metodu, iletişim kutusundaki bilgilerle yeni bir liman oluşturur ve tabloya ekler.
- `liman_sil` metodu, seçili bir limanı tablodan siler.

#### 6. **KaptanlarEkranı Sınıfı:**

- Bu sınıf, kaptanların görüntülediği bölümün arayüzünü oluşturur.
- `__init__` metodu, arayüzü oluşturmak için `initUI` metodunu çağırır.
- `initUI` metodu, kaptan listesi için bir tablo oluşturur ve eklemek ve silmek için düğmeler ekler.
- `kaptan_ekle_dialog_ac` metodu, yeni bir kaptan eklemek için bir iletişim kutusu açar.
- `yeni_kaptan_kaydet` metodu, iletişim kutusundaki bilgilerle yeni bir kaptan oluşturur ve tabloya ekler.
- `kaptan_sil` metodu, seçili bir kaptanı tablodan siler.

#### 7. **MürettebatEkranı Sınıfı:**

- Bu sınıf, mürettebatın görüntülediği bölümün arayüzünü oluşturur.
- `__init__` metodu, arayüzü oluşturmak için `initUI` metodunu çağırır.
- `initUI` metodu, mürettebat listesi için bir tablo oluşturur ve eklemek ve silmek için düğmeler ekler.
- `murettebat_ekle_dialog_ac` metodu, yeni bir mürettebat eklemek için bir iletişim kutusu açar.
- `yeni_murettebat_kaydet` metodu, iletişim kutusundaki bilgilerle yeni bir mürettebat oluşturur ve tabloya ekler.
- `murettebat_sil` metodu, seçili bir mürettebatı tablodan siler.

- 

## 4.2 Görev dağılımı

- Yapılan her bir eylem görev dağılımı olmaksızın birlikte yapılmıştır.

## 4.3 Karşılaşılan zorluklar ve çözüm yöntemleri

- Arayüz geliştirmek oldukça zorlu bir süreçti fakat dersin hocası bizlere katkı sağlaması açısından bir eğitim videosu yolladı. Farklı bir kütüphane olsa da bizlere fikir verdi ve temelini anladık. İnternette de biraz araştırma yaparak sorunu çözdük.

## 4.4 Proje isterlerine göre eksik yönler

- Projede bazı listelerde 2 tane ekle butonu var ve bunları her kaldırdığımızda sorun yaşadık. Fakat sonrasında bunun kullanım kolaylığını artırdığını fark ettik ve düzenlemeyi bıraktık.

# 5 TEST VE DOĞRULAMA

## 5.1 Yazılımın test süreci

- Test sınıfı oluşturulmadı