



T.C
KOCAELİ SAęLIK VE TEKNOLOJİ ÜNİVERSİTESİ
MÜHENDİSLİK VE DOęA BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR/YAZILIM MÜHENDİSLİęİ

PROJE KONUSU: 2023-2024 BAHAR DÖNEMİ
BLM232/YZM228 PROGRAMLAMA LAB II
ÖDEVİ- VERİ TABANI YÖNETİM SİSTEMLERİ PROJESİ

ÖęRENCİ ADI:Emir Dursun
ÖęRENCİ NUMARASI:220502001
GITHUB: github.com/emirdrs

ÖęRENCİ ADI:Serhat Arslaner
ÖęRENCİ NUMARASI:220502043
GITHUB: github.com/serhatarslaner

DERS SORUMLUSU:
PROF. DR./DR. ÖęR. ÜYESİ Prof. Dr. Nevcihan DURU

TARİH:04.05.2024

1 GİRİŞ

1.1 Projenin amacı

- Projemizdeki amacımız,kendi tercihimiz olan Python yazılım dili ile bir şirketin gemi seferlerini yönetmek için bir veritabanı yönetim sistemi oluşturmak.

2 GEREKSİNİM ANALİZİ

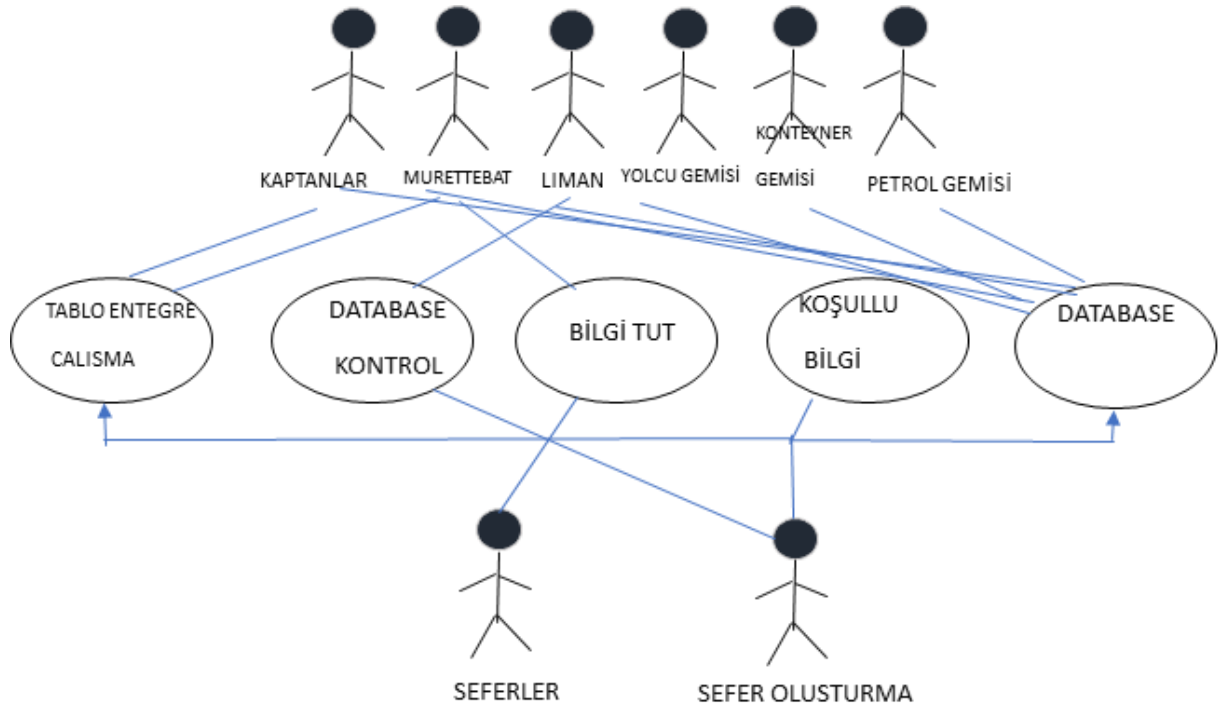
2.1 Arayüz gereksinimleri

- Veritabanı yönetim sistemi ilk açıldığında tüm veri tabanlarına yeni veri girişleri ve veri güncellemeleri yapabileceğimiz bir form ekranı açılacak.
- Sistem ilk açıldığında,veri tabanlarının son halinin bulunduğu pencereye yönlendirecek bir buton bulunacak.
- Sistem açıldığında ilk ekranda veri tabanlarına bölünmüş şekilde girilebilecek değerlerin olduğu bir ekran bulunacak,değerler girildikten sonra EKLE veya GÜNCELLE isminde 2 adet seçenek bulunacak,bu seçeneklerden birisine basıp onaylandığı zaman istenen işlem sağlanacak.
- Aynı şekilde bir veri tabanından istenen ayırıcı değere sahip olan satırı silmek için bir SİL butonu bulunacak.
- Sistemin veri tabanlarını gösterdiği ekranda (2.Sayfa) bir önceki sayfaya geri dönmemizi sağlayacak bir buton bulunacak.

2.2 Fonksiyonel gereksinimler

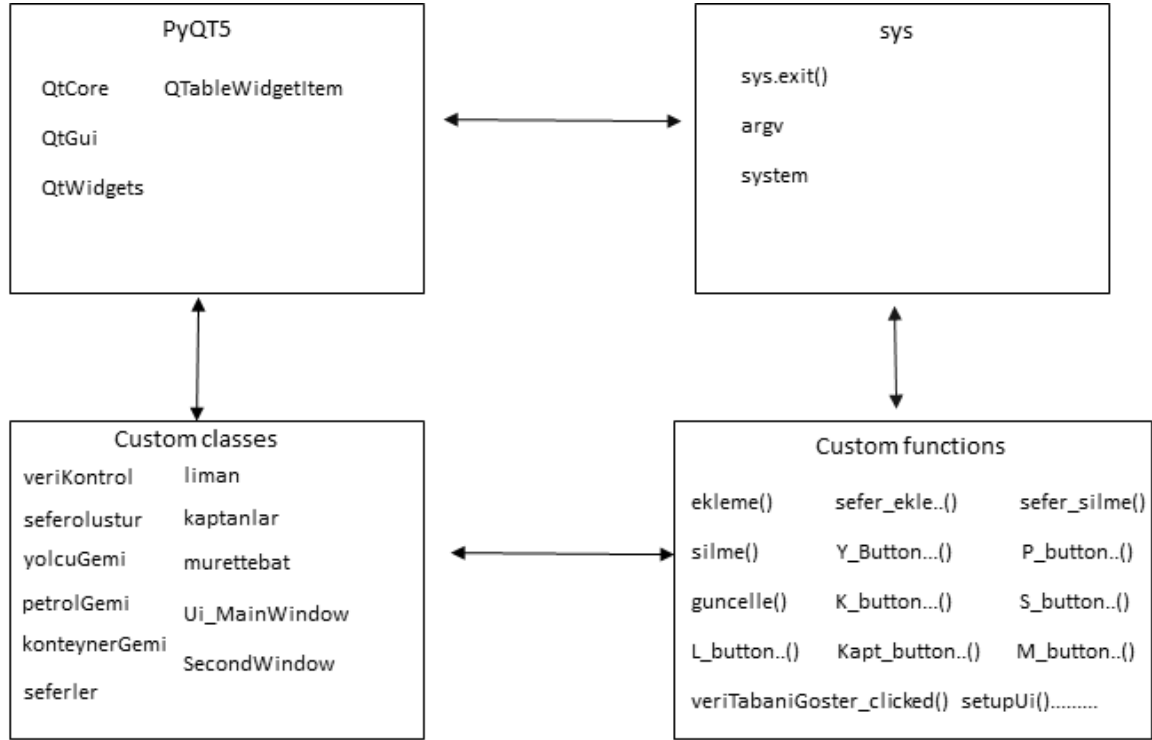
- Silme işleminde primary key kullanılarak silme işlemi yapılacak,örneğin seri_no gibi satırları birbirinden ayıran değerlerle işlem yapılacak.
- Bir limana birden fazla gemi uğrayabilmeli.
- Verilerin veri tabanı ve arayüz ekranı arasındaki yönetim sınıf nesneleri aracılığı ile yapılacak.
- SQL server veri tabanı kullanılmalı.
- Kaptanların lisansları ve mürettebatın hangi görevde çalıştığı bilgisi tutulmalı.
- Gemiler zaman içerisinde farklı seferlere çıkabilir.

2.3 Use-Case diyagramı



3 TASARIM

3.1 Mimari Tasarım



3.2 Kullanılacak teknolojiler

- Python kodlama dilinde yazılacak.
- Python üzerinde arayüz ve font ekranı tasarımı için PyQt5 kütüphanesi.
- PyQt5 ile font ekranı tasarlamamız için QTDesigner.
- Visual Studio Code
- Uygulamayı kapatma kontrolü için sys kütüphanesi.
- SQL veri tabanı işlemlerimiz için sqlite3 kütüphanesi.

3.3 Veritabanı Tasarımı



3.4 Kullanıcı Arayüz Tasarımı

YOLCU GEMİSİ	KONTEYNER GEMİSİ	SEFER OLUŞTUR
seri_no <input type="text"/>	seri_no <input type="text"/>	sefer_id <input type="text"/>
ad <input type="text"/>	ad <input type="text"/>	kaptan_id <input type="text"/>
agirlik <input type="text"/>	agirlik <input type="text"/>	murettabat_id <input type="text"/>
yapim_yili <input type="text"/>	yapim_yili <input type="text"/>	gemi_seri_no <input type="text"/>
yolcu_kapasitesi <input type="text"/>	konteyner_sayisi <input type="text"/>	gidilecek_liman <input type="text"/>
<input type="checkbox"/> EKLE <input type="button" value="ONAYLA"/> <input type="checkbox"/> GUNCELLE	<input type="checkbox"/> EKLE <input type="button" value="ONAYLA"/> <input type="checkbox"/> GUNCELLE	<input type="checkbox"/> EKLE <input type="button" value="ONAYLA"/> <input type="checkbox"/> GUNCELLE
PETROL GEMİSİ	KAPTANLAR	SEFER SİL
seri_no <input type="text"/>	kaptan_id <input type="text"/>	sefer_id <input type="text"/>
ad <input type="text"/>	kaptan_ad <input type="text"/>	<input type="button" value="SİL"/>
agirlik <input type="text"/>	kaptan_soyad <input type="text"/>	
yapim_yili <input type="text"/>	kaptan_adres <input type="text"/>	
petrol_kapasite <input type="text"/>	kaptan_vatandaslik <input type="text"/>	
<input type="checkbox"/> EKLE <input type="button" value="ONAYLA"/> <input type="checkbox"/> GUNCELLE	kaptan_dogumtarihi <input type="text"/>	
	kaptan_iseGiris <input type="text"/>	
	kaptan_lisans <input type="text"/>	
	<input type="checkbox"/> EKLE <input type="button" value="ONAYLA"/> <input type="checkbox"/> GUNCELLE	
MURETTABAT	LIMAN	SEFERLER
murettabat_id <input type="text"/>	liman_adi <input type="text"/>	sefer_id <input type="text"/>
murettabat_ad <input type="text"/>	liman_ukesi <input type="text"/>	yola_cikis_tarihi <input type="text"/>
murettabat_soyad <input type="text"/>	nufus <input type="text"/>	donus_tarihi <input type="text"/>
murettabat_adres <input type="text"/>	pasaport_sorgu <input type="text"/>	cikis_limani <input type="text"/>
murettabat_vatandaslik <input type="text"/>	demirleme_ucret <input type="text"/>	<input type="checkbox"/> EKLE <input type="button" value="ONAYLA"/> <input type="checkbox"/> GUNCELLE
murettabat_dogumtarihi <input type="text"/>	<input type="checkbox"/> EKLE <input type="button" value="ONAYLA"/> <input type="checkbox"/> GUNCELLE	
murettabat_gorev <input type="text"/>		
murettabat_iseGiris <input type="text"/>		
<input type="checkbox"/> EKLE <input type="button" value="ONAYLA"/> <input type="checkbox"/> GUNCELLE		
		sefer_id <input type="text"/>
		<input type="button" value="SİL"/>

Veri tabanı yönetim sistemi ilk açıldığında,kullanıcının karşısına tüm veritabanlarında bulunan verileri güncelleme,yeni veri ekleme ve veri silme işlemlerini yapabileceği bir ekran çıkacak.Bu ekranda yeni bir veri eklemesi örneği(Ekle tuşu seçilip onayla tuşuna basılır.)

YOLCU GEMİSİ	
seri_no	2154
ad	yıldırım
agirlik	100000
yapim_yili	3.01.2009
yolcu_kapasitesi	1200
<input checked="" type="checkbox"/> EKLE	<input type="button" value="ONAYLA"/> <input type="checkbox"/> GUNCELLE

Veri tabanı yönetim sisteminin ilk sayfasında sağda bulunan Veritabanı Göster tuşuna basıldıktan sonra sistemde bulunan tüm veri tabanlarının gösterimi :

VERİTABANI BİLGİ																																																																																												
<div>YOLCU GEMİSİ</div> <table><tr><th>seri_no</th><th>ad</th><th>agirlik</th></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	seri_no	ad	agirlik																												<div>KONTEYNER GEMİSİ</div> <table><tr><th>seri_no</th><th>ad</th><th>agirlik</th></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	seri_no	ad	agirlik																												<div>PETROL GEMİSİ</div> <table><tr><th>seri_no</th><th>ad</th><th>agirlik</th></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	seri_no	ad	agirlik																											
seri_no	ad	agirlik																																																																																										
seri_no	ad	agirlik																																																																																										
seri_no	ad	agirlik																																																																																										
<div>KAPTANLAR</div> <table><tr><th>kaptan_id</th><th>kaptan_ad</th><th>kaptan_soyad</th></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	kaptan_id	kaptan_ad	kaptan_soyad																												<div>SEFERLER</div> <table><tr><th>sefer_id</th><th>yola_cikis_tarihi</th><th>donus_tarihi</th></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	sefer_id	yola_cikis_tarihi	donus_tarihi																												<div>MURETTEBAT</div> <table><tr><th>murettebat_id</th><th>murettebat_ad</th><th>murettebat_soyad</th></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	murettebat_id	murettebat_ad	murettebat_soyad																											
kaptan_id	kaptan_ad	kaptan_soyad																																																																																										
sefer_id	yola_cikis_tarihi	donus_tarihi																																																																																										
murettebat_id	murettebat_ad	murettebat_soyad																																																																																										
<div>→ KONTROL ARAYUZU</div>	<div>LİMANLAR</div> <table><tr><th>liman_adi</th><th>liman_ulkesi</th><th>nufus</th></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	liman_adi	liman_ulkesi	nufus																												<div>SEFER OLUŞTUR</div> <table><tr><th>sefer_id</th><th>kaptan_id</th><th>murettebat_id</th></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	sefer_id	kaptan_id	murettebat_id																																																									
liman_adi	liman_ulkesi	nufus																																																																																										
sefer_id	kaptan_id	murettebat_id																																																																																										

Bu ekranda veritabanlarının güncel hali gösterilir.Geri dönmek için sol alttaki Kontrol Arayüzü butonu kullanılır.

YOLCU GEMİSİ			
	ad	agirlik	yapim_yili
1	yıldırım	100000.0	3.01.2009

seri_no
2154

SIL

[illegible]

Son olarak veri güncelleme işlemi için tekrar her veri tabanı için ayırıcı değişken girilir,sonrasında GÜNCELLE butonu seçilerek onaylaya basılır.

Güncelleme işlemi öncesi veri:

YOLCU GEMİSİ			
	ad	agirlik	yapim_yili
1	yıldırım	100000.0	3.01.2009

Güncelleme işlemi:

YOLCU GEMİSİ	
seri_no	2154
ad	kabasakal
agirlik	160000
yapim_yili	3.11.2019
yolcu_kapasitesi	50000
<input type="checkbox"/> EKLE	<input type="button" value="ONAYLA"/> <input checked="" type="checkbox"/> GÜNCELLE

Onayla tuşuna bastıktan sonraki verinin son hali:

YOLCU GEMİSİ			
	seri_no	ad	agirlik
1	2154	kabasakal	160000.0

Kullanılan kütüphaneler

```
1 import sqlite3
2 import sys
3 from PyQt5 import QtCore, QtGui, QtWidgets
4 from PyQt5.QtWidgets import QApplication, QWidget, QVBoxLayout, QHBoxLayout, QLabel, QLineEdit, QPushButton, QTableWidget, QTableWidgetItem
5
6
```

```
db = sqlite3.connect('gezginGemiSirketi.db')

db.execute('''CREATE TABLE IF NOT EXISTS VOLCUGEMISI
(
    seri_no INTEGER PRIMARY KEY,
    ad TEXT NOT NULL,
    agirlik REAL NOT NULL,
    yapim_yili INTEGER NOT NULL,
    yolcu_kapasitesi INTEGER
)''')

db.execute('''CREATE TABLE IF NOT EXISTS PETROLGEMISI
(
    seri_no INTEGER PRIMARY KEY,
    ad TEXT NOT NULL,
    agirlik REAL NOT NULL,
    yapim_yili INTEGER NOT NULL,
    petrol_kapasite INTEGER
)''')

db.execute('''CREATE TABLE IF NOT EXISTS KONTEYNERGEMISI
(
    seri_no INTEGER PRIMARY KEY,
    ad TEXT NOT NULL,
    agirlik REAL NOT NULL,
    yapim_yili INTEGER NOT NULL,
    konteyner_sayisi INTEGER,
    maks_agirlik INTEGER
)''')

db.execute('''CREATE TABLE IF NOT EXISTS SEFERLER
(
    sefer_id INTEGER PRIMARY KEY AUTOINCREMENT,
    yola_cikis_tarihi TEXT NOT NULL,
    donus_tarihi TEXT NOT NULL,
    cikis_limani TEXT NOT NULL
)''')
```

DATABASE OLUSTURMA -1

```

db.execute('''CREATE TABLE IF NOT EXISTS LIMANLAR
(
    liman_adi TEXT ,
    liman_ulkesi TEXT ,
    nufus INTEGER NOT NULL,
    pasaport_sorgu TEXT NOT NULL,
    demirleme_ucret INTEGER NOT NULL,
    PRIMARY KEY(liman_adi,liman_ulkesi)
)''')

db.execute('''CREATE TABLE IF NOT EXISTS KAPTANLAR
(
    kaptan_id INTEGER PRIMARY KEY ,
    kaptan_ad TEXT NOT NULL,
    kaptan_soyad TEXT NOT NULL,
    kaptan_adres TEXT NOT NULL,
    kaptan_vatandaslik TEXT NOT NULL,
    kaptan_dogum_tarihi INTEGER NOT NULL,
    kaptan_iseGiris INTEGER NOT NULL,
    kaptan_lisans TEXT NOT NULL
)''')

db.execute('''CREATE TABLE IF NOT EXISTS MURETTABAT
(
    murettebat_id INTEGER PRIMARY KEY ,
    murettebat_ad TEXT NOT NULL,
    murettebat_soyad TEXT NOT NULL,
    murettebat_adres TEXT NOT NULL,
    murettebat_vatandaslik TEXT NOT NULL,
    murettebat_dogum_tarihi INTEGER NOT NULL,
    murettebat_gorev TEXT NOT NULL,
    murettebat_iseGiris INTEGER NOT NULL
)''')

db.execute('''CREATE TABLE IF NOT EXISTS SEFEROLUSTUR
(
    sefer_id INTEGER PRIMARY KEY ,
    kaptan_id TEXT NOT NULL,
    murettebat_id TEXT NOT NULL,
    gemi_seri_no TEXT NOT NULL,
    gidilecek_liman TEXT NOT NULL
)''')

db.commit()

```

DATABASE OLUSTURMA - 2

```

class yolcuGemi(veriKontrol):
    def ekleme(self,seri_no,ad,agirlik,yapim_yili,yolcu_kapasitesi):
        try:
            self.sorgu.execute('insert into YOLCUGEMISI values(?,?,?,?,?)',(seri_no,ad,agirlik,yapim_yili,yolcu_kapasitesi))
            self.db.commit()
        except sqlite3.IntegrityError:
            pass

    def silme(self,seri_no):
        self.sorgu.execute("Delete from YOLCUGEMISI where seri_no=?", (seri_no,))
        self.db.commit()

    def guncelle(self,seri_no,ad,agirlik,yapim_yil,yolcu_kapastesi):
        self.sorgu.execute("UPDATE YOLCUGEMISI SET ad=?,agirlik=?,yapim_yili=?,yolcu_kapasitesi=? WHERE seri_no=? ", (ad,agirlik,yapim_yil,yolcu_kapastesi,seri_no))
        self.db.commit()

```

YOLCU GEMİSİ CLASS - VERİTABANINA EKLEME – SİLME – GÜNCELLEME İŞLEMLERİNİ YAPAR.

```

class petrolGemi(veriKontrol):
    def ekleme(self,seri_no,ad,agirlik,yapim_yili,petrol_kapasite):
        try:
            self.sorgu.execute('insert into PETROLGEMISI values(?,?,?,?,?)',(seri_no,ad,agirlik,yapim_yili,petrol_kapasite))
            self.db.commit()
        except sqlite3.IntegrityError:
            pass

    def silme(self,seri_no):
        self.sorgu.execute("Delete from PETROLGEMISI where seri_no=?", (seri_no,))
        self.db.commit()

    def guncelle(self,seri_no,ad,agirlik,yapim_yili,petrol_kapasite):
        self.sorgu.execute("UPDATE PETROLGEMISI SET ad=?,agirlik=?,yapim_yili=?,petrol_kapasite=? WHERE seri_no=? ", (ad,agirlik,yapim_yili,petrol_kapasite,seri_no))
        self.db.commit()

```

PETROL GEMİSİ CLASS – VERİTABANINA EKLEME-SİLME-GÜNCELLEME İŞLEMLERİNİ YAPAR.

```

class konteynerGemi(veriKontrol):
    def ekleme(self,seri_no,ad,agirlik,yapim_yili,konteyner_sayisi,maks_agirlik):
        try:
            self.sorgu.execute('insert into KONTEYNERGEMISI values(?,?,?,?,?)',(seri_no,ad,agirlik,yapim_yili,konteyner_sayisi,maks_agirlik))
            self.db.commit()
        except sqlite3.IntegrityError:
            pass

    def silme(self,seri_no):
        self.sorgu.execute("Delete from KONTEYNERGEMISI where seri_no=?", (seri_no,))
        self.db.commit()

    def guncelle(self,seri_no,ad,agirlik,yapim_yili,konteyner_sayisi,maks_agirlik):
        self.sorgu.execute("UPDATE KONTEYNERGEMISI SET ad=?,agirlik=?,yapim_yili=?,konteyner_sayisi=?,maks_agirlik=? WHERE seri_no=? ",(ad,agirlik,yapim_yili,konteyner_sayisi,maks_agirlik,seri_no))
        self.db.commit()

```

KONTEYNER GEMİSİ CLASS – VERİTABANINA EKLEME-SİLME- GÜNCELLEME-İŞLEMLERİNİ YAPAR.

```

class kaptanlar(veriKontrol):
    def ekleme(self,kaptan_id,kaptan_ad,kaptan_soyad,kaptan_adres,kaptan_vatandaslik,kaptan_dogum_tarihi,kaptan_iseGiris,kaptan_lisans):
        try:
            self.sorgu.execute('insert into KAPTANLAR values(?,?,?,?,?,?,?)',(kaptan_id,kaptan_ad,kaptan_soyad,kaptan_adres,kaptan_vatandaslik,kaptan_dogum_tarihi,kaptan_iseGiris,kaptan_lisans))
            self.db.commit()
        except sqlite3.IntegrityError:
            pass

    def silme(self,kaptan_id):
        self.sorgu.execute("Delete from KAPTANLAR where kaptan_id=?", (kaptan_id,))
        self.db.commit()

    def guncelle(self,kaptan_id,kaptan_ad,kaptan_soyad,kaptan_adres,kaptan_vatandaslik,kaptan_dogum_tarihi,kaptan_iseGiris,kaptan_lisans):
        self.sorgu.execute("UPDATE KAPTANLAR SET kaptan_ad=?,kaptan_soyad=?,kaptan_adres=?,kaptan_vatandaslik=?,kaptan_dogum_tarihi=?,kaptan_iseGiris=?,kaptan_lisans=? WHERE kaptan_id=? ",(kaptan_ad,kaptan_soyad,kaptan_adres,kaptan_vatandaslik,kaptan_dogum_tarihi,kaptan_iseGiris,kaptan_lisans,kaptan_id))
        self.db.commit()

```

KAPTAN CLASS – VERİTABANINA EKLEME- SİLME-GÜNCELLEME-İŞLEMLERİNİ YAPAR.

```

class seferolustur(veriKontrol):
    def ekleme(self,sefer_id,kaptan_id,murettebat_id,gemi_seri_no,gidilecek_liman):
        try:
            self.sorgu.execute('insert into SEFEROLUSTUR values(?,?,?,?,?)',(sefer_id,kaptan_id,murettebat_id,gemi_seri_no,gidilecek_liman))
            self.db.commit()
        except sqlite3.IntegrityError:
            pass

    def silme(self,sefer_id):
        self.sorgu.execute("Delete from SEFEROLUSTUR where sefer_id=?", (sefer_id,))
        self.db.commit()

    def guncelle(self,sefer_id,kaptan_id,murettebat_id,gemi_seri_no,gidilecek_liman):
        self.sorgu.execute("UPDATE SEFEROLUSTUR SET kaptan_id=?,murettebat_id=?,gemi_seri_no=?,gidilecek_liman=? WHERE sefer_id=? ",(kaptan_id,murettebat_id,gemi_seri_no,gidilecek_liman,sefer_id))
        self.db.commit()

```

SEFER OLUŞTURMA CLASS – VERİTABANINA EKLEME SİLME VE GÜNCELLEME İŞLEMLERİNİ YAPAR.

```

class liman(veriKontrol):
    def ekleme(self,liman_adi,liman_ulkesi,nufus,pasaport_sorgu,demirleme_ucret):
        try:
            self.sorgu.execute('insert into LIMANLAR values(?,?,?,?,?)',(liman_adi,liman_ulkesi,nufus,pasaport_sorgu,demirleme_ucret))
            self.db.commit()
        except sqlite3.IntegrityError:
            pass

    def silme(self,liman_adi,liman_ulkesi):
        self.sorgu.execute("Delete from LIMANLAR where liman_adi=?,liman_ulkesi=?", (liman_adi,liman_ulkesi,))
        self.db.commit()

    def guncelle(self,liman_adi,liman_ulkesi,nufus,pasaport_sorgu,demirleme_ucret):
        self.sorgu.execute("UPDATE LIMANLAR SET liman_ulkesi=?,nufus=?,pasaport_sorgu=?,demirleme_ucret=? WHERE liman_adi=? ",(liman_ulkesi,nufus,pasaport_sorgu,demirleme_ucret,liman_adi))
        self.db.commit()

```

LİMAN CLASS – VERİTABANINA EKLEME SİLME VE GÜNCELLEME İŞLEMLERİNİ YAPAR.


```

class seferler(veriKontrol):
    def ekleme(self,sefer_id,yola_cikis_tarihi,donus_tarihi,cikis_limani):
        try:
            self.sorgu.execute('insert into SEFERLER values(?,?,?,?)',(sefer_id,yola_cikis_tarihi,donus_tarihi,cikis_limani))
            self.db.commit()
        except sqlite3.IntegrityError:
            pass

    def silme(self,sefer_id):
        self.sorgu.execute("Delete from SEFERLER where sefer_id=?", (sefer_id,))
        self.db.commit()

    def guncelle(self,sefer_id,yola_cikis_tarihi,donus_tarihi,cikis_limani):
        self.sorgu.execute("UPDATE SEFERLER SET yola_cikis_tarihi=?,donus_tarihi=?,cikis_limani=? WHERE sefer_id=? ", (yola_cikis_tarihi,donus_tarihi,cikis_limani,sefer_id))
        self.db.commit()

```

SEFERLER CLASS – VERİTABANINA EKLEME, SİLME VE GÜNCELLEME İŞLEMLERİNİ YAPAR.

```

class veriKontrol():
    def __init__(self,tableName):
        self.db = sqlite3.connect('gezginGemiSirketi.db')
        self.sorgu = self.db.cursor()
        self.tableName = tableName

    def listele(self):
        listeleme = self.sorgu.execute('select * from {}'.format(self.tableName)).fetchall()
        return listeleme

```

VERİKONTROL CLASS - DATABASE ÜZERİNDEN GEREKLİ VERİLERİ TABLOYA ATAMA YAPMAMIZI SAĞLAR

```

class murettebat(veriKontrol):
    def ekleme(self,murettebat_id,murettebat_ad,murettebat_soyad,murettebat_adres,murettebat_vatandaslik,murettebat_dogum_tarihi,murettebat_gorev,murettebat_iseGiris):
        try:
            self.sorgu.execute('insert into MURETTABAT values(?,?,?,?,?,?,?)',(murettebat_id,murettebat_ad,murettebat_soyad,murettebat_adres,murettebat_vatandaslik,murettebat_dogum_tarihi,murettebat_gorev,murettebat_iseGiris))
            self.db.commit()
        except sqlite3.IntegrityError:
            pass

    def silme(self,murettebat_id):
        self.sorgu.execute("Delete from MURETTABAT where murettebat_id=?", (murettebat_id,))
        self.db.commit()

    def guncelle(self,murettebat_id,murettebat_ad,murettebat_soyad,murettebat_adres,murettebat_vatandaslik,murettebat_dogum_tarihi,murettebat_gorev,murettebat_iseGiris):
        self.sorgu.execute("UPDATE MURETTABAT SET murettebat_ad=?,murettebat_soyad=?,murettebat_adres=?,murettebat_vatandaslik=?,murettebat_dogum_tarihi=?,murettebat_gorev=?,murettebat_iseGiris=? where murettebat_id=?", (murettebat_ad,murettebat_soyad,murettebat_adres,murettebat_vatandaslik,murettebat_dogum_tarihi,murettebat_gorev,murettebat_iseGiris,murettebat_id))

```

MURETTABAT CLASS – EKLEME, SİLME VE GUNCELLEME İŞLEMLERİNİ YAPAR

```

seferekletable=seferolustur("SEFEROLUSTUR")
yolcutable=yolcuGemi("YOLCUGEMISI")
konteynertable=konteynerGemi("KONTEYNERGEMISI")
petroltable=petrolGemi("PETROLGEMISI")
sefertable=seferler("SEFERLER")
limantable=liman("LIMANLAR")
kaptantable=kaptanlar("KAPTANLAR")
murettebattable=murettebat("MURETTABAT")

```

TABLO OBJELERİ OLUŞTURMA

```

def setupUi(self, MainWindow):
    MainWindow.setObjectName("MainWindow")
    MainWindow.resize(1116, 890)
    MainWindow.setCursor(QtGui.QCursor(QtCore.Qt.ArrowCursor))
    self.centralwidget = QtWidgets.QWidget(MainWindow)
    self.centralwidget.setObjectName("centralwidget")
    self.label_2 = QtWidgets.QLabel(self.centralwidget)
    self.label_2.setGeometry(QtCore.QRect(640, 230, 81, 16))
    self.label_2.setObjectName("label_2")
    self.label_3 = QtWidgets.QLabel(self.centralwidget)
    self.label_3.setGeometry(QtCore.QRect(630, 0, 121, 31))
    self.label_3.setObjectName("label_3")
    self.label_4 = QtWidgets.QLabel(self.centralwidget)
    self.label_4.setGeometry(QtCore.QRect(590, 520, 61, 20))
    self.label_4.setObjectName("label_4")
    self.label_5 = QtWidgets.QLabel(self.centralwidget)
    self.label_5.setGeometry(QtCore.QRect(140, 520, 101, 16))
    self.label_5.setObjectName("label_5")
    self.label_6 = QtWidgets.QLabel(self.centralwidget)
    self.label_6.setGeometry(QtCore.QRect(90, 230, 111, 16))
    self.label_6.setObjectName("label_6")
    self.label_7 = QtWidgets.QLabel(self.centralwidget)
    self.label_7.setGeometry(QtCore.QRect(1000, 530, 81, 16))
    self.label_7.setObjectName("label_7")
    self.label_8 = QtWidgets.QLabel(self.centralwidget)
    self.label_8.setGeometry(QtCore.QRect(100, 10, 71, 16))
    self.label_8.setObjectName("label_8")
    self.label_10 = QtWidgets.QLabel(self.centralwidget)
    self.label_10.setGeometry(QtCore.QRect(10, 40, 47, 13))
    self.label_10.setObjectName("label_10")
    self.label_11 = QtWidgets.QLabel(self.centralwidget)
    self.label_11.setGeometry(QtCore.QRect(10, 70, 21, 16))
    self.label_11.setObjectName("label_11")
    self.label_12 = QtWidgets.QLabel(self.centralwidget)
    self.label_12.setGeometry(QtCore.QRect(10, 100, 47, 13))
    self.label_12.setObjectName("label_12")
    self.label_13 = QtWidgets.QLabel(self.centralwidget)
    self.label_13.setGeometry(QtCore.QRect(0, 130, 51, 16))
    self.label_13.setObjectName("label_13")
    self.label_14 = QtWidgets.QLabel(self.centralwidget)
    self.label_14.setGeometry(QtCore.QRect(0, 160, 81, 16))
    self.label_14.setObjectName("label_14")
    self.seri_noYolcu = QtWidgets.QLineEdit(self.centralwidget)
    self.seri_noYolcu.setGeometry(QtCore.QRect(80, 40, 113, 16))
    self.seri_noYolcu.setObjectName("seri_noYolcu")
    self.adYolcu = QtWidgets.QLineEdit(self.centralwidget)

```

Pyqt5 MAIN PENCERE OLUŞTURMA.

```

def K_button_click_ekle_guncelle(self,seri_no,ad,agirlik,yapim_yili,konteyner_sayisi,maks_agirlik,):

    if self.konteynerEkleCheck.isChecked() and self.konteynerGuncelleCheck.isChecked():
        return

    if self.konteynerEkleCheck.isChecked():

        konteynertable.ekleme(seri_no,ad,agirlik,yapim_yili,konteyner_sayisi,maks_agirlik)

    if self.konteynerGuncelleCheck.isChecked():
        konteynertable.guncelle(seri_no,ad,agirlik,yapim_yili,konteyner_sayisi,maks_agirlik)

def K_button_click_sil(self,seri_no):

    konteynertable.silme(seri_no)

```

KONTEYNER GEMİSİ İÇİN TIKLAMA İŞLEMİ YARDIMI İLE ARAYÜZ ÜZERİNDEN VERİTABANINA VERİ EKLEME SİLME VE GÜNCELLEME İŞLEMLERİ.

```

def L_button_click_ekle_guncelle(self,liman_adi,liman_ulkesi,nufus,pasaport_sorgu,demirleme_ucret):

    if self.limanEkleCheck.isChecked() and self.limanGuncelleCheck.isChecked():
        return

    if self.limanEkleCheck.isChecked():

        limantable.ekleme(liman_adi,liman_ulkesi,nufus,pasaport_sorgu,demirleme_ucret)

    if self.limanGuncelleCheck.isChecked():
        limantable.guncelle(liman_adi,liman_ulkesi,nufus,pasaport_sorgu,demirleme_ucret)

def L_button_click_sil(self,liman_adi,liman_ulkesi):

    limantable.silme(liman_adi,liman_ulkesi)

```

LİMANLAR İÇİN TIKLAMA İŞLEMİ YARDIMI İLE ARAYÜZ ÜZERİNDEN VERİTABANINA VERİ EKLEME SİLME VE GÜNCELLEME İŞLEMLERİ.

```

def Kapt_button_click_ekle_guncelle(self,kaptan_id,kaptan_ad,kaptan_soyad,kaptan_adres,kaptan_vatandaslik,kaptan_dogum_tarihi,kaptan_iseGiris,kaptan_lisans):

    if self.kaptanEkleCheck.isChecked() and self.kaptanGuncelleCheck.isChecked():
        return

    if self.kaptanEkleCheck.isChecked():
        #Kaptanlar
        kaptantable.ekleme(kaptan_id,kaptan_ad,kaptan_soyad,kaptan_adres,kaptan_vatandaslik,kaptan_dogum_tarihi,kaptan_iseGiris,kaptan_lisans)

    if self.kaptanGuncelleCheck.isChecked():
        kaptantable.guncelle(kaptan_id,kaptan_ad,kaptan_soyad,kaptan_adres,kaptan_vatandaslik,kaptan_dogum_tarihi,kaptan_iseGiris,kaptan_lisans)

def Kapt_button_click_sil(self,kaptan_id):

    kaptantable.silme(kaptan_id)

```

KAPTANLAR İÇİN TIKLAMA İŞLEMİ YARDIMI İLE ARAYÜZ ÜZERİNDEN VERİTABANINA VERİ EKLEME SİLME VE GÜNCELLEME İŞLEMLERİ.

```

def M_button_click_ekle_guncelle(self,murettebat_id,murettebat_ad,murettebat_soyad,murettebat_adres,murettebat_vatandaslik,murettebat_dogum_tarihi,murettebat_gorev,murettebat_lisans):

    if self.murettebatEkleCheck.isChecked() and self.murettebatGuncelleCheck.isChecked():
        return

    if self.murettebatEkleCheck.isChecked():
        murettebattable.ekleme(murettebat_id,murettebat_ad,murettebat_soyad,murettebat_adres,murettebat_vatandaslik,murettebat_dogum_tarihi,murettebat_gorev,murettebat_lisans)

    if self.murettebatGuncelleCheck.isChecked():
        murettebattable.guncelle(murettebat_id,murettebat_ad,murettebat_soyad,murettebat_adres,murettebat_vatandaslik,murettebat_dogum_tarihi,murettebat_gorev,murettebat_lisans)

def M_button_click_sil(self,murettebat_id):

    murettebattable.silme(murettebat_id)

def veriTabaniGoster_clicked(self):
    self.second_window = SecondWindow()

```

MURETTEBAT İÇİN TIKLAMA İŞLEMİ YARDIMI İLE ARAYÜZ ÜZERİNDEN VERİTABANINA VERİ EKLEME SİLME VE GÜNCELLEME İŞLEMLERİ.

```
def Y_button_click_ekle_guncelle(self,seri_no,ad,agirlik,yapim_yil,yolcu_kapasitesi):  
    if self.yolcuEkleCheck.isChecked() and self.yolcuGuncelleCheck.isChecked():  
        return  
  
    if self.yolcuEkleCheck.isChecked():  
        yolcutable.ekleme(seri_no,ad,agirlik,yapim_yil,yolcu_kapasitesi)  
    if self.yolcuGuncelleCheck.isChecked():  
        yolcutable.guncelle(seri_no,ad,agirlik,yapim_yil,yolcu_kapasitesi)  
  
def Y_button_click_sil(self,seri_no):  
    yolcutable.silme(seri_no)
```

**YOLCU GEMİSİ İÇİN TIKLAMA İŞLEMİ
YARDIMI İLE ARAYÜZ ÜZERİNDEN
VERİTABANINA VERİ EKLEME SİLME VE
GÜNCELLEME İŞLEMLERİ.**

```
def S_button_click_ekle_guncelle(self,sefer_id,yola_cikis_tarihi,donus_tarihi,cikis_limani):  
    if self.seferEkleCheck.isChecked() and self.seferGuncelleCheck.isChecked():  
        return  
  
    if self.seferEkleCheck.isChecked():  
        sefertable.ekleme(sefer_id,yola_cikis_tarihi,donus_tarihi,cikis_limani)  
  
    if self.seferGuncelleCheck.isChecked():  
        sefertable.guncelle(sefer_id,yola_cikis_tarihi,donus_tarihi,cikis_limani)  
  
def S_button_click_sil(self,seri_no):  
    sefertable.silme(seri_no)
```

**SEFER BİLGİLERİ İÇİN TIKLAMA İŞLEMİ
YARDIMI İLE ARAYÜZ ÜZERİNDEN
VERİTABANINA VERİ EKLEME SİLME VE
GÜNCELLEME İŞLEMLERİ.**

```

class Ui_MainWindow(object):

    def sefer_ekle_click_guncelle(self,sefer_id,kaptan_id,murettebat_id,gemi_seri_no,gidilecek_liman):

        if self.seferOlusturEkleC.isChecked() and self.seferOlusturGuncelleC.isChecked():
            return

        if self.seferOlusturEkleC.isChecked():

            seferekletable.ekleme(sefer_id,kaptan_id,murettebat_id,gemi_seri_no,gidilecek_liman)
        if self.seferOlusturGuncelleC.isChecked():

            seferekletable.guncelle(sefer_id,kaptan_id,murettebat_id,gemi_seri_no,gidilecek_liman)

    def sefer_silme(self,sefer_id):

        seferekletable.silme(sefer_id)

```

SEFER OLUŞTURMA İÇİN TIKLAMA İŞLEMİ YARDIMI İLE ARAYÜZ ÜZERİNDEN VERİTABANINA VERİ EKLEME SİLME VE GÜNCELLEME İŞLEMLERİ .

```

def P_button_click_ekle_guncelle(self,seri_no,ad,agirlik,yapim_yili,petrol_kapasite):

    if self.petrolEkleCheck.isChecked() and self.petrolGuncelleCheck.isChecked():
        return

    if self.petrolEkleCheck.isChecked():

        petroltable.ekleme(seri_no,ad,agirlik,yapim_yili,petrol_kapasite)

    if self.petrolGuncelleCheck.isChecked():

        petroltable.guncelle(seri_no,ad,agirlik,yapim_yili,petrol_kapasite)

def P_button_click_sil(self,seri_no):

    petroltable.silme(seri_no)

```

PETROL GEMİSİ İÇİN TIKLAMA İŞLEMİ YARDIMI İLE ARAYÜZ ÜZERİNDEN VERİTABANINA VERİ EKLEME SİLME VE GÜNCELLEME İŞLEMLERİ.

```

def tablo_guncelleyici(self):
    yolcuverisi=yolcutable.listele()
    if yolcuverisi !=[]:

        for row_index,row_data in enumerate(yolcuverisi):

            self.yolcuGemisiVeritabani.insertRow(self.yolcuGemisiVeritabani.rowCount())
            for column_index,veri in enumerate(row_data):
                item=QTableWidgetItem(str(veri))
                self.yolcuGemisiVeritabani.setItem(row_index,column_index,item)

    konteynerG=konteynertable.listele()
    if konteynerG !=[]:

        for row_index,row_data in enumerate(konteynerG):

            self.konteynerGVeritabani.insertRow(self.konteynerGVeritabani.rowCount())
            for column_index,veri in enumerate(row_data):
                item=QTableWidgetItem(str(veri))
                self.konteynerGVeritabani.setItem(row_index,column_index,item)

    petrolG=petroltable.listele()
    if petrolG !=[]:

        for row_index,row_data in enumerate(petrolG):

            self.petrolGemisiVeritabani.insertRow(self.petrolGemisiVeritabani.rowCount())
            for column_index,veri in enumerate(row_data):
                item=QTableWidgetItem(str(veri))
                self.petrolGemisiVeritabani.setItem(row_index,column_index,item)

    seferT=sefertable.listele()
    if seferT !=[]:

        for row_index,row_data in enumerate(seferT):

            self.seferlerVeritabani.insertRow(self.seferlerVeritabani.rowCount())
            for column_index,veri in enumerate(row_data):
                item=QTableWidgetItem(str(veri))
                self.seferlerVeritabani.setItem(row_index,column_index,item)

```

**VERİTABANINA KAYDEDİLEN VERİLERİN
AYNI ZAMANDA ARAYÜZ ÜZERİNDE
VERİTABANI HAKKINDA BİLGİ VEREN
SAYFASI (2. PENCERE) İÇİN TABLO
GÜNCELLEME FONKSİYONLARI VE
METODLARI.**


```

limanT=limantable.listele()
if limanT !=[]:

    for row_index,row_data in enumerate(limanT):

        self.limanVeritabani.insertRow(self.limanVeritabani.rowCount())
        for column_index,veri in enumerate(row_data):
            item=QTableWidgetItem(str(veri))
            self.limanVeritabani.setItem(row_index,column_index,item)

kaptanT=kaptantable.listele()
if kaptanT !=[]:

    for row_index,row_data in enumerate(kaptanT):

        self.kaptanVeritabani.insertRow(self.kaptanVeritabani.rowCount())
        for column_index,veri in enumerate(row_data):
            item=QTableWidgetItem(str(veri))
            self.kaptanVeritabani.setItem(row_index,column_index,item)

murettebatT=murettebattable.listele()
if murettebatT !=[]:

    for row_index,row_data in enumerate(murettebatT):

        self.murettebatVeritabani.insertRow(self.murettebatVeritabani.rowCount())
        for column_index,veri in enumerate(row_data):
            item=QTableWidgetItem(str(veri))
            self.murettebatVeritabani.setItem(row_index,column_index,item)

seferOlusturT = seferekletable.listele()
if seferOlusturT !=[]:

    for row_index,row_data in enumerate(seferOlusturT):

        self.seferOlusturVeritabani.insertRow(self.seferOlusturVeritabani.rowCount())
        for column_index,veri in enumerate(row_data):
            item=QTableWidgetItem(str(veri))
            self.seferOlusturVeritabani.setItem(row_index,column_index,item)

```

**VERİTABANINA KAYDEDİLEN VERİLERİN
AYNI ZAMANDA ARAYÜZ ÜZERİNDE
VERİTABANI HAKKINDA BİLGİ VEREN
SAYFASI (2. PENCERE) İÇİN TABLO
GÜNCELLEME FONKSİYONLARI VE
METODLARI.**

```

def __init__(self):
    super().__init__()

    MainWindow.resize(1025, 835)
    self.centralwidget = QtWidgets.QWidget(MainWindow)
    self.centralwidget.setObjectName("centralwidget")
    self.label = QtWidgets.QLabel(self.centralwidget)
    self.label.setGeometry(QtCore.QRect(20, -40, 851, 131))
    font = QtGui.QFont()
    font.setPointSize(16)
    font.setBold(True)
    font.setWeight(75)
    self.label.setFont(font)
    self.label.setObjectName("label")
    self.yolcuGemisiVeriTabani = QtWidgets.QTableWidget(self.centralwidget)
    self.yolcuGemisiVeriTabani.setGeometry(QtCore.QRect(30, 90, 301, 192))
    self.yolcuGemisiVeriTabani.setObjectName("yolcuGemisiVeriTabani")
    self.yolcuGemisiVeriTabani.setColumnCount(5)
    self.yolcuGemisiVeriTabani.setRowCount(0)
    item = QtWidgets.QTableWidgetItem()
    self.yolcuGemisiVeriTabani.setHorizontalHeaderItem(0, item)
    item = QtWidgets.QTableWidgetItem()
    self.yolcuGemisiVeriTabani.setHorizontalHeaderItem(1, item)
    item = QtWidgets.QTableWidgetItem()
    self.yolcuGemisiVeriTabani.setHorizontalHeaderItem(2, item)
    item = QtWidgets.QTableWidgetItem()
    self.yolcuGemisiVeriTabani.setHorizontalHeaderItem(3, item)
    item = QtWidgets.QTableWidgetItem()
    self.yolcuGemisiVeriTabani.setHorizontalHeaderItem(4, item)
    self.konteynerGVeritabani = QtWidgets.QTableWidget(self.centralwidget)
    self.konteynerGVeritabani.setGeometry(QtCore.QRect(360, 90, 291, 192))
    self.konteynerGVeritabani.setObjectName("konteynerGVeritabani")
    self.konteynerGVeritabani.setColumnCount(6)
    self.konteynerGVeritabani.setRowCount(0)
    item = QtWidgets.QTableWidgetItem()
    self.konteynerGVeritabani.setHorizontalHeaderItem(0, item)
    item = QtWidgets.QTableWidgetItem()
    self.konteynerGVeritabani.setHorizontalHeaderItem(1, item)
    item = QtWidgets.QTableWidgetItem()
    self.konteynerGVeritabani.setHorizontalHeaderItem(2, item)

```

**SecondWindow (Arayüz Üzerinde ki 2. Pencere)
için tasarım kodları arayüz oluşturma.**

```

self.konteynerGVeritabani.setHorizontalHeaderItem(5, item)
self.petrolGemisiVeritabani = QtWidgets.QTableWidget(self.centralwidget)
self.petrolGemisiVeritabani.setGeometry(QtCore.QRect(690, 90, 291, 192))
self.petrolGemisiVeritabani.setObjectName("petrolGemisiVeritabani")
self.petrolGemisiVeritabani.setColumnCount(5)
self.petrolGemisiVeritabani.setRowCount(0)
item = QtWidgets.QTableWidgetItem()
self.petrolGemisiVeritabani.setHorizontalHeaderItem(0, item)
item = QtWidgets.QTableWidgetItem()
self.petrolGemisiVeritabani.setHorizontalHeaderItem(1, item)
item = QtWidgets.QTableWidgetItem()
self.petrolGemisiVeritabani.setHorizontalHeaderItem(2, item)
item = QtWidgets.QTableWidgetItem()
self.petrolGemisiVeritabani.setHorizontalHeaderItem(3, item)
item = QtWidgets.QTableWidgetItem()
self.petrolGemisiVeritabani.setHorizontalHeaderItem(4, item)
self.kaptanVeritabani = QtWidgets.QTableWidget(self.centralwidget)
self.kaptanVeritabani.setGeometry(QtCore.QRect(30, 340, 301, 192))
self.kaptanVeritabani.setObjectName("kaptanVeritabani")
self.kaptanVeritabani.setColumnCount(8)
self.kaptanVeritabani.setRowCount(0)
item = QtWidgets.QTableWidgetItem()
self.kaptanVeritabani.setHorizontalHeaderItem(0, item)
item = QtWidgets.QTableWidgetItem()
self.kaptanVeritabani.setHorizontalHeaderItem(1, item)
item = QtWidgets.QTableWidgetItem()
self.kaptanVeritabani.setHorizontalHeaderItem(2, item)
item = QtWidgets.QTableWidgetItem()
self.kaptanVeritabani.setHorizontalHeaderItem(3, item)
item = QtWidgets.QTableWidgetItem()
self.kaptanVeritabani.setHorizontalHeaderItem(4, item)
item = QtWidgets.QTableWidgetItem()
self.kaptanVeritabani.setHorizontalHeaderItem(5, item)
item = QtWidgets.QTableWidgetItem()
self.kaptanVeritabani.setHorizontalHeaderItem(6, item)
item = QtWidgets.QTableWidgetItem()
self.kaptanVeritabani.setHorizontalHeaderItem(7, item)
self.seferlerVeritabani = QtWidgets.QTableWidget(self.centralwidget)
self.seferlerVeritabani.setGeometry(QtCore.QRect(360, 340, 291, 192))
self.seferlerVeritabani.setObjectName("seferlerVeritabani")
self.seferlerVeritabani.setColumnCount(4)
self.seferlerVeritabani.setRowCount(0)
item = QtWidgets.QTableWidgetItem()

```

**SecondWindow (Arayüz Üzerinde ki 2. Pencere)
için tasarım kodları arayüz oluşturma.**

```

def retranslateUi(self, MainWindow):

    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))
    self.label.setText(_translate("MainWindow", " "
VERITABANI BILGI"))
    item = self.yolcuGemisiVeritabani.horizontalHeaderItem(0)
    item.setText(_translate("MainWindow", "seri_no"))
    item = self.yolcuGemisiVeritabani.horizontalHeaderItem(1)
    item.setText(_translate("MainWindow", "ad"))
    item = self.yolcuGemisiVeritabani.horizontalHeaderItem(2)
    item.setText(_translate("MainWindow", "agirlik"))
    item = self.yolcuGemisiVeritabani.horizontalHeaderItem(3)
    item.setText(_translate("MainWindow", "yapim_yili"))
    item = self.yolcuGemisiVeritabani.horizontalHeaderItem(4)
    item.setText(_translate("MainWindow", "yolcu_kapasitesi"))
    item = self.konteynerGVeritabani.horizontalHeaderItem(0)
    item.setText(_translate("MainWindow", "seri_no"))
    item = self.konteynerGVeritabani.horizontalHeaderItem(1)
    item.setText(_translate("MainWindow", "ad"))
    item = self.konteynerGVeritabani.horizontalHeaderItem(2)
    item.setText(_translate("MainWindow", "agirlik"))
    item = self.konteynerGVeritabani.horizontalHeaderItem(3)
    item.setText(_translate("MainWindow", "yapim_yili"))
    item = self.konteynerGVeritabani.horizontalHeaderItem(4)
    item.setText(_translate("MainWindow", "konteyner_sayisi"))
    item = self.konteynerGVeritabani.horizontalHeaderItem(5)
    item.setText(_translate("MainWindow", "maks_agirlik"))
    item = self.petrolGemisiVeritabani.horizontalHeaderItem(0)
    item.setText(_translate("MainWindow", "seri_no"))
    item = self.petrolGemisiVeritabani.horizontalHeaderItem(1)
    item.setText(_translate("MainWindow", "ad"))
    item = self.petrolGemisiVeritabani.horizontalHeaderItem(2)
    item.setText(_translate("MainWindow", "agirlik"))
    item = self.petrolGemisiVeritabani.horizontalHeaderItem(3)
    item.setText(_translate("MainWindow", "yapim_yili"))
    item = self.petrolGemisiVeritabani.horizontalHeaderItem(4)
    item.setText(_translate("MainWindow", "petrol_kapasite"))
    item = self.kaptanVeritabani.horizontalHeaderItem(0)
    item.setText(_translate("MainWindow", "kaptan_id"))
    item = self.kaptanVeritabani.horizontalHeaderItem(1)
    item.setText(_translate("MainWindow", "kaptan_ad"))
    item = self.kaptanVeritabani.horizontalHeaderItem(2)
    item.setText(_translate("MainWindow", "kaptan_soyad"))
    item = self.kaptanVeritabani.horizontalHeaderItem(3)
    item.setText(_translate("MainWindow", "kaptan_adres"))
    item = self.kaptanVeritabani.horizontalHeaderItem(4)

```

**SecondWindow (Arayüz Üzerinde ki 2. Pencere)
için tasarım kodları arayüz oluşturma. –
retranslateUI -**

```
def kontrolArayuzu_clicked(self):  
    self.MainWindow = QtWidgets.QMainWindow()  
    self.ui = Ui_MainWindow()  
    self.ui.setupUi(MainWindow)
```

**VERİTABANI BİLGİ (2. PENCERE)
ÜZERİNDEN 1. PENCEREYE GEÇİŞ
YAPMAMIZI SAĞLAR**

```
if __name__ == "__main__":  
    app = QtWidgets.QApplication(sys.argv)  
    MainWindow = QtWidgets.QMainWindow()  
    ui = Ui_MainWindow()  
    ui.setupUi(MainWindow)  
    MainWindow.show()  
    sys.exit(app.exec_())
```

**UYGULAMANIN ÇALIŞMASI İÇİN GEREKLİ
OLAN KODLAR**

4 UYGULAMA

4.3 Kodlanan bileşenlerin açıklamaları

- Verikontrol,seferolustur,yolcuGemi,petrolGemi,konteynerGemi Seferler,liman,kaptanlar ve murettabat classları içerisinde bulunan ekleme,silme ve güncelleme fonksiyonları sayesinde veritabanına veri aktarımı yapabiliyoruz.
 - Ui_MainWindow classının içerisinde bulunan sefer_ekle_click_guncelle, sefer_silme, Y_button_click_ekle_guncelle , Y_button_click_sil, K_button_click_ekle_guncelle, K_button_click_sil, P_button_click_ekle_guncelle gibi fonksiyonlar sayesinde arayüz üzerinde bulunan ekle güncelle checkboxların yönetimini sağlıyoruz.
 - MainWindow ve SecondWindow Classları sayesinde Arayüz Tasarımını gerçekleştiriyoruz.
 - kontrolArayuzu_clicked fonksiyonu sayesinde veritabanı bilgisi arayüzü üzerinde ki kontrolArayuzune tıklayarak MainWindowa geçiş yapmamızı sağlar.
 - veriTabaniGoster_clicked fonksiyonu sayesinde veritabanı bilgi arayüzüne erişim sağlıyoruz.

4.4 Görev dağılımı

- Bileşenlerinin tasarım ve geliştirme aşamalarında Kütüphanelerin öğrenilmesi , Algoritma hazırlanması ve deneme yöntemleri ortak gerçekleştirilmiştir.
- Raporun hazırlanması süreçlerinde ortak çalışma yapılmıştır.

4.5 Karşılaşılan zorluklar ve çözüm yöntemleri

- MainWindow üzerinden SecondWindowa geçiş işlemi – pyqt5 kütüphanesinin gerekli metodları öğrenilip düzeltilmiştir.
- Database üzerinde ki verilerin aynı zamanda arayüz üzerinde bir tablo üzerinden görüntüleme – pyqt5 kütüphanesinin gerekli metodları uygulanıp gerekli işlemlerle desteklenip uygulanmıştır.

4.6 Proje isterlerine göre eksik yönler

- Liman veritabanı üzerinde silme işlemi sadece liman_adı üzerinden gerçekleştirilmektedir.

5 TEST VE DOĞRULAMA

5.3 Yazılımın test süreci

- Proje kapsamında Yazılan Tüm Sınıflar ve fonksiyonlar test edilmiştir

5.4 Yazılımın doğrulanması

- Proje kapsamında Testi Yapılan Sınıflar ve Fonksiyonlar üzerinde tespit edilen anormallikler giderilmiştir.

```
test = petrolGemi()  
test.ekleme  
test.guncelle  
test.silme  
test.listele
```

```
test1 = seferler()  
test1.ekleme  
test1.guncelle  
test1.silme  
test1.listele  
test2 = konteynerGemi()  
test2.ekleme  
test2.guncelle  
test2.silme  
test2.listele
```

```
test3 = kapitanlar()  
test3.ekleme  
test3.guncelle  
test3.silme  
test3.listele
```

```
test4 = yolcuGemi()  
test4.ekleme  
test4.guncelle  
test4.silme  
test4.listele
```

```
test5 = liman()  
test5.ekleme  
test5.guncelle  
test5.silme  
test5.listele
```

```
test6 = seferolustur()  
test6.ekleme  
test6.guncelle  
test6.silme  
test6.listele
```

```
test7 = murettebat()  
test7.ekleme  
test7.guncelle  
test7.silme  
test7.listele
```

CLASSLARIN METODLARI TEST EDİLMİŞTİR.