



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

**PROJE KONUSU:**  
**VERİ TABANI YÖNETİM SİSTEMLERİ PROJESİ**

**ÖęRENCİ ADI:**  
**DAMLA KEKLİK**  
**ECENAZ BOZDOęAN**  
**ÖęRENCİ NUMARASI:**  
**230501033**  
**220502040**

**DERS SORUMLUSU:**  
**DR. ÖęR. ÜYESİ NUR BANU ALBAYRAK**

**TARİH:04.05.2024**

# 1 GİRİŞ

## 1.1 Projenin amacı

Gemilerin, limanların, kaptanların ve mürettebatın bilgilerini saklamak ve bu bilgilere erişebilmek için bir veritabanı oluşturmak projenin temel amacıdır. Proje, kullanıcıların gemi ve sefer bilgilerini eklemesine, güncellemesine, silmesine ve sorgulamasına olanak sağlar ve veritabanı bağlantısı kurarak, kullanıcıların veritabanındaki bilgilere erişimini sağlar ve bu bilgiler üzerinde işlemler yapabilmesini sağlar.

### Gerçekleştirilmesi Beklenenler:

- 1.Gemilerin, limanların, kaptanların ve mürettebatın bilgilerini veritabanında saklamak için uygun tabloların oluşturulması.
- 2.Kullanıcı arayüzüyle etkileşimli bir şekilde, gemi ve sefer bilgilerini eklemek, güncellemek, silmek ve sorgulamak için işlevlerin geliştirilmesi ve bu arayüzün kullanıcı dostu olması.
- 3.Veritabanı işlemlerinde oluşabilecek hataların kullanıcıya uygun şekilde gösterilmesi ve işlenmesi.
- 4.Veritabanı bağlantısının güvenli bir şekilde kurulması ve veritabanı işlemlerinin gerçekleştirilmesi için gerekli olan kütüphanelerin (örneğin: pyodbc) kullanılması.
- 5.Projenin modüler bir yapıda geliştirilmesi ve kod tekrarının önlenmesi için uygun fonksiyonlar ve sınıfların kullanılması.

## 2 GEREKSİNİM ANALİZİ

### 2.1 Arayüz gereksinimleri

Kullanıcıların gemi ve sefer bilgilerini görüntülemesini, eklemesini, güncellemesini ve silebilmesini sağlayan bir ana menü.

İşlem sonuçları ekranı: Kullanıcının gerçekleştirdiği işlemlerin sonuçlarını görebileceği bir ekran.

### Donanım Arayüzü Gereksinimleri:

Standart bir bilgisayar veya dizüstü bilgisayar.

Fare ve klavye gibi standart giriş aygıtları.

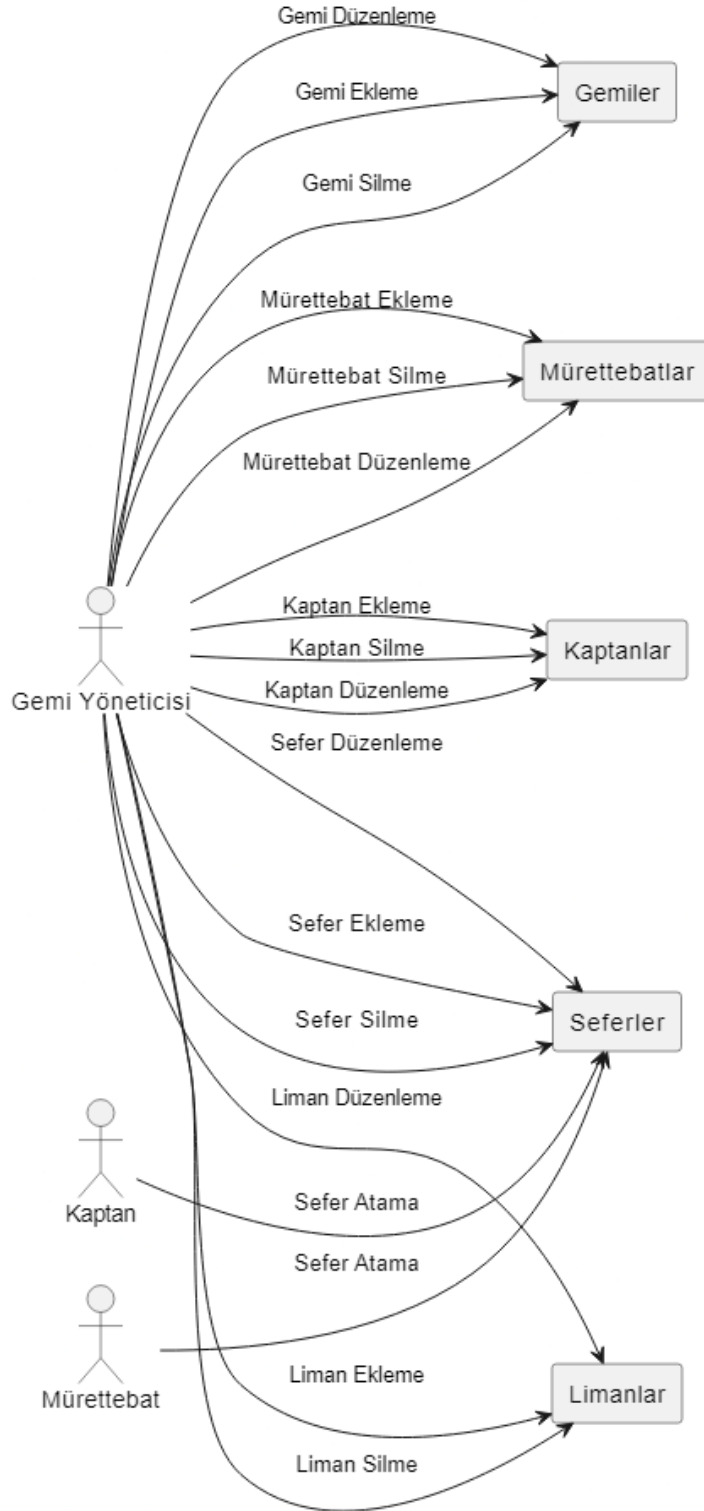
Ekran boyutunun uyumluluğu: Kullanıcı arayüzü ekran boyutuna uygun olmalı.

### 2.2 Fonksiyonel Gereksinimler

Kullanıcı, yeni gemi bilgilerini sisteme ekleyebilmelidir; varolan gemi bilgilerini güncelleyebilmelidir; sistemdeki gemi bilgilerini listeleyebilmelidir; belirli bir gemiye ait bilgileri silme işlemi yapabilmelidir; yeni sefer bilgilerini sisteme ekleyebilmelidir; varolan sefer bilgilerini güncelleyebilmelidir; sistemdeki sefer bilgilerini listeleyebilmelidir; belirli bir sefere ait bilgileri silme işlemi yapabilmelidir; yeni liman bilgilerini sisteme ekleyebilmelidir; varolan liman bilgilerini güncelleyebilmelidir; sistemdeki liman bilgilerini listeleyebilmelidir; belirli bir limana ait bilgileri silme işlemi yapabilmelidir; yeni kaptan bilgilerini sisteme ekleyebilmelidir; varolan kaptan bilgilerini

güncelleyebilmelidir; sistemdeki kaptan bilgilerini listeleyebilmelidir; belirli bir kaptanın bilgilerini silme işlemi yapabilmelidir; yeni mürettebat bilgilerini sisteme ekleyebilmelidir; varolan mürettebat bilgilerini güncelleyebilmelidir; sistemdeki mürettebat bilgilerini listeleyebilmelidir; belirli bir mürettebatın bilgilerini silme işlemi yapabilmelidir.

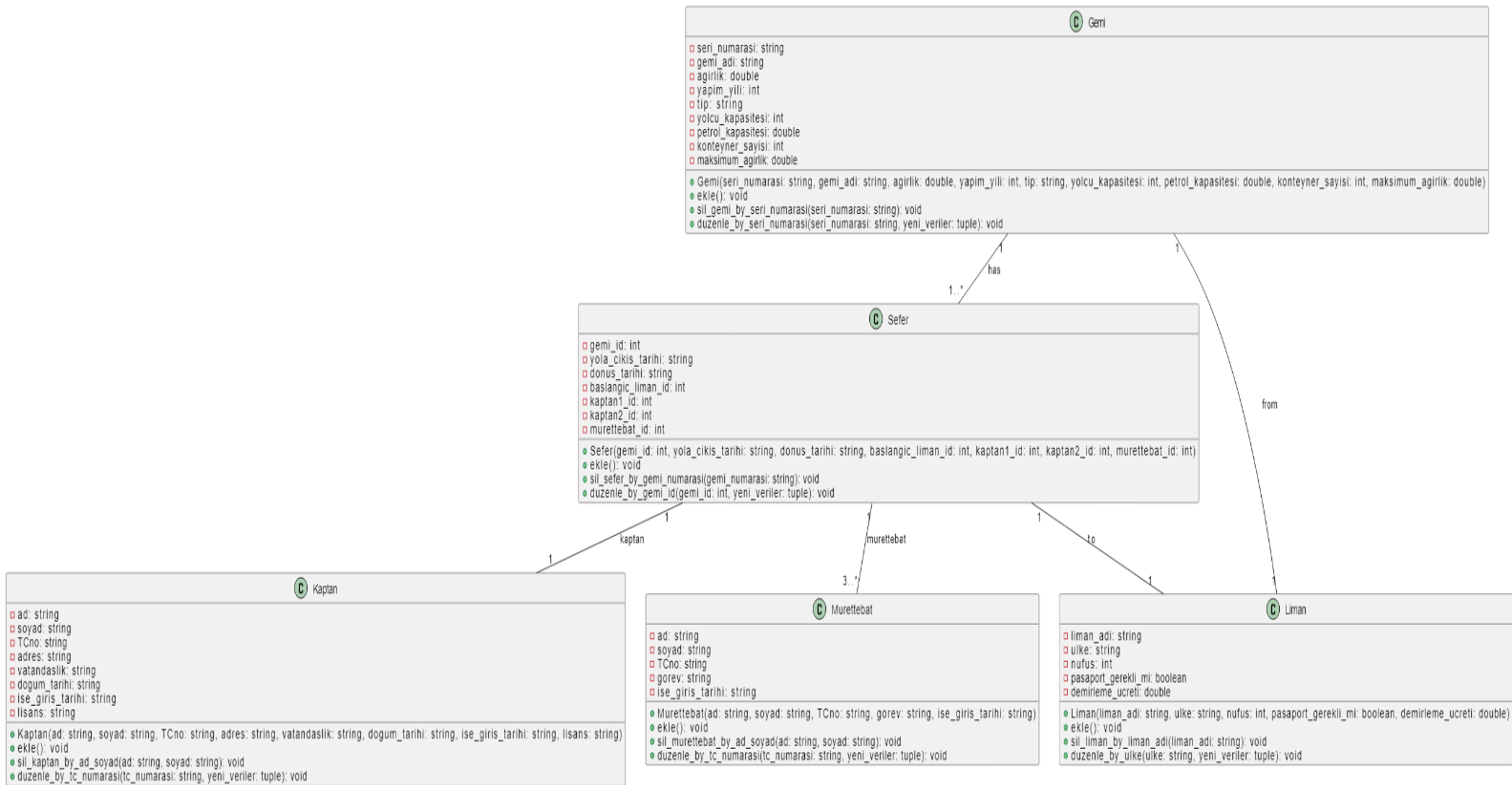
## 2.3 Use-Case diyagramı

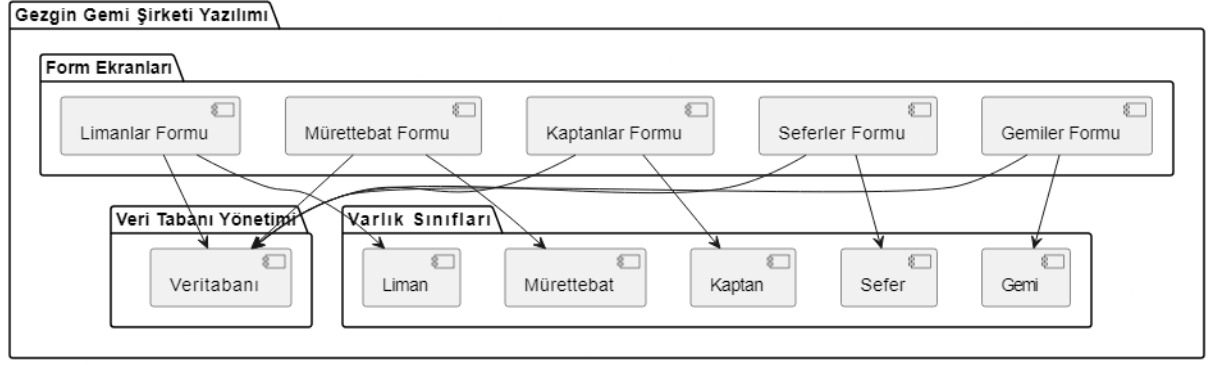


## 3 TASARIM

### 3.1 Mimari tasarım

Projeyi kullanıcı arayüzü, iş mantığı ve veritabanı erişimi gibi farklı katmanlara ayırarak daha düzenli ve bakımı kolay hale gelmesini sağlayan katmanlı mimari kullandık. Kullanıcı arayüzü, iş mantığı ve veritabanı etkileşimleri MVC tasarım desenine uygun olacak şekilde ayırdık. Veritabanı işlemleri için özel bir katman kullandık. Bu katman, veritabanı bağlantısını yönetecek ve veritabanı işlemlerini gerçekleştirecek. İş mantığı kurallarını içeren bir servis katmanı oluşturduk. Bu katman, kullanıcı arayüzünden gelen istekleri işleyecek ve gerektiğinde veritabanı işlemlerini gerçekleştirdi. Kullanıcı arayüzü, kullanıcıların sisteme erişip etkileşime girebilecekleri bir arayüz sağlayacak. Bu, kullanıcıların gemi, liman, kaptan, mürettebat ve sefer gibi verilere erişmelerini sağlıyor. Ayrıca, projenin gerektirdiği veritabanı tabloları ve ilişkilerini tasarladık. Bu, verilerin nasıl saklanacağını ve ilişkilendirileceğini belirlerken veri bütünlüğünü korur. Bağımlılık enjeksiyonu kullanarak bileşenler arasındaki bağımlılıkları minimize edeceğiz, böylece bileşenlerin test edilebilirliğini artırdık.





## 3.2 Kullanılacak teknolojiler

Yazılım Dil ve Çevreleri:

Python

SQL

Harici Kütüphaneler:

pyodbc

tkinter

unittest

Diğer Teknolojiler:

Microsoft SQL Server

Git

## 3.3 Veri tabanı tasarımı

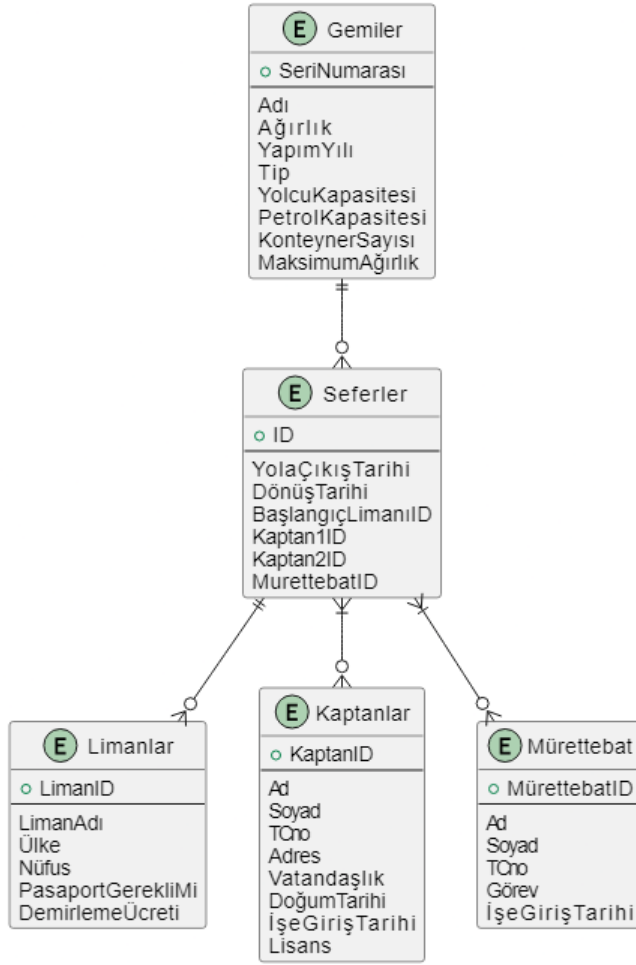
**Tablolar ve Alanlar:** Projenin gereksinimlerine uygun olarak her bir veri tipi için ayrı bir tablo oluşturulur. Gemiler için bir tablo, limanlar için başka bir tablo oluşturulur. Her tablonun içinde ilgili alanlar bulunur. Örneğin, gemi tablosunda seri numarası, gemi adı, ağırlık gibi alanlar bulunur.

**İlişkiler:** Tablolar arasındaki ilişkiler doğru bir şekilde tanımlanır. Örneğin, gemi tablosu ile sefer tablosu arasında bir ilişki vardır. Bu ilişki, gemi tablosundaki bir geminin sefer tablosundaki kayıtlarını referans alır.

**Anahtarlar:** Her tabloda birincil anahtarlar (primary key) tanımlanır. Bu, her kaydın benzersiz bir şekilde tanımlanmasını sağlar. Ayrıca, tablolar arasındaki ilişkiler için dış anahtarlar (foreign key) belirlenir.

**Normalizasyon:** Veri tabanı normalizasyon prensiplerine uygun bir şekilde tasarlanır. Bu, veri bütünlüğünü sağlamak için gereken en uygun yapıya sahip olmasını sağlar.

**Endeksler:** Performansı artırmak için tablolara uygun endeksler eklenir. Bu, veriye erişimin hızlanmasına ve sorgu performansının iyileştirilmesine yardımcı olur.



### 3.4 Kullanıcı arayüzü tasarımı

Arayüz, kullanıcıların kolayca erişebileceği ve kullanabileceği şekilde tasarlanmıştır. Menüler, düğmeler ve giriş alanları gibi bileşenler, kullanıcıların işlemleri hızlıca gerçekleştirmesini sağlayacak şekilde düzenlenmiştir. Arayüzdeki bilgiler, önem sırasına göre düzenlenmiştir. Kullanıcılar için en önemli bilgiler ve işlevler, daha erişilebilirdir..

**Yazılımı çalıştırmak için kullanıcılar genellikle şu adımları izler:**

- 1.Yazılımı başlatmak için uygun bir komut veya simgeye tıklama.
- 2.Eğer gerekiyorsa, kullanıcı girişi yapma veya diğer başlangıç ayarlarını yapma.
- 3.Ana ekran veya başlangıç ekranı görüntülenir.
- 4.Kullanıcılar arayüz üzerinden istedikleri işlemleri gerçekleştirirler. Bu işlemler, menülerden seçim yapmayı, düğmelere tıklamayı veya giriş alanlarına veri girmeyi içerebilir.
- 5.İşlemler tamamlandıktan sonra, kullanıcılar genellikle bir onay ile karşılaşır veya işlemin sonucunu gösteren bir ileti alırlar.
- 6.Yazılım, kullanıcıların çıkış yapmasına veya başka bir işlem gerçekleştirmesine izin verir.

Gemiler ve Seferler Yönetim Sistemi

### Gemi İşlemleri

Seri Numarası:

Gemi Adı:

Ağırlık (ton):

Yapım Yılı:

Tip:

Yolcu Kapasitesi:

Petrol Kapasitesi (ton):

Konteyner Sayısı:

Maksimum Ağırlık (ton):

### Liman İşlemleri

Liman Adı:

Ülke:

Nüfus:

Pasaport Gerekli mi:

Demirleme Ücreti:

### Sefer İşlemleri

Gemi ID:

Yola Çıkış Tarihi:

Dönüş Tarihi:

Başlangıç Liman ID:

Kaptan 1 ID:

Kaptan 2 ID:

Mürettebat ID:

### Kaptan İşlemleri

Ad:

Soyad:

TC Numarası:

Adres:

Vatandaşlık:

Doğum Tarihi (YYYY-MM-DD):

İşe Giriş Tarihi (YYYY-MM-DD):

Lisans:

### Mürettebat İşlemleri

Ad:

Soyad:

TC Numarası:

Görev:

İşe Giriş Tarihi (YYYY-MM-DD):

Kullanıcı arayüzü tasarımında da belirtildiği gibi arayüz kullanıcının anlayabileceği şekilde ve önem sırasında yapılmıştır.

## 4 UYGULAMA

### 4.1 Kodlanan bileşenlerin açıklamaları

#### DatabaseManager Sınıfı:

init(): Veritabanı bağlantısı ve imleci başlatır.

connect(): Veritabanına bağlanmayı sağlar.

disconnect(): Veritabanı bağlantısını kapatır.

execute\_query(): Veritabanına sorgu gönderir ve sonuçları işler.

ekle(): Yeni gemi eklemek için veritabanına sorgu gönderir.

#### Gemi Sınıfı:

init(): Gemi özelliklerini başlatır.

ekle(): Yeni bir gemi eklemek için veritabanına sorgu gönderir.

sil\_gemi\_by\_seri\_numarasi(): Seri numarasına göre gemi silmek için veritabanına sorgu gönderir.

duzenle\_by\_seri\_numarasi(): Seri numarasına göre gemi bilgilerini güncellemek için veritabanına sorgu gönderir.

### **Sefer Sınıfı:**

init(): Sefer özelliklerini başlatır.

ekle(): Yeni bir sefer eklemek için veritabanına sorgu gönderir.

sil\_sefer\_by\_gemi\_numarasi(): Gemi numarasına göre sefer silmek için veritabanına sorgu gönderir.

duzenle\_by\_gemi\_id(): Gemi ID'sine göre sefer bilgilerini güncellemek için veritabanına sorgu gönderir.

### **Liman Sınıfı:**

init(): Liman özelliklerini başlatır.

ekle(): Yeni bir liman eklemek için veritabanına sorgu gönderir.

sil\_liman\_by\_liman\_adi(): Liman adına göre limanı silmek için veritabanına sorgu gönderir.

duzenle\_by\_ulke(): Ülkeye göre liman bilgilerini güncellemek için veritabanına sorgu gönderir.

### **Kaptan Sınıfı:**

init(): Kaptan özelliklerini başlatır.

ekle(): Yeni bir kaptan eklemek için veritabanına sorgu gönderir.

sil\_kaptan\_by\_ad\_soyad(): Ad ve soyada göre kaptanı silmek için veritabanına sorgu gönderir.

duzenle\_by\_tc\_numarasi(): TC numarasına göre kaptan bilgilerini güncellemek için veritabanına sorgu gönderir.

### **Murettebat Sınıfı:**

init(): Mürettebat özelliklerini başlatır.

ekle(): Yeni bir mürettebat üyesi eklemek için veritabanına sorgu gönderir.

sil\_murettebat\_by\_ad\_soyad(): Ad ve soyada göre mürettebat üyesini silmek için veritabanına sorgu gönderir.

duzenle\_by\_tc\_numarasi(): TC numarasına göre mürettebat bilgilerini güncellemek için veritabanına sorgu gönderir.

## **4.2 Görev dağılımı**

### **Bileşenlerin tasarım ve geliştirme aşamalarındaki görev dağılımı:**

Yapabilmek için kendi içimizde görev dağılımı yaptık ve öyle gerçekleştirdik.

### **Raporun hazırlanması sürecindeki görev dağılımı:**

Raporda da şemaları ve açıklama görevini farklı kişilere verdik ve bu görev dağılımını yaptık.

## **4.3 Karşılaşılan zorluklar ve çözüm yöntemleri**

Veritabanı bağlantısı sırasında beklenmedik hatalar veya bağlantı kesilmeleri yaşandı. Bu durumdan ötürü Microsoft SQL Server 2019 kullandık çünkü son sürüm kullandığımızda bağlantı hataları oluşabildi.

Hataları bulmak ve onlara uygun çözümleri bulmak zorlayıcı oldu ama üstesinden geldik.



Performans optimizasyonu için veritabanı indekslerinin düzenli olarak kontrol edilmesi, sorguların optimize edilmesi ve gereksiz kod parçalarının kaldırılmasını sağladık.

#### **4.4 Proje isterlerine göre eksik yönler**

Projenin test kısmında testin sonucunun terminalde verilmesi uzun sürüyor.

### **TEST VE DOĞRULAMA**

#### **4.4 Yazılımın test süreci**

Test uygulaması, database\_operations modülünde tanımlı olan Gemi, Sefer, Liman, Kaptan ve Murettebat sınıflarının metodlarını test etmek için oluşturulmuştur. Her bir test metodunda, ilgili sınıfın bir örneği oluşturulmakta ve ilgili metotlar çağrılarak işlemler gerçekleştirilmektedir.

Örneğin, test\_gemi\_ekle metodu, Gemi sınıfının ekle metodunu test etmektedir. Bu metodda, bir gemi örneği oluşturulmakta ve ekle metodu çağrılarak bir gemi verisinin veritabanına eklenip eklenmediği kontrol edilmektedir.

Benzer şekilde, test\_sefer\_ekle metodu, Sefer sınıfının ekle metodunu test etmektedir. Bu metodda, bir sefer örneği oluşturulmakta ve ekle metodu çağrılarak bir sefer verisinin veritabanına eklenip eklenmediği kontrol edilmektedir.

Diğer test metodları da aynı şekilde, ilgili sınıfların metodlarını test etmek için kullanılmaktadır. Bu testler sayesinde, database\_operations modülündeki bileşenlerin doğru şekilde çalışıp çalışmadığı kontrol edilmektedir.

Test uygulaması tekrar tekrar test etmeye imkan tanıyacak şekilde geliştirilmiştir.

#### **4.5 Yazılımın doğrulanması**

Test uygulaması kullanılarak yapılan testler sonucunda yazılımın doğruluğu sağlanmıştır. Her bir test metodunun başarıyla tamamlanması, ilgili bileşenlerin (sınıfların) beklenen şekilde çalıştığını göstermektedir. Çıktı olarak başarılı olunca başarılı olduğunu olmayınca da olmadığını terminalde belirtir ve başarısız olunca bununla ilgili açıklamayı da terminalde yapmaktadır.

<https://github.com/ajnasss/project5.git>

<https://github.com/Damla-keklik/proje2.git>