



T.C

**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
YAZILIM MÜHENDİSLİĞİ**

Veri Tabanı Yönetim Sistemleri Projesi

Mert Bülbül

220502006

<https://github.com/Mertb2627>

Can Şafak Çakır

220502002

<https://github.com/cakirz>

DERS SORUMLUSU

Dr. Öğr. Üyesi Nur Banu ALBAYRAK

Projenin amacı, Gezgin Gemi Şirketi'nin seferlerini, gemilerini, limanlarını, kaptanlarını ve mürettebatını yönetmek için bir yazılım geliştirmektir. Bu yazılım, şirketin operasyonlarını planlamasını ve izlemesini sağlayacaktır.

Projenin gerçekleştirilmesi beklenenler:

Seferlerin yönetimi.

Limanların yönetimi.

Kaptan ve mürettebatın yönetimi.

Veri tabanı ve form ekranları arasındaki veri yönetimi.

2. GEREKSİNİM ANALİZİ

Kullanıcı Arayüzü Gereksinimleri:

Kullanıcıların gemileri, seferleri, limanları, kaptanları ve mürettebatı ekleyebilecekleri bir ekran oluşturulmalıdır.

Kullanıcılar, limanlara gemi ekleyebilmeli ve limanlar hakkında bilgi ekleyebilmelidir.

Gemi seferleri planlanırken, kullanıcılar seferin başlangıç ve bitiş tarihlerini, hangi geminin hangi seferde olduğunu ve hangi limanlarda durulacağını belirtebilmelidir.

Donanım Arayüzü Gereksinimleri:

Yazılımın çalışması için standart bir bilgisayar yeterlidir.

Veritabanı yönetimi için uygun bir SQL veritabanı sunucusu gereklidir.

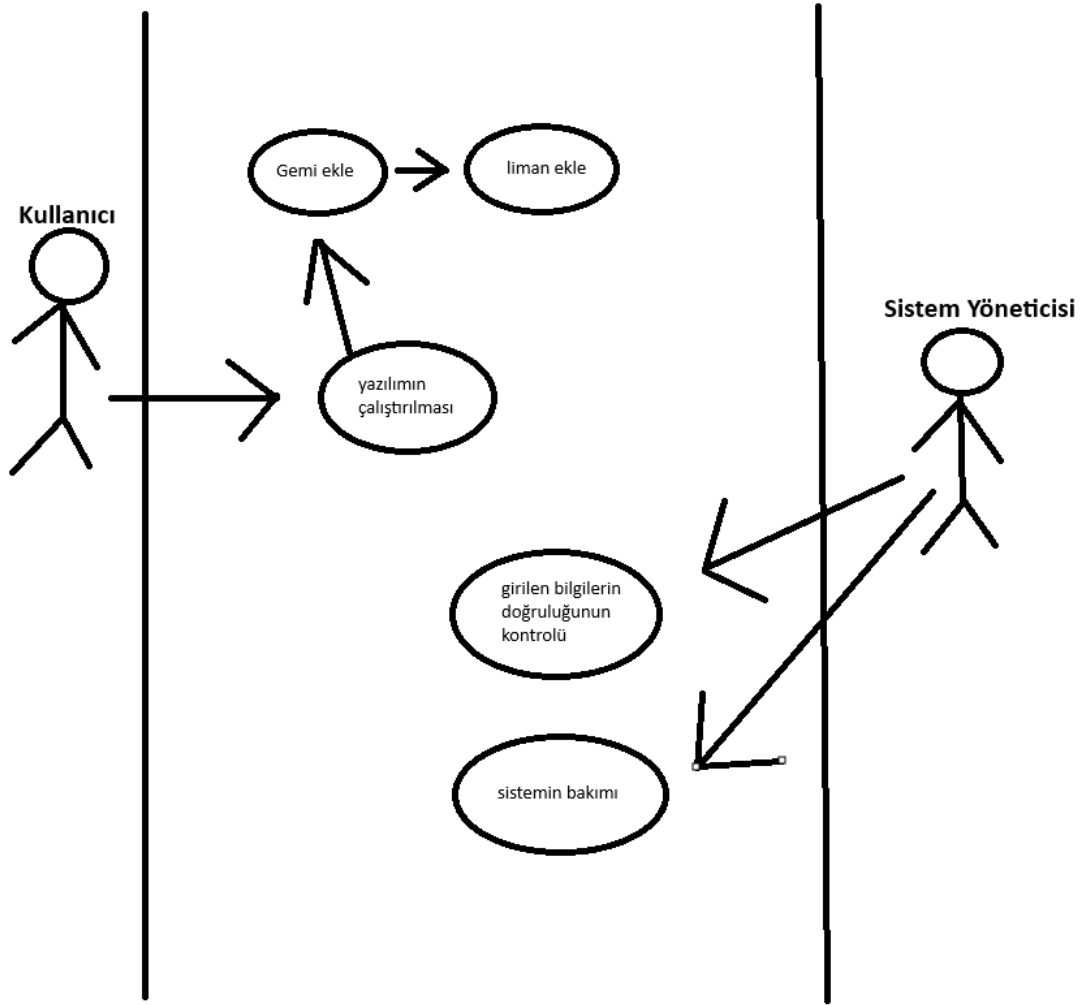
Fonksiyonel Gereksinimler

Kullanıcılar, gemileri, seferleri, limanları, kaptanları ve mürettebatı ekleyebilmeli, düzenleyebilmeli ve silebilmelidir.

Seferlerin planlanması sırasında, her sefer için gerekli olan gemi, kaptan ve mürettebat bilgileri sağlanmalıdır.

Limanlar için eklenen bilgilerin doğruluğu sağlanmalıdır.

Kullanıcılar, ekledikleri verileri kolayca görüntüleyebilmeli ve gerektiğinde raporlar alabilmelidir.



3. TASARIM

3.1 Mimari Tasarım

Yazılım, katmanlı mimari kullanarak tasarlanacaktır. Bu katmanlar genellikle kullanıcı arayüzü, iş mantığı ve veritabanı erişimini içerecektir.

Kullanıcı arayüzü, otomasyon için gerekli sınıflar ve veritabanı işlemleri ayrı modüller olacak şekilde tasarlanacaktır.

Veritabanı erişimi için ORM (Object-Relational Mapping) kullanılacaktır.

3.2 Kullanılacak Teknolojiler

Yazılım python kullanılarak tasarlanacak ve sqllite tkinter gibi kütüphanelerden yararlanılacaktır.

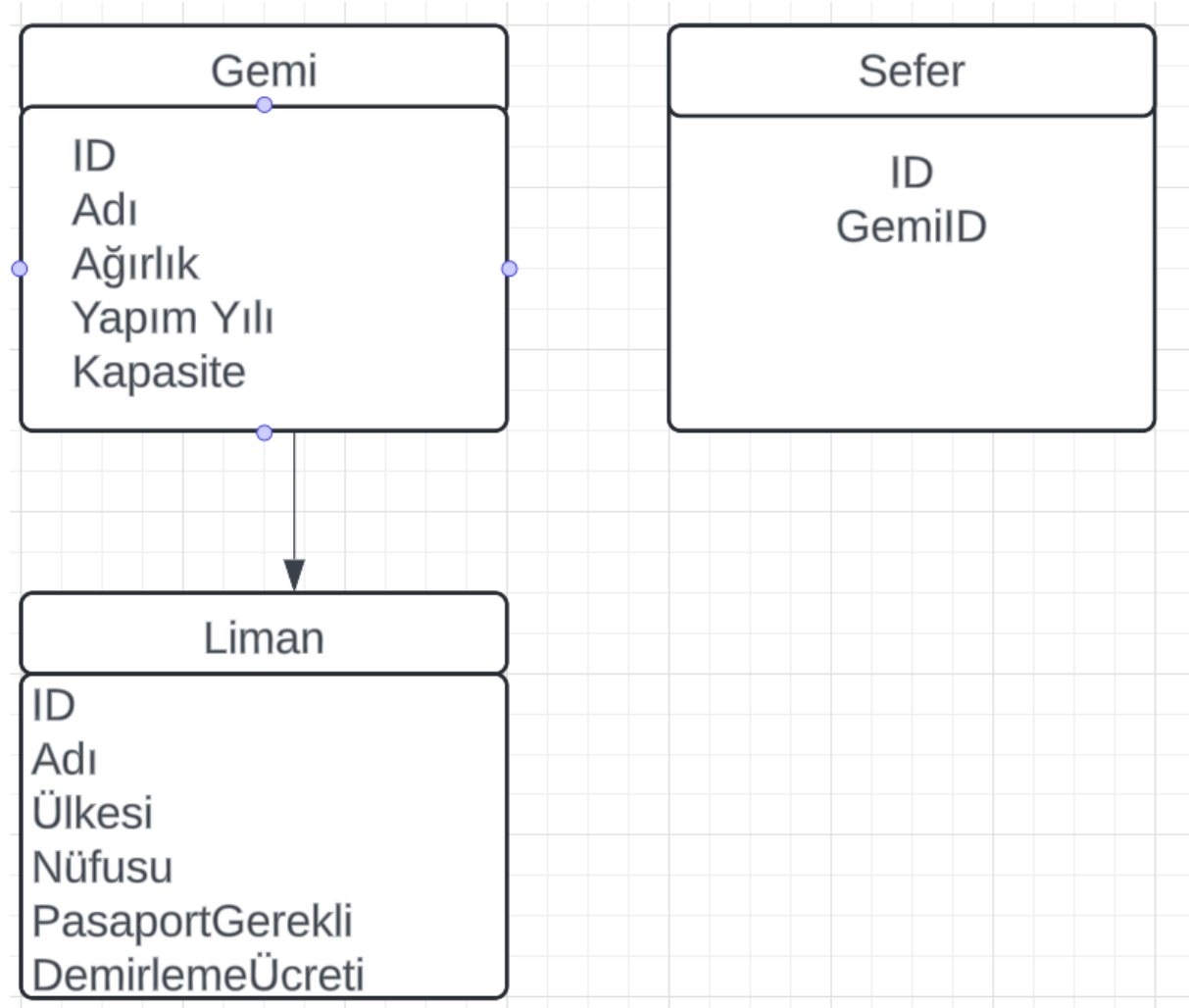
3.3 Veritabanı Tasarımı

Veritabanı, DB Browser for SQLite kullanılarak tasarlanacaktır.

İlişkisel veritabanı modeli kullanılacak ve tablolar arasında ilişkiler kurulacaktır.

Her varlık için ayrı bir tablo oluşturulacak ve veriler bu tablolarda saklanacaktır.

Modül ve Sınıf Diyagramları





3.4 Kullanıcı Arayüzü Tasarımı

Arayüz, kullanıcıların verilere erişimini kolaylaştırmak için düzenli ve kullanıcı dostu olacaktır.

Uygulamanın Çalıştırılması:

Projeyi derleyin ve çalıştırın.

Kullanıcı arayüzü üzerinden gemileri ve limanları ekleyin, düzenleyin veya silin.

Gerekli verileri girerek seferlerin planlanmasını sağlayın.

4.1 Kodlanan Bileşenlerin Açıklamaları

GemiVeritabani Sınıfı: SQLite veritabanıyla iletişim kurar. Gemi ve liman bilgilerinin eklenmesi, güncellenmesi ve silinmesi gibi işlevleri içerir.

LimanEklemeFormu Sınıfı: Yeni liman eklemek için bir kullanıcı arayüzü sağlar. Kullanıcıdan limanın adı, ülkesi, nüfusu, pasaport gereksinimi ve demirleme ücreti gibi bilgileri alır ve bu bilgileri veritabanına ekler.

GemiEklemeFormu Sınıfı: Yeni gemi eklemek için bir kullanıcı arayüzü sağlar. Kullanıcıdan gemi tipi, seri numarası, adı, ağırlığı, yapım yılı ve ekstra özelliklere göre gemi bilgilerini alır ve bu bilgileri veritabanına ekler.

SeferAyrıntılarıFormu Sınıfı: Bir seferin ayrıntılarını girmek için bir kullanıcı arayüzü sağlar. Kullanıcıdan liman bilgilerini alır ve bu bilgileri veritabanına ekler.

4.2 Görev Dağılımı

Tasarım ve Geliştirme Aşamalarındaki Görev Dağılımı: Ortak geliştirme Ortak tasarım

Raporun Hazırlanması Sürecindeki Görev Dağılımı: Ortak hazırlık.

4.3 Karşılaşılan Zorluklar ve Çözüm Yöntemleri

Karşılaşılan Zorluklar: Veritabanı tasarımı ve sınıflar arasındaki ilişkilerin doğru bir şekilde kurulması sırasında zorluklar yaşandı.

Çözüm Yöntemleri: Zorluklarla başa çıkmak için, proje ekibi sürekli iletişim halindeydi. Ayrıca bu teknik zorlukla başa çıkmak için dokümantasyon ve çevrimiçi kaynaklardan yararlanıldı.

4.4 Proje İsterlerine Göre Eksik Yönler

Eksik Yönler: Bu kod örneğinde, kullanıcı arayüzüne ek olarak seferlerin yönetimi ve seferler arası ilişkilerin işlenmesi eksiktir.

5.2 Yazılımın Doğrulanması

Test Sonuçları:

TestGemiVeritabani sınıfı altında yapılan testler başarıyla geçildi.

TestGemiEklemeFormu sınıfı altındaki tüm testler başarıyla geçildi.

Eksik veya Hatalı Çalışan Bileşenler:

Kaptan ve mürettebat sınıfları eklenmedi.

Tam ve Doğru Çalışan Bileşenler:

Gemi veritabanı işlevleri (gemi ekleme, liman ekleme) tam ve doğru bir şekilde çalışıyor.

Gemi ekleme formu kullanıcıdan girdileri doğru bir şekilde alıyor ve veritabanına eklemek için iletişim kuruyor.

Sefer ayrıntıları formu, liman bilgilerini doğru bir şekilde alıyor ve veritabanına eklemek için iletişim kuruyor.

Kaynakça:

<https://chatgpt.com>

<https://www.w3schools.com/>

<https://www.udemy.com/>

<https://www.veribilimiokulu.com/python-sql-server-baglantisi/>

<https://derslik.kerteriz.net/python-dersleri/veritabani-islemleri/python-mssql-veritabani>

<https://learn.microsoft.com/tr-tr/azure/azure-sql/database/connect-query-python?view=azuresql>

<https://onurkece.com.tr/pyhton-ile-sql-server-baglantisi/>