

**T.C**

**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ**

**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**

**YAZILIM MÜHENDİSLİĞİ**

**VERİ TABANI YÖNETİM SİSTEMLERİ PROJESİ**

**ALPEREN ALSAN**

**220502024**

**İBRAHİM TEKİN**

**220502037**

**DERS SORUMLUSU:**

**DR. ÖĞR. ÜYESİ NUR BANU ALBAYRAK**

**05.05.2024**

1. **GİRİŞ**
   1. **Projenin amacı**

* Projemizin amacı, bir gemi şirketinin gemilerinin seferlerlerini kontrol edebilmesini sağlamaktır
* Gemilerin türlerini, yüklerini, destianasyonlarını bir veritabanında tutmaktır

1. **GEREKSİNİM ANALİZİ**
   1. **Arayüz gereksinimleri**

* Kullanıcı arayüzünü oldukça basit tutmaya çalıştık
  1. **Fonksiyonel gereksinimler**
* Her bir geminin; Seri numarası, Ağırlığı, Adı, Yapım yılı tutulmalı
* Eğer yolcu gemisi ise yolcu kapasitesi, tutulmalı.
* Petrol tankeri ise petrol kapasitesi(litre) tutulmalı.
* Konteyner gemisi ise konteyner kapasitesi tutulmalı.
* Her bir sefere aynı anda bir geminin çıkması gereklidir.
* Sefer kayıtları için ID, yola çıkış tarihi, dönüş tarihi ve çıkılan limanın verileri tutulmalıdır.
* Geçmişte yapılmış ve gelecekte yapılması olası seferlerin bilgilerinin tutulması gereklidir.
  1. **Use-Case diyagramı**
* diyagram, çizgi, plan, teknik çizim içeren bir resim

  Açıklama otomatik olarak oluşturulduUse-Case diyagramı

1. **TASARIM**
   1. **Mimari tasarım**

* Kod, Tkinter kütüphanesini kullanarak bir GUI (Grafiksel Kullanıcı Arayüzü) oluşturmak için tasarlanmıştır.
* SQLite veritabanı ile etkileşim kurar. sqlite3 modülü, ilişkisel bir veritabanı yönetim sistemi olan SQLite'ı Python programlarına entegre etmek için kullanılır. Veritabanı, gemiler, seferler, limanlar, kaptanlar ve mürettebatlar gibi denizcilik verilerini depolar.
* Kod, gemiler, seferler, limanlar, kaptanlar ve mürettebatlar gibi farklı varlık türlerini temsil etmek için sınıfları kullanır. Her sınıf, ilgili varlık türünün özelliklerini ve davranışlarını içerir.
* Her varlık için, yani gemiler, seferler, limanlar, kaptanlar ve mürettebatlar için, CRUD işlemleri (Create, Read, Update, Delete) sağlanmıştır. Bu işlemler, veritabanındaki kayıtların oluşturulması, okunması, güncellenmesi ve silinmesini sağlar.
* Kod, kullanıcıların yapabileceği hataları ele almak için hata işleme mekanizmaları içerir. Örneğin, bir gemi eklenirken veya bir sefer oluşturulurken, kullanıcıya olası hataları bildiren bir iletişim kutusu gösterilir.
* metin, ekran görüntüsü, dikdörtgen, yazı tipi içeren bir resim

  Açıklama otomatik olarak oluşturulduKod, farklı işlevleri gerçekleştiren farklı sınıflara ve formlara ayrılmıştır. Örneğin, gemi ekleme, sefer ekleme, liman ekleme gibi işlevler için ayrı formlar bulunur. Bu modüler yapı, kodun daha okunabilir, sürdürülebilir ve genişletilebilir olmasını sağlar.
  1. **Kullanılacak teknolojiler**
* Projemiz Python programlama dili ile geliştirilmiş olup bununla beraber sql veri tabanı teknolojiside kullanılmıştır
* Projemizde Python dilinin “tkinter” ve “sqlite3” kütüphanelerinden faydalanılmıştır.
* Sqlite3 ile veri tabanı bilgileri tutulmuştur.
  1. **Veri tabanı tasarımı**
* Veritabanı, gemiler, seferler, limanlar, kaptanlar ve mürettebatlar gibi farklı varlık türlerini temsil eden tablolar içerir. Her bir tablo, ilgili varlık türünün özelliklerini içeren sütunlardan oluşur.
* diyagram, metin, taslak, çizgi içeren bir resim

  Açıklama otomatik olarak oluşturulduTablolar arasındaki ilişkiler, örneğin bir geminin hangi seferlere katıldığını veya bir seferde hangi kaptanların görev aldığını belirler. Bu ilişkiler, birleştirilmiş anahtarlar kullanılarak tanımlanır.
* Her tabloda, her bir satırın benzersiz bir şekilde tanımlanmasını sağlayan birincil anahtarlar (primary keys) tanımlanır. Ayrıca, tablolar arasındaki ilişkileri belirlemek için birleştirilmiş anahtarlar (foreign keys) kullanılır. Bu anahtarlar, veri bütünlüğünü sağlar.
  1. **Kullanıcı arayüzü tasarımı**
* **metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

  Açıklama otomatik olarak oluşturuldu**Kullanıcı arayüzünü oldukça anlaşılabilir kılmaya çalıştık.

1. **UYGULAMA**
   1. **Kodlanan bileşenlerin açıklamaları**

* Bileşenlerin (fonksiyonlar, sınıflar ya da modüller) açıklanması
* Gemiler, Seferler, Limanlar, Kaptanlar ve Murettabat gibi sınıflar, veritabanındaki ilgili tablolarla etkileşim için yöntemlere (methods) sahiptir.
* gemi\_ekle, sefer\_ekle, liman\_ekle, kaptan\_ekle ve murettabat\_ekle gibi fonksiyonlar, kullanıcı girişi alır ve bu girdileri veritabanına eklemek için ilgili sınıfların yöntemlerini çağırır.
* Kod, tkinter kütüphanesinden (tkinter) GUI bileşenlerini ve veritabanı işlemleri için sqlite3 modülünü kullanır. Ayrıca, kullanıcıya mesaj kutuları göstermek için messagebox modülünü kullanır.
* GUI bileşenleri ve işlevler, kullanıcının etkileşime geçmesini sağlar. Örneğin, bir gemi eklemek için kullanılan bir düğme tıklanınca, gemi\_ekle fonksiyonu çağrılır ve kullanıcının girdileri alınır.
* Kod, veritabanı işlemleri için SQL sorgularını oluşturmak ve bu sorguları yürütmek için sqlite3 modülünü kullanır. Bu işlemler, kullanıcıdan alınan verileri veritabanına eklemek, silmek veya güncellemek içindir.
* Kod, tkinter kütüphanesini kullanarak bir GUI oluşturur. Bu GUI, kullanıcının gemi, sefer, liman, kaptan ve mürettebat gibi verileri görüntülemesini, eklemesini, düzenlemesini veya silebilmesini sağlar.
  1. **Görev dağılımı**
* Bileşenleri tasarlarken görev dağılımını eşit yapmaya özen gösterdik.
* Sorumlulukları bölerek Yükümüzü hafifletmeye çalıştık
* Raporun hazırlanması sürecinde birbirimizle sürekli iletişim halinde olarak özellikle diyagramlar konusunda birbirimize güzel fikirler verdik.
  1. **Karşılaşılan zorluklar ve çözüm yöntemleri**
* Veri Tabanı ile ilgili ilk projemiz olduğundan Veri tabanları konusu biraz bizi zorladı
  1. **Proje isterlerine göre eksik yönler**
* Proje isterlerine göre herhangi bir eksik yön bırakılmamıştır.

1. **TEST VE DOĞRULAMA**
   1. **Yazılımın test süreci**

* Yazılım için bir test süreci deneme yanılma yoluyla bizim tarafımızdan manuel olarak test edilmiştir.
* Veri Tabanını Silme ve Düzenleme Dışında Harici Bir Test Uygulaması yazılmamıştır.
  1. **Yazılımın doğrulanması**
* Herhangi bir test kodu olmadığı için Yazılımın doğrulanmasıda bizler tarafından manuel olarak yapılmıştır.
* Test sonucunda eksik ya da hatalı çalışan bileşenleri düzeltmeye çalıştık. Bizler şu anki pozisyonda bu amacımıza ulaştığımızı düşünmekteyiz.

<https://github.com/IbrahimTekin03>

https://github.com/Alpyinwonderlands