

LabSense

Tunahan KANBAK, Murat UYANIK

Tasarım Belgesi **Yazılım Geliştirme Projesi**

Bilgi Teknolojileri Sertifika Programı
İDEA: ODTÜ Sanal Kampüsü

28 Haziran 2022

Öz

Tasarım Belgesinde LabSense projesinin gerçekleştirimi için gerekli olan yazılımsal altyapının detayları verilmektedir. Bu kapsamda, analiz belgesinde yer verilen kullanım durumları detaylandırılarak akış diyagramları oluşturulmuştur. Bu akış diyagramlarında verilen aktör/objeler ve bu objelerin fonksiyonları ise ilişki diyagramında gösterilmiştir. Ek olarak, veritabanında yer alan tablolar, ilişkiler EER diyagramı ile gösterilmiş olup bu diyagram içerisinde yer alan veriler, veri sözlüğü kullanılarak detaylandırılmıştır. Son olarak, LabSense projesi için oluşturulan modüllerin içeriklerinden ve bu modüllerde yer alacak fonksiyonların detayların bahsedilmiştir.

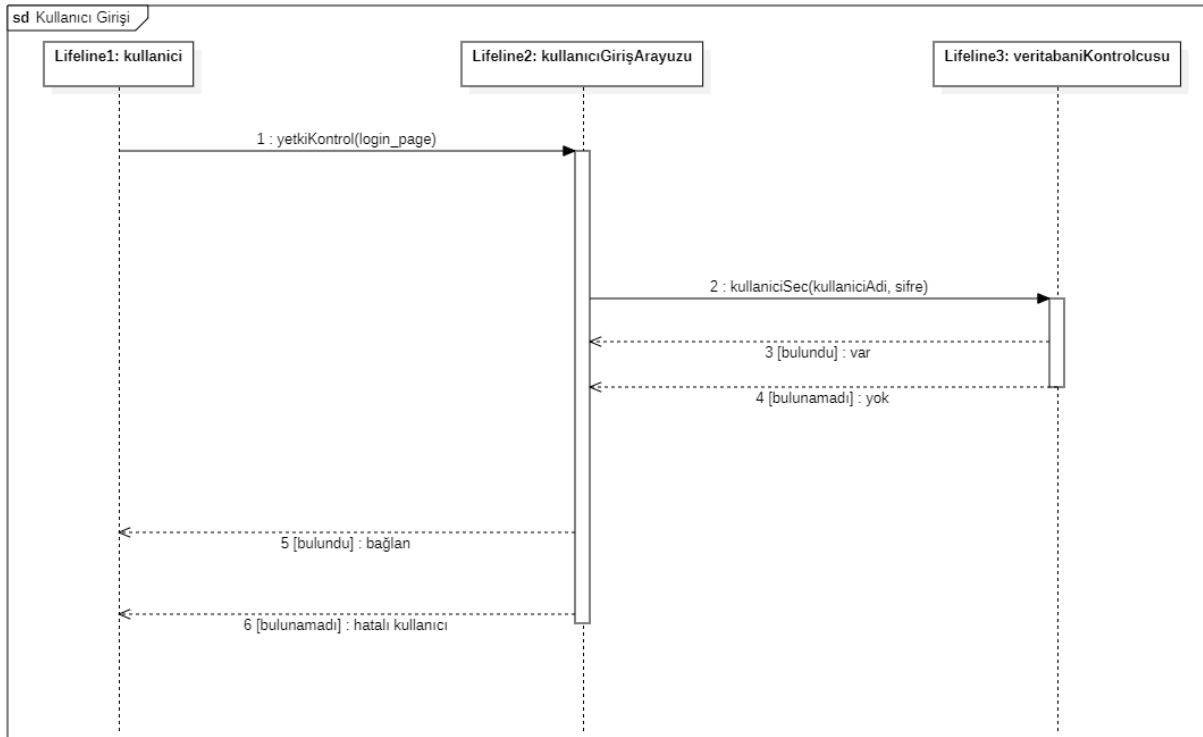
Proje Tanımı

LabSense müşteriler ile laboratuvar arasındaki iletişimi sağlayacak bir veri tabanı uygulamasıdır.

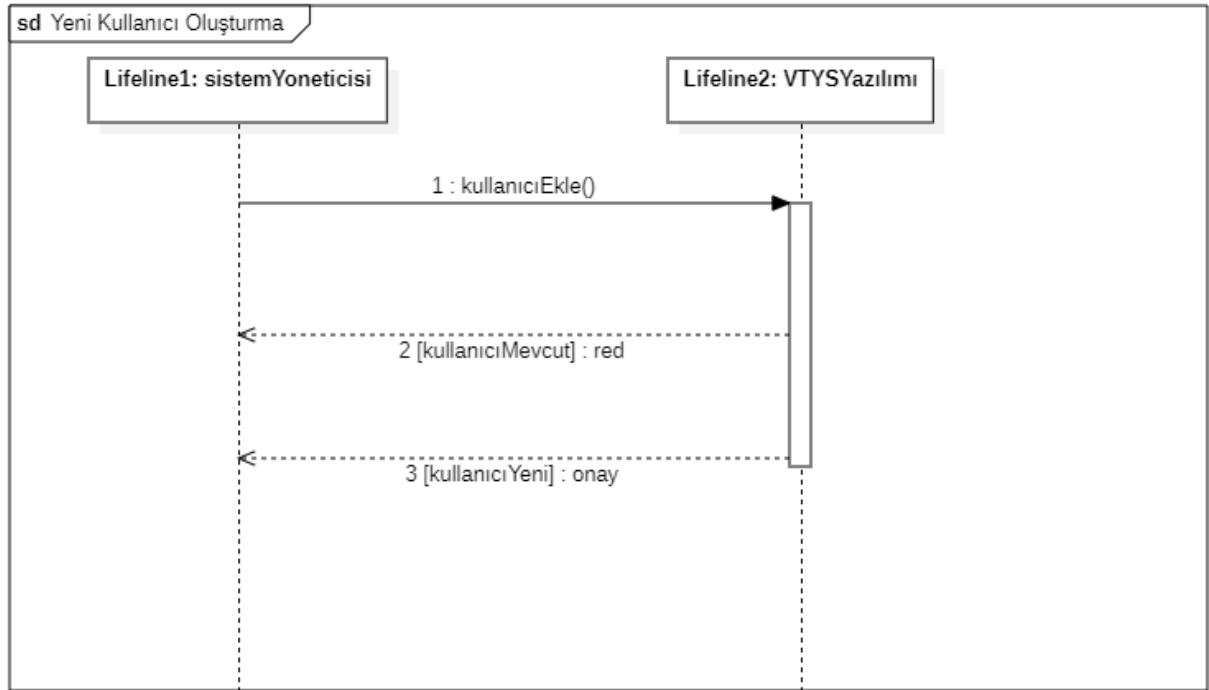
Müşteriler iki farklı deney tipi için deney taleplerini girebilecek ve deney sonuçları için kolaylıkla görüntüleme ve kıyaslama yapabileceklerdir. Laboratuvar çalışanları ise yapmış oldukları deneyleri kendileri için hazırlanmış arayüz üzerinden veri tabanına girebileceklerdir.

Mimari Tasarım

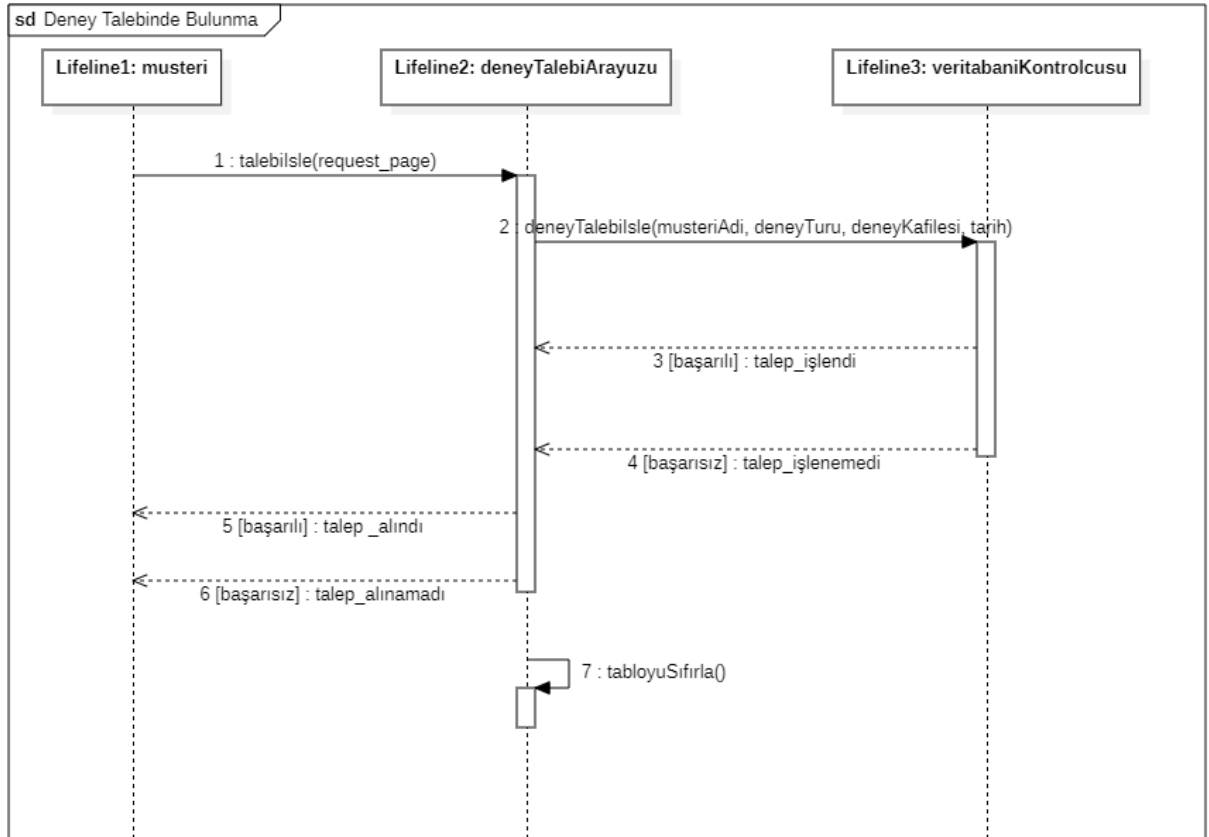
Analiz dokümanında yer verilen kullanım durumu şemasında yer alan kullanım durumlarına ait akış diyagramları (ing. Sequence diagram) aşağıda sırasıyla verilmiştir.



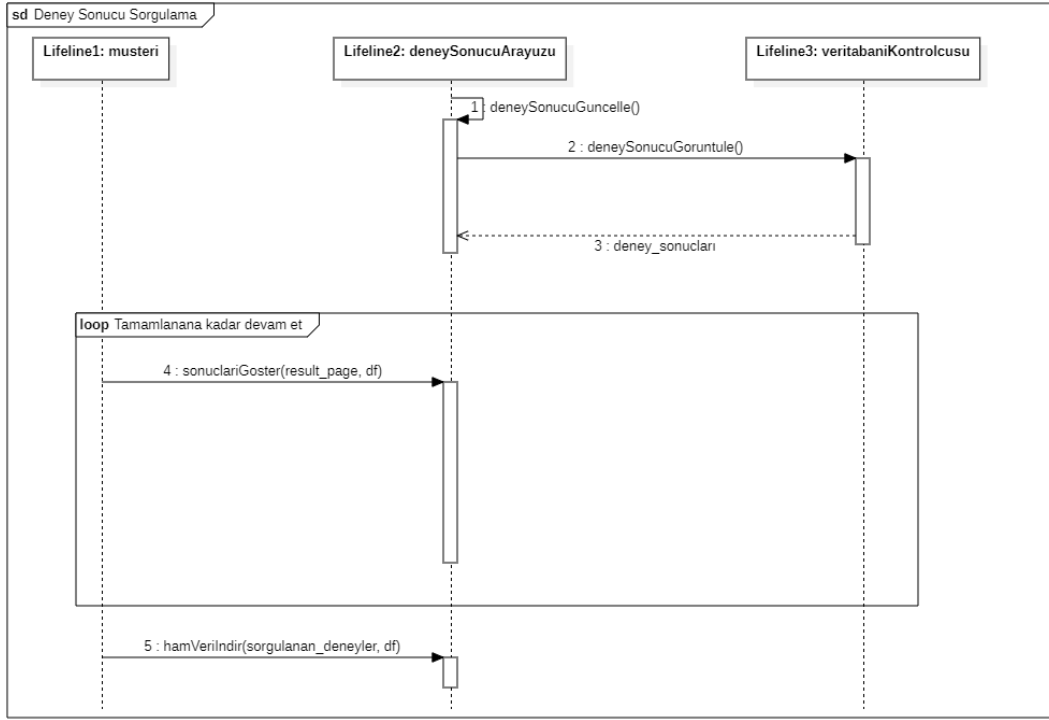
Şekil 1: Kullanıcı Girişi Akış Diyagramı



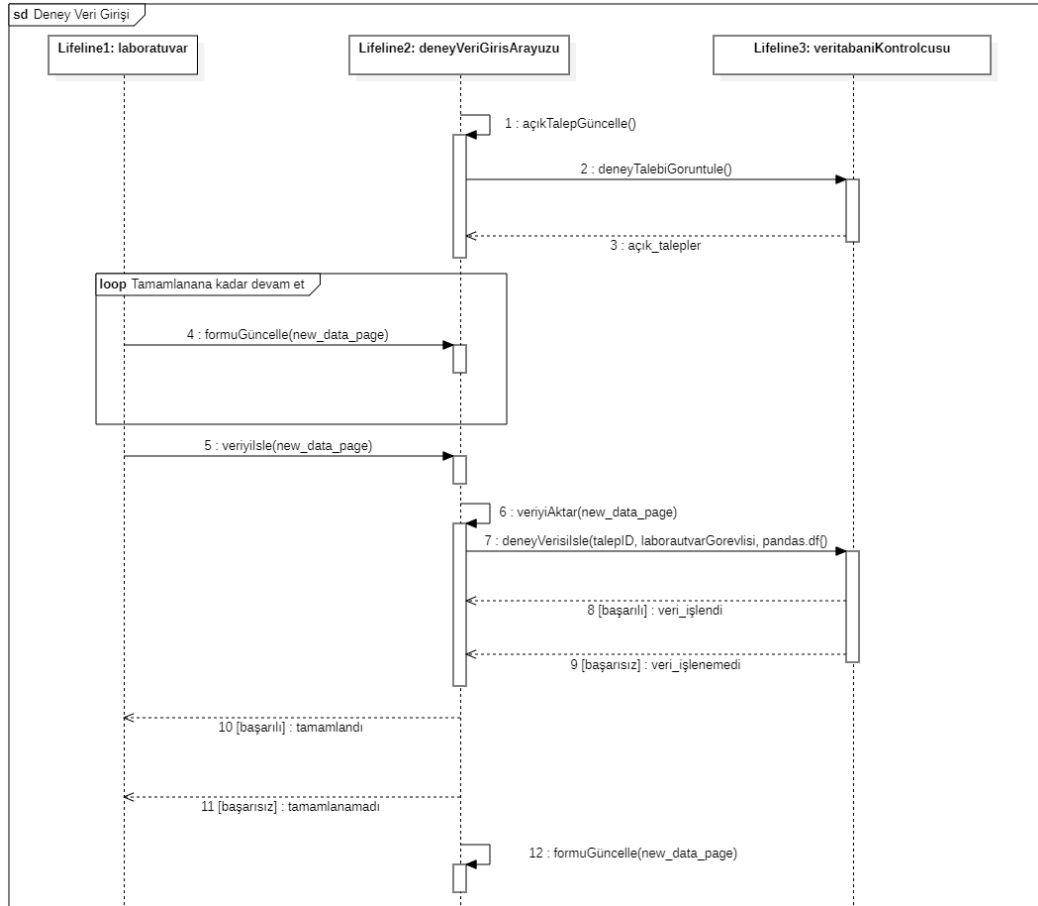
řekil 2: Yeni Kullanıcı Oluřturma Akıř Diyagramı



řekil 3:Deney Talebinde Bulunma Akıř Diyagramı

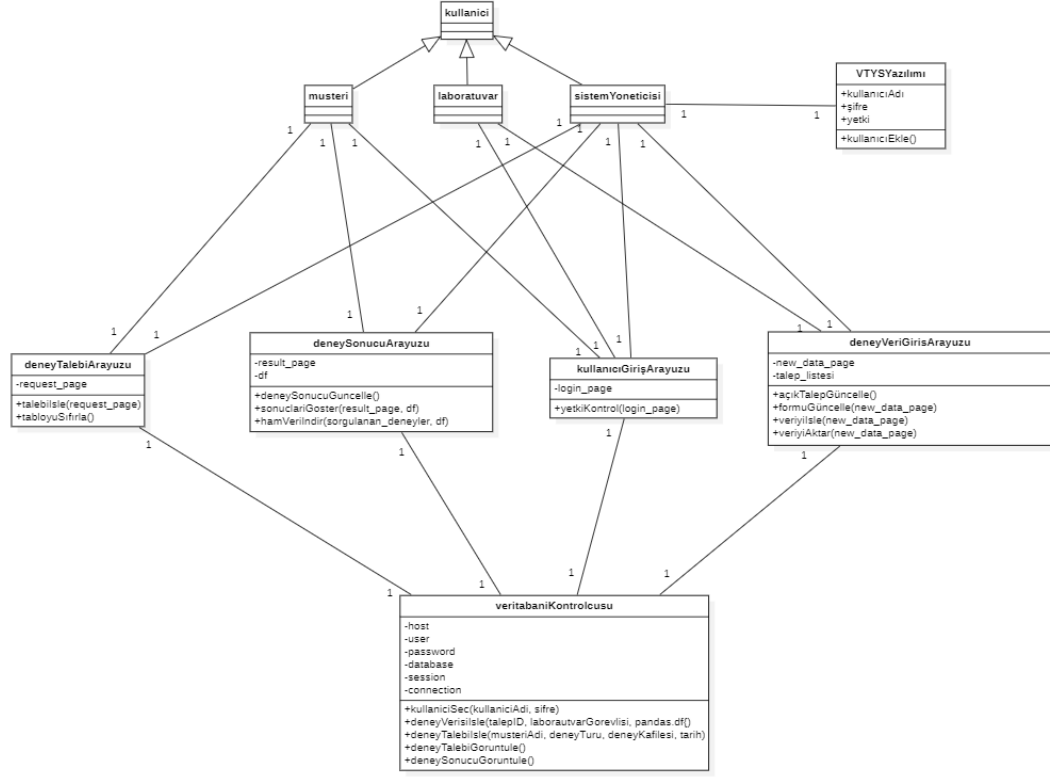


Şekil 4: Deney Sonucu Sorgulama Akış Diyagramı



Şekil 5: Deney Veri Girişi Akış Diyagramı

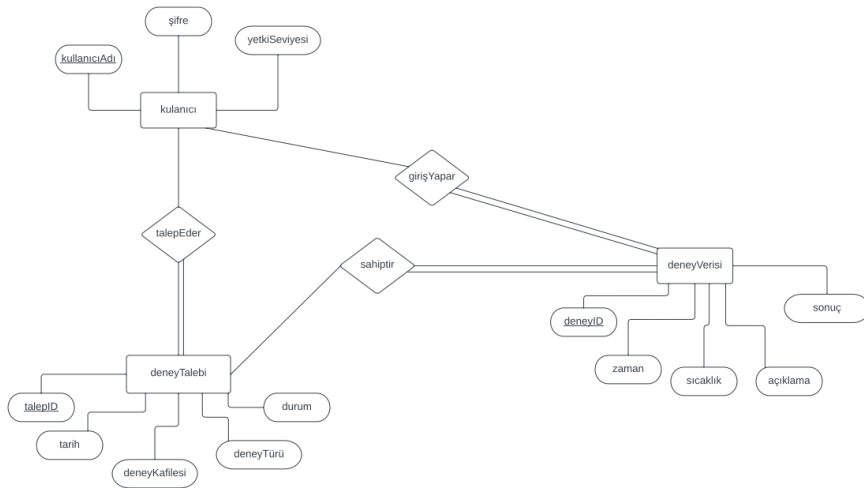
Akış diyagramlarına yer verilen sınıflar ve bu sınıfların metot ve nitelikleri aşağıdaki sınıf diyagramında detaylandırılmıştır.



Şekil 6: Labsense Sınıf Diyagramı

Veri Tasarımı

Veritabanı yönetim sistemi olarak MySQL veritabanının kullanımı tercih edilmiştir. Veritabanı tanımlanırken ilişkisel tablolar da kullandığı için veritabanı motoru olarak “InnoDB” tercih edilmiştir. Veritabanı ilişkisel olarak kurgulanmış olup veritabanı yapısını gösterir ER (ing. Entity Relationship) diyagramı aşağıda paylaşılmıştır.



Şekil 7: Labsense Veritabanı ER Diyagramı

ER Diyagram ile detaylandırılan tablolarda kullanılan her veri biriminin tanımının yer aldığı Veri Sözlüğü aşağıdaki tabloda verilmiştir.

Tablo 1: Labsense Veritabanı Veri Sözlüğü

Alan Adı	Veri Tipi	Uzunluk	Açıklama
kullaniciAdi	varchar	50	kullanıcı adı
sifre	varchar	50	sisteme giriş şifresi
yetki	enum		arayüzlere erişim yetki seviyesi (admin, customer, technician)
talepID	int	11	talep numarası
tarih	date		talebin yapıldığı tarih
deneyKafilesi	varchar	50	deney kafilesi
deneyTürü	enum		deneyin türü('A','B')
durum	varchar	10	açık/kapatıldı
deneyID	int	11	deney numarası
zaman	datetime		deneyin yapılma zamanı
sicaklik	double		deney sıcaklığı
sonuc	double		deney sonucu
açıklama	text		deneye eklenecek notlar

Arayüz Tasarımı

LabSense yazılımının gerçekleştirimi için kullanılacak modüller aşağıda listelenmiştir. Modüller kendi içerisinde veritabanı ve önyüz (ing. Front-end) olarak bölünmüştür:

- Veritabanı modülü:
 - labDBmanager.py
- Önyüz modülleri:
 - app.py
 - main.py
 - login_page.py [kullanıcıGirişArayuzu]
 - query_page.py [deneySonucuArayuzu]
 - new_data_page.py [deneyVeriGirisArayuzu]
 - request_page.py [deneyTalebiArayuzu]

“labDBmanager.py” modülü, önyüzün veritabanı ile iletişimini sağlamak için tasarlanmış bir veritabanı adaptörüdür.

- Değişkenler:
 - host: MySQL sunucu adı
 - user: MySQL kullanıcı adı
 - password: kullanıcı şifresi
 - database: veritabanı adı

- session: veritabanı oturumu
- connection: veritabanı bağlantısı
- Fonksiyonlar
 - openDB (): veritabanına bağlantıyı sağlar.
 - closeDB (): veritabanı bağlantısını keser.
 - kullanıcıSec(kullanıcıAdi, sifre): kullanıcının seçilip yetki seviyesinin kontrol edilmesini sağlar ve yetki seviyesini döndürür.
 - deneyTalebilsle(musteriAdi, deneyTuru, deneyKafilesi, tarih): müşteri tarafından girilen talepleri kaydının yapılması sağlar ve bir talepID döndürür, taleper() fonksiyonunu subfonksiyon olarak çağırır.
 - talepEder (musteriAdi, talepID): taleper tablosuna verileri işler.
 - deneyTalebiGoruntule(): müşteri taleplerinin laboratuvar tarafından sorgulanmasının yapıldığı fonksiyondur. Açık 'talepID' leri liste olarak döner.
 - deneyVerisilsle(talepID, laboratuvarGorevlisi, pandas.df{keys:zaman, sicaklik, sonuc, aciklama}): yapılan deneylerin laboratuvar tarafından girilen sonuçlarının tablolara eklenmesi, deneytalebi tablosundaki durum değişkeni kapatıldı olarak güncellenecek. DeneyID ile sahiptir ve giriş yapar tabloları güncellenir.
 - deneySonucuGoruntule(): müşterilerin talep etmiş olduğu deney sonuçlarını döner. Veritabanında yer alan deneySonucu tablosunun, sahiptir ve deneyTalebi tabloları ile birleştirilmesi sonucu elde edilen tüm verileri bir dataframe olarak döner.

“app.py” modülü, dash kütüphanesinin constructor’ı olan Dash fonksiyonunu çalıştırıp diğer modüllerde kullanılmak üzere uygulama (app) değişkenini tanımlar.

“main.py” modülü, önyüz elementlerinin toplu tutulduğu bir modül olup sistemin açılışından sorumlu modüldür. Bu modül içerisinde yer alan fonksiyon ve değişkenler aşağıda verilmiştir:

- Değişkenler:
 - app.layout: Uygulamanın genel şemasını tanımlayan değişkendir.

“login_page” modülü, giriş sayfasında yer alan önyüz elementlerinin ve bu elementler ile gerçekleşecek etkileşimleri kontrol eden modüldür. Bu modül içerisinde yer alan fonksiyon ve değişkenler aşağıda verilmiştir:

- Değişkenler:
 - login_page: Giriş sayfası içerisinde yer alan önyüz elementlerini ve bu elementleri detay özelliklerini tanımlayan değişkendir(dbc.Container).

- Fonksiyonlar:
 - yetkiKontrol(login_page): Kullanıcı girişi esnasında kullanıcı bilgilerinin kontrol edilerek yetki seviyesinin belirlendiği (veritabanından) ve belirlenen yetki seviyesine göre yazılımın ilgili kısımlarını aktifleştiren fonksiyondur. Fonksiyon girdisi olarak tanımlanan “login_page” değişkeni kompleks bir önyüz elementi olup kullanıcı bilgilerini (kullanici_adi, sifre) barındırmaktadır. Bu fonksiyon iki farklı sinyal ile tetiklenmektedir ve bunlar “login_page” objesi içerisinde yer alan giriş yap butonu ve çıkış yap butonudur.

“query_page” modülü, müşteri kullanıcılarının deney sonucu sorgulayabildikleri önyüz elementleri ve bu elementlerin etkileşimlerini içeren modüldür. Bu modül içerisinde yer alan fonksiyon ve değişkenler aşağıda verilmiştir:

- Değişkenler:
 - query_page: Deney sonucu sorgulama sayfası içerisinde yer alan önyüz elementlerini ve elementlerin detay özelliklerini tanımlayan değişkendir(dbc.Container).
 - df: Veritabanından çekilen sonuçların tutulduğu pandas.dataframe objesidir.
- Fonksiyonlar:
 - deneySonucuGuncelle(): “query_page” sayfası her aktifleştirildiğinde devreye giren deneySonucuGuncelle fonksiyonu “query_page”de yer alan deney sonucu arama çubuğunun seçeneklerini veritabanından kontrol ederek günceller.
 - sonuclariGoster(result_page, df): Kullanıcının belirttiği deney listesini “df” üzerinde filtreleyerek sonuç görsellerini kullanıcıya “result_page” elementini değiştirerek gösterir. Kullanıcının sorguladığı deneyler “result_page” elementi içerisinde çekilir. Bu fonksiyon tek bir sinyal ile tetiklenmekte olup bu da arama bölgesinde farklı deney adı seçilmesidir.
 - hamVerilindir(sorgulanan_deneyler, df): “sorgulanan_deneyler” listesini girdi olarak alan “hamVerilindir” fonksiyonu, kullanıcı tarafından sorgulanan deney sonuçlarını “csv” formatında kullanıcı bilgisayarına indirmek için kullanılır. Bu fonksiyon önyüzde yer alan indir butonundan sinyal alarak devreye girer.

“new_data_page.py” modülü, laboratuvar kullanıcısının yeni veri girişini yapabileceği önyüz elementlerini barındıran ve bu elementlerin etkileşimlerini kontrol eden modüldür. Bu modül içerisinde yer alan fonksiyon ve değişkenler aşağıda verilmiştir:

- Değişkenler:
 - new_data_page: Veri giriş sayfası içerisinde yer alan önyüz elementlerini ve bu elementlerin detay özelliklerini tanımlayan değişkendir(dbc.Container).
 - acik_deneyler: Veritabanından çekilen deney talepleri listesinin tutulduğu list objesidir.
- Fonksiyonlar:

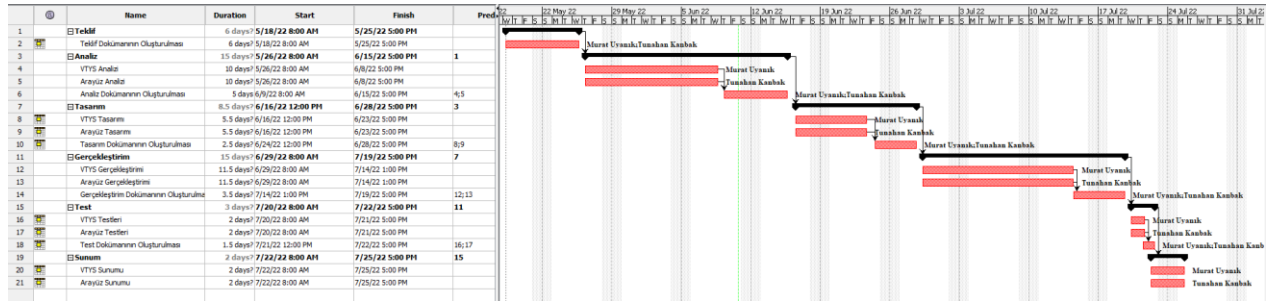
- açıkTalepGüncelle(): Kullanıcının “new_data_page.py” sayfasına erişim sağladığında güncel veritabanından açık taleplerin listesi çekilir ve “acik_deneyler” değişkenine atanır.
- formuGüncelle(new_data_page): Kullanıcı “new_data_page” objesinde yer alan veri giriş tablosunda değişiklik yapmak istediğinde tablo formatının güncellenmesini sağlar. Bu fonksiyon üç farklı sinyal ile tetiklenmekte olup bunlar “new_data_page” objesi içerisinde yer alan satır ekle, satır sil ve veri işle butonlarıdır.
- veriyiisle(new_data_page): Kullanıcı “new_data_page” objesinde yer alan veri tablosunu, veritabanına iletmek istediğinde “new_data_page” içerisindeki elementleri işleyerek (zaman, sıcaklık, not, sonuc, sonucu_giren, sahip_talep) veritabanına yönlendirir. Bu fonksiyon tek bir sinyal ile tetiklenmekte olup bu da veri işle butonudur.
- veriyiAktar(new_data_page): Veriyiisle fonksiyonunun new_data_page objesine işlediği verileri çekerek veri uygunluğunu kontrol eder. Uygun veri veritabanına iletilir ve veritabanının cevabına göre kullanıcıya bilgilendirme yapar.

“request_page.py” modülü, müşteri kullanıcısının yeni bir deney talebinde bulunabileceği önyüz elementlerinin bulunduğu ve bu elementlerin etkileşimlerini kontrol eden modüldür. Bu modül içerisinde yer alan fonksiyon ve değişkenler aşağıda verilmiştir:

- Değişkenler:
 - request_page: Deney talebi sayfası içerisinde yer alan önyüz elementlerini ve bu elementlerin detay özelliklerini tanımlayan değişkendir(dbc.Container).
- Fonksiyonlar:
 - talebilsle(request_page): “request_page” objesi içerisinde yer alan talep bilgilerinin (tarih, deney_kafilesi, deney_turu, talebi_yapan) işlenerek veritabanına yönlendirilmesini sağlar. Veritabanının sonucuna göre kullanıcıyı (talepID bilgisi ile birlikte) bilgilendirir. Bu fonksiyon tek bir sinyal ile tetiklenmekte olup bu da talebi gönder butonudur.
 - tabloyuSıfırla(): talebilsle fonksiyonunun veriyi veritabanına başarılı olarak aktarmasının ardından tablo içeriğini sıfırlar.

Proje Planı ve Uygulama Takvimi

Proje Çağlayan modeli ile geliştirilmektedir. Projenin ilerleyen fazlarında yer alan Tasarım ve Gerçekleştirim aşamaları için yapılan iş bölümü aşağıda paylaşıldığı gibidir:



Kaynakça

<https://idea.metu.edu.tr/mod/book/view.php?id=360&chapterid=599> BTSP Ders Notları

<https://idea.metu.edu.tr/mod/book/view.php?id=360&chapterid=608> BTSP Geçmiş Dönem Projeleri