

Kocaeli Üniversitesi
Bilgisayar Mühendisliği Bölümü
Yazılım Laboratuvarı - 1
Proje 1: Dinamik Sınav Takvimi ve
Oturma Planı Sistemi

230202040-Ömer Faruk Toycu
230202069-Ali Berke Erenler

1. Özeti

Bu proje, üniversitelerdeki sınav takvimi oluşturma sürecinin karmaşıklığını ve yüksek hata payını ele almaktadır. Sürecin otomatize edilmesi ve optimize edilmesi amacıyla, Python programlama dili ve PyQt5 arayüz kütüphanesi kullanılarak bir masaüstü uygulaması geliştirilmiştir. Sistemin veri yönetimi ve tutarlılığı, ilişkisel bir MySQL veritabanı ile sağlanmıştır. Geliştirilen uygulama, "Admin" ve "Bölüm Koordinatörü" olmak üzere iki farklı kullanıcı rolünü desteklemektedir. Admin, sistem genelinde kullanıcıları yönetirken; bölüm koordinatörleri, kendi bölümlerine ait derslikleri, dersleri ve öğrencileri yönetebilmektedir. Sistem, standartlaştırmamış Excel dosyalarından ders ve öğrenci kayıtlarını akıllı bir şekilde işleyerek veritabanına aktarma (parsing) yeteneğine sahiptir. Veri girişi tamamlandıktan sonra, tanımlanan kısıtlara (öğrenci çakışması, derslik kapasitesi vb.) uygun olarak dinamik bir sınav takvimi oluşturulması hedeflenmektedir. Bu çalışma, modüler bir yazılım mimarisi kullanarak hem geliştirilebilir hem de kullanıcı dostu bir otomasyon sistemi sunmayı amaçlamaktadır.

2. Giriş

Üniversitelerde sınav takvimlerinin hazırlanması, çok sayıda değişken ve kısıt içeren karmaşık bir optimizasyon problemdir. Aynı anda yüzlerce dersin, binlerce öğrencinin ve sınırlı sayıda

dersliğin yönetilmesi gerekmektedir. Bu sürecin manuel olarak yürütülmesi, yoğun iş gücü gerektirmesinin yanı sıra, öğrenci çakışmaları, derslik kapasite aşımları ve adil olmayan sınav dağılımları gibi ciddi hatalara yol açabilmektedir.

Bu projenin temel amacı, Kocaeli Üniversitesi Mühendislik Fakültesi bünyesindeki beş bölüm (Bilgisayar, Yazılım, Elektrik, Elektronik, İnşaat) için bu süreci otomatize eden, dinamik ve kullanıcı dostu bir masaüstü uygulaması geliştirmektir. Proje, "Yazılım Laboratuvarı I" dersi kapsamında, masaüstü programlama, veritabanı tasarıımı ve yönetimi becerilerini geliştirmeyi hedeflemektedir.

Proje kapsamında geliştirilen sistem, farklı yetkilere sahip kullanıcı rollerini (Admin, Bölüm Koordinatörü) yönetebilmekte, Excel dosyaları üzerinden toplu veri alımını desteklemekte ve belirlenen kısıtlara göre optimize edilmiş bir sınav programı oluşturma altyapısını sunmaktadır.

Teknoloji seçimi olarak, arayüz geliştirme için Python dilinin PyQt5 kütüphanesi tercih edilmiştir. Python'un sunduğu pandas gibi güçlü veri işleme kütüphaneleri, Excel'den veri alma gibi karmaşık işlemleri kolaylaştırmıştır. Veri depolama ve ilişkisel bütünlüğün sağlanması için ise endüstri standartı olan MySQL veritabanı yönetim sistemi kullanılmıştır.

3. Yöntem

3.1. Sistem Mimarisi

Uygulamanın kod tabanı, her biri belirli bir görevde odaklanmış birden fazla Python dosyasına bölünmüştür:

- **main.py (Ana Kontrolcü):**
Uygulamayı başlatan ve

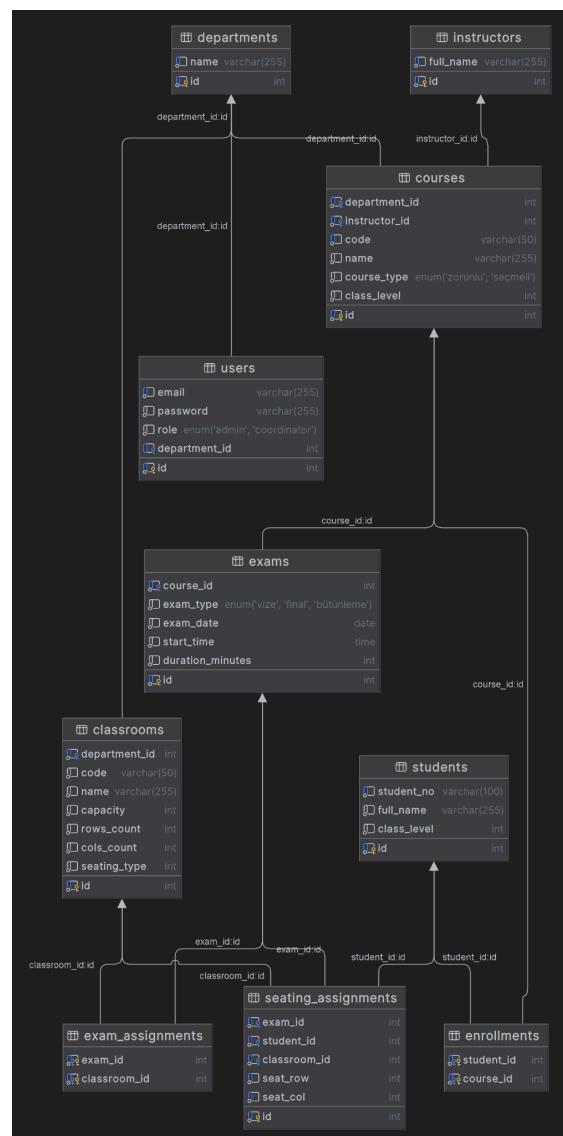
pencereler arası geçişini yöneten ana "orquestra şefi"dir. Kullanıcı girişinin başarılı olduğunda, gelen rol bilgisine göre Admin veya Koordinatör panelini açmakla sorumludur.

- **ui/ (Arayüz Katmanı):**
Kullanıcının gördüğü tüm pencereler (`login_window.py`, `admin_dashboard.py`, `coordinator_dashboard.py`) bu klasör altında ayrı dosyalarda tutulur. Bu dosyalar, pencerenin görsel tasarımını (butonlar, tablolar) ve bu görsel elementlerin olaylarını (tıklama vb.) yönetir.
- **database.py (Veri Erişim Katmanı):** Uygulamanın veritabanı ile konuşduğu tek yerdir. Tüm SQL sorguları ve bağlantı yönetimi bu dosyada merkezileştirilmiştir. Arayüz katmanı, veritabanına asla doğrudan erişmez; bunun yerine bu dosyadaki `verify_user()`, `add_classroom()` gibi fonksiyonları çağrıır.
- **excel_processor.py (Akıllı Veri İşleme Modülü):** Projenin en gelişmiş modüllerinden biridir. Sadece Excel'i okumakla kalmaz, aynı zamanda standart dışı (örn: başlığı olmayan, sütun adları farklı) dosyaları dahı **akıllıca işlemen** geçirir (**robust parsing**). `pandas` ve `re` (RegEx) kütüphanelerini kullanarak veriyi temizler, sınıf bilgisi gibi eksik verileri satır içeriğinden çıkarır ve veritabanına **toplu** aktarımı uygun hale getirir.
- **config.py:** Veritabanı bağlantı bilgileri (kullanıcı adı, şifre) gibi hassas yapılandırma ayarlarını koddan ayırarak güvenliği ve yönetim kolaylığını artırır.

3.2. Veritabanı Tasarımı

Sistemin temeli, MySQL üzerinde tasarlanan ilişkisel veritabanı şemasıdır. Veri bütünlüğünü ve işlem güvenliğini desteklemesi amacıyla **InnoDB** depolama motoru tercih edilmiştir.

Veritabanı, kullanıcıları, bölümleri, derslikleri, öğretim üyelerini, dersleri, öğrencileri ve bu varlıklar arasındaki ilişkileri (örn: `enrollments` tablosu ile öğrencilerin hangi dersleri aldığı) tutan tablolardan oluşur.



3.3. Temel İşlevler

Kullanıcı Yönetimi: Sistem, iki farklı rolü destekler. "Admin" rolü, `admin_dashboard.py` arayüzü üzerinden yeni koordinatörler ekleyebilir ve tüm kullanıcıları listeleyip silebilir. "Bölüm Koordinatörü" rolü ise sisteme giriş yaptığından, `database.py` içerisindeki sorguların `WHERE department_id = %s` koşulu sayesinde sadece kendi bölümyle ilgili verileri görebilir.

Derslik Yönetimi (CRUD): Koordinatör, `coordinator_dashboard.py` arayüzündeki "Derslik Yönetimi" sekmesi üzerinden kendi bölümune ait derslikleri ekleyebilir, bilgilerini güncelleyebilir ve silebilir (CRUD operasyonları). Ayrıca, bir dersliğin ID'si ile arama yaparak, o dersliğin `rows_count`, `cols_count` ve `seating_type` verilerine göre oluşturulan oturma düzenini görsel olarak görebilir.

Akıllı Excel Veri Yükleme: Projenin en kritik özelliklerinden biri, standart bir formata sahip olmayan Excel dosyalarından veri okuyabilmektir. `excel_processor.py` modülü:

- Ders Yükleme**
`(process_courses_excel):`
Fonksiyon, standart dışı sütun adlarını ("KOD" vb.) tanıyarak ve başlıksız dosyaları analiz ederek ders listelerini esnek bir şekilde işler. "1. Sınıf" gibi ara satırlardan sınıf bilgisini çıkarır. Veritabanına `add_instructor` ve `add_course` fonksiyonları ile satır satır ekleme yapar; bu fonksiyonlar (get-or-create) mükerrer kaydı engeller.
- Öğrenci Yükleme**
`(process_students_excel):`
Bu fonksiyon, öğrenci-ders kayıtlarını **yüksek performanslı**

toplu (batch) yöntemle işler. Esnek sütun tanıma ('Ders', 'NO' vb.) ve sınıf metnini ('5. Sınıf' -> 5) açıklama özelliklerine sahiptir. Veritabanı performansını maksimize etmek için satır satır işlem yerine: a. Öğrenci ve ders verileri Excel'den önbelleğe alınır. b. Veritabanına minimum sayıda toplu sorgu atılarak mevcut kayıtların ID haritaları (maps) belleğe yüklenir. c. Sadece yeni öğrenciler ve *yeniden* ders kayıtları (`enrollments`) **tek bir toplu executemany komutu** ile veritabanına eklenir.

FONKSİYON
`process_courses_excel(dosya_yolu, bölüm_id):`
BAŞLA
`excel_dosyasını_oku`
`sütun_adlarını_esnek_eşleştir ('DERS KODU', 'DERSİN ADI', vb.)`

EĞER
`beklenen_sütunlar_HALA_yoksa:`

`dosayı_başlıksız_oku`

`mevcut_sınıf = BOŞ`

DOSYADAKİ_HER_SATIR için:

EĞER `satır_sınıf_bilgisi_ise ("1. Sınıf" gibi):`

`mevcut_sınıf =`
`satırda_sınıf_sayısını_çıkar`

`SONRAKİ_SATIRA_GEC`

EĞER `satır_geçersiz_veri_ise (boş, 'nan', başlık_metni):`

`SONRAKİ_SATIRA_GEC`

```

hoca_id =
database.add_instructor(hoca_adı) //  

get-or-create mantığı

```

```

sınıf = (koda_göre_belirle) VEYA  

mevcut_sınıf

```

```

ders_id =
database.add_course(ders_bilgileri,  

hoca_id, sınıf) // get-or-create

```

// ... Başlıklı dosya için benzer bir döngü

...

başarılı_ve_hatalı_kayıt_sayısını_döndür

HATA_OLUŞURSA:

HATA_MESAJI_döndür

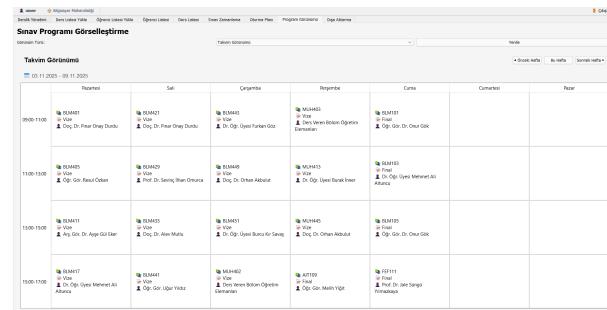
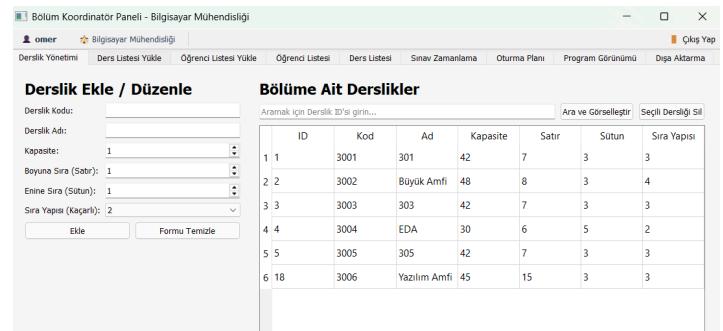
BİTİR

4. Sonuç

Bu proje kapsamında, üniversiteler için "Dinamik Sınav Takvimi Oluşturma Sistemi"nin temel altyapısı ve veri yönetim modülleri başarıyla tasarlanmış ve kodlanmıştır. Python (PyQt5) ve MySQL kullanılarak, farklı kullanıcı rollerini yönetebilen, modüler ve genişletilebilir bir masaüstü uygulaması geliştirilmiştir.

Projenin mevcut durumu, sınav takvimi oluşturma algoritması için gerekli olan tüm verinin (derslikler, dersler, öğrenciler ve kayıtları) sisteme alınmasını sağlamaktadır.

Bu proje, masaüstü programlama, ilişkisel veritabanı yönetimi ve karmaşık veri işleme konularında önemli bir tecrübe kazandırmıştır.



5. Kaynakça

Arayüz Kütüphaneleri

-<https://pypi.org/project/PyQt5/>

-<https://pandas.pydata.org/docs/>

-<https://dev.mysql.com/doc/connector-python/en/>

-<https://docs.python.org/3/library/re.html>

-https://www.youtube.com/watch?v=rZcdhles6vQ&list=PLCC34OHNcOtpmCA8s_dpPMvQLyHbxocY