



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

PROJE KONUSU:
PROGRAM DERS İLİŞKİ MATRİSİ PROJESİ

ÖęRENCİ ADI:
MERT BÜLBÜL 220502006
<https://github.com/Mertb2627>
İBRAHİM TEKİN 220502037
<https://github.com/IbrahimTekin03>
CAN ŞAFAK ÇAKIR 220502002
<https://github.com/cakirz>
ECENAZ BOZDOęAN 220502040
<https://github.com/ajnasss>

DERS SORUMLUSU:
DR. ÖęR. ÜYESİ ELİF PINAR HACİBEYOęLU

TARİH:30.12.2024

1 GİRİŞ

1.1 Projenin amacı

Bu proje, öğrenci başarılarını doğru bir şekilde değerlendirmek ve her öğrencinin performansını detaylı bir biçimde analiz etmek amacıyla geliştirilmiştir. Öğrenci notları, derslerin ağırlıklı katkıları ile hesaplanarak her bir öğrencinin genel başarı yüzdesi ortaya konulacaktır. Eğitim sürecindeki başarının daha verimli izlenebilmesi için, başarı oranları belirli program çıktılarıyla ilişkilendirilerek karşılaştırılacaktır.

Gerçekleşmesi Beklenenler

Proje sonunda, öğrenci başarılarının değerlendirilmesi için aşağıdaki sonuçlar elde edilecektir:

- Ağırlıklı Değerlendirme Tablosu:** Öğrencilerin notları, belirlenen ağırlıklarla hesaplanarak başarıları değerlendirilecektir.
- Öğrenci Başarı Raporları:** Her öğrenci için, ders notları ve başarı yüzdesi gibi bilgileri içeren kişisel raporlar oluşturulacaktır.
- Başarı Oranı Hesaplaması:** Program çıktıları ile öğrenci başarıları ilişkilendirilerek başarı oranları hesaplanacaktır.
- Excel Çıktıları:** Tüm veriler ve raporlar, düzenli bir şekilde Excel dosyasına aktarılacak ve her öğrenciye özel başarı raporları sunulacaktır.

Bu proje, eğitimdeki başarıyı objektif bir şekilde ölçmeyi ve her öğrencinin gelişimini doğru bir biçimde izlemeyi amaçlamaktadır.

2 GEREKSİNİM ANALİZİ

2.1 Arayüz gereksinimleri

- Kullanıcı Dostu Arayüz:** Kullanıcı, öğrenci notlarını ve ağırlıklarını kolayca girebilmeli ve raporları görüntüleyebilmelidir.
- Veri Girişi:** Excel dosyasından veri alınıp, bu verilerin doğru şekilde işlenebilmesi için form ve giriş alanları sunulmalıdır.
- Rapor Görüntüleme:** Kullanıcı, her öğrencinin notları ve başarı oranlarını içeren raporları Excel formatında indirilebilmelidir.
- Hata Bildirimi:** Eğer giriş verileri hatalı ya da eksikse, kullanıcıya bilgilendirme yapılmalıdır.

2.2 Fonksiyonel Gereksinimler

1. Veri Girişi:

- Sistem, kullanıcıdan öğrenci notları ve ders ağırlıkları gibi verileri içeren bir Excel dosyasını alabilmelidir.
- Kullanıcı, veri dosyasındaki her öğrencinin ders notlarını doğru bir şekilde girmelidir.
- Sistem, Excel dosyasındaki verileri okuyarak öğrenci notlarını ve ağırlıklı değerlendirme tablosunu doğru şekilde işleyebilmelidir.

2. Ağırlıklı Değerlendirme Hesaplama:

- Sistem, derslerin ağırlıklı değerlendirmelerini kullanıcı tarafından belirtilen yüzdelik ağırlıklar ile çarparak her ders için ağırlıklı puanları hesaplamalıdır.
- Ağırlıklı değerlendirme tablosu, öğrenci bazında her ders için notlar ve toplam başarıyı içermelidir.
- Sistem, tüm dersler için ağırlıklı notları ve toplam başarıyı hesapladıktan sonra "Ağırlıklı Toplam" değerini otomatik olarak hesaplamalıdır.

3. Başarı Oranı Hesaplama:

- Sistemde her öğrencinin derslerdeki başarı oranı, toplam başarıyı ve dersin maksimum değerini göz önünde bulundurarak hesaplanmalıdır.
- Öğrencilerin her ders için başarı yüzdesi, ilgili dersin ağırlıklandırılmış notu ile hesaplanmalıdır.

4. Çıktı Raporları:

- Sistem, her öğrencinin derslerindeki notları, toplam başarıyı, başarı yüzdesini ve maksimum başarı değerlerini içeren bir rapor oluşturmalıdır.
- Ayrıca, her öğrencinin başarı oranını belirten ek bir rapor ("Başarı Oranı") oluşturulmalıdır.
- Tüm bu raporlar, kullanıcı tarafından istenilen Excel formatında dışa aktarılmalıdır.

5. Veri Doğrulama ve Hata Kontrolü:

- Kullanıcı verisi eksik veya hatalı olduğunda, sistem uygun hata mesajları ile kullanıcıyı bilgilendirmelidir.
- Notların ve ağırlıkların doğru formatta ve geçerli aralıkta olup olmadığı kontrol edilmelidir.

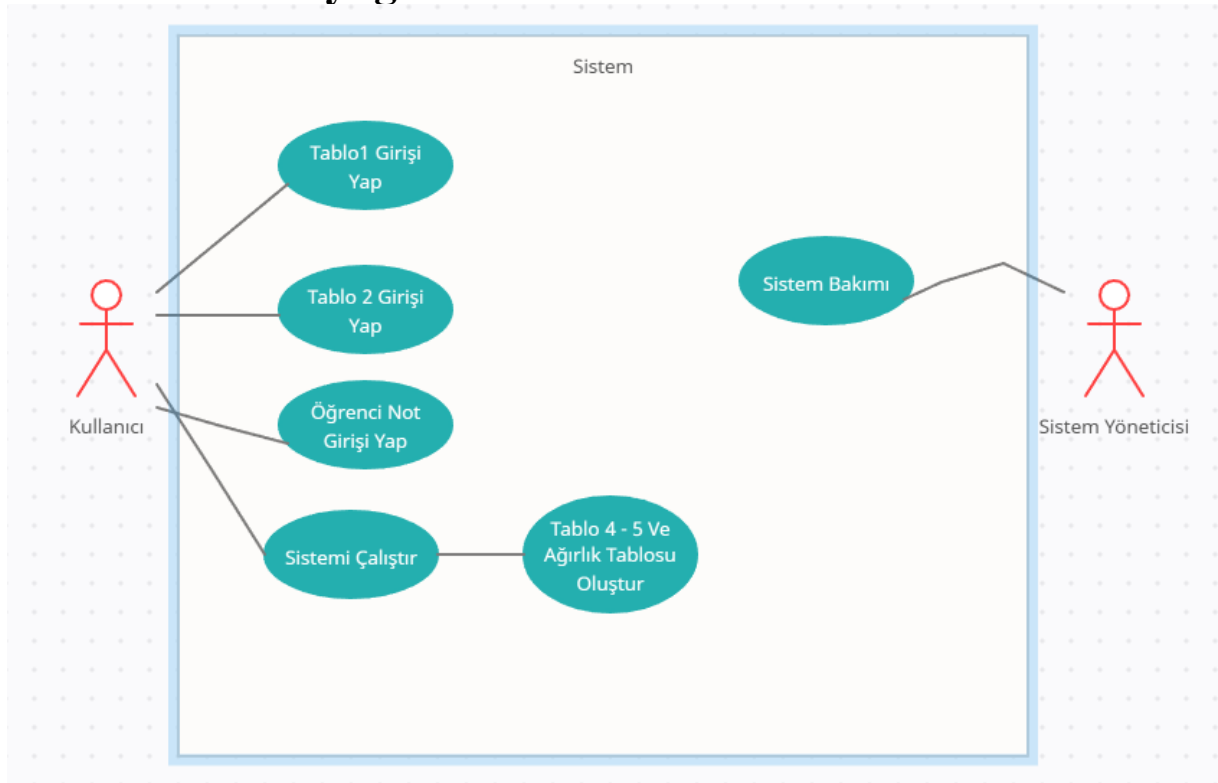
6. Sayfa İsimlendirme ve Veri Düzenleme:

- Her öğrenci için oluşturulan raporlar, öğrenci isimlerine göre sayfalar halinde düzenlenmelidir.
- Sayfa isimleri, geçersiz karakterlerden temizlenerek düzenli bir şekilde oluşturulmalıdır.

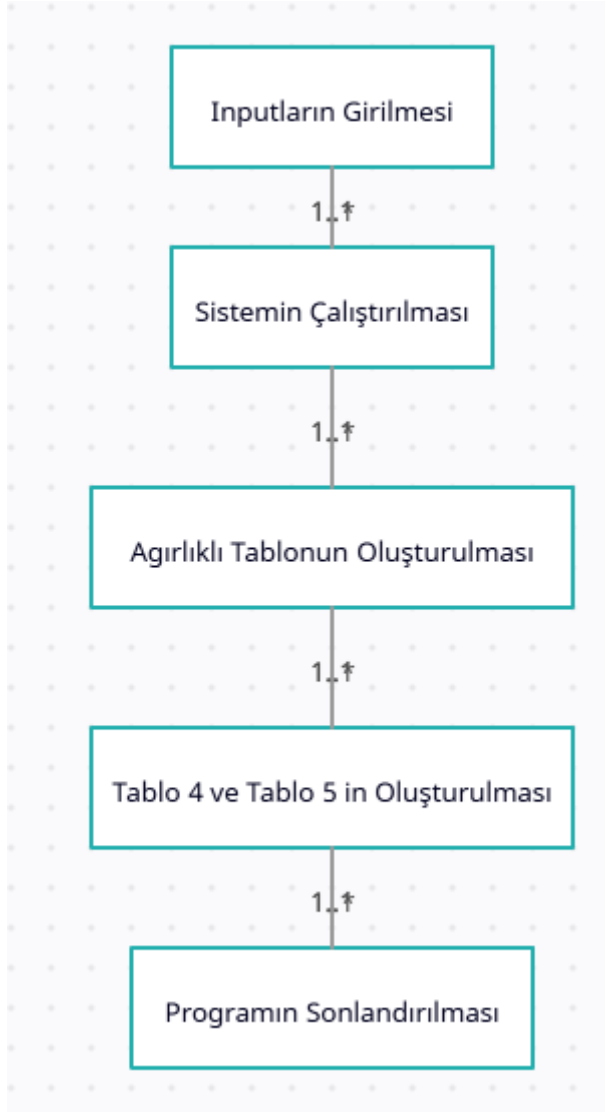
7. Çıktıların İndirilmesi:

- Oluşturulan raporlar ve başarı oranı hesaplamaları, her öğrenci için ayrı sayfalarda yer alacak şekilde "Output_Tables.xlsx" dosyasına kaydedilmelidir.
- Sistem, dosya oluşturulmasının ardından kullanıcıya dosyanın başarıyla kaydedildiğini bildirmelidir.

2.3 Use-Case diyagramı



2.4 Akış diyagramı



3 TASARIM

3.1 Mimari tasarım

1. Veri Girişi Katmanı

- Öğrenci notları, ders ağırlıkları ve ilişki değerleri Excel dosyasından okunur.
- **pandas** kütüphanesi ile veriler doğrulanır ve işlenir.

2. İşlem Katmanı

- **Ağırlıklı Değerlendirme:** Ders ağırlıkları ve notlara göre ağırlıklı toplamlar hesaplanır.
- **Başarı Oranı Hesaplama:** Her öğrenci ve program çıktısı için başarı yüzdesi belirlenir.
- **İlişki Değeri:** Tablo1'deki ilişki katsayılarına göre başarı oranları uyarlanır.

3. Veri Çıktısı Katmanı

- Sonuçlar her öğrenci için ayrı sayfalarda Excel dosyasına yazılır.

- Ağırlıklı toplamalar, maksimum değerler ve başarı yüzdeleri raporlanır.

4. Hata Yönetimi ve Doğrulama

- Eksik veya hatalı veriler kontrol edilerek işlenir.
- Hata durumlarında kullanıcı bilgilendirilir.

Bu yapı, veri akışını düzenler ve tüm işlemleri modüler bir şekilde gerçekleştirir.

3.2 Kullanılacak teknolojiler

- **Python:** Veri işleme ve analiz için.
- **Kütüphaneler:** pandas, openpyxl, numpy.
- **Excel (XLSX):** Girdi ve çıktı dosyaları.
- **Geliştirme Ortamı:** PyCharm.
- **Diğer Teknolojiler:** Github

3.3 Veri tabanı tasarımı

Veri tabanı yapmadık.

3.4 Kullanıcı arayüzü tasarımı

Bu proje için bir kullanıcı arayüzü tasarımı yerine komut satırı tabanlı bir çözüm tercih edilmiştir. Kullanıcı, gerekli dosyaları (örneğin Data.xlsx) belirtilen dizine koyarak ve Python betiğini çalıştırarak işlemleri gerçekleştirebilir. Elde edilen sonuçlar otomatik olarak Excel dosyalarına yazılır ve kullanıcıya görsel bir analiz sunulur.

Kullanıcı Ne Yapmalı?

- **Girdi Dosyasını Hazırlayın:** Girdi dosyası olan Data.xlsx içinde gerekli sayfalar (Tablo1, Tablo2, TabloNot) doğru biçimde doldurulmalıdır.
- **Betiği Çalıştırın:** Python kodu çalıştırıldığında çıktı dosyaları otomatik olarak oluşturulur.
- **Sonuçları İnceleyin:** Çıktılar, öğrencilerin başarı oranlarını ve program çıktıları ile olan ilişkilerini içerir. Excel dosyası üzerinden bu sonuçlar incelenebilir.

Önemli Bilgiler ve İşlevler

- Kullanıcı, Excel şablonlarını doğru formatta sağlamalıdır.
- Sonuçlar, bireysel öğrenci performansları ve genel başarı oranlarını içerir.
- Başarı oranları, program çıktıları ile ilişkilendirilmiş olarak analiz edilir.

Yazılımı Çalıştırma Adımları

1. **Python Yüklü Olmalı:** Bilgisayarda Python ve gerekli kütüphaneler (pandas, openpyxl) kurulu olmalıdır.
2. **Girdi Dosyasını Hazırlayın:** Data.xlsx şablonunu doldurun ve belirtilen dizine kaydedin.
3. **Python Betiğini Çalıştırın:** Betiği çalıştırarak sonuç dosyalarının oluşturulmasını bekleyin.
4. **Çıktı Dosyalarını İnceleyin:** Weighted_Table.xlsx ve Output_Tables.xlsx dosyalarını açarak detaylı analiz sonuçlarını inceleyin.

Bu adımlar kullanıcılar için kolay, erişilebilir ve anlaşılır bir işlem süreci sunar.

4 UYGULAMA

4.1 Kodlanan bileşenlerin açıklamaları

1. Dosya Yönetimi

- **İşlev:** Projenin giriş ve çıkış dosyalarını belirler.
- **Açıklama:** Kullanıcı, veri işleme sürecinde kullanılacak dosya adlarını kolayca değiştirebilir.

```
input_file = "Data.xlsx"  
weighted_file = "Weighted_Table.xlsx"  
output_file = "Output_Tables.xlsx"
```

2. Girdi Verilerinin Okunması

- **İşlev:** Excel dosyasından verileri okur ve analiz için uygun bir yapıya dönüştürür.
- **Açıklama:** `pandas` kütüphanesi kullanılarak "Tablo1", "Tablo2" ve "TabloNot" adlı sayfalardan veri okunur.

```
excel_data = pd.ExcelFile(input_file)  
sheet2_data = excel_data.parse("Tablo2")  
tablo_not_data = excel_data.parse("TabloNot")  
sheet1_data = excel_data.parse("Tablo1", header=1)
```

3. Ağırlıklı Değerlendirme Tablosu

- **İşlev:** Ders ağırlıklarına göre öğrencilerin performansını değerlendirir.
- **Açıklama:** "Tablo2"den alınan ağırlıklar ve notlar çarpılarak her öğrencinin ağırlıklı başarı toplamı hesaplanır.

```
weights_raw = sheet2_data.iloc[0, 1:6]  
weights = pd.to_numeric(weights_raw.replace(r'[^\d.]', "", regex=True),  
errors='coerce').fillna(0).astype(float)  
grades = sheet2_data.iloc[2:, 1:6].apply(pd.to_numeric,
```

```
errors='coerce').fillna(0)
weighted_grades = grades * weights.values / 100
weighted_grades['Ağırlıklı Toplam'] = weighted_grades.sum(axis=1)
```

4. Öğrenci Sonuçlarının Hesaplanması

- **İşlev:** Her öğrencinin notlarını ağırlıklı değerlendirerek başarı yüzdesini hesaplar.

- **Açıklama:** Öğrencilerin notları ders ağırlıklarıyla çarpılarak toplam başarı ve başarı oranları bulunur.

```
for i, student in enumerate(students):
    student_grades = student_data.iloc[i].values
    weighted_grades = weights_data.values * student_grades
    weighted_grades_sum = weighted_grades.sum(axis=1)
    success_percentage = (weighted_grades_sum / max_values) * 100
```

5. Başarı Oranı Hesaplama

- **İşlev:** Program çıktıları ile ilişkilendirilmiş başarı oranlarını hesaplar.

- **Açıklama:** "Tablo1"deki ilişki değerlerine göre her dersin başarı oranı program çıktılarıyla ilişkilendirilir.

```
for j in range(len(program_outcomes)):
    related_outcomes = sheet1_data.iloc[j, 1:6].apply(pd.to_numeric,
errors='coerce').fillna(0)
    related_success = success_percentages * related_outcomes
    tablo5_values.append(related_success.sum() / num_outcomes)
```

4.2 Görev dağılımı

Bileşenlerin tasarım ve geliştirme aşamalarındaki görev dağılımı:

Yapabilmek için herkes kendi çapında yapmaya çalıştı sonunda beraber son haline getirdik.

Raporun hazırlanması sürecindeki görev dağılımı:

Raporda da şemaları ve açıklama görevini farklı kişilere dağıtarak görev dağılımını gerçekleştirdik.

4.3 Karşılaşılan zorluklar ve çözüm yöntemleri

1. **Veri Formatlama:** Excel dosyasındaki tutarsız veri formatları, ağırlıkların ve notların işlenmesini zorlaştırdı.
 - o **Çözüm:** pandas ile veri temizleme ve eksik değer doldurma yöntemleri uygulandı.
2. **Ağırlık Hesaplamaları:** Metin formatındaki ağırlık değerleri nedeniyle hesaplamalar karmaşık hale geldi.
 - o **Çözüm:** metinler temizlendi ve sayısal değerlere dönüştürüldü.

3. **Tablo Başlıkları:** Sütun isimlerinin tutarsızlığı veri işlemeyi engelledi.
 - **Çözüm:** Başlıklar dinamik olarak kod içinde yeniden düzenlendi.
4. **Sayfa İsimlendirme:** Geçersiz karakterler içeren sayfa isimleri sorun yarattı.
 - **Çözüm:** Geçersiz karakterleri temizleyen bir fonksiyon kullanıldı.
5. **Başarı Oranı Hesaplamaları:** İlişki katsayılarının ve başarı oranlarının doğru hesaplanması karmaşıktı.
 - **Çözüm:** Formüller dikkatle optimize edilerek hesaplamalar tamamlandı.
6. **Excel Performansı:** Çok sayfalı dosya yazımı sırasında performans sorunları oluştu.
 - **Çözüm:** Dosya yazma işlemleri optimize edildi.

Bu zorluklar çözüm odaklı yaklaşımlarla giderilmiş ve proje başarılı bir şekilde tamamlanmıştır.

4.4 Proje isterlerine göre eksik yönler

Crawler yazmadık.

Database eklemedik.

TEST VE DOĞRULAMA

4.5 Yazılımın test süreci

Yazılımın test süreci, aşağıdaki adımlarla gerçekleştirilmiştir:

1. Fonksiyonel Testler:

- **Amaç:** Kodun her bir fonksiyonunun doğru çalıştığını doğrulamak.
- **Uygulama:**
 - **Ağırlıklı Tablo Testi:** Ağırlıklı not hesaplamalarının doğru yapıldığını kontrol etmek için farklı veri setleri kullanıldı.
 - **Öğrenci Sonuçları Testi:** Her bir öğrenci için üretilen Excel sayfalarının doğru hesaplamalar içerdiği test edildi.
 - **Başarı Oranı Testi:** Program çıktılarının başarı oranlarının doğru hesaplandığı manuel kontrolle doğrulandı.

2. Veri Doğrulama Testleri:

- **Amaç:** Excel'den alınan verilerin eksiksiz ve doğru işlendiğini kontrol etmek.
- **Uygulama:**
 - Eksik veya hatalı verilerin doğru şekilde ele alındığı test edildi.
 - Geçersiz veri formatlarının temizlenip temizlenmediği kontrol edildi.

3. Performans Testleri:

- **Amaç:** Büyük veri setlerinde yazılımın çalışma hızını değerlendirmek.
- **Uygulama:**
 - Çok sayıda öğrenci ve tablo içeren dosyalarda yazılımın performansı ölçüldü.
 - Performans darboğazları optimize edildi.

4. Kullanılabilirlik Testleri:

- **Amaç:** Kullanıcının yazılımı kolayca çalıştırıp kullanabildiğini doğrulamak.
- **Uygulama:**
 - Yazılımın çıktılarının okunabilir ve düzenli olduğu kontrol edildi.
 - Çalıştırma adımlarının kullanıcı dostu olup olmadığı değerlendirildi.

5. Hata Yönetimi Testleri:

- **Amaç:** Beklenmedik durumlarda yazılımın hataları doğru şekilde ele aldığını kontrol etmek.
- **Uygulama:**
 - Geçersiz Excel dosyaları ve eksik veri durumları simüle edilerek yazılımın hata mesajları test edildi.

Sonuç olarak, yazılım her türlü senaryoda başarıyla test edilmiş ve kullanım için hazır hale getirilmiştir.

4.6 Yazılımın doğrulanması

Yazılımın doğrulanması, işlevsellik ve doğruluk açısından test edilmiştir:

1. **Fonksiyonel Doğrulama:** Ağırlıklı notlar ve başarı oranlarının doğru hesaplanıp hesaplanmadığı kontrol edilmiştir.
2. **Veri Doğrulama:** Excel verilerinin doğru işlenmesi ve hatalı verilerin temizlenmesi sağlanmıştır.
3. **Hata Yönetimi:** Hatalı girişlerde yazılımın düzgün çalıştığı ve doğru hata mesajları verdiği test edilmiştir.
4. **Performans:** Yazılımın büyük veri setlerinde de düzgün çalıştığı test edilmiştir.
5. **Kullanıcı Dostu:** Kullanıcıların yazılımı kolayca kullanabildiği doğrulanmıştır.

Sonuçta, yazılım gereksinimlere uygun ve doğru çalışmaktadır.

Kaynakça:

- <https://chatgpt.com/>
- <https://kerteriz.net/python-ile-excel-dosya-islemleri/>
- <https://blog.aspose.com/tr/cells/read-excel-files-using-python/>
- <https://www.w3schools.com/python/>