



T.C
KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR / YAZILIM MÜHENDİSLİĞİ

PROJE KONUSU:
BAŞARI HESAPLAMA ARAYÜZÜ

ÖĞRENCİ ADI:
ÖĞRENCİ NUMARASI:
Sema Su YILMAZ - 220502016
Zehra YARDIMCI - 220502038
Senem ADALAN - 220502045
Şevval ÖZEREN - 220501028
Hilal AYDIN - 220501030

DERS SORUMLUSU:
PROF. DR. TARIK DURU / DR. ÖĞR. ÜYESİ ELİF PINAR
HACİBEYOĞLU

TARİH:
22 OCAK 2025

İÇİNDEKİLER

1. GİRİŞ.....	3
1.1. Projenin Amacı.....	3
2. GEREKSİNİM ANALİZİ.....	3
2.1. Arayüz Gereksinimleri.....	3
2.2. Fonksiyonel Gereksinimler.....	4
2.3. Use-Case Diyagramı.....	5
3. TASARIM.....	6
3.1. Mimari Tasarım.....	6
3.2. Kullanılacak Teknolojiler.....	6
3.3. Veri Tabanı Tasarımı.....	7
3.4. Kullanıcı Arayüzü Tasarımı.....	9
4. UYGULAMA.....	12
4.1. Kodlanan Bileşenlerin Açıklamaları.....	12
4.2. Görev Dağılımı.....	14
4.3. Karşılaşılan Zorluklar ve Çözüm Yöntemleri.....	14
4.4. Proje İsterlerine Göre Eksik Yönler.....	14
5. TEST VE DOĞRULAMA.....	14
5.1. Yazılımın Test Süreci.....	14
5.2. Yazılımın Doğrulanması.....	16
6. GİTHUB BAĞLANTILARI.....	17

1. GİRİŞ

1.1 Projenin Amacı

Bu projenin amacı, bir eğitim programı çerçevesinde ders çıktıları, program çıktıları ve öğrenci değerlendirme kriterleri arasındaki ilişkiyi sistematik ve kolay yönetilebilir bir arayüz üzerinden sağlamak, kullanıcıların bu verileri hızlıca girmesine ve düzenlemesine imkan tanımaktır. Bu proje, eğitim kurumlarının kalite standartlarını karşılamasına ve çıktı odaklı eğitim modellerinin etkin bir şekilde uygulanmasına yardımcı olacaktır. Projede yazılım geliştirme kısmı Python dili ile programlanmış ve tablo verileri için SQL Server veri tabanı kullanılmıştır.

2. GEREKSİNİM ANALİZİ

2.1 Arayüz Gereksinimleri

Ders Ekleme ve Öğrenci Listesi

- Dersler adları yazılarak eklenir.
- Eklenen derslerden istenen seçilerek devam edilir.
- Öğrenci listesinde öğrencilerin numaraları bulunur.

Program Çıktıları ve Öğrenme Çıktıları Yükleme

- Hem program çıktıları hem de ders çıktıları kullanıcıdan manuel giriş alınarak eklenir.

Değerlendirme Kriterleri ve Ağırlıkları Giriş Ekranı

- Minimum 5 değerlendirme kriteri girilir.
- Her bir kriterle birlikte ağırlığı girilir.
- Toplam ağırlık 100 olmalıdır. Aksi halde uyarı vermektedir.
- Kaydet butonu ile girilen bilgiler saklanmaktadır.

Tablo 1 ve Tablo 2 Girişi

- Kullanıcıdan manuel olarak alınan girişlerle tablo 1 ve tablo 2 oluşturulur.
- Giriş değerleri [0, 1] aralığında olmalıdır.

Tablo Notlar Girişi

- Kullanıcıdan manuel giriş alınarak kaydedilmektedir.

Tablo 4 ve Tablo 5 Görüntülenmesi

- Öğrenci listesi tablosundan seçilen öğrencinin tablo 4 ve tablo 5'i oluşturulup görüntülenmektedir.

Reset ve Ders Seçimi

- Kullanıcı, işlemlerini tamamladıktan sonra yeni bir ders için hesaplama yapmak üzere "Ders Seçim" ekranına geri dönebilmektedir.

2.2 Fonksiyonel Gereksinimler

Veri Tabanı Bağlantısı (İG-1)

- SQL Server ile bağlantı kurulmasını sağlar.
- "RelationMatrix" adlı bir veri tabanı oluşturur ya da mevcut olup olmadığını kontrol eder.

Tablo Yönetimi (İG-2)

- "Lessons", "Students", "CourseOutcomes", "ProgramOutcomes" gibi tabloların varlığını kontrol eder ve eksik tabloları oluşturur.
- Tablo yapılarında gerekli sütunları kontrol eder ve ekler.

Veri Ekleme ve Silme (İG-3)

- "Lessons", "Students", "CourseOutcomes", "ProgramOutcomes" gibi tablolara veri ekleme ve silme işlemleri yapılır.
- İlişkisel tablolar ("ProgramCourseRelations", "CourseEvaluationRelations") için ilişki değerlerini ekler.

Excel Dosyası Oluşturma (İG-4)

- Derslere ait verilerle ilişkili olarak farklı tablolar (Table1, Table2, Table3, Table4, Table5) oluşturur ve bu verileri Excel dosyalarına kaydeder.
- Excel dosyasında her ders için ayrı bir sayfa oluşturulur.

Not Hesaplama (İG-5)

- Öğrencilerin başarı oranlarını ve ders çıktılarıyla ilişkilerini hesaplar.
- Hesaplamalar sonucunda Excel dosyasına "Total", "Max", "% Success" gibi sütunlar eklenir.

İlişkisel Veri İşlemleri (İG-6)

- Ders çıktıları ve program çıktıları arasındaki ilişki değerlerini tutar ve günceller.
- Değerlendirme kriterleriyle ilişkili verileri işler ve tablolar arasında ilişkilendirir.

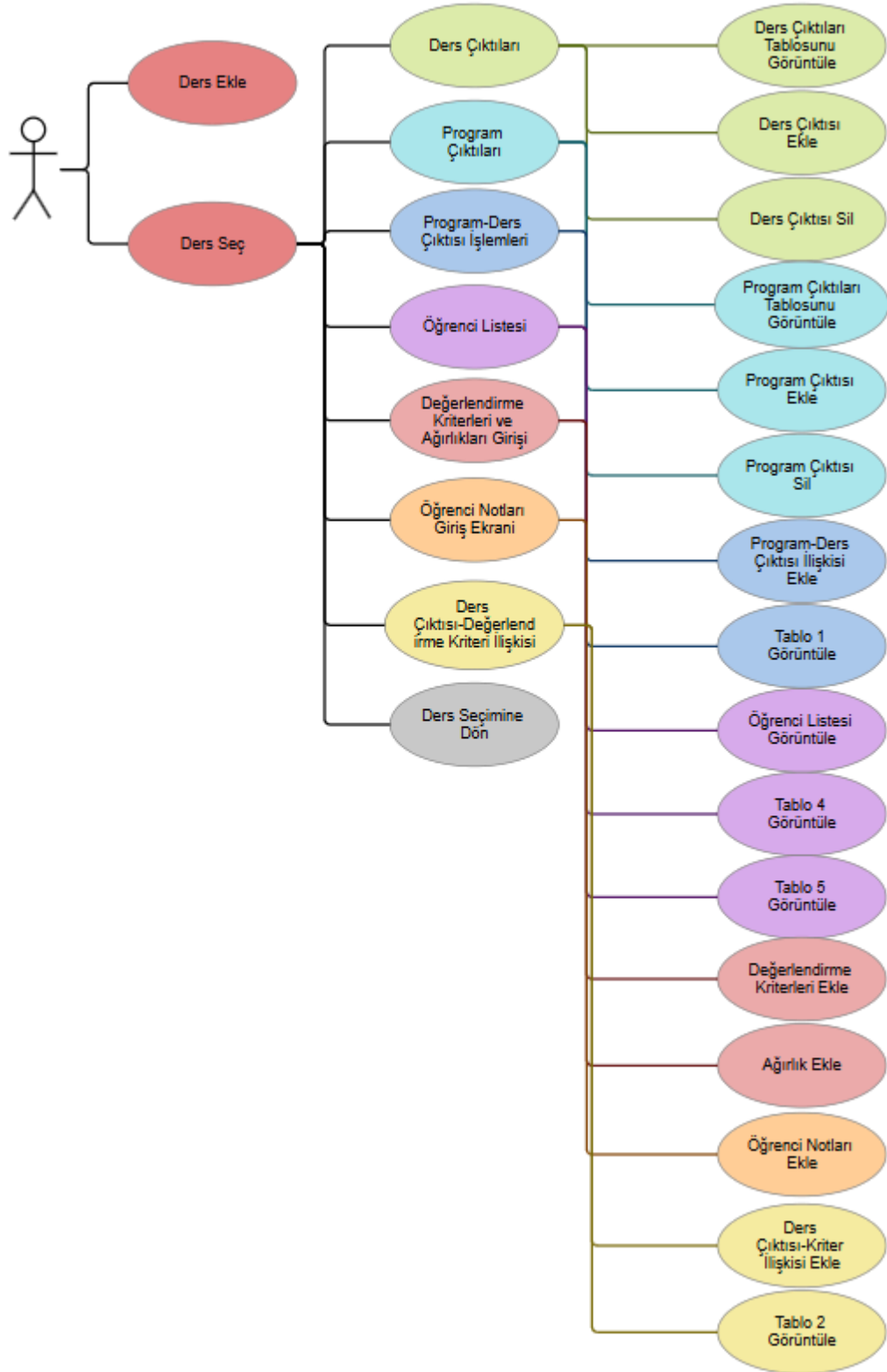
Veri Filtreleme (İG-7)

- Ders ID'sine göre veri filtreleme yapılır.
- Öğrenci numarasıyla öğrenciye özgü veriler çekilir.

Hata Kontrolü (İG-8)

- Veri girişinde hatalı veya eksik veri olup olmadığını kontrol eder.
- Ağırlık toplamalarının %100 olması gerektiği durumlarda uyarılar gösterir.

2.3 Use-Case Diyagramı

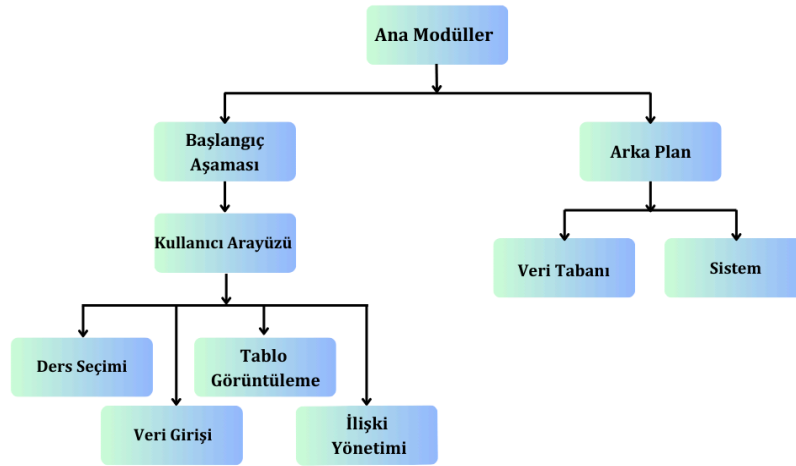


3. TASARIM

3.1 Mimari Tasarım

- Veri Tabanı Katmanı
 - Veri Tabanı Bağlantısı: ODBC kullanılarak SQL Server ile iletişim kurar.
 - Tabloların oluşturulması ve yönetimini sağlar.
 - Veri ekleme, veri çekme, veri silme ve güncellemeler bu verilere dayanır.
- İşlem Katmanı
 - İlişkisel verilerin işlenmesini gerçekleştirir.
 - Not hesaplama ve veri filtreleme algoritmaları uygulanır.
 - Excel dosyalarının oluşturulması ve işlenmesi sağlanır.
- Kullanıcı Katmanı
 - Kullanıcı etkileşimlerini yönetir ve veriyi görselleştirir.
 - Ders seçimi ve tablo sonuçlarının gösterilmesi gibi kısımları içerir.

MODÜL DİYAGRAMI



3.2 Kullanılacak Teknolojiler

- Bu projede Python programlama dili kullanılmıştır.
- Pyodbc kütüphanesi, Open Database Connectivity (ODBC) protokolünü kullanarak veri tabanlarına bağlanmak için kullanılan bir kütüphanedir. Bu kütüphane sayesinde Python uygulamalarında SQL tabanlı Microsoft SQL Server ile kolayca iletişim kurulabilir.
- Openpyxl kütüphanesi kullanılmıştır. Microsoft Excel dosyalarıyla çalışmak için kullanılan bir kütüphanedir. Bu kütüphane sayesinde Excel dosyaları okunabilir, oluşturulabilir ve düzenlenebilir.
- Veri analizi işlemleri için Pandas kütüphanesi kullanılmıştır. Masaüstü kullanıcı arayüzü Tkinter kütüphanesi aracılığıyla yapılmıştır. Görsel dosyaları işlemek ve Tkinter'de göstermek için PIL kütüphanesi kullanılmıştır.

3.3 Veri Tabanı Tasarımı

RelationMatrix Veri Tabanı

- SQL veri tabanında RelationMatrix adlı bir veri tabanı oluşturuldu.
- CourseOutcomes, ProgramOutcomes, ProgramCourseRelations, EvaluationCriteria, CourseEvaluationRelations ve Lessons olmak üzere 6 tablo oluşturulmuştur.

CourseOutcomes (Ders Çıktıları) Tablosu

- Ders çıktılarının metinsel bir açıklamasını içerir.
- Her ders çıktısına bir id atanır, bu id otomatik olarak artar ve birincil anahtar olarak kullanılır.

ProgramOutcomes (Program Çıktıları) Tablosu

- Program çıktılarının metinsel bir açıklamasını içerir.
- Her program çıktısına bir id atanır, bu id otomatik olarak artar ve birincil anahtar olarak kullanılır.

ProgramCourseRelations (Program ve Ders Çıktıları İlişkisi) Tablosu

- Program çıktıları ile ders çıktıları arasındaki ilişkiyi tanımlar.
- Program çıktısının kimliği, ProgramOutcomes.id ile ilişkilidir.
- Ders çıktısının kimliği, CourseOutcomes.id ile ilişkilidir.
- RelationValue, 0 ile 1 arasında olmalıdır.

EvaluationCriteria (Değerlendirme Kriterleri) Tablosu

- Ders veya program çıktılarının değerlendirilmesi için kullanılan kriterleri tanımlar.
- Kriterin adı ve kriterin ağırlığını tutar.

CourseEvaluationRelations (Ders Çıktıları ve Değerlendirme Kriterleri İlişkisi) Tablosu

- Ders çıktıları ile değerlendirme kriterleri arasındaki ilişkiyi tanımlar.
- Ders çıktısının kimliği, CourseOutcomes.id ile ilişkilidir.
- Değerlendirme kriterinin adı, EvaluationCriteria Criteria ile ilişkilidir.
- RelationValue, ilişkinin değerini tutar.

Lessons Tablosu

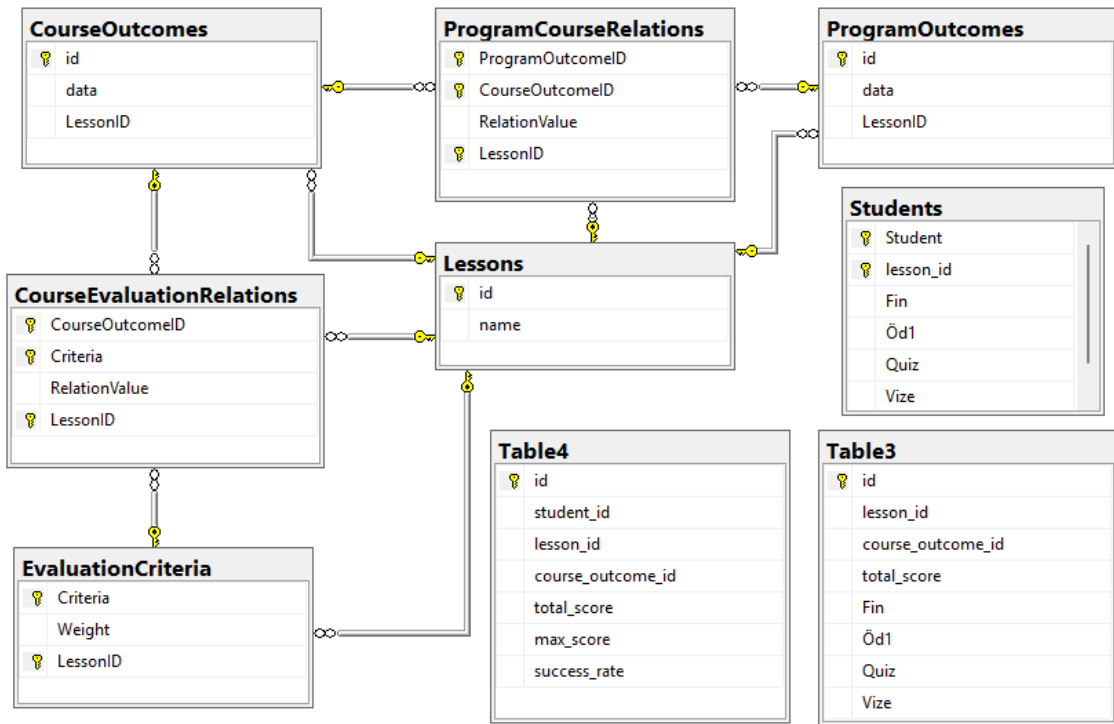
- Sistemdeki tüm dersleri depolayan temel tablodur.
- Ders adlarını ve kimliklerini tutar. Diğer tüm tablolar bu tabloya bağlıdır.

Students Tablosu

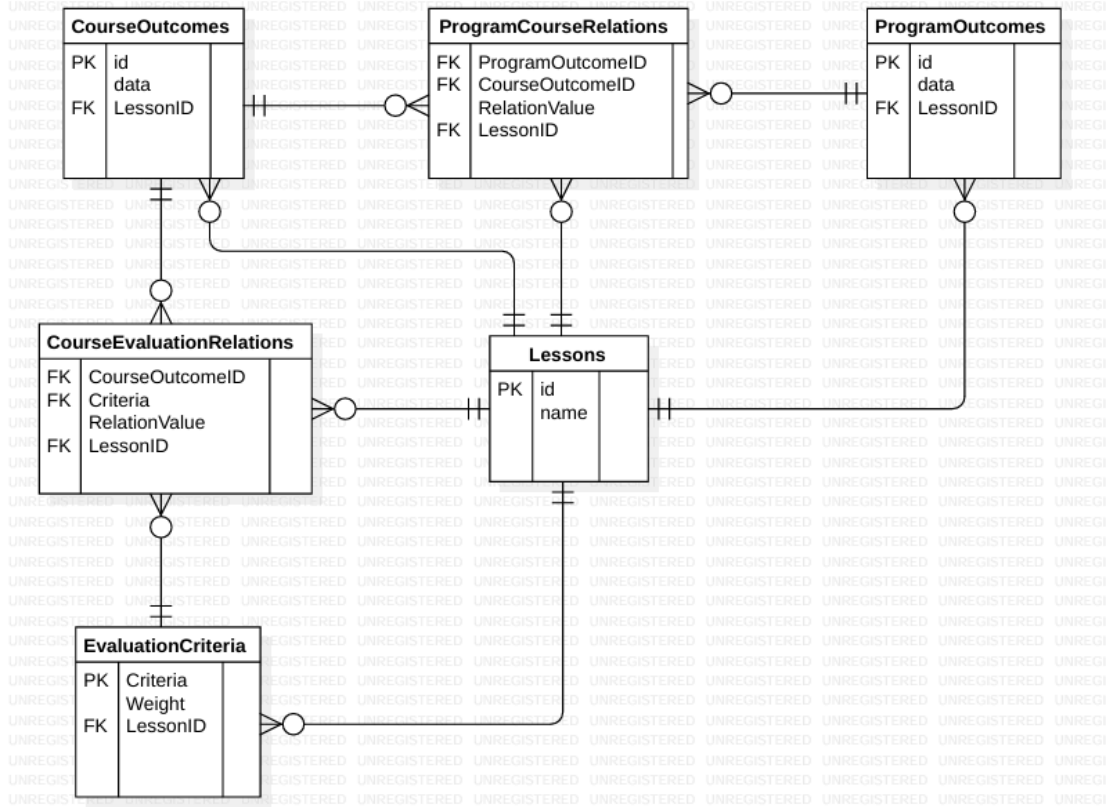
- Öğrenci bilgilerini içerir.
- Öğrenci numaraları ve her bir bölümden (Fin, Quiz, Vize) aldığı notlar tutulmaktadır.

Table3 ve Table4 Tablosu

- Table3 ve Table4'deki değerler tutulmaktadır.



İlişkisel Veri Tabanı Gösterimi



ER Diyagramı

3.4 Kullanıcı Arayüzü Tasarımı



**KOCAELİ SAĞLIK VE TEKNOLOJİ
ÜNİVERSİTESİ**

Ders Seçimi Yapınız

101 - Bilgisayar Muhendisliğine Giriş

Segeceğiniz ders mevcut değilse, aşağıdan ekleyebilirsiniz:

Ders Ekle

Devam Et

Ana ekranda ders seçimi yapılır. Eğer ders mevcut değilse eklenir. Devam et butonuyla sisteme girilir.

**Ders Seçimine
Dön**

Ders Çıktıları

Program Çıktıları

**Program - Ders
Çıktısı İşlemleri**

Öğrenci Listesi

**Değerlendirme
kriterleri
ve ağırlıkları
giriş ekranı**

**Öğrenci Notları
Giriş Ekranı**

Ders Öğrenme Çıktıları

id	data	Lessor
1	bmg için c	101
2	bmg için c	101
3	bmg için c	101
4	bmg için c	101

Ders Çıktısı eklerken metin giriniz,
silme için seçilen Ders Çıktısının id'sini giriniz.

Ders Çıktısı Ekle **Seçilen Çıktıyı Sil**

Ders çıktıları kısmından ilgili dersle ilgili çıktılar bir tabloda görüntülenir. Aynı zamanda ekleme ve silme işlemleri yapılabilir.

Ders Seçimine Dön

Ders Çıktıları

Program Çıktıları

Program - Ders Çıktısı İşlemleri

Öğrenci Listesi

Değerlendirme kriterleri ve ağırlıkları giriş ekranı

Öğrenci Notları Giriş Ekranı

Program Çıktısı İşlemleri

id	data	Lessor
1	bm giriş için p1	101
2	bm giriş için p2	101
3	bm giriş için p3	101

Program Çıktısı eklerken metin giriniz, silmek için seçilen Ders Çıktısının id'sini giriniz.

Program Çıktısı Ekle

Seçilen Çıktıyı Sil

Program çıktıları kısmından ilgili programla ilgili çıktılar bir tabloda görüntülenir. Aynı zamanda ekleme ve silme işlemleri gerçekleştirilir.

Ders Seçimine Dön

Ders Çıktıları

Program Çıktıları

Program - Ders Çıktısı İşlemleri

Öğrenci Listesi

Değerlendirme kriterleri ve ağırlıkları giriş ekranı

Öğrenci Notları Giriş Ekranı

Tablo 1 Görüntüle

Program Outcom	1	2.0	3.0	4.0	Rel Value
1	0.2	0.0	1.0	0.3	0.38
2	1	0.7	0.3	0.0	0.5
3	0.4	1.0	0.1	1.0	0.62

Ders ID'sini giriniz:

Program Çıktısı ID'sini giriniz:

Ders Çıktısı ID'sini giriniz:

İlişki değerini giriniz:

Kaydet

Tablo 1 oluştur

Program-Ders çıktısı ilişkileri kısmından ilgili ders çıktısı ve program çıktısının ilişki değerleri eklenir. Sonrasında bu değerler tablo1'de gösterilir.

Kriter Adı	Ağırlık (%)
Kriter 1: vize	30
Kriter 2: final	40
Kriter 3: quiz1	10
Kriter 4: quiz2	10
Kriter 5: odev	10

Kaydet

Değerlendirme kriterleri ve ağırlıkları giriş ekranı kısmında kriterler ve bunların ağırlıkları belirlenerek kaydedilir.

Öğrenci Not Giriş Ekranı

Öğrenci No giriniz: 220501000

final:	80
odev:	100
quiz1:	80
quiz2:	90
vize:	70

Kaydet

Öğrenci notları giriş ekranı kısmında öğrencinin numarası ve bu öğrencinin belirlenen değerlendirme kriterlerinden aldığı notlar eklenerek kaydedilir.

KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ - Ders Verileri Giriş Ekranı

Seçilen Ders: 1 - YAZILIM LAB 1

Ders seçim ekranından çıktınız.

Ders Seçimine Dön

Ders Çıktıları

Program Çıktıları

Program - Ders Çıktısı İşlemleri

Öğrenci Listesi

Değerlendirme kriterleri ve ağırlıkları giriş ekranı

Öğrenci Notları Giriş Ekranı

Ders Çıktısı ve Değerlendirme Kriteri Girişi

Öğrenci Listesi

Student

22018

YAZILIM LAB 1 - Table 4

Student ID	Course Outcome	Final	Ödev1	Ödev2	Proje	Vize	Total	Max	% Success
22018	Yazılım Mühendisliği	0	0	0	24	24	30	80	
22018	Yazılım mühendisliği	7	0	0	0	7	10	70	
22018	Meslek hayatında	0	0	8.5	0	8.5	10	85	
22018	Yazılım geliştirme	0	8	0	0	8	10	80	
22018	Yazılım yaşam döngüsü	36	0	0	0	36	40	90	

YAZILIM LAB 1 - Table 5

Student ID	Program Outcome	Yazılım Mühendisliği	Yazılım mühendisliği	Meslek hayatında	Yazılım geliştirme	Yazılım yaşam döngüsü	Success Rate
22018	Bilgisayar mühendisliği	80	0	40	0	80	
22018	Mühendislik problemleri	0	51	0	0	85	
22018	Farklı uygulamalar	0	0	80	0	80	
22018	Problemlerin çözümü	0	0	0	0	0	
22018	Mühendislik uygulamaları	0	0	0	0	0	
22018	Problemlerin çözümü	0	0	0	0	0	
22018	Mühendislik projesi	0	0	0	0	0	

Eklenen öğrenciler için tablo 4 ve tablo 5 görüntülenir.

4. UYGULAMA

4.1 Kodlanan Bileşenlerin Açıklamaları

- **create_connection():** SQL Server ile bağlantı oluşturur ve bağlantı nesnesini döndürür.
- **check_database():** "RelationMatrix" veri tabanının varlığını kontrol eder yoksa oluşturur.
- **get_connection(database=None):** SQL Server veri tabanına bağlanır ve ODBC sürücüsünü kullanarak bir bağlantı nesnesi oluşturur.
- **check_tables():** Gerekli tabloların mevcut olup olmadığını kontrol eder. Eksik tablolar varsa oluşturur.
- **insert_data_into_table(table_name, data, lesson_id):** Belirtilen tabloya veri ekler. Örneğin, bir ders çıktısı veya program çıktısı eklemek için kullanılır.
- **delete_data_from_table_by_id(table_name, record_id):** Belirtilen tablodan belirli bir kaydı siler.
- **get_data_from_table(table_name):** Veri tabanındaki bir tablodan tüm verileri çeker ve döndürür.
- **get_data_from_table_with_filter(table_name, l_id):** Belirli bir ders ID'sine göre veri filtreler ve sonuçları döndürür.
- **create_table3():** Derslere ilişkin ağırlıklı değerlendirme kriterlerini hesaplayarak bir Excel dosyası oluşturur.
- **create_table4():** Öğrenci başarı oranlarını ve kriter bazlı notlarını hesaplayarak Excel dosyasına kaydeder.
- **create_table5():** Program çıktıları ve başarı oranları ile ilgili bir rapor oluşturur ve Excel dosyasına kaydeder.
- **create_notes():** Öğrenciler için not tablosu oluşturur ve her ders için ayrı bir Excel sayfasına kaydeder.
- **display_data_in_treeview(table_name, frame, x, y, l_id):** Belirtilen bir tabloyu Tkinter Treeview bileşeninde gösterir.
- **on_row_select(event, frame, tree, student_no, l_id):** Kullanıcının Treeview içinde seçtiği satırın detaylarını görüntüler.
- **check_lessons():** Lessons tablosunun var olup olmadığını kontrol eder, yoksa oluşturur. Lessons tablosundaki tüm dersleri alır ve formatlanmış bir liste döndürür.
- **add_course():** Kullanıcının giriş yaptığı yeni bir ders adını alır ve Lessons tablosuna ekler. Ekleme sonrası Lessons tablosundaki güncel verileri alarak dropdown menüsünü yeniler. Başarılı bir eklemeden sonra giriş alanını temizler.
- **proceed_to_next():** Seçili dersin kontrolünü yapar. Eğer bir ders seçilmişse ders bilgilerini alır ve bir sonraki adıma geçmek için gerekli işlemleri yapar. Seçim yapılmadıysa bir hata mesajı gösterir.
- **insert_table(table_name, data, lesson_id, frame):** Verilen table_name tablosuna belirtilen data ve lesson_id değerlerini ekler. Güncel tablo verilerini ilgili bir çerçeveye (frame) yükleyerek gösterir.
- **del_from_table(table_name, id, lesson_id, frame):** Verilen id ile belirtilen tablo (table_name) içindeki kaydı siler. Güncel tablo verilerini çerçevede yeniden gösterir.
- **show_frame1():** Kullanıcıyı ana ders seçim ekranına yönlendirir.

- **insert_relation_value(program_outcome_id, course_outcome_id, relation_value, lesson_id):** Program çıktıları ile ders çıktıları arasındaki ilişkileri ve lesson_id ile bağlanan ilişki değerini veri tabanına ekler.
- **fetch_table_data(table_name):** Veri tabanından belirtilen tablodaki verileri çeker.
- **fetch_lesson_names():** Veri tabanındaki tüm derslerin adlarını ve ID'lerini döndürür.
- **fetch_evaluation_data():** Değerlendirme kriterleri ve ağırlıklarını çeker.
- **fetch_evaluation_relations():** Ders çıktıları ile değerlendirme kriterleri arasındaki ilişki verilerini döndürür.
- **fetch_relations():** ProgramCourseRelations tablosundaki ilişkileri alır ve Program çıktıları, Ders çıktıları, İlişki değerleri ile LessonID bilgilerini döndürür.
- **create_table1():** Bu fonksiyon bir Excel dosyası oluşturur ve her ders için bir sayfa ekler. ProgramOutcomes ve CourseOutcomes veri tabanından verileri çeker ve dersle ilişkilendirir. Bu verilerle program çıktıları ve ders çıktıları arasındaki ilişkilerle tabloyu doldurur. Ayrıca her ders için toplam ilişki değeri hesaplanır ve tabloya eklenir. Son olarak, tablo "table1.xlsx" olarak kaydedilir.
- **create_table2():** Bu fonksiyon bir Excel dosyası oluşturur ve her ders için bir sayfa ekler. Ders çıktıları ve değerlendirme kriterleri gibi verileri alır ve her ders için tabloyu doldurur. Değerlendirme ilişkilerini veri tabanından çeker ve dersle ilişkilendirir. Her dersin toplam değerlendirme değerini hesaplar ve tabloyu "table2.xlsx" olarak kaydeder.
- **show_excel(table, frame, lesson_name):** Bu fonksiyon belirtilen tabloyu (table1 veya table2) oluşturur ve ardından belirtilen derse ait verileri gösterir. İlgili Excel dosyasını okur, eksik verileri çıkarır ve verileri bir Treeview widget'ı içinde görüntüler. Verilere açıklama ekler ve hücrelerdeki yorumları uygun şekilde gösterir.
- **show_frame2(l_id, text):** Bu fonksiyon belirtilen dersle ilgili bir ekran açar. Bu ekran, dersle ilgili menüler ve seçenekler içerir. Menüdeki her bir buton, başka bir ekranı gösteren fonksiyonlara bağlanır.
- **show_other_frames(targetf, frame2, selected_id):** Bu fonksiyon belirli bir targetf değeri göre farklı Frame'ler oluşturur ve onları görüntüler. Hedef değerine göre belirli bir başlık, veri görüntüleme ve ekleme/silme işlemleri için gerekli widget'lar ekler.
 - **targetf == 3:** Dersin öğrenme çıktıları ekranını oluşturulur ve ilgili veri çekilip bir tabloya yerleştirilir. Ayrıca ders çıktısı eklemek ve silmek için butonlar eklenir.
 - **targetf == 4:** Program çıktıları işlemleri ekranı oluşturulur ve ilgili veri çekilip tabloya yerleştirilir. Program çıktısı ekleme ve silme butonları da eklenir.
 - **targetf == 5:** Ders ile program çıktıları arasındaki ilişkileri işleyen ekran oluşturulur. Ders ID'si ve Program Çıktısı ID'si gibi veriler girilerek bir ilişki oluşturulur.
 - **targetf == 6:** Öğrenci listesi ekranı gösterilir. Bu ekran Students tablosundaki tüm verileri görüntüler.
 - **targetf == 7:** Değerlendirme kriterleri ve ağırlıkları giriş ekranı oluşturulur. Bu ekran kullanıcının kriter adı ve ağırlıklarını girmesine olanak tanır. Veriler veri tabanına kaydedilir.
 - **targetf == 8:** Öğrenci not giriş ekranı oluşturulur. Bu ekran öğrenci numarasına göre değerlendirme kriterlerine dayalı not girmeye olanak sağlar.
 - **targetf == 9:** "Ders Çıktısı ve Değerlendirme Kriteri Giriş Ekranı" ekranın açılmasını sağlar. Bu ekran ders ve değerlendirme kriteri ilişkilerini girmek için gerekli alanları içerir.

4.2 Görev Dağılımı

Tüm aşamalar birlikte gerçekleştirilmiştir.

4.3 Karşılaşılan Zorluklar ve Çözüm Yöntemleri

Geliştirme sürecinde herhangi bir problemle karşılaşılmamıştır.

4.4 Proje İsterlerine Göre Eksik Yönler

Projede gerçekleşmesi beklenen görevlerin tümü tamamlanmıştır.

5. TEST VE DOĞRULAMA

5.1 Yazılımın Test Süreci

Bu birim testi, veri tabanı işlemleriyle ilgili çeşitli fonksiyonların doğru çalışıp çalışmadığını test etmek amacıyla yazılmıştır. Testler, veri tabanı bağlantısı oluşturma, veri tabanı ve tablo kontrolü yapma, veri ekleme ve silme işlemlerini kapsar. Her bir test, ilgili fonksiyonun doğru SQL sorgularını çalıştırıp çalıştırmadığını ve beklenen işlemleri gerçekleştirip gerçekleştirmediğini kontrol eder.

```
import unittest
from unittest.mock import patch, MagicMock
import pyodbc

from main import (
    create_connection,
    check_database,
    check_tables,
    insert_data_into_table,
    delete_data_from_table_by_id
)

class TestDatabaseFunctions(unittest.TestCase):
```

test_create_connection testi, create_connection fonksiyonunun doğru şekilde çalışıp çalışmadığını kontrol eder. pyodbc.connect fonksiyonu, bağlantı kurma işlemini simüle etmek için MagicMock() ile değiştirilir. Test, bağlantının oluşturulup oluşturulmadığını ve MagicMock nesnesi olarak döndürüldüğünü doğrular.

```
@patch('main.pyodbc.connect')
def test_create_connection(self, mock_connect):
    """create_connection fonksiyonunun doğru şekilde pyodbc bağlantısı oluşturup oluşturmadığını test eder."""
    mock_connect.return_value = MagicMock()
    conn = create_connection()
    self.assertTrue(mock_connect.called)
    self.assertIsInstance(conn, MagicMock)
```

test_check_database testi, `check_database` fonksiyonunun, veri tabanı varlığını kontrol eden doğru SQL sorgularını çalıştırıp çalıştırmadığını test eder. Bu fonksiyon, veri tabanının mevcut olup olmadığını kontrol eder ve yoksa yeni bir veri tabanı oluşturur. Test, `cursor().execute()` fonksiyonunun doğru SQL komutlarını çalıştırdığını doğrular.

```
@patch('main.get_connection')
def test_check_database(self, mock_get_connection):
    """check_database fonksiyonunun doğru sorgular çalıştırdığını test eder."""
    mock_conn = MagicMock()
    mock_get_connection.return_value = mock_conn

    check_database()

    mock_conn.cursor().execute.assert_any_call(
        "SELECT database_id FROM sys.databases WHERE Name = 'RelationMatrix'"
    )
    mock_conn.cursor().execute.assert_any_call("CREATE DATABASE RelationMatrix")
```

test_check_tables testi `check_tables` fonksiyonunun doğru şekilde tablo kontrolü yapip yapmadığını test eder. Bu fonksiyon, tabloyu kontrol eder ve eksikse tabloyu oluşturur. Test, `cursor().execute()` fonksiyonunun doğru SQL komutlarını çalıştırdığını ve tabloyu oluşturduğunu doğrular.

```
@patch('main.get_connection')
def test_check_tables(self, mock_get_connection):
    """check_tables fonksiyonunun doğru şekilde tablo kontrollerini yaptığını test eder."""
    mock_conn = MagicMock()
    mock_get_connection.return_value = mock_conn

    check_tables()

    mock_conn.cursor().execute.assert_any_call(
        """
        IF NOT EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'Students')
        CREATE TABLE Students (
            Student INT,
            lesson_id INT,
            PRIMARY KEY (Student, lesson_id)
        );
        """
    )
```

test_insert_data_into_table testi, `insert_data_into_table` fonksiyonunun veriyi doğru şekilde tabloya eklediğini test eder. Bu test, belirli bir tabloya veri eklerken doğru SQL komutunun çalıştırıldığını doğrular. Test, `cursor().execute()` fonksiyonunun `INSERT INTO` komutuyla doğru veriyi eklediğini kontrol eder.

```
@patch('main.get_connection')
def test_insert_data_into_table(self, mock_get_connection):
    """insert_data_into_table fonksiyonunun doğru veriyi eklediğini test eder."""
    mock_conn = MagicMock()
    mock_get_connection.return_value = mock_conn

    insert_data_into_table('CourseOutcomes', 'Example Data', 1)

    mock_conn.cursor().execute.assert_called_with(
        "INSERT INTO CourseOutcomes (data, LessonID) VALUES (?, ?)", ('Example Data', 1)
    )
```

test_delete_data_from_table_by_id testi, `delete_data_from_table_by_id` fonksiyonunun veri tabanından doğru veriyi sildiğini kontrol eder. Verilen bir ID ile veri silme işlemi yapılırken DELETE FROM komutunun doğru şekilde çalışıp çalışmadığını kontrol eder.

```
@patch('main.get_connection')
def test_delete_data_from_table_by_id(self, mock_get_connection):
    """delete_data_from_table_by_id fonksiyonunun doğru veriyi sildiğini test eder."""
    mock_conn = MagicMock()
    mock_get_connection.return_value = mock_conn

    delete_data_from_table_by_id('CourseOutcomes', 1)

    mock_conn.cursor().execute.assert_called_with(
        "DELETE FROM CourseOutcomes WHERE id = ?", (1,)
    )
```

Bu arayüz testi, Tkinter uygulamasının doğru şekilde yapılandırıldığını kontrol etmeyi amaçlar. Testler, pencerenin başlığının ve boyutlarının doğru ayarlandığını doğrular. `test_window_title` fonksiyonu, pencerenin başlığının beklenen metni içerip içermediğini kontrol ederken, `test_window_size` fonksiyonu pencerenin boyut ve konum bilgilerinin doğru şekilde ayarlandığını test eder. Bu sayede arayüzün temel görsel özelliklerinin doğru bir şekilde yapılandırıldığı kontrol eder.

```
import unittest
from tkinter import Tk
from main import root

class TestGUI(unittest.TestCase):

    def test_window_title(self):
        """Tkinter pencere başlığını test eder."""
        self.assertEqual(root.title(), "KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ - Ders Verileri Giriş Ekranı")

    def test_window_size(self):
        """Tkinter pencere boyutunu test eder."""
        self.assertEqual(root.geometry(), "1500x790+0+0")

if __name__ == '__main__':
    unittest.main()
```

5.2 Yazılımın Doğrulanması

- **Fonksiyonel Doğrulama:** Testler, Tkinter arayüzünün belirlenen gereksinimlere uygun olarak yapılandırıldığını doğrular. Örneğin, pencere başlığının ve boyutlarının önceden belirlenen değerlerle uyumlu olup olmadığı kontrol edilir. Bu, yazılımın istenilen fonksiyonelliği sunduğunu garanti eder.
- **Hata Önleme:** Pencere başlığı veya boyutları yanlış ayarlandığında testler hatayı tespit ederek erken müdahaleye olanak tanır. Bu da kullanıcı deneyimini olumsuz etkileyebilecek sorunların önüne geçilmesini sağlar.
- **Belirlenmiş Gereksinimlerin Karşılanması:** Testlerde kullanılan başlık ve boyut değerleri, uygulamanın gereksinimlerinden alınır. Bu doğrultuda testler uygulamanın ilgili gereksinimlere uygunluğunu doğrular.

6. GİTHUB BAĞLANTILARI

<https://github.com/SemaSuYILMAZ>

<http://github.com/Zehrayardimci>

<https://github.com/senemadalan>

<https://github.com/sevvalozrn>

<https://github.com/HilallAydinn>