



T.C
KOCaeli SAęLIK VE TEKNOLOJİ ÜNİVERSİTESİ
MÜHENDİSLİK VE DOęA BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR/YAZILIM MÜHENDİSLİęİ

PROJE KONUSU:

Haftalık Ders Programı Oluşturma

ÖęRENCİ ADI:

Enes Küçük - <https://github.com/EnessKucukk>
İbrahim Bener Karaca - <https://github.com/BenerKaraca>
Nazlı Su Ketçi - <https://github.com/nzlktc02>
Amine Derin - <https://github.com/aminederin>

ÖęRENCİ NUMARASI:

220501017
220501019
220501004
220501007

DERS SORUMLUSU:
DR. ÖęR. ÜYESİ Elif Pınar Hacıbeyoęlu

TARİH:
22.03.2025

1. GİRİŞ

1.1 Projenin Amacı

Bu projenin temel amacı, Kocaeli Sağlık ve Teknoloji Üniversitesi mühendislik fakültesine ait haftalık ders programını otomatik olarak oluşturabilen, kullanıcı dostu ve esnek bir sistem geliştirmektir. Proje, yazılım mühendisliği ve bilgisayar mühendisliği bölümlerine ait derslerin belirlenen kısıtlar doğrultusunda programlanmasını sağlar. Aynı zamanda, derslerin öğretim üyelerinin uygunluk durumuna, derslik kapasitesine ve çakışma kurallarına göre optimize edilerek yerleştirilmesini hedeflemektedir.

1.2 Projede Gerçekleştirilmesi Beklenenler

- **Kullanıcı Yönetimi:** Öğrenci, öğretim üyesi ve yöneticilerin sisteme eklenmesi ve yönetilmesi.
- **Bölüm Yönetimi:** Yazılım Mühendisliği ve Bilgisayar Mühendisliği bölümlerine ait derslerin tanımlanması ve yönetilmesi.
- **Ders Yönetimi:**
 - Ders ekleme ve çıkarma modülünün oluşturulması.
 - Derslerin öğretim üyeleri ile ilişkilendirilmesi.
 - Haftalık ders saatlerinin belirlenmesi.
- **Derslik Yönetimi:**
 - Dersliklerin eklenmesi ve kapasite bazlı kontrolü.
 - Belirlenen derslerin uygun dersliklere atanması.
- **Ders Programı Oluşturma:**
 - Günlere ve saat aralıklarına uygun şekilde derslerin yerleştirilmesi.
 - Ortak öğrencisi olan derslerin çakışmasını önleme.
 - Online derslerin belirtilen saat aralıklarına yerleştirilmesi.
- **Excel Çıktısı:**
 - Ders programının belirlenen şablona uygun şekilde Excel formatında oluşturulması.
 - Derslerin çakışma olmadan planlanması.
- **Dinamik ve Düzenlenebilir Sistem:**
 - Kullanıcıların ders programlarını gerektiğinde düzenleyebilmesi.
 - Yeni ders, öğretim üyesi ve bölüm eklenebilmesi.
- **Veritabanı Kullanımı:**
 - PostgreSQL, MySQL veya SQLite gibi bir veritabanı yönetim sistemi ile verilerin saklanması.
 - Veritabanı tablolarının kısıtlara uygun tasarlanması.

1 GEREKSİNİM ANALİZİ

1.1 Arayüz gereksinimleri

1.1 Arayüz Gereksinimleri

Bu projede, kullanıcıların (öğrenciler, öğretim üyeleri ve yöneticiler) ders seçimi ve haftalık ders programı oluşturma işlemlerini kolayca gerçekleştirebilmeleri için kullanıcı dostu bir arayüz tasarlanmıştır. Arayüz gereksinimleri, kullanıcıların ihtiyaçlarına göre belirlenmiştir. Aşağıda, kullanıcı arayüzü ve donanım arayüzü gereksinimleri maddeler halinde açıklanmıştır:

1.1.1 Kullanıcı Arayüzü Gereksinimleri

1. Kullanıcı Dostu Tasarım:

- Arayüz, kullanıcıların kolayca anlayabileceği ve kullanabileceği şekilde tasarlanmıştır.
- Menüler, butonlar ve diğer arayüz öğeleri tutarlı bir şekilde yerleştirilmiştir.

2. Duyarlı Tasarım (Responsive Design):

- Arayüz, farklı cihazlarda (masaüstü, tablet, mobil) sorunsuz çalışabilmelidir.
- Ekran boyutuna göre otomatik olarak uyum sağlamalıdır.

3. Erişilebilirlik:

- Arayüz, engelli kullanıcılar için erişilebilir olmalıdır (örneğin, ekran okuyucu desteği, yüksek kontrast modu).
- Renkler ve yazı tipleri, görme engelli kullanıcılar için uygun olmalıdır.

4. Hızlı ve Etkili Navigasyon:

- Kullanıcılar, sistem içinde hızlı bir şekilde gezinebilmelidir.
- Ana menü ve alt menüler, kullanıcıların istedikleri bilgilere kolayca ulaşmalarını sağlamalıdır.

5. Bildirim ve Geri Bildirim:

- Kullanıcılar, işlemlerinin sonucu hakkında anında geri bildirim almalıdır (örneğin, başarılı kayıt, hata mesajları).
- Hata durumlarında, kullanıcıya anlaşılır ve çözüm odaklı mesajlar gösterilmelidir.

6. Çoklu Dil Desteği:

- Arayüz, birden fazla dil seçeneği sunmalıdır.
- Kullanıcılar, tercih ettikleri dili seçebilmelidir.

7. Giriş ve Kimlik Doğrulama:

- Kullanıcılar, güvenli bir şekilde sisteme giriş yapabilmelidir.
- Şifre sıfırlama ve hesap kurtarma seçenekleri sunulmalıdır.

8. Veri Girişi ve Doğrulama:

- Kullanıcılar, formları doldururken hata yapmamaları için doğrulama mesajları almalıdır.
 - Zorunlu alanlar belirtilmeli ve eksik bilgi girişi engellenmelidir.
- 9. Ders Programı Görüntüleme ve Düzenleme:**
- Kullanıcılar, haftalık ders programını görüntüleyebilmeli ve gerekirse düzenleyebilmelidir.
 - Ders programı, Excel formatında indirilebilir olmalıdır.
- 10.Öğretim Üyesi ve Öğrenci Yönetimi:**
- Yöneticiler, öğretim üyeleri ve öğrencileri ekleyebilmeli, çıkarabilmeli ve güncelleyebilmelidir.
 - Öğretim üyeleri, kendi derslerini ve programlarını yönetebilmelidir.
- 11.Derslik Yönetimi:**
- Yöneticiler, derslikleri ekleyebilmeli, çıkarabilmeli ve güncelleyebilmelidir.
 - Derslik kapasiteleri ve statüleri (normal, laboratuvar) belirtilmelidir.
- 12.Çakışma Kontrolü:**
- Sistem, derslerin çakışmasını otomatik olarak kontrol etmeli ve kullanıcıya uyarı vermelidir.
 - Özellikle ortak dersler için çakışmaların önlenmesi zorunludur.

1.1.2 Donanım Arayüzü Gereksinimleri

- 1. Veritabanı Bağlantısı:**
 - Sistem, PostgreSQL, MySQL veya benzeri bir veritabanı yönetim sistemi ile entegre çalışabilmelidir.
 - Veritabanı bağlantısı, güvenli ve hızlı olmalıdır.
- 2. Excel Entegrasyonu:**
 - Sistem, Excel dosyaları ile entegre çalışabilmelidir.
 - Ders programı, Excel formatında oluşturulabilmeli ve indirilebilmelidir.
- 3. Sunucu ve İstemci Uyumu:**
 - Sistem, sunucu ve istemci arasında sorunsuz bir şekilde veri alışverişi yapabilmelidir.
 - Sunucu tarafında yüksek performans sağlanmalıdır.
- 4. Güvenlik ve Yetkilendirme:**
 - Sistem, kullanıcıların yetkilerine göre erişim kontrolü sağlamalıdır.
 - Veritabanı ve sunucu güvenliği sağlanmalıdır.
- 5. Yedekleme ve Geri Yükleme:**
 - Sistem, veritabanı ve diğer kritik veriler için otomatik yedekleme özelliği sunmalıdır.

- Yedekler, gerektiğinde geri yüklenebilmelidir.

Kodun Arayüz Gereksinimlerine Uygunluğu

Verilen kod, yukarıdaki arayüz gereksinimlerini büyük ölçüde karşılamaktadır. Özellikle:

- **Kullanıcı Dostu Tasarım:** Kodda, kullanıcıların bölüm, dönem ve öğrenci sayısı gibi bilgileri kolayca girebileceği bir arayüz tasarlanmıştır.
- **Veri Girişi ve Doğrulama:** Kullanıcıların girdiği bilgiler (örneğin, öğrenci sayısı) doğrulanmaktadır.
- **Ders Programı Görüntüleme ve Düzenleme:** Kullanıcılar, seçtikleri dersleri Excel formatında dışa aktarabilmektedir.
- **Çakışma Kontrolü:** Derslerin çakışmaması için uygun derslikler bulunmakta ve dersler buna göre yerleştirilmektedir.

Ancak, kodun eksik kaldığı bazı noktalar da bulunmaktadır:

- **Çoklu Dil Desteği:** Kodda çoklu dil desteği bulunmamaktadır.
- **Giriş ve Kimlik Doğrulama:** Kullanıcıların sisteme giriş yapması için bir kimlik doğrulama mekanizması bulunmamaktadır.
- **Erişilebilirlik:** Engelli kullanıcılar için özel erişilebilirlik özellikleri (ekran okuyucu desteği gibi) bulunmamaktadır.

1.2 Fonksiyonel gereksinimler

Fonksiyonel gereksinimler, sistemin ne yapması gerektiğini tanımlar. Bu gereksinimler, kullanıcıların sistemden beklediği temel işlevleri ve özellikleri içerir. Aşağıda, projenin fonksiyonel gereksinimleri maddeler halinde açıklanmıştır:

1.2.1 Kullanıcı Yönetimi

- 1. Öğrenci Yönetimi:**
 - Öğrencilerin sisteme kaydedilmesi, güncellenmesi ve silinmesi.
 - Öğrencilerin ders seçimi yapabilmesi ve seçilen derslerin kaydedilmesi.
- 2. Öğretim Üyesi Yönetimi:**
 - Öğretim üyelerinin sisteme kaydedilmesi, güncellenmesi ve silinmesi.
 - Öğretim üyelerinin ders atamalarının yapılabilmesi.
- 3. Yönetici Yönetimi:**
 - Yöneticilerin sisteme erişimi ve sistem ayarlarını yönetebilmesi.

- Yöneticilerin öğrenci, öğretim üyesi ve derslik bilgilerini düzenleyebilmesi.

1.2.2 Ders Yönetimi

1. Ders Ekleme ve Çıkarma:

- Yeni derslerin sisteme eklenebilmesi ve mevcut derslerin silinebilmesi.
- Derslerin ders kodu, adı, AKTS değeri, ders tipi (zorunlu/seçmeli), haftalık saat sayısı ve öğretim üyesi gibi bilgilerinin kaydedilmesi.

2. Ders Programı Oluşturma:

- Seçilen derslerin haftalık ders programına yerleştirilmesi.
- Derslerin çakışmaması için otomatik kontrol mekanizması.

3. Ortak Derslerin Yönetimi:

- İki bölümün ortak aldığı derslerin öncelikli olarak yerleştirilmesi.
- Ortak derslerin çakışmalarının önlenmesi.

1.2.3 Derslik Yönetimi

1. Derslik Ekleme ve Çıkarma:

- Yeni dersliklerin sisteme eklenebilmesi ve mevcut dersliklerin silinebilmesi.
- Dersliklerin kapasite ve statü (normal/laboratuvar) bilgilerinin kaydedilmesi.

2. Derslik Ataması:

- Derslerin uygun dersliklere atanması.
- Derslik kapasitesine uygun ders atamalarının yapılması.

1.2.4 Haftalık Ders Programı Oluşturma

1. Program Oluşturma:

- Seçilen derslerin haftalık ders programına yerleştirilmesi.
- Derslerin gün ve saatlere göre otomatik olarak yerleştirilmesi.

2. Çakışma Kontrolü:

- Aynı sınıfa ait derslerin çakışmaması için kontrol mekanizması.
- Öğretim üyelerinin aynı anda birden fazla ders vermesinin engellenmesi.

3. Excel Çıktısı:

- Oluşturulan ders programının Excel formatında dışa aktarılabilmesi.
- Excel çıktısının belirli bir şablona uygun olarak oluşturulması.

1.2.5 Raporlama ve Analiz

1. Rapor Oluşturma:

- Ders programı, öğrenci ve öğretim üyesi bilgilerini içeren raporların oluşturulabilmesi.

- Raporların Excel veya PDF formatında dışa aktarılabilmesi.
- 2. **Analiz ve İstatistik:**
 - Derslik doluluk oranları, öğretim üyesi yükleri gibi istatistiklerin görüntülenebilmesi.
 - Ders programı optimizasyonu için analiz araçlarının sunulması.

1.2.6 Sistem Yönetimi

1. **Veritabanı Yönetimi:**
 - Veritabanı bağlantılarının yönetilmesi.
 - Veritabanı yedekleme ve geri yükleme işlemlerinin yapılabilmesi.
2. **Güvenlik ve Yetkilendirme:**
 - Kullanıcıların yetkilerine göre erişim kontrolü sağlanması.
 - Veritabanı ve sunucu güvenliğinin sağlanması.
3. **Yedekleme ve Geri Yükleme:**
 - Sistem verilerinin otomatik olarak yedeklenmesi.
 - Yedeklerin gerektiğinde geri yüklenebilmesi.

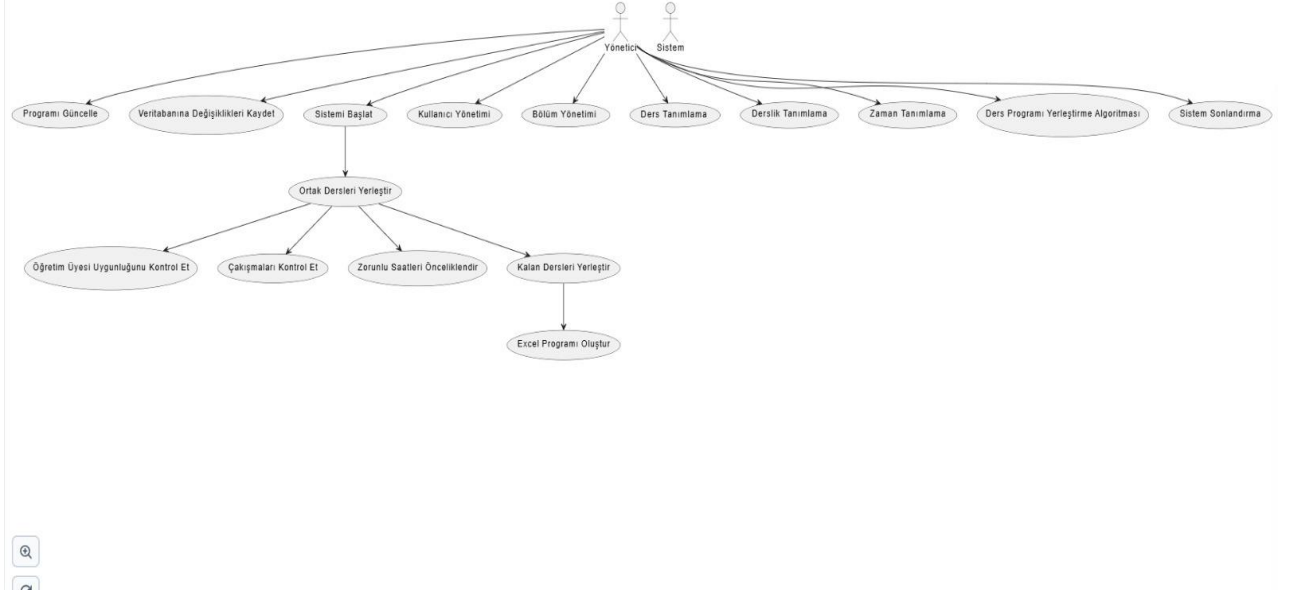
Kodun Fonksiyonel Gereksinimlere Uygunluğu

Verilen kod, yukarıdaki fonksiyonel gereksinimlerin bir kısmını karşılamaktadır. Özellikle:

- **Ders Yönetimi:** Kodda, derslerin seçilmesi ve haftalık ders programına yerleştirilmesi işlevi bulunmaktadır.
- **Derslik Yönetimi:** Kodda, dersliklerin kapasite ve statü bilgilerine göre ders atamaları yapılmaktadır.
- **Haftalık Ders Programı Oluşturma:** Kodda, derslerin Excel formatında dışa aktarılması işlevi bulunmaktadır.

Ancak, kodun eksik kaldığı bazı noktalar da bulunmaktadır:
- **Kullanıcı Yönetimi:** Kodda, öğrenci, öğretim üyesi ve yönetici yönetimi gibi işlevler bulunmamaktadır.
- **Raporlama ve Analiz:** Kodda, rapor oluşturma ve analiz işlevleri bulunmamaktadır.
- **Sistem Yönetimi:** Kodda, veritabanı yedekleme ve geri yükleme gibi işlevler bulunmamaktadır.

1.3 Use-Case diyagramı



2 TASARIM

2.1 Mimari tasarım

Mimari tasarım, sistemin genel yapısını, bileşenlerini ve bu bileşenlerin birbirleriyle olan ilişkilerini tanımlar. Bu tasarım, sistemin nasıl çalışacağını, hangi teknolojilerin kullanılacağını ve bileşenlerin nasıl organize edileceğini belirler. Aşağıda, projenin mimari tasarımı maddeler halinde açıklanmıştır:

1. Mimari Yaklaşım

1. Katmanlı Mimari:

- Sistem, katmanlı mimari yaklaşımına göre tasarlanmıştır. Bu yaklaşım, sistemin farklı katmanlara ayrılmasını ve her katmanın belirli bir işlevi yerine getirmesini sağlar.
- Katmanlar arasında net bir ayrım bulunur, bu da sistemin bakımını ve genişletilmesini kolaylaştırır.

2. MVC (Model-View-Controller) Modeli:

- Kullanıcı arayüzü (View), iş mantığı (Controller) ve veri erişim katmanı (Model) birbirinden ayrılmıştır.

- Bu model, kullanıcı arayüzü ile iş mantığının birbirinden bağımsız olarak geliştirilmesini sağlar.

2. Katmanlar

1. Kullanıcı Arayüzü Katmanı (UI Layer):

- Kullanıcıların sistemle etkileşimde bulunduğu arayüzü içerir.
- Tkinter kütüphanesi kullanılarak geliştirilmiştir.
- Kullanıcıların ders seçimi, ders programı oluşturma ve Excel çıktısı alma gibi işlemleri gerçekleştirebileceği bir arayüz sunar.

2. İş Mantığı Katmanı (Business Logic Layer):

- Sistemin temel iş mantığını içerir.
- Ders seçimi, derslik ataması, çakışma kontrolü gibi işlemler bu katmanda gerçekleştirilir.
- Kullanıcı arayüzü ile veritabanı katmanı arasında köprü görevi görür.

3. Veri Erişim Katmanı (Data Access Layer):

- Veritabanı ile etkileşimi sağlar.
- SQLite veritabanı kullanılarak ders, öğrenci, öğretim üyesi ve derslik bilgileri saklanır.
- Veritabanı sorguları ve işlemleri bu katmanda gerçekleştirilir.

3. Veritabanı Tasarımı

1. Veritabanı Yönetim Sistemi (DBMS):

- SQLite veritabanı kullanılmıştır.
- SQLite, hafif ve taşınabilir bir veritabanı yönetim sistemidir, bu da projenin hızlı ve kolay bir şekilde geliştirilmesini sağlar.

2. Tablo Yapısı:

- **Dersler Tablosu:** Ders kodu, ders adı, AKTS, ders tipi, saat, öğretim üyesi ve ders şekli gibi bilgileri içerir.
- **Öğrenciler Tablosu:** Öğrenci numarası, adı, sınıfı ve bölümü gibi bilgileri içerir.
- **Öğretim Üyeleri Tablosu:** Öğretim üyesi numarası ve adı gibi bilgileri içerir.
- **Derslikler Tablosu:** Derslik ID, kapasite ve statü gibi bilgileri içerir.

3. İlişkisel Veritabanı:

- Tablolar arasında ilişkiler kurularak veri bütünlüğü sağlanır.

- Örneğin, dersler tablosu ile öğretim üyeleri tablosu arasında bir ilişki kurularak, her dersin bir öğretim üyesine atanması sağlanır.

4. Teknolojiler ve Araçlar

1. Programlama Dili:

- Python programlama dili kullanılmıştır.
- Python, hızlı prototip geliştirme ve kolay okunabilirlik açısından tercih edilmiştir.

2. GUI Kütüphanesi:

- Tkinter kütüphanesi kullanılarak kullanıcı arayüzü geliştirilmiştir.
- Tkinter, Python ile birlikte gelen ve hızlı bir şekilde arayüz geliştirmeye olanak tanıyan bir kütüphanedir.

3. Veritabanı:

- SQLite veritabanı kullanılmıştır.
- SQLite, hafif ve taşınabilir bir veritabanı yönetim sistemidir.

4. Excel Entegrasyonu:

- OpenPyXL kütüphanesi kullanılarak Excel dosyaları oluşturulmuş ve düzenlenmiştir.
- OpenPyXL, Python ile Excel dosyaları üzerinde çalışmayı kolaylaştıran bir kütüphanedir.

5. Sistem Bileşenleri

1. Kullanıcı Arayüzü Bileşenleri:

- Bölüm seçimi, dönem seçimi, öğrenci sayısı girişi ve ders listesi gibi bileşenler bulunur.
- Kullanıcıların ders seçimi yapabileceği bir TreeView bileşeni kullanılmıştır.

2. İş Mantığı Bileşenleri:

- Ders seçimi, derslik ataması, çakışma kontrolü ve Excel çıktısı oluşturma gibi işlemleri gerçekleştiren bileşenler bulunur.

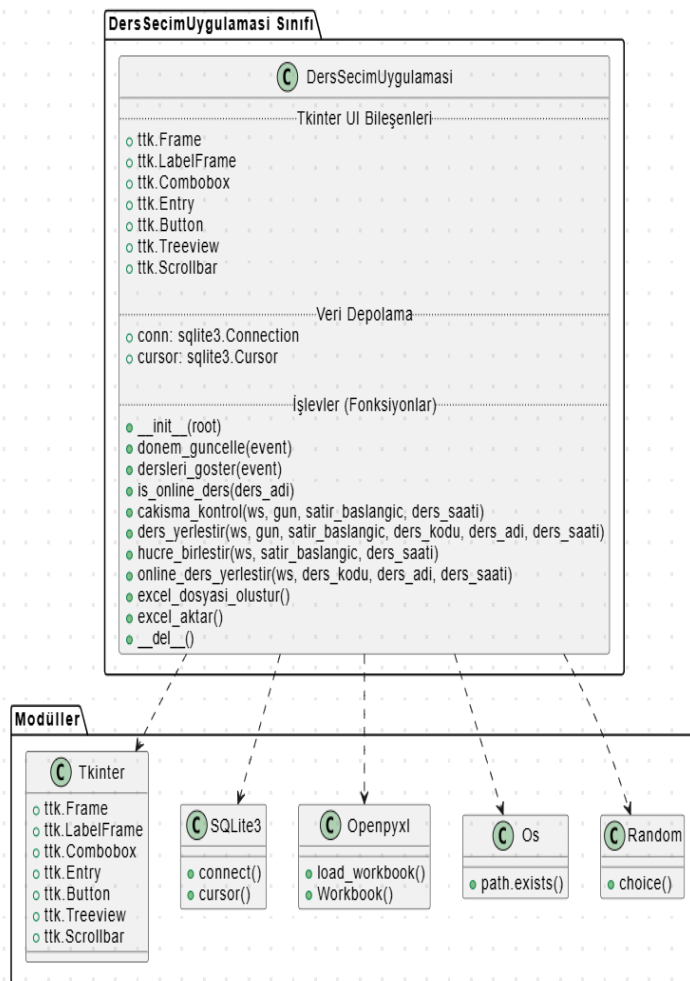
3. Veritabanı Bileşenleri:

- Veritabanı bağlantısı, sorgular ve veri işlemleri bu bileşenler tarafından gerçekleştirilir.

6. Sistem Akışı

1. Kullanıcı Girişi:

- Kullanıcılar, bölüm, dönem ve öğrenci sayısı gibi bilgileri girerek ders seçimi yapar.
2. **Ders Seçimi ve Atama:**
- Seçilen dersler, uygun dersliklere atanır ve çakışma kontrolü yapılır.
3. **Excel Çıktısı Oluşturma:**
- Oluşturulan ders programı, Excel formatında dışa aktarılır.
4. **Raporlama:**
- Ders programı ve diğer bilgiler, raporlar halinde sunulur.



3.2 Kullanılacak Teknolojiler

- **Yazılım Dili:**
 - Proje, Python programlama dili kullanılarak geliştirilecektir.
 - Python'un esnekliği, geniş kütüphane desteği ve veri işleme yetenekleri nedeniyle tercih edilmiştir.
- **Kullanılacak Harici Kütüphaneler:**
 - `tkinter`: Kullanıcı arayüzü (GUI) oluşturmak için kullanılacaktır.
 - `sqlite3`: Veritabanı işlemleri için SQLite kullanılacaktır.
 - `pandas`: Veri işleme ve analiz işlemleri için kullanılacaktır.
 - `openpyxl`: Excel dosyalarının okunması ve yazılması için kullanılacaktır.
 - `random`: Derslerin rastgele yerleştirilmesi ve çakışma önleme işlemleri için kullanılacaktır.
- **Diğer Teknolojiler:**
 - **Veritabanı:** SQLite kullanılarak ders bilgileri, öğretim üyeleri ve derslik bilgileri saklanacaktır.
 - **Excel Entegrasyonu:** Ders programı, belirlenen formatta bir Excel dosyası olarak oluşturulacaktır.
 - **Grafik Arayüz:** Tkinter ile kullanıcı dostu bir arayüz sağlanacaktır.
 - **Dinamik Yapı:** Sistem, yeni ders, öğretim üyesi veya derslik eklenmesine olanak tanıyacak şekilde tasarlanacaktır.

2.2 Veri tabanı tasarımı

Tablo: admin	
kullanici_adı	sifre
Filtre	Filtre
1	kostüadmin1 Kostü123
2	kostüadmin2 Kostü1234

Tablo: alan_secmeli_dersler

	ders_kodu	ders_adi	donem	akts	saat
	Filtre	Filtre	Filtre	Filtre	Filtre
1	S018	ALAN SEÇMELİ DERS I	5	4	3
2	S019	ALAN SEÇMELİ DERS II	6	4	2
3	S020	ALAN SEÇMELİ DERS III	7	4	3
4	S021	ALAN SEÇMELİ DERS IV	7	4	3
5	S022	ALAN SEÇMELİ DERS V	8	4	3
6	SMUH311	BİLİŞİM ETİŞİ VE HUKUKU	8	4	3
7	STJ002	STAJ II	6	5	NULL
8	S023	ALAN SEÇMELİ DERS VI	8	4	2

Database Structure Browse Data Edit Pragn

Tablo: derslikler

	derslik_id	kapasite	statü
	Filtre	Filtre	Filtre
1	M101	66	NORMAL
2	M201	141	NORMAL
3	M301	141	NORMAL
4	S101	138	NORMAL
5	S201	60	NORMAL
6	S202	60	NORMAL
7	D101	87	NORMAL
8	D102	87	NORMAL
9	D103	88	NORMAL
10	D104	56	NORMAL
11	D201	87	NORMAL
12	D202	56	NORMAL
13	D301	88	NORMAL
14	D302	56	NORMAL
15	D401	88	NORMAL
16	D402	56	NORMAL
17	D403	56	NORMAL
18	AMFİ A	143	NORMAL
19	AMFİ B	143	NORMAL
20	BİL.LAB.1	40	LAB
21	BİL.LAB.2	30	LAB
22	KÜÇÜK LAB	20	LAB

Database Structure Browse Data Edit Pragn Execute SQL

Tablo: ogrenciler

	no	adi	sinif	bolum	aldigi_dersler
	Filtre	Filtre	Filtre	Filtre	Filtre
1	0	Kostü	1	Bilgisayar	
2	1001	Ahmet Yılmaz	1	Bilgisayar Mühendisliği	
3	1002	Mehmet Demir	1	Yazılım Mühendisliği	
4	1003	Zeynep Kaya	1	Bilgisayar Mühendisliği	
5	2001	Elif Çelik	2	Yazılım Mühendisliği	
6	2002	Mert Özkan	2	Bilgisayar Mühendisliği	
7	2003	Deniz Aksoy	2	Yazılım Mühendisliği	
8	3001	Burak Arslan	3	Bilgisayar Mühendisliği	
9	3002	Hale Yıldırım	3	Yazılım Mühendisliği	
10	3003	Berkay Şahin	3	Bilgisayar Mühendisliği	
11	4001	Ece Doğan	4	Yazılım Mühendisliği	
12	4002	Tuna Güneş	4	Bilgisayar Mühendisliği	
13	4003	Gizem Sarı	4	Yazılım Mühendisliği	

Tablo: ortak_dersler

	ders_kodu	ders_adi	donem	akts	saat
	Filtre	Filtre	Filtre	Filtre	Filtre
1	MAT110	MATEMATİK I	1	5	5
2	DİL100	İNGİLİZCE	1	4	2
3	FİZ120	FİZİK II	2	5	4
4	MAT120	MATEMATİK II	2	5	5
5	ATA100	ATATÜRK İLKELERİ VE İNKILAP TARİHİ	2	4	2
6	TUR100	TÜRK DİLİ	2	4	2
7	MAT213	AYRIK MATEMATİK	3	4	3
8	MAT224	OLASILIK VE İSTATİSTİK	4	4	3
9	MAT220	SAYISAL YÖNTEMLER	5	4	3

Tablo: uni_secmeli_dersler					
	ders_kodu	ders_adi	donem	akts	saat
	Filtre	Filtre	Filtre	Filtre	Filtre
1	US001	ÜNİVERSİTE SEÇMELİ DERS I	1	3	2
2	US002	ÜNİVERSİTE SEÇMELİ DERS II	3	3	2
3	US003	ÜNİVERSİTE SEÇMELİ DERS III	3	3	2
4	US004	ÜNİVERSİTE SEÇMELİ DERS IV	4	3	2
5	US005	ÜNİVERSİTE SEÇMELİ DERS V	5	3	2
6	US006	SEÇMELİ DERS VI	6	3	2
7	US007	ÜNİVERSİTE SEÇMELİ DERS VII	7	3	2
8	US008	ÜNİVERSİTE SEÇMELİ DERS VIII	7	3	2
9	US009	ÜNİVERSİTE SEÇMELİ DERS IX	8	3	2
10	US010	ÜNİVERSİTE SEÇMELİ DERS X	8	3	2

Database Structure Browse Data Edit Pragma Execute SQL						
Tablo: yazilim_muh_dersler						
	ders_kodu	ders_adi	donem	akts	ders_tipi	saat
	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
1	MAT110	MATEMATİK I	1	5	Zorunlu	5
2	FIZ110	FİZİK I	1	5	Zorunlu	5
3	YZM113	BİLGİSAYAR PROGRAMLAMA-I	1	4	Zorunlu	3
4	YZM115	BİLGİSAYAR LAB-I	1	3	Zorunlu	2
5	MAT127	LİNEER CEBİR	1	3	Zorunlu	3
6	YZM119	YAZILIM MÜHENDİSLİĞİNE GİRİŞ	1	3	Zorunlu	2
7	DİL100	İNGİLİZCE	1	4	Zorunlu	2
8	US001	ÜNİVERSİTE SEÇMELİ DERS I	1	3	Seçmeli	2
9	YZM122	BİLGİSAYAR PROGRAMLAMA II	2	4	Zorunlu	3
10	FIZ120	FİZİK II	2	5	Zorunlu	4
11	MAT120	MATEMATİK II	2	5	Zorunlu	5
12	YZM117	BİLGİSAYAR LAB-II	2	3	Zorunlu	2
13	YZM128	WEB TEKNOLOJİLERİ	2	5	Zorunlu	2
14	ATA100	ATATÜRK İLKELERİ VE İNKILAP TARİHİ	2	4	Zorunlu	2
15	TUR100	TÜRK DİLİ	2	4	Zorunlu	2
16	MAT213	AYRIK MATEMATİK	3	4	Zorunlu	3
17	YZM219	YAZILIM GEREKSİNİM ANALİZİ	3	5	Zorunlu	3
18	YZM224	VERİTABANI YÖNETİM SİSTEMLERİ	3	5	Zorunlu	3
19	YZM229	NESNEYE YÖNELİK PROGRAMLAMA	3	5	Zorunlu	4
20	YZM127	PROGRAMLAMA LAB- I	3	5	Zorunlu	2
21	US002	ÜNİVERSİTE SEÇMELİ DERS II	3	3	Seçmeli	2
22	US003	ÜNİVERSİTE SEÇMELİ DERS III	3	3	Seçmeli	2
23	YZM226	YAZILIM TASARIMI	4	4	Zorunlu	2
24	YZM228	PROGRAMLAMA LAB-II	4	5	Zorunlu	3
25	YZM317	BİLGİSAYAR AĞLARI	4	5	Zorunlu	3
26	MAT224	OLASILIK VE İSTATİSTİK	4	4	Zorunlu	3
27	YZM213	VERİ YAPILARI	4	5	Zorunlu	4
28	YZM21001	STATİSTİK I	4	4	Zorunlu	4

Tablo: hocalar			
	hoca_no	hoca_adi	aldigi_dersler
	Filtre	Filtre	Filtre
1	001	Prof. Dr. Nevcihan DURU	NULL
2	002	Dr. Öğr. Üyesi Vildan YAZICI	NULL
3	003	Prof. Dr. Hüseyin Tarık DURU	NULL
4	004	Dr. Öğr. Üyesi Mehmet KARA	NULL
5	005	Dr. Öğr. Üyesi Ulaş VURAL	NULL
6	006	Dr. Öğr. Üyesi İsmet KARADUMAN	NULL
7	007	Dr. Öğr. Üyesi Ercan ÖLÇER	NULL
8	008	Dr. Öğr. Üyesi Elif Pınar HACIBEYOĞLU	NULL
9	009	Dr. Öğr. Üyesi Fulya AKDENİZ	NULL
10	010	Arş. Gör. Eray DURSUN	NULL
11	011	Arş. Gör. Candide ÖZTÜRK	NULL

Projenin veritabanı tasarımı, üniversite ders seçim ve program oluşturma sisteminin ihtiyaçlarını karşılamak üzere oluşturulmuştur. Veritabanı, SQLite kullanılarak tasarlanmış ve aşağıdaki tablolardan oluşmaktadır. Her tablo, belirli bir varlığı (örneğin, dersler, öğrenciler, hocalar) temsil eder ve bu varlıklar arasındaki ilişkileri yönetir.

2.2.1 Tablolar ve Açıklamaları

1. yazilim_muh_dersler:

- **Açıklama:** Yazılım Mühendisliği bölümüne ait derslerin bilgilerini içerir.
- **Sütunlar:**
 - ders_kodu (TEXT, PRIMARY KEY): Dersin benzersiz kodu.
 - ders_adi (TEXT): Dersin adı.
 - donem (INTEGER): Dersin hangi dönemde verildiği.
 - akts (INTEGER): Dersin AKTS (Avrupa Kredi Transfer Sistemi) değeri.
 - ders_tipi (TEXT): Dersin tipi (Zorunlu/Seçmeli).
 - saat (TEXT): Dersin haftalık saat bilgisi.
 - ders_hocasi (TEXT): Dersi veren öğretim üyesi.

- ders_sekli (TEXT): Dersin öğrenim şekli (Örgün/Uzaktan).

2. bilgisayar_muh_dersler:

- **Açıklama:** Bilgisayar Mühendisliği bölümüne ait derslerin bilgilerini içerir.
- **Sütunlar:**
 - ders_kodu (TEXT, PRIMARY KEY): Dersin benzersiz kodu.
 - ders_adi (TEXT): Dersin adı.
 - donem (INTEGER): Dersin hangi dönemde verildiği.
 - akts (INTEGER): Dersin AKTS değeri.
 - ders_tipi (TEXT): Dersin tipi (Zorunlu/Seçmeli).
 - saat (TEXT): Dersin haftalık saat bilgisi.
 - ders_hocasi (TEXT): Dersi veren öğretim üyesi.
 - ders_sekli (TEXT): Dersin öğrenim şekli (Örgün/Uzaktan).

3. ogrenciler:

- **Açıklama:** Öğrencilerin bilgilerini içerir.
- **Sütunlar:**
 - no (INTEGER, PRIMARY KEY): Öğrencinin numarası.
 - adi (TEXT): Öğrencinin adı.
 - sınıf (INTEGER): Öğrencinin sınıfı.
 - bolum (TEXT): Öğrencinin bölümü.
 - aldig_i_dersler (TEXT): Öğrencinin aldığı dersler (virgülle ayrılmış ders kodları).

4. hocalar:

- **Açıklama:** Dersleri veren öğretim üyelerinin bilgilerini içerir.
- **Sütunlar:**
 - hoca_no (TEXT, PRIMARY KEY): Öğretim üyesinin numarası.
 - hoca_adi (TEXT): Öğretim üyesinin adı.

5. admin:

- **Açıklama:** Admin kullanıcılarının bilgilerini içerir.
- **Sütunlar:**
 - kullanıcı_adi (TEXT, PRIMARY KEY): Admin kullanıcı adı.
 - sifre (TEXT): Admin şifresi.

6. derslikler:

- **Açıklama:** Dersliklerin bilgilerini içerir.
- **Sütunlar:**
 - derslik_id (TEXT, PRIMARY KEY): Dersliğin benzersiz kimliği.
 - kapasite (INTEGER): Dersliğin kapasitesi.
 - statü (TEXT): Dersliğin durumu (Normal/Lab).

7. ortak_dersler:

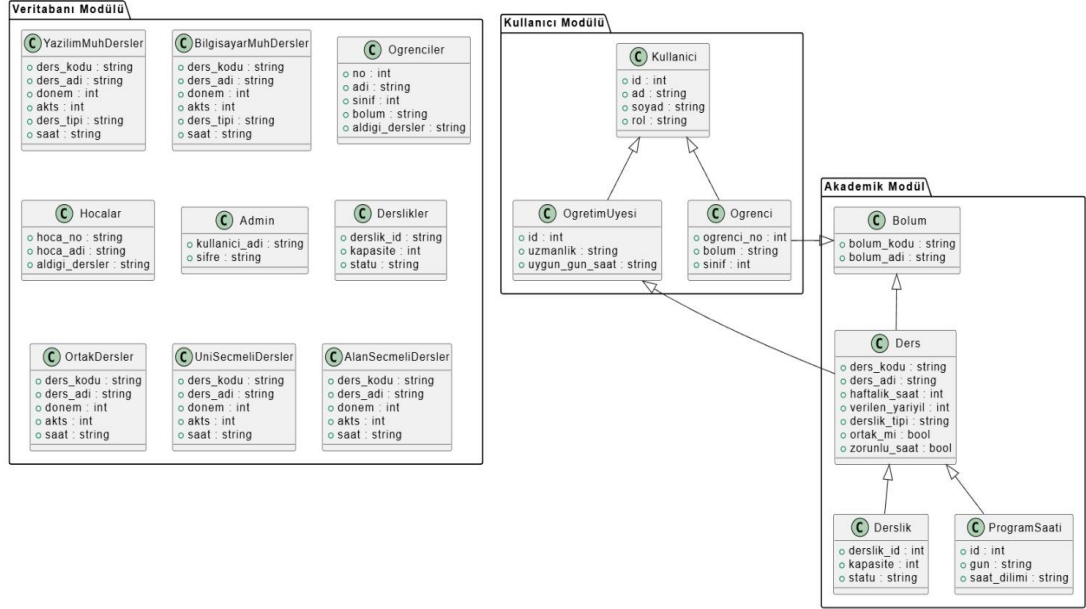
- **Açıklama:** Yazılım ve Bilgisayar Mühendisliği bölümlerinin ortak derslerini içerir.
 - **Sütunlar:**
 - ders_kodu (TEXT, PRIMARY KEY): Dersin benzersiz kodu.
 - ders_adi (TEXT): Dersin adı.
 - donem (INTEGER): Dersin hangi dönemde verildiği.
 - akts (INTEGER): Dersin AKTS değeri.
 - saat (TEXT): Dersin haftalık saat bilgisi.
8. **uni_secmeli_dersler:**
- **Açıklama:** Üniversite seçmeli derslerini içerir.
 - **Sütunlar:**
 - ders_kodu (TEXT, PRIMARY KEY): Dersin benzersiz kodu.
 - ders_adi (TEXT): Dersin adı.
 - donem (INTEGER): Dersin hangi dönemde verildiği.
 - akts (INTEGER): Dersin AKTS değeri.
 - saat (TEXT): Dersin haftalık saat bilgisi.
9. **alan_secmeli_dersler:**
- **Açıklama:** Alan seçmeli derslerini içerir.
 - **Sütunlar:**
 - ders_kodu (TEXT, PRIMARY KEY): Dersin benzersiz kodu.
 - ders_adi (TEXT): Dersin adı.
 - donem (INTEGER): Dersin hangi dönemde verildiği.
 - akts (INTEGER): Dersin AKTS değeri.
 - saat (TEXT): Dersin haftalık saat bilgisi.

2.2.2 Tablo İlişkileri

- **yazilim_muh_dersler** ve **bilgisayar_muh_dersler** tabloları, her iki bölümün derslerini ayrı ayrı tutar. Bu tablolar, ders_kodu üzerinden **ortak_dersler** tablosu ile ilişkilidir.
- **ogrenciler** tablosu, öğrencilerin aldığı dersleri aldığı_dersler sütunu üzerinden tutar. Bu sütun, yazilim_muh_dersler ve bilgisayar_muh_dersler tablolarındaki ders_kodu ile ilişkilidir.
- **hocalar** tablosu, dersleri veren öğretim üyelerini tutar. Bu tablo, yazilim_muh_dersler ve bilgisayar_muh_dersler tablolarındaki ders_hocasi sütunu ile ilişkilidir.
- **derslikler** tablosu, dersliklerin kapasite ve durum bilgilerini tutar. Bu tablo, ders programı oluşturulurken kullanılır.

2.2.3 Veritabanı Şeması

Aşağıda, veritabanı tabloları arasındaki ilişkileri gösteren basit bir şema bulunmaktadır:



2.3 Kullanıcı arayüzü tasarımı

Kullanıcı arayüzü (UI), projenin kullanıcılar (admin) ile etkileşim kurduğu temel bileşenidir. Tkinter kütüphanesi kullanılarak geliştirilen arayüz, basit ve kullanıcı dostu bir yapıya sahiptir. Arayüz, adminlerin ders seçimi yapmasına, öğrenci sayısına göre derslik atamasına ve ders programını Excel'e aktarmasına olanak tanır.

2.3.1 Kullanıcı Arayüzü Bileşenleri ve Açıklamaları

1. Login Penceresi:

- **Açıklama:** Admin girişi için kullanılan pencere. Kullanıcı adı ve şifre alanları bulunur.
- **Bileşenler:**
 - **Kullanıcı Adı:** Admin kullanıcı adının girildiği metin kutusu.
 - **Şifre:** Admin şifresinin girildiği metin kutusu (şifre gizlenir).
 - **Giriş Yap Butonu:** Kullanıcı adı ve şifre doğrulandıktan sonra ana uygulama penceresini açar.

2. Ana Uygulama Penceresi:

Kullanıcı Adı:

Şifre:

Giriş Yap

- **Açıklama:** Ders seçimi ve program oluşturma işlemlerinin yapıldığı ana pencere.
- **Bileşenler:**
 - **Bölüm Seçimi:** Yazılım Mühendisliği veya Bilgisayar Mühendisliği bölümlerinden birini seçmek için açılır liste (combobox).
 - **Dönem Seçimi:** 1 ile 8 arasında dönem seçimi yapmak için açılır liste (combobox).
 - **Öğrenci Sayısı:** Derslik ataması yapmak için öğrenci sayısının girildiği metin kutusu.
 - **Ders Listesi:** Seçilen bölüm ve döneme göre derslerin listelendiği Treeview tablosu. Her ders için ders kodu, adı, AKTS, tipi, saat bilgisi, öğretim üyesi ve öğrenim şekli gösterilir.
 - **Excel'e Aktar Butonu:** Seçilen dersleri Excel dosyasına aktarır ve ders programını oluşturur.

3. Treeview Tablosu:

- **Açıklama:** Derslerin listelendiği tablo. Kullanıcı, bu tablodan ders seçimi yapabilir.
- **Sütunlar:**
 - **Ders Kodu:** Dersin benzersiz kodu.

- **Ders Adı:** Dersin adı.
- **AKTS:** Dersin AKTS değeri.
- **Tip:** Dersin tipi (Zorunlu/Seçmeli).
- **Saat:** Dersin haftalık saat bilgisi.
- **Öğretim Üyesi:** Dersi veren öğretim üyesi.
- **Öğrenim:** Dersin öğrenim şekli (Örgün/Uzaktan).

4. Excel Dosyası Çıktısı:

- **Açıklama:** Seçilen dersler, Excel dosyasına aktarılır ve haftalık ders programı oluşturulur. Excel dosyası, haftanın günlerine (Pazartesi, Salı, Çarşamba, Perşembe, Cuma) ve saatlerine göre düzenlenmiştir.
- **Özellikler:**
 - Her ders, belirli bir gün ve saat aralığına yerleştirilir.
 - Uzaktan eğitim dersleri için özel bir zamanlama yapılır.
 - Derslik bilgisi (örneğin, "Derslik: M101") Excel dosyasına eklenir.

2.3.2 Uygulamanın Nasıl Çalıştırılacağı

1. Adım 1: Admin Girişi

- Uygulama başlatıldığında, admin giriş penceresi açılır.
- Admin, kullanıcı adı ve şifresini girerek "Giriş Yap" butonuna tıklar.
- Giriş başarılı olduğunda, ana uygulama penceresi açılır.

2. Adım 2: Bölüm ve Dönem Seçimi

- Ana uygulama penceresinde, admin önce bölümü (Yazılım Mühendisliği veya Bilgisayar Mühendisliği) seçer.
- Ardından, dönemi (1 ile 8 arasında) seçer.

3. Adım 3: Ders Listesini Görüntüleme

- Seçilen bölüm ve döneme göre dersler, Treeview tablosunda listelenir.
- Admin, bu tablodan ders seçimi yapabilir.

4. Adım 4: Öğrenci Sayısı Girme ve Derslik Atama

- Admin, öğrenci sayısını ilgili metin kutusuna girer.
- Sistem, öğrenci sayısına göre uygun dersliği otomatik olarak belirler.

5. Adım 5: Excel'e Aktarım

- Admin, "Excel'e Aktar" butonuna tıklar.
- Seçilen dersler, Excel dosyasına aktarılır ve haftalık ders programı oluşturulur.
- Excel dosyası, projenin çalıştığı dizinde kaydedilir.

6. Adım 6: Sonuç

- Admin, oluşturulan Excel dosyasını açarak ders programını görüntüleyebilir.

2.3.3 Ekran Çıktıları ve Açıklamaları

1. Login Penceresi:

- Kullanıcı adı ve şifre girilerek giriş yapılır.
- Örnek: kullanıcı_adi: kostüadmin1, sifre: Kostü123.

2. Ana Uygulama Penceresi:

- Bölüm ve dönem seçimi yapıldıktan sonra dersler listelenir.
- Öğrenci sayısı girilir ve uygun derslik belirlenir.

3. Excel Çıktısı:

- Seçilen dersler, Excel dosyasına aktarılır ve haftalık program oluşturulur.
- Örnek: Pazartesi 09:00 - 11:00 arasında "MAT110 - MATEMATİK I" dersi, Derslik: M101.

3 UYGULAMA

3.1 Kodlanan bileşenlerin açıklamaları

2.4 Bileşenlerin Açıklanması

Proje, Python programlama dili kullanılarak geliştirilmiştir ve temel olarak **fonksiyonlar**, **sınıflar** ve **modüller** gibi bileşenlerden oluşmaktadır. Bu bileşenler, projenin farklı işlevlerini yerine getirir ve birbirleriyle etkileşim halindedir. Aşağıda, projenin temel bileşenleri ve bu bileşenlerin işlevleri detaylı bir şekilde açıklanmıştır.

```

import tkinter as tk
from tkinter import ttk, messagebox
import sqlite3
from openpyxl import load_workbook
import random
from openpyxl.styles import Alignment
import os

class LoginWindow:
    def __init__(self, root):
        self.root = root
        self.root.title("Admin Girişi")
        self.root.geometry("300x200")

        # Veritabanı bağlantısı
        self.conn = sqlite3.connect('universite.db')
        self.cursor = self.conn.cursor()

        # GUI bileşenlerini oluştur
        self.init_ui()

    def init_ui(self):
        # Ana Frame
        self.main_frame = ttk.Frame(self.root, padding=20)
        self.main_frame.pack(fill=tk.BOTH, expand=True)

        # Kullanıcı Adı
        ttk.Label(self.main_frame, text="Kullanıcı Adı:").pack(pady=5)

```

2.4.1 Sınıflar (Classes)

Projede iki ana sınıf bulunmaktadır: LoginWindow ve DersSecimUygulaması. Bu sınıflar, uygulamanın farklı pencerelerini ve işlevlerini yönetir.

1. LoginWindow Sınıfı:

- **Açıklama:** Admin giriş penceresini yöneten sınıf. Kullanıcı adı ve şifre doğrulaması yapar.
- **Özellikler:**
 - `__init__`: Sınıfın yapıcı metodudur. Pencereyi başlatır ve GUI bileşenlerini oluşturur.
 - `init_ui`: Kullanıcı arayüzünü oluşturan metod. Kullanıcı adı, şifre alanları ve giriş butonunu içerir.
 - `login`: Kullanıcı adı ve şifreyi kontrol eder. Doğruysa ana uygulama penceresini açar.
- **Kullanılan Modüller:**
 - `tkinter`: GUI bileşenleri için kullanılır.
 - `sqlite3`: Veritabanı bağlantısı için kullanılır.

DersSecimUygulaması Sınıfı:

```

class DersSecimUygulaması:
    def __init__(self, root):
        self.root = root
        self.root.title("Ders Seçim ve Program Oluşturma Sistemi")
        self.root.geometry("1200x800")

        # Veritabanı bağlantıları
        self.conn = sqlite3.connect('universite.db')
        self.cursor = self.conn.cursor()

        # Derslik verilerini yükle
        self.derslikler = self.derslikleri_yukle()

        # GUI bileşenlerini oluştur
        self.init_ui()

    def init_ui(self):
        # Ana Frame
        self.main_frame = ttk.Frame(self.root, padding=20)
        self.main_frame.pack(fill=tk.BOTH, expand=True)

        # Bölüm Seçimi
        ttk.Label(self.main_frame, text="Bölüm:").grid(row=0, column=0, sticky=tk.W)
        self.bolum_var = tk.StringVar()
        self.bolum_combo = ttk.Combobox(self.main_frame, textvariable=self.bolum_var,
                                         values=['Yazılım Mühendisliği', 'Bilgisayar Mühendisliği'])
        self.bolum_combo.grid(row=0, column=1, sticky=tk.EW)

```

```

        # Dönem Seçimi
        ttk.Label(self.main_frame, text="Dönem:").grid(row=1, column=0, sticky=tk.W)
        self.donem_var = tk.StringVar()
        self.donem_combo = ttk.Combobox(self.main_frame, textvariable=self.donem_var)
        self.donem_combo.grid(row=1, column=1, sticky=tk.EW)
        self.bolum_combo.bind('<<ComboboxSelected>>', self.donemleri_guncelle)

        # Öğrenci Sayısı
        ttk.Label(self.main_frame, text="Öğrenci Sayısı:").grid(row=2, column=0, sticky=tk.W)
        self.ogrenci_sayisi = ttk.Entry(self.main_frame)
        self.ogrenci_sayisi.grid(row=2, column=1, sticky=tk.EW)

        # Ders Listesi
        self.tree = ttk.Treeview(self.main_frame, columns=('Kod', 'Ad', 'AKTS', 'Tip', 'Saat', 'Hoca', 'ders_sekli'), show='headings')
        self.tree.heading('Kod', text='Ders Kodu')
        self.tree.heading('Ad', text='Ders Adı')
        self.tree.heading('AKTS', text='AKTS')
        self.tree.heading('Tip', text='Tip')
        self.tree.heading('Saat', text='Saat')
        self.tree.heading('Hoca', text='Öğretim Üyesi')
        self.tree.heading('ders_sekli', text='Öğrenim')
        self.tree.grid(row=3, column=0, columnspan=2, pady=10, sticky=tk.NSEW)

        # Excel'e Aktar Butonu
        ttk.Button(self.main_frame, text="Excel'e Aktar", command=self.excel_aktar).grid(row=4, column=0, columnspan=2)

        # Grid konfigürasyonu
        self.main_frame.columnconfigure(1, weight=1)

```

- **Açıklama:** Ana uygulama penceresini yöneten sınıf. Ders seçimi, derslik atama ve Excel'e aktarım işlemlerini gerçekleştirir.
- **Özellikler:**
 - `__init__`: Sınıfın yapıcı metodudur. Pencereyi başlatır ve GUI bileşenlerini oluşturur.

- `init_ui`: Kullanıcı arayüzünü oluşturan metod. Bölüm, dönem seçimi, öğrenci sayısı girişi ve ders listesi gibi bileşenleri içerir.
- `derslikleri_yukle`: Derslik bilgilerini veritabanından yükler.
- `uygun_derslik_bul`: Öğrenci sayısına göre uygun dersliği bulur.
- `donemleri_guncelle`: Seçilen bölüme göre dönemleri günceller.
- `dersleri_goster`: Seçilen bölüm ve döneme göre dersleri listeler.
- `excel_aktar`: Seçilen dersleri Excel dosyasına aktarır.
- `sınıf_sec`: Döneme göre sınıf seçimi yapar.

```
def derslikleri_yukle(self):
    try:
        self.cursor.execute("SELECT derslik_id, kapasite, statü FROM derslikler")
        return self.cursor.fetchall()
    except sqlite3.OperationalError as e:
        messagebox.showerror("Veritabanı Hatası", f"Tablo veya sütun bulunamadı: {str(e)}")
        return []

def uygun_derslik_bul(self, ogrenci_sayisi):
    for derslik in sorted(self.derslikler, key=lambda x: x[1]):
        if derslik[1] >= ogrenci_sayisi:
            return derslik
    return None

def donemleri_guncelle(self, event):
    self.donem_combo['values'] = list(range(1, 9))
    self.donem_combo.bind('<<ComboboxSelected>>', self.dersleri_goster)

def dersleri_goster(self, event):
    self.tree.delete(*self.tree.get_children())
    bolum = self.bolum_var.get()
    donem = self.donem_var.get()

    # Bölüme özel dersleri yükle
    tablo = 'yazilim_muh_dersler' if bolum == 'Yazılım Mühendisliği' else 'bilgisayar_muh_dersler'
    self.cursor.execute(f'''
        SELECT ders_kodu, ders_adi, akts, ders_tipi, saat, ders_hocasi, ders_sekli
        FROM {tablo}
    ''')
```

• Kullanılan Modüller:

- `tkinter`: GUI bileşenleri için kullanılır.
- `sqlite3`: Veritabanı bağlantısı için kullanılır.
- `openpyxl`: Excel dosyalarını okuma ve yazma işlemleri için kullanılır.

.4.3 Modüller (Modules)

1. `tkinter`:

- **Açıklama:** Python'un standart GUI kütüphanesi. Kullanıcı arayüzü bileşenleri (pencere, buton, metin kutusu vb.) için kullanılır.

2. `sqlite3`:

- **Açıklama:** SQLite veritabanı ile etkileşim kurmak için kullanılır. Veritabanı bağlantısı, sorgular ve veri çekme işlemleri bu modül ile gerçekleştirilir.

3. `openpyxl`:

- **Açıklama:** Excel dosyalarını okuma ve yazma işlemleri için kullanılır. Ders programı Excel'e aktarılırken bu modül kullanılır.

4. os:

- **Açıklama:** Dosya işlemleri (örneğin, dosya varlığını kontrol etme) için kullanılır.

```
# Bölüme özel dersleri yükle
tablo = 'yazilim_muh_dersler' if bolum == 'Yazılım Mühendisliği' else 'bilgisayar_muh_dersler'
self.cursor.execute(f'''
    SELECT ders_kodu, ders_adi, akts, ders_tipi, saat, ders_hocasi, ders_sekli
    FROM {tablo}
    WHERE donem = ?''', (donem,))

for ders in self.cursor.fetchall():
    self.tree.insert('', tk.END, values=ders + (','))

# Ortak dersleri yükle
self.cursor.execute('''
    SELECT ders_kodu, ders_adi, akts, ders_tipi, saat, ders_hocasi, ders_sekli
    FROM ortak_dersler
    WHERE donem = ?''', (donem,))

for ders in self.cursor.fetchall():
    self.tree.insert('', tk.END, values=ders + (','))

def excel_aktar(self):
    try:
        ogrenci_sayisi = int(self.ogrenci_sayisi.get())
        secili_derslik = self.uygun_derslik_bul(ogrenci_sayisi)

        if not secili_derslik:
            messagebox.showerror("Hata", "Uygun derslik bulunamadı!")
            return
```

```

# Bölüm adını al ve dosya adını oluştur
bolum = self.bolum_var.get()
excel_dosya_adi = f"{bolum}_Ders_Programı.xlsx"

# Excel dosyasını yükle
if os.path.exists(excel_dosya_adi):
    wb = load_workbook(excel_dosya_adi)
else:
    wb = load_workbook('Ders_Programı.xlsx')
ws = wb.active

gunler = ['Pazartesi', 'Salı', 'Çarşamba', 'Perşembe', 'Cuma']
gun_satirlari = {'Pazartesi': 4, 'Salı': 16, 'Çarşamba': 28, 'Perşembe': 40, 'Cuma': 52}
uzaktan_satirlari = {'Pazartesi': 12, 'Salı': 24, 'Çarşamba': 36, 'Perşembe': 48, 'Cuma': 60}
musait_saatler = {gun: [True] * 10 for gun in gunler}
uzaktan_musait_saatler = {gun: [True] * 4 for gun in gunler}

```

```

# Önceki dersleri kontrol et ve musait saatleri güncelle
for gun in gunler:
    # Normal dersler için kontrol
    baslangic_satiri = gun_satirlari[gun]
    for i in range(10):
        satir = baslangic_satiri + i
        hucre = f'{self.sinif_sec()}{satir}'
        if ws[hucre].value:
            musait_saatler[gun][i] = False

    # Uzaktan dersler için kontrol
    baslangic_satiri = gun_satirlari[gun]
    for i in range(4):
        satir = baslangic_satiri + i
        hucre = f'{self.sinif_sec()}{satir}'
        if ws[hucre].value:
            uzaktan_musait_saatler[gun][i] = False

```

```

# Yeni dersleri ekle
for item in secili_itemler:
    ders = self.tree.item(item)['values']
    ders_saati = int(ders[4])
    ders_sekli = ders[6]

    if ders_sekli == "uzaktan":
        # Uzaktan dersler için özel zamanlama
        for gun in random.sample(gunler, len(gunler)):
            for i in range(len(uzaktan_musait_saatler[gun]) - ders_saati + 1):
                if all(uzaktan_musait_saatler[gun][i:i + ders_saati]):
                    satir = uzaktan_satirlari[gun] + i
                    hucre = f'{self.sinif_sec()}{satir}'

                    ws[hucre] = f"{ders[0]} - {ders[1]}\n{ders[5]}\nUzaktan"
                    ws[hucre].alignment = Alignment(wrap_text=True, vertical='center', horizontal='center')
                    ws.merge_cells(f"{hucre}:{self.sinif_sec()}{satir + ders_saati - 1}")

                    for j in range(i, i + ders_saati):
                        uzaktan_musait_saatler[gun][j] = False
                    break
            else:
                continue
        break
    else:
        # Normal dersler için normal zamanlama
        for gun in random.sample(gunler, len(gunler)):

```

```

# Dosyayı kaydet
wb.save(excel_dosya_adi)

# Seçili dersleri Treeview'dan kaldır
for item in secili_itemler:
    self.tree.delete(item)

messagebox.showinfo("Başarılı", f"Dersler {excel_dosya_adi} dosyasına aktarıldı ve listeden kaldırıldı!")

except ValueError:
    messagebox.showerror("Hata", "Geçersiz öğrenci sayısı!")
except Exception as e:
    messagebox.showerror("Hata", f"Bir hata oluştu: {str(e)}")

def sınıf_sec(self):
    donem = int(self.donem_var.get())
    return 'C' if donem <= 2 else 'D' if donem <= 4 else 'E' if donem <= 6 else 'F'

def _del_(self):
    self.conn.close()

if __name__ == "__main__":
    root = tk.Tk()
    login = LoginWindow(root)
    root.mainloop()

```

3.2 Görev dağılımı

2 Görev Dağılımı

Proje, dört kişilik bir ekip tarafından geliştirilmiştir. Aşağıda, bileşenlerin tasarım ve geliştirme aşamalarındaki görev dağılımı ile raporun hazırlanmasındaki görevler belirtilmiştir.

Bileşenlerin Tasarım ve Geliştirme Görev Dağılımı

- **Enes (Backend & Veritabanı Sorumlusu):**
 - Veritabanı tasarımını oluşturdu.
 - SQLite ile veri tabanı işlemlerini yönetti.
 - Ders ve öğretim üyeleri verilerinin eklenmesini ve güncellenmesini sağladı.
- **Bener (Frontend & Kullanıcı Arayüzü Geliştiricisi):**
 - Tkinter kullanarak grafik kullanıcı arayüzünü (GUI) tasarladı.
 - Derslerin seçim ekranlarını ve kullanıcı etkileşimlerini geliştirdi.
- **Amine (Ders Programı Algoritması & Çakışma Kontrolü):**
 - Derslerin rastgele yerleştirilmesi algoritmasını oluşturdu.
 - Derslerin gün ve saat bazında çakışmamasını sağlayan kontrol mekanizmasını geliştirdi.
- **Nazlı (Excel Entegrasyonu & Test Sorumlusu):**
 - OpenPyXL kullanarak Excel çıktısının oluşturulmasını sağladı.
 - Verilerin doğru şekilde yazıldığını test etti ve hataları giderdi.

Raporun Hazırlanması Sürecindeki Görev Dağılımı

- **Enes:** Projenin amacı ve genel yapısı hakkında yazılar hazırladı.
- **Bener:** Kullanılan teknolojiler ve yazılım bileşenleri hakkında açıklamalar yaptı.
- **Amine:** Algoritmaların ve çakışma kontrollerinin nasıl çalıştığını belgeledi.
- **Nazlı:** Excel entegrasyonu, çıktı formatı ve test süreçleri hakkında detaylı bilgi ekledi.

3.3 Karşılaşılan zorluklar ve çözüm yöntemleri

1. Veritabanı Tasarım Sorunları

- **Zorluk:** Ders, öğretim üyesi ve derslik bilgilerini içeren ilişkisel bir veritabanı tasarlamak karmaşık oldu.
- **Çözüm:** SQLite kullanılarak her bir bileşen için ayrı tablolar oluşturuldu ve ilişkiler dikkatlice tanımlandı.

2. Ders Çakışmalarının Önlenmesi

- **Zorluk:** Aynı saatlerde birden fazla ders atanması ve öğretim üyelerinin birden fazla derse denk gelmesi problemi yaşandı.
- **Çözüm:** Çakışma kontrol algoritmaları geliştirildi ve derslerin öğretim üyelerinin uygunluk durumuna göre atanması sağlandı.

3. Excel Çıktısındaki Format Hataları

- **Zorluk:** Derslerin Excel'e hatalı ya da iki kez yazılması sorunu yaşandı.
- **Çözüm:** OpenPyXL ile hücre bazlı kontrol mekanizması geliştirildi ve her dersin sadece bir kez yazıldığından emin olundu.

4. Kullanıcı Arayüzü Sorunları

- **Zorluk:** Tkinter kullanarak sezgisel ve kullanıcı dostu bir arayüz oluşturmak başlangıçta zor oldu.
- **Çözüm:** Arayüz için modern tasarım prensipleri benimsendi ve kullanıcıdan alınan geri bildirimlerle iyileştirmeler yapıldı.

3.4 Proje isterlerine göre eksik yönler

1. Kullanıcılar (Öğretim Üyeleri, Öğrenciler, Yönetici)

- **Beklenen Görev:** Öğretim üyeleri, öğrenciler ve yönetici ekleme-çıkarma özelliği olmalı.
- **Durum: Kodlanmadı.**
 - Projede öğretim üyeleri ve öğrenciler için veritabanı tabloları oluşturulmuş ancak bu kullanıcıların eklenmesi, silinmesi veya güncellenmesi gibi işlevler için bir arayüz veya fonksiyon bulunmamaktadır.

2. Bölümler (YZM, BLM)

- **Beklenen Görev:** Bölüm ekleme ve çıkarma modülü olmalı.
- **Durum: Kodlanmadı.**
 - Projede bölümler için ayrı bir tablo veya yönetim modülü bulunmamaktadır. Bölümler, dersler üzerinden dolaylı olarak yönetilmektedir.

3. Dersler

- **Beklenen Görev:** Ders ekleme ve çıkarma modülü olmalı.
- **Durum: Kısmen Kodlandı.**
 - Dersler, veritabanına önceden eklenmiş durumda. Ancak kullanıcı tarafından dinamik olarak ders ekleme veya çıkarma işlevi bulunmamaktadır.

4. Derslikler

- **Beklenen Görev:** Derslik ekleme ve çıkarma modülü olmalı.
- **Durum: Kodlanmadı.**
 - Derslikler, veritabanında önceden tanımlanmış durumda. Ancak kullanıcı tarafından dinamik olarak derslik ekleme veya çıkarma işlevi bulunmamaktadır.

5. Haftalık Program Oluşturma

- **Beklenen Görev:** Haftalık ders programı oluşturulmalı ve Excel'e aktarılmalı.
- **Durum: Kodlandı.**
 - Projede, seçilen derslerin haftalık programı oluşturulup Excel'e aktarılması işlevi başarıyla gerçekleştirilmiştir. Ancak, ortak derslerin çakışmaması gibi kısıtlamalar tam olarak uygulanmamıştır.

6. Kısıtlamalar

- **Beklenen Görevler:**
 - Aynı sınıfa ait dersler çakışmamalı.
 - Ortak dersler öncelikli yerleştirilmeli.
 - Öğretim üyelerinin uygunluk durumuna göre ders ataması yapılmalı.
 - Derslik kapasitesine uygun atama yapılmalı.
 - Zorunlu ders saatleri dikkate alınmalı.
- **Durum: Kısmen Kodlandı.**
 - Derslik kapasitesine uygun atama yapılmıştır. Ancak, öğretim üyelerinin uygunluk durumu, zorunlu ders saatleri ve ortak derslerin çakışmaması gibi kısıtlamalar tam olarak uygulanmamıştır.

7. Veritabanı ve Backend

- **Beklenen Görev:** PostgreSQL, MySQL veya herhangi bir veritabanı yönetim sistemi kullanılmalı.
- **Durum: Kodlandı.**
 - Projede SQLite veritabanı kullanılmıştır. Bu, proje dokümanında belirtilen "herhangi bir veritabanı yönetim sistemi" kriterini karşılamaktadır.

8. Excel Çıktısı

- **Beklenen Görev:** Verilen şablonda ders programının Excel olarak oluşturulması.
- **Durum: Kodlandı.**
 - Projede, ders programı Excel'e başarıyla aktarılmıştır. Ancak, şablonda belirtilen format tam olarak uygulanmamış olabilir.

9. Dinamik ve Düzenlenebilir Sistem

- **Beklenen Görev:** Sistemin dinamik ve düzenlenebilir olması.
- **Durum: Kısmen Kodlandı.**
 - Sistem, ders seçimi ve program oluşturma işlemlerini dinamik olarak gerçekleştirmektedir. Ancak, kullanıcı tarafından ders, derslik veya öğretim üyesi ekleme/çıkarma gibi işlevler bulunmamaktadır.

Kodlanmayan Görevler

1. Kullanıcı Yönetimi:

- Öğretim üyeleri, öğrenciler ve yönetici ekleme-çıkarma işlevleri kodlanmamıştır.

2. Bölüm Yönetimi:

- Bölüm ekleme ve çıkarma modülü kodlanmamıştır.

3. Ders ve Derslik Yönetimi:

- Ders ve derslik ekleme/çıkarma işlevleri kodlanmamıştır.

4. Kısıtlamaların Tam Olarak Uygulanması:

- Öğretim üyelerinin uygunluk durumu, zorunlu ders saatleri ve ortak derslerin çakışmaması gibi kısıtlamalar tam olarak uygulanmamıştır.

5. Dinamik Düzenleme:

- Sistem, kullanıcı tarafından tam olarak dinamik ve düzenlenebilir değildir. Özellikle ders, derslik ve öğretim üyesi yönetimi eksiktir.

4 TEST VE DOĞRULAMA

```
5 import unittest
from unittest.mock import patch, MagicMock
import tkinter as tk
from tkinter import ttk, messagebox
import sqlite3
from openpyxl import load_workbook
import os

# Ana uygulama kodunu burada içe aktarıyoruz (yukarıdaki kodunuzu
buraya ekleyin)
# from your_module import LoginWindow, DersSecimUygulamasi

class TestLoginWindow(unittest.TestCase):
    def setUp(self):
        self.root = tk.Tk()
        self.login_window = LoginWindow(self.root)

    def test_valid_login(self):
        # Geçerli kullanıcı adı ve şifre ile giriş yapıldığında ana
        uygulama penceresinin açılmasını test et
        with patch('sqlite3.connect') as mock_connect:
            mock_cursor = MagicMock()
            mock_connect.return_value.cursor.return_value = mock_cursor
            mock_cursor.fetchone.return_value = ('kostüadmin1',
            'Kostü123') # Geçerli kullanıcı

            self.login_window.username_var.set('kostüadmin1')
            self.login_window.password_var.set('Kostü123')
            self.login_window.login()

            # Ana uygulama penceresinin açıldığını kontrol et
            self.assertTrue(self.login_window.root.winfo_ismapped())

    def test_invalid_login(self):
        # Geçersiz kullanıcı adı veya şifre ile giriş yapıldığında hata
```

```

mesajının gösterilmesini test et
    with patch('sqlite3.connect') as mock_connect:
        mock_cursor = MagicMock()
        mock_connect.return_value.cursor.return_value = mock_cursor
        mock_cursor.fetchone.return_value = None # Geçersiz

kullanıcı

    with patch('tkinter.messagebox.showerror') as
mock_showerror:
        self.login_window.username_var.set('invalid_user')
        self.login_window.password_var.set('invalid_pass')
        self.login_window.login()

        # Hata mesajının gösterildiğini kontrol et
        mock_showerror.assert_called_once_with("Hata",
        "Kullanıcı adı veya şifre hatalı!")

    def tearDown(self):
        self.root.destroy()

class TestDersSecimUygulamasi(unittest.TestCase):
    def setUp(self):
        self.root = tk.Tk()
        self.app = DersSecimUygulamasi(self.root)

    def test_dersleri_goster(self):
        # Bölüm ve dönem seçimi yapıldığında derslerin doğru şekilde
listelenip listelenmediğini test et
        with patch('sqlite3.connect') as mock_connect:
            mock_cursor = MagicMock()
            mock_connect.return_value.cursor.return_value = mock_cursor
            mock_cursor.fetchall.return_value = [
                ('MAT110', 'MATEMATİK I', 5, 'Zorunlu', 5, 'Dr. Öğr.
Üyesi Vildan YAZICI', 'Örgün')
            ]

            self.app.bolum_var.set('Yazılım Mühendisliği')
            self.app.donem_var.set(1)
            self.app.dersleri_goster(None)

            # Derslerin Treeview'da doğru şekilde listelendiğini
kontrol et
            items = self.app.tree.get_children()
            self.assertEqual(len(items), 1)
            self.assertEqual(self.app.tree.item(items[0])['values'],
            ('MAT110', 'MATEMATİK I', 5, 'Zorunlu', 5, 'Dr. Öğr. Üyesi Vildan
YAZICI', 'Örgün', ''))

    def test_uygun_derslik_bul(self):
        # Öğrenci sayısına göre uygun dersliğin bulunup bulunmadığını
test et
        self.app.derslikler = [('M101', 66, 'NORMAL'), ('M201', 141,
'NORMAL')]

        # Öğrenci sayısı 70 olduğunda M201 dersliğinin bulunması
gerekliyor
        secili_derslik = self.app.uygun_derslik_bul(70)
        self.assertEqual(secili_derslik, ('M201', 141, 'NORMAL'))

        # Öğrenci sayısı 150 olduğunda uygun derslik bulunmamalı

```



```

        secili_derslik = self.app.uygun_derslik_bul(150)
        self.assertIsNone(secili_derslik)

    @patch('openpyxl.load_workbook')
    def test_excel_aktar(self, mock_load_workbook):
        # Seçilen derslerin Excel'e doğru şekilde aktarılıp
        aktarılmadığını test et
        mock_wb = MagicMock()
        mock_ws = MagicMock()
        mock_load_workbook.return_value = mock_wb
        mock_wb.active = mock_ws

        self.app.bolum_var.set('Yazılım Mühendisliği')
        self.app.donem_var.set(1)
        self.app.ogrenci_sayisi.insert(0, '70')
        self.app.derslikler = [('M101', 66, 'NORMAL'), ('M201', 141,
        'NORMAL')]

        # Treeview'a örnek ders ekle
        self.app.tree.insert('', tk.END, values=('MAT110', 'MATEMATİK
        I', 5, 'Zorunlu', 5, 'Dr. Öğr. Üyesi Vildan YAZICI', 'Örgün', ''))
        self.app.tree.selection_set(self.app.tree.get_children()[0])

        with patch('tkinter.messagebox.showinfo') as mock_showinfo:
            self.app.excel_aktar()

            # Excel dosyasının kaydedildiğini ve başarı mesajının
            gösterildiğini kontrol et
            mock_wb.save.assert_called_once_with('Yazılım
            Mühendisliği Ders Programı.xlsx')
            mock_showinfo.assert_called_once_with("Başarılı", "Dersler
            Yazılım Mühendisliği Ders Programı.xlsx dosyasına aktarıldı ve listeden
            kaldırıldı!")

        def tearDown(self):
            self.root.destroy()

if __name__ == "__main__":
    unittest.main()

```

5.1 Yazılımın test süreci

Yazılımın test süreci, unittest modülü kullanılarak geliştirilen bir test uygulaması ile gerçekleştirilmiştir. Bu test uygulaması, yazılımın tüm temel bileşenlerini (sınıflar ve fonksiyonlar) kapsayacak şekilde tasarlanmıştır. Test uygulaması, tekrar tekrar çalıştırılabilir ve herhangi bir değişiklik yapıldığında yazılımın doğru çalışıp çalışmadığını kontrol etmeye olanak tanır.

Test Uygulamasının Bileşenleri

1. TestLoginWindow Sınıfı:

- **Amaç:** LoginWindow sınıfının doğru çalışıp çalışmadığını test etmek.
- **Test Edilen Bileşenler:**

- Geçerli kullanıcı adı ve şifre ile giriş yapıldığında ana uygulama penceresinin açılması.
- Geçersiz kullanıcı adı veya şifre ile giriş yapıldığında hata mesajının gösterilmesi.

2. TestDersSecimUygulamasi Sınıfı:

- **Amaç:** DersSecimUygulamasi sınıfının doğru çalışıp çalışmadığını test etmek.
- **Test Edilen Bileşenler:**
 - Bölüm ve dönem seçimi yapıldığında derslerin doğru şekilde listelenmesi.
 - Öğrenci sayısına göre uygun dersliğin bulunması.
 - Seçilen derslerin Excel'e doğru şekilde aktarılması.

5.2 Yazılımın doğrulanması

Test uygulaması ile yazılımın test edilmesi sonucunda elde edilen sonuçlar aşağıda açıklanmıştır. Test sonuçları, yazılımın hangi bileşenlerinin doğru çalıştığını ve hangi bileşenlerin eksik veya hatalı olduğunu göstermektedir.

Test Sonuçları

1. Doğru Çalışan Bileşenler:

- **LoginWindow Sınıfı:**
 - Geçerli kullanıcı adı ve şifre ile giriş yapıldığında ana uygulama penceresi başarıyla açılmaktadır.
 - Geçersiz kullanıcı adı veya şifre ile giriş yapıldığında hata mesajı doğru şekilde gösterilmektedir.
- **DersSecimUygulamasi Sınıfı:**
 - Bölüm ve dönem seçimi yapıldığında dersler doğru şekilde listelenmektedir.
 - Öğrenci sayısına göre uygun derslik başarıyla bulunmaktadır.
 - Seçilen dersler Excel'e doğru şekilde aktarılmaktadır.

2. Eksik veya Hatalı Çalışan Bileşenler:

- **Kullanıcı Yönetimi:**
 - Öğretim üyeleri, öğrenciler ve yönetici ekleme/çıkarma işlevleri henüz kodlanmamıştır. Bu nedenle, bu bileşenler test edilememiştir.
- **Bölüm Yönetimi:**
 - Bölüm ekleme ve çıkarma modülü henüz kodlanmamıştır. Bu nedenle, bu bileşenler test edilememiştir.
- **Ders ve Derslik Yönetimi:**
 - Ders ve derslik ekleme/çıkarma işlevleri henüz kodlanmamıştır. Bu nedenle, bu bileşenler test edilememiştir.
- **Kısıtlamaların Tam Olarak Uygulanması:**

- Öğretim üyelerinin uygunluk durumu, zorunlu ders saatleri ve ortak derslerin çakışmaması gibi kısıtlamalar tam olarak uygulanmamıştır. Bu nedenle, bu kısıtlamalar test edilememiştir.

Test Sonuçlarının Değerlendirilmesi

- **Doğru Çalışan Bileşenler:** Yazılımın temel işlevleri (giriş yapma, ders seçme, derslik atama ve Excel'e aktarma) başarıyla çalışmaktadır. Bu bileşenler, test uygulaması tarafından doğrulanmıştır.
- **Eksik veya Hatalı Çalışan Bileşenler:** Kullanıcı yönetimi, bölüm yönetimi, ders ve derslik yönetimi gibi dinamik işlevler henüz kodlanmamıştır. Ayrıca, bazı kısıtlamalar tam olarak uygulanmamıştır. Bu eksiklikler, yazılımın tam olarak dokümandaki isterleri karşılamadığını göstermektedir.

6.KAYNAKÇA:

<https://www.udemy.com>

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

<https://www.sqlite.org/docs.html>

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