

Yazılım Lab. I Proje I

Öğrenci Ders Seçim Sistemi

EMRE KAYA

Öğrenci No

210201029

I. ÖZET

Bu doküman Kocaeli Üniversitesi 20022-2023 bahar dönemi Mühendislik fakültesi Bilgisayar Mühendisliği Yazılım Lab Laboratuvarı Proje 1 için hazırlanmıştır. Rapor: Özet, Giriş, Yöntem ve Sonuç alt başlıklarından oluşmaktadır.

II. GİRİŞ

Proje Dersi Kayıt Sistemi sanal ortamda, veri tabanı yönetim sistemleri kullanılarak bir üniversitede açılan proe dersleri için talep-kriter temelinde öğrenci-hoca eşleştirmesi yapan bir sistemin temel aksiyonları yerine getirmek üzere tasarlanmış bir projedir. Bu isterlerin gerçekleştirilmesi sırasında Normalizasyon kurallarına uygun veri tabanı sistemleri kullanılmıştır.

III. YÖNTEM

A. Teknolojiler

Bu projenin veritabanı sistemi olarak PostgreSQL kullanılmıştır. PostgreSQL, proje dahilinde kullanılması gereken isterlerden olmasının yanı sıra. Veritabanı işlemlerini yönetmek ve öğrenci-hoca eşleşmelerini yapmak, sistemin bir sunucu yapısında dahi çalışması gibi genelleştirilmiş bir kullanıma izin vermek, gerekli verileri uygun şartlarda verimli bir biçimde saklamak için tercih edilmiştir. Veritabanı, kullanıcı bilgilerini, ders bilgilerini, talep durumlarını. Normalizasyon kurallarına uygun bir şekilde tasarlanan veritabanı, verilerin tutarlı ve erişilebilir bir şekilde saklanmasını sağlar.

Bu projede Yazılım dili olarak Python tercih edilmiştir. Bu tercihin yapılmasında kullanılan genel Kriterler şunlardır : Kolay Okunabilirlik ve Dil Tasarımı: Python, okunabilir ve sade bir dil tasarımına sahiptir.

Geniş Kütüphane Desteği: Python, zengin bir standart kütüphane ve birçok üçüncü taraf kütüphane sunar. projenin farklı gereksinimlerini karşılamak için veritabanı işlemleri için Psycopg2, kullanıcı arayüzü için GTK ve PDF okuma için pyesseract kütüphanelerinin kullanımına imkan sağlamıştır.

Çapraz Platform Desteği: Python, farklı işletim sistemlerinde (Linux, Windows, macOS) çalışabilir. Bu kriter, projenin Linux da geliştirilmesine imkan sağlamıştır.

B. Kütüphaneler

Python GTK Kütüphanesi, Görsel Arayüz: Projenin kullanıcı arayüzü Python GTK kütüphanesi kullanılarak geliştirilmiştir. Bu sayede, öğrenci, öğretmen, yönetici gibi kullanıcıların birbirleri arasında gerçekleştirmesi gereken eylemleri etkili biçimde yapabilmelerine olanak sağlayan bir sistem oluşturulmuştur. Kullanıcı dostu, kullanımı kolay bir arayüz tasarlanmıştır, bu sistemden öğrencilerin ders taleplerini oluşturmasını, hocaların dersleri seçmelerin ve sistem yöneticisinin tüm işlemleri (parametre yönetimi, veri görselleştirme ve yönetimi) yönetmesini kolaylaştırır.

Pytesseract OCR Kütüphanesi: Proje, öğrencilerin transkript belgelerini Python Gtk ile geliştirilmiş bir pdf okuma alanına gelen dosya yolundan okumaktadır, PDF formatında verileri okuma ve veritabanındaki datalara dönüştürme amacıyla Pytesseract OCR kütüphanesi kullanılmıştır. Öğrencilerin daha önce aldıkları derslerin bilgileri bu kütüphane sayesinde otomatik olarak okunur ve ayrılarak veritabanına kaydedilir.

Psycopg2 Kütüphanesi: Projede, PostgreSQL veritabanı ile bağlantı kurmak ve veritabanı işlemlerini yönetmek için Psycopg2 kütüphanesini kullanılmıştır. kütüphane, veritabanı ile iletişim kurmak, basit düzeyde verileri eklemek, güncellemek ve sorgulamak için kullanılır.

C. Proje Akışı

- Kullanıcı ilk başta ilgili panel üzerinden uygun butondan (Öğrenci, hoca ve yönetici) rolünü seçmelidir
- Öğrenci, hoca ve yönetici fark etmeksizin aynı ekrandan ilk başta kendi kullanıcı ad ve şifresiyle giriş yapmaktadır
- Öğrenci, sisteme giriş yaptıktan sonra kendi transkript belgesinin pdf dosyasını öğrenci ekranından yüklemektedir. Gtk'nın sunduğu sürükleyip bırak eylemine bağlı olarak bırakılan dosyanın mutlak dosya yolunu dışarıdan sisteme alınması sağlanmıştır. Yüklenen transkript içeriğindeki öğrencinin önceden almış olduğu ders bilgilerin sistemde OCR aracılığıyla okunarak metne dönüştürülmesi daha sonrasında ise bu verilerin veritabanında ilgili alanlara kaydedilmesi sağlanmaktadır.

- OCR aracılığıyla okunma işlemi :
Öncelikle, giriş olarak bir PDF dosyasının yolunu (filePath) Gtk dan alınır.
PDF dosyası, sayfa sayfa görüntülere dönüştürülür. Her bir sayfadan metinler ayrı ayrı alınacaktır.
customConfig adlı bir yapılandırma dizesi tanımlanır. Bu yapılandırma, OCR işleminin nasıl gerçekleştirileceğini belirtir: " r'-psm 6 -oem 3' " Özel bir ayarlar dizisini içerir.
bu dize, OCR işleminin nasıl gerçekleştirileceğini belirler. Örneğin, hangi dilin kullanılacağı, karakter tanıma hassasiyeti ve diğer ayarlar bu yapılandırma ile kontrol edilir. Bu ayarlar, OCR işlemi sırasında kullanılan algoritmanın davranışını etkiler.
deneme yanılma yöntemiyle mümkün varyasyonlardan test transkriptlerinde en verimli sonucu alan dize olduğu için bu yapı tercih edilmiştir.
Pytesseract.imageToString fonksiyonu kullanılarak. Her bir sayfa üzerindeki metin verisi çıkartılır.
- OCR ile okunan verilerin ayıklanması :
Kocaeli Üniversitesi Bilgisayar Mühendisliği bölümünde okuyan bir öğrenci olarak kendi transkriptimi test verisi olarak kullandım ve bu transkripti öğrenci bilgi sisteminden edindim. İhtiyacım olan veriler Öğrencinin aldığı dersler, isimleri ve öğrencinin ders başarı Notu. bu verilerin elde edilmesi için kullanılan algoritma şu şekildedir :
Öncelikle her bir alt satıra inme karakterinden veri parçalanır. Ardından Koü ders numara sistemindeki derslerin 6-7 basamaklı bir sayı olarak numaralandırılmasından yararlanılarak regex kütüphanesinin de yardımıyla bu veriyi içeren satırlar seçilir. Bu satırlar içerisinde Üniversite ders notu sisteminde bulunan AA, BA, BB, CC, DC, DD, FD, FF, KK, gibi iki büyük harfin yan yana gelmesiyle oluşan notlar için yazılan özel regex koduyla :
"(?![/w()])[A-Z]2(?![/w()])"
ayıklama yapılır ve bulunan ders için alınan ders notu eşleştirilmesi yapılır. (G için özel sorgu yazılmıştır)
Ders ismi, bu iki verinin arasında kalan kısımda bulunmaktadır. Fakat OCR sistemi her ne kadar PDF de okusa ara katmanda resime çevirdiğinden ve türkçe karakter desteği çok iyi çalışmadığından ötürü ders isimlerinde bozulmalar olmuştur. Bu sebepten normalde " " karakterinden ayırma yapılması çok mantıklı değilken anlamsız karakterlerin yok edilmesi fazlasıyla zor olacağından ve kalan verinin de gerçek anlamını yitireceğinden ötürü hatalı karakterler de içerse yalnızca " " karakterinden ayrılarak aradaki veri isim olarak alınmıştır.
- Ayıklanan verinin öğrenci panelinde görselleştirilmesi :
Bir Gtk dialog içerisinde Gtk Liststore widgeti

kullanılarak sqllablosundaki veriler görselleştirilmiştir.

- Sistem transkriptini yüklemiş olan öğrenciler Bir Gtk dialog da Gtk Liststore içerisinde almak istediği derse göre dersi veren hocaları listelemektedir burada gösterilen veriler Ders Numarası, Ders Adı, Hoca Adu, Hoca ilgi alanları, Hoca Sicil Numarası. ayrıca öğrenci dialogda bulunan bir comboBox dan seçerek ilgi alanlarına göre ve derslere göre hoca listesinde filtreleme yapabilmektedir.
- Öğrenci alacağı her bir ders için sistem yöneticisinin başta belirlediği sayıda farklı hocalardan talep oluşturabilmektedir. Bunu öğrenci panelinde bulunan talep et dialogundaki listStore widgetinde bulunan checkBoxButton sayesinde yapmaktadır. Herhangi bir hoca tarafından talebi onaylanmadığı sürece yine aynı ekranda talep ettiği butondaki tiki kaldırarak talebini geri çekebilmektedir.
- Öğrenci Talep ekranından herhangi bir hocadan talep oluştururken hocaya iletmek istediği bir mesajı sistemncde açılan ayrı bir pencereden, yöneticinin belirlediği karakter sınırıyla sistemden gönderebilmektedir.
Aynı şekilde hoca da panelinde bulunan öğrenciye dersi alması konusunda talepte bulunma ekranından talep oluştururken öğrenciye iletmek istediği mesajı yöneticinin belirlediği karakter sınırında öğrenciye iletebilmektedir.
her iki rolde bulunan kullanıcı kendi panelinde bulunan mesajları görüntüle arayüzü sayesinde kendisinin gönderdiği ve kendisine gelen tüm mesajları bir ekranda görebilmektedir.
- Hoca panelinden giriş yapıp, kendisine gelen dersini alma istekleri arasından kontenjanı olması halinde ilgili öğrencinin talebini ilk onaylayan hoca olması kaydıyla bu öğrenciye bu hocadan ders eklenmektedir. Kabul edilen derslerin verisi AcceptedLessons adlı tabloda tutulmaktadır. öğrencinin panelinde aldığı ders ile ilgili bilgiler, veren hoca ile ilgili bilgiler görselleştirilmektedir.
- Hoca sisteme giriş yaptıktan sonra kendi ilgi alanlarını seçerek sisteme kaydedebilmektedir. Bunu hoca paneline bulunan ilgi Alanı ekle bölümündeki bir ilgi alanı checkbox yardımı ile yapabilmektedir. Bu sayede öğrenciler hocaalrın ilgi alanlarını görüp ders seçimlerinde bunu dikkate alabilmektedirler.
- Hoca sistemden kendi dersine talep oluşturan öğrencilerin listesini görüntüleyebilmektedir. Bunun yanısıra aynı panelden bu öğrencinin isteğini kabul edip etmeme konusundaki kararını verip öğrenciyi dersine alabilir veya reddede bilir.

- Hoca kendisinin de verdiği bir ders için henüz bir hoca tarafından onaylanmamış talebi olan veya talep oluşturmamış öğrencileri de görüntüleyebilmektedir. Bunun gerçekleşmesi için kullanılan algoritma şöyledir : Öncelikle aramamız bir hocaya bağlı olacağından ötürü o hocanın verdiği tüm derslerin verisi talodan alınır. ardından sisteme kayıtlı tüm öğrenciler tablodan çekilir. Her bir öğrenci için öncelikle bu öğrencinin kabul aldığı dersler tablodan çekilir. Eğer hocanın verdiği tüm dersler arasında öğrencinin henüz hiçbir hocadan kabul almadığı bir ders varsa bu dersin numarası ve öğrencinin numarası bağdaştırılarak bir veri yapısına görselleştirmek üzere aktarılır. Bulunan bu öğrenci bulunan ders için sorgulanan hoca tarafından derse davet edilebilir olmaktadır.
- Hoca öğrenciye dersini alması için talep oluştururken aynı zamanda o öğrencinin aldığı dersleri de açılan ayrı bir pencerede görüntüleyebilmektedir.
- Hoca listelediği öğrencilerden herhangi birisine dersini alması için talepte bulunabilmektedir. Öğrenci ekranından sorgulandığında görünen ekranda öğrencinin talebi onaylaması durumunda öğrenciye ders eklenecektir. ancak öğrenciye red seçeneği sunulmamaktadır.
- Hoca ancak yöneticinin belirlediği kontenjan sayısında öğrenci kabul edebilmektedir. Sistem aşım olmasının önüne geçmektedir ve kullanıcıyı dersi onaylayan hoca da olsa, hocanın tabelini onaylamaya çalışan öğrenci de olsa uyarılmaktadır.
- Sistemde yöneticinin belirlediği bir zaman aşımı bulunmaktadır, bu zaman sayacı program başladığı anda ayrı bir thread de çalışmaktadır ve verilen parametre verisine kadar sayımı gerçekleştirmektedir. sayımın bitmesi durumunda yetkisi dahilinde thread çağırıldığı noktadaki bir bitiş değişkeni üzerinde değişiklik yaparak bunu belirtir. Sistemde yönetici zaman verisini her güncellediğinde yeniden başlatmamak için aynı zamanda bir maxZaman, geçen süre karşılaştırılması yapılarak ikinci bir kontrol katmanı oluşturulmuştur. Zaman aşımının dolması durumunda hoca ve öğrencinin talepte bulunma, kabul ve onay işlemleri kapatılmaktadır.

IV. TABLOLAR

A. Kullanılan Rüm Tablolar

Schema	acceptedlessons	Type	Owner
public	activelessons	table	postgres
public	login	table	postgres
public	messages	table	postgres
public	req	table	postgres
public	root	table	postgres
public	students	table	postgres
public	studentslessons	table	postgres
public	teachers	table	postgres

B. Root Tablosu

rec	reqTeacherLimit	msgagelenlimit	timeout
0	1	100	300

C. activelessons Tablosu

rec	lessonno	lessonname	regno
1	20020035	yazlab I	10
2	20020056	mat I	11

D. login Tablosu

userno	password	userid	userrole
1	root	root	root
2	1234	emre	student
3	1234	emre	teacher

E. messages Tablosu

rec	studentno	regno	messagetext
1	210201029	10	Wirt Your Message Here
2	210201029	10	Wirt Your Message Here

F. req Tablosu

rec	studentno	regno	lessonno
1	210201146	11	20020035
2	12	21020176	2002004

G. students Tablosu

userno	studentno	name	surname	transkriptpath	note	interests
2	210201029	EMRE	KAYA	EmreKayaTranskript.pdf	3.52	OS AI
2	210201029	EMRE	KAYA	EmreKayaTranskript.pdf	3.52	OS AI

H. students Tablosu

userno	studentno	lessonno	note	name
2	210201029	20020035	AA	mat I

I. teachers Tablosu

userno	regno	name	surname	maxstudent	interests
2	10	EMRE	KAYA	100	OS AI
4	11	FATİH	KAYA	100	

V. SONUÇ

Sonuç olarak, PostgreSQL veritabanı yönetim sistemi, Python programlama dili ve GTK arayüzü kullanan bir öğrenci - hoca ders kayıt sisteminin başarılı bir uygulaması gerçekleştirilmiştir. Proje, öğrencilerin ilgi alanlarına uygun proje dersleri seçmelerine ve hocaların ders taleplerini yönetmelerine olanak sağlarken, veritabanı tasarlanması hayata geçirilmesi ve uygulanması, OCR teknolojisi ve masaüstü uygulama geliştirme konularında pratik ve deneyim kazandırmıştır.

REFERENCES

- [1] <https://python-gtk-3-tutorial.readthedocs.io/en/latest/>
- [2] <https://www.gtk.org/docs/language-bindings/python>
- [3] <https://wiki.archlinux.org/title/PostgreSQL>
- [4] <https://www.postgresql.org/docs/>
- [5] <https://docs.python.org/3/>
- [6] <https://tesseract-ocr.github.io/tessdoc/Documentation.html>

A. Rasthree Öğrenci atama işlemi için Kod Akışı.

