

Not Takip Sistemi (WEB)

Uğuralp KIVANÇ 231307081

İbrahim Alperen KESKİN 231307052

Furkan Kerim OCAK 231307030

Bilişim Sistemleri Mühendisliği

Veritabanı Yönetim Sistemleri

Kocaeli Üniversitesi Kocaeli/Türkiye

Geliştirme Dili: Java

Geliştirme Ortamı: Visual Studio Code (VS Code)

Mimari Yaklaşım: 3 Katmanlı (Three-Tier) Mimari

Giriş

Günümüzde eğitim kurumlarında öğrenci notlarının manuel yollarla takip edilmesi birçok sorunu beraberinde getirmektedir. Bu sorunlar arasında zaman kaybı, verilerin düzensizliği ve kullanıcı hataları öne çıkmaktadır. Bu bağlamda, öğrenci notlarının daha sistematik ve düzenli biçimde işlenmesini sağlayan bir masaüstü yazılımı gereksinimi doğmuştur. Geliştirilen bu proje, bu ihtiyacı karşılamak üzere hazırlanmış sade ve işlevsel bir çözümdür.

Projenin Amacı

Bu projenin temel amacı, öğrenci bilgileriyle birlikte sınav notlarının sisteme girilmesi, gerektiğinde güncellenmesi, silinmesi ve kolaylıkla listelenmesini sağlamaktır. Kullanıcılara, sade bir arayüz üzerinden öğrenci verilerini işleyebilecekleri bir yapı sunulmuştur.

Kullanılan Teknolojiler

- Java:** Nesneye dayalı ve platformdan bağımsız programlama dili
- React:** Kullanıcı arayüzleri geliştirmek için kullanılan, bileşen tabanlı ve yeniden kullanılabilir yapıya sahip modern bir JavaScript kütüphanesidir.

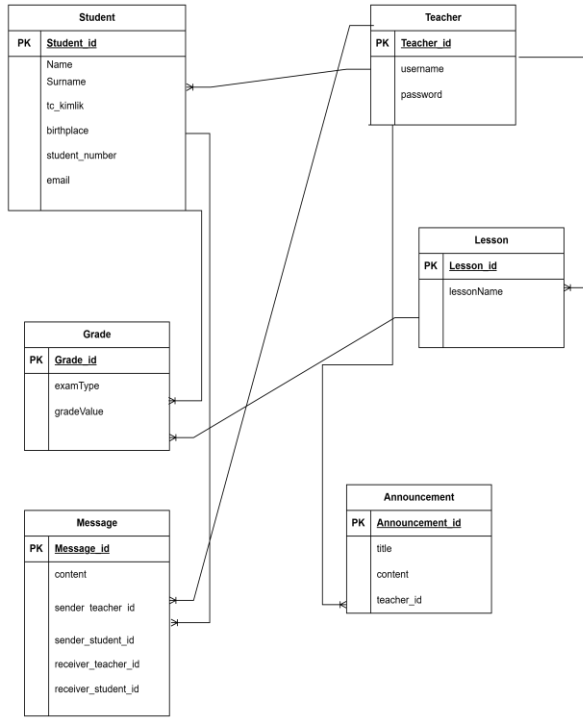
- JTable & DefaultTableModel:** Tablo tabanlı veri gösterimi ve yönetimi
- IDE:** Visual Studio Code (VS Code)
- Veri Tabanı Modeli (Entity Yapısı)**
Bu uygulamada herhangi bir harici veritabanı (örneğin SQLite veya MySQL) kullanılmamaktadır. Bunun yerine, veriler uygulama belleğinde geçici olarak **DefaultTableModel** üzerinden yönetilir. Bu yapı, verilerin tablo şeklinde saklanmasını ve düzenlenmesini sağlar.

Öğrenci (Student) Entity

Alan Adı	Türü	Açıklama
Ad	String	Öğrencinin adı
Soyad	String	Öğrencinin soyadı
Vize Notu	Double	Öğrencinin vize sınavından aldığı not
Final Notu	Double	Öğrencinin final sınavından aldığı not
Ortalama	Double	Vize ve final notlarıyla hesaplanan ort

Entity Özellikleri

- Tüm veriler bir Object[] dizisi halinde tabloya satır olarak eklenir.
- Bu yapıya karşılık gelen tablo **JTable** içinde gösterilir.
- Veriler, DefaultTableModel kullanılarak kullanıcı arayüzüyle senkronize şekilde çalışır.
- Her öğrenci kaydı ayrı bir satır olarak tutulur.
- Ortalama değeri, girilen vize ve final notlarına bağlı olarak anlık hesaplanır.
- Not: Kalıcı veri saklama yapılmadığı için veriler uygulama kapandığında kaybolur. Gelişmiş sürümlerde bu yapılar bir dosya veya veritabanı ile ilişkilendirilebilir.



Tasarım Yaklaşımı

Bu uygulama, öğrencilerin ders notlarını görüntüleyebilmesi ve öğretmenlerin not girişlerini kolayca yapabilmesi amacıyla geliştirilen web tabanlı bir sistemdir. Proje, kullanıcı dostu bir arayüz ile hem öğrencilere hem öğretmenlere yönelik işlevsel özellikler sunacak şekilde yapılandırılmıştır. Tasarımda modern web geliştirme ilkeleri esas alınarak, sürdürülebilir ve genişletilebilir bir mimari hedeflenmiştir.

Uygulama Mimarisi

Proje, **frontend** ve **backend** olmak üzere iki temel katmandan oluşur:

- **Frontend (Kullanıcı Arayüzü):** React tabanlıdır. Bileşen (component) temelli yapı benimsenmiştir.
- **Backend (Sunucu Tarafı):** Node.js ya da Python Flask/Django gibi framework'ler kullanılabilir (proje içeriğine göre netleştirilebilir).
- **Veritabanı:** JSON veri dosyaları üzerinden ya da küçük ölçekli projelerde kullanılabilecek düz yapılarla veri yönetimi sağlanmıştır. Geliştirilebilir versiyonunda ilişkisel bir veritabanı (örneğin: SQLite veya PostgreSQL) önerilir.

Kullanılan Yapılar

- **React:** Modern, component-based yapı ile sayfa güncellemeleri minimum veri yükü ile sağlanır.
- **Props & State Management:** Veriler, bileşenler arası "props" ile aktarılır, local state kullanımı ile UI güncel tutulur.
- **TailwindCSS / CSS Modules:** Arayüzde sade ve erişilebilir tasarım prensipleri benimsenmiştir.

- **RESTful API (Opsiyonel):** Arka uç katmanla veri alışverişi yapılması durumunda JSON tabanlı REST servisler tasarlanabilir.

Tasarım Katmanları

■ Kullanıcı Arayüzü (UI)

- Öğrenciler için: Giriş ekranı, not görüntüleme paneli
- Öğretmenler için: Not ekleme/güncelleme formu, öğrenci listesi
- Yönetici için: Öğretmen ve öğrenci yönetimi (opsiyonel) **Veri Yapısı**
- Student: Öğrenci bilgileri (id, ad, sınıf vb.)
- Teacher: Öğretmen bilgileri
- Course: Ders bilgileri
- Grade: Notlar
- Announcement: Duyurular
- Message: İki yönlü mesajlaşma
- Kullanıcı dostu arayüz desteği

• İlişkiler ve Veri Yönetimi

- Her öğrenci bir veya birden fazla not alabilir.
- Her öğretmen bir veya daha fazla ders verebilir.
- Notlar, öğrenci ve dersle ilişkilidir.
- Duyurular öğretmenler tarafından oluşturulur.
- Mesajlar hem öğretmen-öğrenci hem öğrenci-öğretmen yönünde çalışır.
- Öğretmenler tek başına kayıt olabilir, öğrencilerin kayıt olabilmesi için öğretmenlere ihtiyaç vardır
- Öğretmenler kayıt yapabilmek için öğrencinin email adresine ihtiyacı vardır
- Öğrencinin girmiş olduğu sınavın notlarını öğretmen girer.

Github Linki: <https://github.com/Abisat5/45.Grup-Not-Takip-Sistemi-Proje-Odevi>

Kaynakça:

- <https://docs.spring.io/spring-boot/>
- <https://spring.io/projects/spring-framework>
- <https://docs.spring.io/spring-data/jpa/reference/>
- <https://reactjs.org/docs/getting-started.html>
- <https://reactrouter.com/en/main>
- <https://www.framer.com/motion/>
- <https://developer.mozilla.org/en-US/docs/Web/CSS>
- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- <https://css-tricks.com/snippets/css/media-queries-for-standard-devices/>
- <https://stackoverflow.com/>
- <https://docs.oracle.com/en/java/javase/>

- <https://maven.apache.org/plugins/maven-compiler-plugin/>
- Chatgpt
- Belirli Youtube Eğitim videoları