

Ders Programı Oluşturma Uygulaması

Zeynep Serva Yıldız
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
201307052
201307052@kocaeli.edu.tr

Zehra Betül Taşkın
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
201307054
201307054@kocaeli.edu.tr

Ahmet Melih Türkmen
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
201307013
201307013@kocaeli.edu.tr

Abstract—Projemiz, okullarda ders programlarını düzenlemek ve takip etmek amacıyla geliştirilen bir C# .NET uygulamasıdır. Uygulama öğretmenlerin, dersliklerin, sınıfların ve derslerin düzenli bir şekilde planlanmasına ve ders program oluşturulmasına olanak tanımaktadır.

Anahtar Kelimeler—C#, .Net Framework, ders programı, graf renklendirme algoritması.

I. GİRİŞ

Günümüzde teknolojinin gelişmesiyle birlikte eğitim alanında da verimliliği sağlamak ve zamandan kazanmak amacıyla çeşitli projeler ve uygulamalar geliştirilmiştir. Öğrencilerin ve öğretmenlerin sayısının yoğun olduğu günümüzde her konuda esneklik sağlayan ve çakışmalardan bizi koruyacak uygulamalarda geliştirilmiştir. Eğitim sistemlerinde, öğrencilerin ve öğretmenlerin verimli bir şekilde eğitim alabilmesi için düzenli bir ders programı oluşturmak önemli bir gerekliliktir. Bu bağlamda, bizde eğitim sistemlerine katkı sağlayacak bu konuya güzide bir bakış açısıyla yaklaşarak C# .Net tabanlı kendi projemizi oluşturduk. Bu program, kullanıcıların öğretmen, derslik, sınıf, ders gibi bilgileri giriş yaparak belirli kısıtlara bağlı olarak öğrencilere uygun, dengeli ve etkili bir ders programı oluşturmalarına yardımcı olmayı amaçlamaktadır.

II. GELİŞTİRME

A. Programın Temel Fonksiyonları:

Veri Girişi:

Kullanıcılar, program arayüzü üzerinden öğretmen bilgilerini, derslikleri, sınıfları ve dersleri sisteme girebilirler. Bu veriler, programın ders programı oluştururken kullanılacak temel bilgileri oluşturur.

Ders Programı Oluşturma:

Kullanıcılar, belirli kısıtlara göre (örneğin, öğretmen branşı, ders dağılımı, öğretmenin müsaitliği vb.) ders programlarını oluşturabilirler. Program, çakışmaları önlemek ve dengeyi sağlamak için otomatik olarak uygun bir ders programı önerisi sunabilir.

Ders Programı İzleme ve Güncelleme:

Kullanıcılar, oluşturulan ders programını izleyebilir ve güncelleyebilirler. Öğretmenlerin veya sınıfların değişen durumlarına göre program, dinamik olarak ayarlanabilir.

Raporlama ve Analiz:

Program, ders programının çeşitli istatistiklerini ve raporlarını kullanıcılara sunar. Bu raporlar, öğretmen branşı,

ders dağılımları, derslikler, sınıflar ve öğretmen yükü gibi faktörleri içerebilir ve kullanıcılara ders programının etkililiği hakkında bilgi sağlar.

Güvenilirlik ve Hata Yönetimi:

Program, kullanıcıların hatalı girişleri önlemek ve güvenilir bir ders programı oluşturmak için hata yönetimi mekanizmalarına sahiptir. Kullanıcılara bilgi ve geri bildirim sağlayarak doğru veri girişi yapmalarına yardımcı olur.

Sonuç olarak, C# .Net ile geliştirilen bu program, eğitim kurumlarının daha organize ve verimli bir şekilde ders programları oluşturmalarına yardımcı olan kapsamlı bir çözümdür. Program, kullanıcı dostu arayüzü ve güçlü fonksiyonları ile eğitim yönetimini kolaylaştırmak için tasarlanmıştır.

B. Kullanılan Teknolojiler ve Programlama Dilleri:

1) C# (C Sharp):

C#, Microsoft tarafından geliştirilen bir programlama dilidir ve genellikle Windows tabanlı uygulamaların geliştirilmesinde kullanılır. C#, nesne odaklı bir dil olup güçlü bir şekilde tip güvenliği ve modern programlama özelliklerini destekler. Programın temel yazılım altyapısı C# dilinde geliştirilmiştir, bu da programın güvenilir, hızlı ve ölçeklenebilir olmasını sağlar.

2) .NET Framework:

.NET Framework, Microsoft'un geliştirdiği bir yazılım platformudur ve C# gibi dillerle yazılmış uygulamaların çalıştığı ortamı sağlar. Bu platform, çeşitli kütüphaneleri, API'ları ve araçları içerir. .NET Framework, programın geliştirilmesi ve çalıştırılması için temel altyapıyı oluşturur. Ayrıca, .NET'in sunduğu zengin sınıf kütüphaneleri, kodun daha etkili ve hızlı bir şekilde yazılmasına olanak tanır.

3) Visual Studio:

Visual Studio, Microsoft'un entegre geliştirme ortamıdır. C# gibi dillerle yazılan projelerin tasarımından derlenmesine ve hata ayıklanmasına kadar birçok geliştirme sürecini destekler. Geliştiricilere zengin bir kod editörü, hata ayıklayıcı, derleyici ve arayüz tasarım araçları sağlar. Ayrıca, Visual Studio'nun zengin eklenti ekosistemi ve entegre takım işbirliği özellikleri, yazılım geliştirme sürecini daha verimli hale getirir.

4) Microsoft SQL Server (MSSQL):

Microsoft SQL Server, ilişkisel veritabanı yönetim sistemidir ve programın veritabanı yönetimini sağlamak için kullanılır. MSSQL, güvenilirlik, performans ve ölçeklenebilirlik açısından zengin bir özellik setine sahiptir.

Bu veritabanı sistemi, programın öğretmen, derslik, sınıf ve ders bilgilerini depolamak, sorgulamak ve yönetmek için kullanılır. Veritabanı işlemleri, programın genel performansını ve veri bütünlüğünü sağlamak için MSSQL üzerinden gerçekleştirilir.

Bu teknolojilerin entegrasyonu, güçlü bir uygulama geliştirme ve yönetme çözümü sunar. C#, .NET Framework, Visual Studio ve MSSQL, programın etkili bir şekilde çalışmasını sağlamak için bir araya gelir ve kullanıcı dostu, güvenilir bir ders programı oluşturma platformu sunar.

C. Kullanıcı Arayüzü:

Projemizde kullanılan arayüz ekranlarımız masaüstü uygulaması şeklinde tasarlandığından dolayı Windows form kullanılarak tasarlanmıştır. Projemizin arayüz kullanımı hakkında çeşitli bilgiler bu bölümde yer alacaktır.

1-Giriş Ekranı

Projemizi ilk çalıştırdığımızda karşımıza gelen arayüzümüzde hangi veritabanı sunucusu ve veritabanı ile çalışacaksa onun bağlantı adresini veriyoruz ya da hazır bir veritabanımız yok ise sunucu adresini yazarak yeni bir veritabanı oluşturarak devam ediyoruz.

Kurulum Sihirbazı

Sunucu Adresi: 11738DH1.SQLEXPRESS01

Veritabanı Adı: dersdb2

Kullanıcı Adı: sa

Şifre:

☒ Windows Authentication ☐ SQL Server Authentication

Bağlan ve Verileri Çek

İzle

Şekil. 1. Veritabanı işlemleri ekranı

2-Ders ve Öğretmen Ekranları

Projemizde gerekli veritabanı bağlantılarını uyguladıktan sonra sistemimizde yer alacak olan derslerimizi, öğretmenlerimizi, dersliklerimizi ve sınıflarımızı ekleyerek gerekli dersler hangi sınıflarda hangi öğretmenler ve dersliklerde olacak şekilde ders atamamızı yapabiliyor olacağız. Sistemimizde yer alacak olan dersler ve öğretmenler konusunda çeşitli kısıtlamalara gidilebilir aynı zamanda hangi gün uygun olacaklarını ya da olmayacaklarını sistem üzerinden belirleyebiliyor olacağız.

Şekil. 2. Ders işlemleri ekranı

Şekil. 3. Sınıf işlemleri ekranı

Şekil. 4. Derslik işlemleri ekranı

Şekil. 5. Öğretmen işlemleri ekranı

Şekil 6. Ders atama ekranı

Şekil 7. Zaman tablosu ekranı

Şekil 8. Kısıt işlemleri ekranı

3-Ders Programı Oluşturma Aracı

Projemizde gerekli haftalık ders dağılım şeklini öğretmenlerin müsaitlik durumunu ve nerede ders vereceklerini sistemimizde belirledikten sonra ders programı oluşturma aracımız ile gerekli konfigürasyonlara sahip haftalık ders çizelgemiz oluşturulmaktadır. Dersleri çizelge üzerinden günlerini veya saatlerini değiştirebilme özelliğimiz mevcuttur ama çakışan derslerimiz aynı zaman aralığında çizelge üzerinde yer alamazlar.

	Pazartesi	Salı	Çarşamba	Perşembe	Cuma
SIF1					
SIF2					

Şekil 9. Ders programı ekranı

D. Metodoloji:

Bu .Net projesi, kullanıcıların öğretmen, ders, derslik ve sınıflarla ilgili CRUD işlemlerini gerçekleştirebilmelerine ek olarak belirli kısıtlara dikkat ederek ders programı oluşturabilmelerine odaklanmaktadır. Projenin geliştirme sürecinde izlenen metodoloji aşağıdaki adımları içerir:

1) Gereksinim Analizi:

Proje başlangıcında, kullanıcılarla etkileşim içinde bulunularak öğretmen, ders, derslik ve sınıflarla ilgili beklentiler ve kısıtlar detaylı bir şekilde belirledik. Hangi kısıtların dikkate alınması gerektiği ve ders programı oluşturulurken nasıl bir denge sağlanması gerektiği üzerine odaklandık.

2) Veritabanı Tasarımı:

MSSQL kullanılarak, öğretmen, ders, derslik ve sınıf bilgilerini depolayan tablolar oluşturduk. Veritabanı tasarımı, ders programı oluşturulurken dikkate alınması gereken kısıtların ve ilişkilerin anlaşılmasına yönelik oluşturduk.

3) Proje Yapısının Oluşturulması:

Visual Studio kullanılarak, projenin temel yapısı oluşturduk. Proje, öğretmen, ders, derslik ve sınıflar için CRUD işlemlerini içeren iş mantığı sınıflarını ve bu sınıfların kullanıcı arayüzüyle etkileşimini içerir.

4) Kullanıcı Arayüzü Tasarımı:

Kullanıcı dostu anlaşılabilir ve basit bir arayüz tasarladık. Kullanıcılar, öğretmen, ders, derslik ve sınıf bilgilerini ekleyebilir, güncelleyebilir, silebilir ve belirli kısıtlara göre ders programı oluşturabilirler. Arayüz, kullanıcıların projeyi kullanırken kısıtları anlamalarını ve bu kısıtlara uygun olarak hareket etmelerini sağlamak amacıyla özenle tasarladık.

5) CRUD ve Ders Programı Oluşturma Metotları:

Her entite için CRUD işlemlerini gerçekleştiren C# metotları oluşturduk. Ders programı oluşturma işlemleri için özel metotlar ekledik. Bu metotlar, kullanıcının belirlediği kısıtlara uygun olarak ders programı önerileri sunabilir ve çakışmaları engellemek için otomatik düzenlemeler yapabilir.

6) Test ve Hata Ayıklama:

Oluşturulan metotlar üzerinde kapsamlı bir test süreci gerçekleştirdik. Kullanıcıların belirlediği kısıtlara uygun olarak ders programı oluşturdukları durumları test ettik. Hata durumları ve kısıtlara uymayan girişler üzerinde titizlikle test yaptık.

Bu metodoloji, projenin öğretmen, ders, derslik ve sınıflar üzerinde etkileşim içinde bulunarak belirli kısıtlara dikkat ederek ders programı oluşturulmasını sağlamak üzere

adım adım bir yaklaşımı temsil etmektedir. Her aşama, kullanıcıların projeyi etkili bir şekilde kullanabilmeleri ve belirlenen kısıtlara uygun olarak ders programı oluşturabilmeleri için titizlikle uygulanmıştır.

III. TARTIŞMA VE SONUÇ

A. Tartışma

Projenin kullanıcı arayüzü, kullanıcıların kolaylıkla öğretmen, ders, derslik ve sınıf bilgilerini yönetmelerini sağlar. Ayrıca, ders programı oluşturma sürecinde kullanıcıların belirlediği kısıtları anlamalarını kolaylaştırmak için kullanıcı dostu bir tasarım benimsemekte ve bu sayede kullanıcı deneyimini artırmaktadır. Proje, kullanıcının belirlediği kısıtlara odaklanarak ders programı oluşturma sürecini yönetmektedir. Kullanıcılar, öğretmen müsaitlik durumu, sınıf kapasitesi gibi faktörleri dikkate alarak ders programını oluşturabilirler. Bu, ders programının daha düzenli ve etkili bir şekilde yapılandırılmasını sağlar. MSSQL kullanılarak sağlanan veritabanı yönetimi, öğretmen, ders, derslik ve sınıf bilgilerini güvenli ve düzenli bir şekilde depolar. Bu, projenin veri bütünlüğünü sağlaması ve kullanıcılara güvenilir bir deneyim sunması açısından önemli bir faktördür. Proje, belirlenen kısıtlar doğrultusunda dinamik bir şekilde ders programı önerileri sunabilir. Bu, kullanıcıların çeşitli senaryolara göre esnek bir şekilde ders programını düzenleyebilmelerine olanak tanır.

B. Sonuç:

Projede kullanılan teknolojiler, güvenilirlik ve performans açısından yüksek standartlarda olup, veri

bütünlüğünü sağlamak adına titiz bir veritabanı yönetimini içermektedir. Bu proje, eğitim yönetimi süreçlerini düzenleme, verimliliği artırma ve kullanıcıların ihtiyaçlarına uygun esneklik sağlama amacıyla tasarlanmıştır. Kullanıcılara veritabanı işlemleri ve ders programı oluşturma sürecindeki etkileşimlerinde kolaylık sunan bu projenin, eğitim kurumları için önemli bir katkı sağlayabileceği düşünülmektedir.

KAYNAKLAR

- [1] De, S., & Vijayakumaran, V. (2020, November). An Efficient Algorithm in Project Management for Resource Scheduling and Conflict Management using Graph Coloring Technique. In 2020 IEEE International Conference for Innovation in Technology (INOCON) (pp. 1-6). IEEE.
- [2] Deogratias, E. (2022). Using Graph Coloring for Effective Timetable Scheduling at Ordinary Secondary Level. International Journal of Curriculum and Instruction, 14(2), 1166-1188.
- [3] Nandal, P., Satyawali, A., Sachdeva, D., & Tomar, A. S. (2021, January). Graph coloring based scheduling algorithm to automatically generate college course timetable. In 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 210-214). IEEE.
- [4] Hata Çözme İşlemleri
- [5] Faydalanılan Konuyla Alakalı Örnek Proje
- [6] How to Insert Data in Table Cells?