

Çizge Renklendirme ile Ders Programı

Hazırlama Proje Raporu

1. Esmâ Gelmez
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, TÜRKİYE
221307099

2. Ömer Faruk Yılmaz
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, TÜRKİYE
191307048

3. Ramazan Geriş
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, TÜRKİYE
191307067

Özet—Bu proje, ders programı oluşturulurken çeşitli kısıtları ele almayı amaçlayan bir mobil uygulama geliştirme projesidir. Greedy Coloring algoritması kullanılarak, ders veren hocaların çakışmaları önlenecek, dersler istenilen günlere atanacak ve sınıf kısıtlamaları gözetilerek optimal bir ders programı hazırlanacaktır. Proje, Android Studio ve Java programlama dili kullanılarak geliştirilecektir.

Anahtar Kelimeler—Ders Programı Oluşturma, Greedy Coloring Algoritması, Mobil Uygulama Geliştirme, Optimizasyon, Eğitim Teknolojileri, Kısıtlamalar, Android Studio, Java, Çizge Renklendirme

I. GİRİŞ

Proje, ders veren hocaların çakışmalarını en aza indirgeyerek daha düzenli ve etkili ders programları oluşturabilmeyi hedeflemektedir. Ayrıca, dersleri istenilen günlere atama, sınıf sayıları ve isimleri gibi kısıtları gözeterek optimize edilmiş bir planlama süreci sunmayı amaçlamaktadır. Bu doğrultuda, Greedy Coloring algoritması kullanılarak çizge renklendirme konsepti entegre edilecek ve Android tabanlı bir mobil uygulama geliştirilecektir.

II. PROJE AKIŞ DİYAGRAMI



[Şekil 1]: Proje Akış Şeması görüntüsü.

Bu projede kullanıcının erişebileceği temel özellikler şunlardır:

- **Kullanıcı Girişi Ekranı:** Kullanıcı, uygulamaya giriş yapmak için kullanıcı adı ve parolasını girebilir.[1, 2, 4]
- **Parola Değiştirme Ekranı:** Kullanıcı, mevcut parolasını değiştirebileceği bir ekranı kullanabilir.[1, 3, 7]
- **Kayıt Ol Ekranı:** Yeni kullanıcılar, uygulamaya kayıt olmak için gerekli bilgileri bu ekran üzerinden girebilir.[1, 3, 7]
- **Menü Ekranı:** Kullanıcının ana ekranıdır. Profil düzenleme, ders ekleme, ders programını düzenleme, ders programını görüntüleme gibi seçenekleri içerir. [1, 3]
- **Profil Düzenleme Ekranı:** Kullanıcı, kişisel bilgilerini düzenleyebileceği bir ekranı kullanabilir. [1, 3, 4]
- **Ders Ekleme Ekranı:** Kullanıcı, yeni dersler ekleyebilir ve bu derslere ilişkin bilgileri girerek kaydedebilir.[1, 3, 7]
- **Ders Programı Düzenleme Ekranı:** Kullanıcı, mevcut ders programını düzenleyebilir ve programındaki dersleri güncelleyebilir.[5, 6, 7]
- **Ders Programını Görüntüleme Ekranı:** Kullanıcı, oluşturulan ders programını görüntüleyebilir ve planlamasını inceleyebilir. [1, 7]
- **Çıkış Ekranı:** Kullanıcı, uygulamadan çıkış yapabilir.[1, 3]

III. YÖNTEM

Proje, Greedy Coloring algoritması ile ders programı hazırlama yöntemini benimsemektedir. Bu algoritma, ders programını oluştururken çeşitli kısıtları ele alarak, aynı hocanın derslerinin çakışmamasını, derslerin istenilen günlere atanmasını ve sınıf sayıları ile isimlerini dikkate alır. Algoritma, çizge renklendirme prensibini kullanarak her bir dersi bir renkle etiketler ve çakışan dersler farklı renklere

atanır. Bu şekilde, çakışma olmayan bir ders programı elde edilir.

A. Greedy Coloring Algoritması

Greedy Coloring algoritması, bir çizgenin düğümlerini (vertex) belirli bir sayıda renkle boyamak için kullanılan bir yaklaşımdır. Temel prensibi, bir düğümü boyamak için en düşük kullanılmış renk numarasını seçmektir.

Aşağıdaki proje kodumuzda, Graph kütüphanesi kullanılarak bir çizge oluşturduk, bu çizgeye düğümler ve bağlantılar eklendi ve Greedy Coloring algoritması ile ders programının renklendirilmiş bir versiyonu elde ettik. Bu algoritmanın çıktıları, her bir dersin hangi renkle etiketlendiğini göstermektedir.

```
public static void main() {
    Graph<String, DefaultEdge> graph = new SimpleGraph<>(DefaultEdge.class);
    graph.addVertex(v: "A"); //ders
    graph.addVertex(v: "B"); //hoca
    graph.addVertex(v: "C"); //sinif
    graph.addVertex(v: "D"); //gun
    graph.addVertex(v: "E"); //saat
    graph.addEdge(sourceVertex: "A", targetVertex: "B");
    graph.addEdge(sourceVertex: "A", targetVertex: "C");
    graph.addEdge(sourceVertex: "A", targetVertex: "D");
    graph.addEdge(sourceVertex: "B", targetVertex: "C");
    graph.addEdge(sourceVertex: "B", targetVertex: "E");
    graph.addEdge(sourceVertex: "D", targetVertex: "E");

    GreedyColoring<String, DefaultEdge> coloring = new GreedyColoring<>(graph);

    Map<String, Integer> colorMap = coloring.getColoring().getColors();

    for (String vertex : graph.vertexSet()) {
        System.out.println("Vertex " + vertex + " has color " + colorMap.get(vertex));
    }
}
```

[Şekil 2]: Proje Akış Şeması görüntüsü.

IV. ÖRNEK SONUÇLAR

Projenin uygulanması sonucunda, Greedy Coloring algoritması kullanılarak oluşturulan ders programları, çeşitli kısıtları gözeterek başarıyla hazırlandı. Aynı hocanın dersleri çakışmamakta, dersler belirlenen günlere atanmakta ve sınıf sayıları ile isimleri dikkate alınarak optimize edilmiş bir planlama elde edilmektedir. Bu, ders programının daha düzenli ve verimli olmasını sağlamaktadır. Ayrıca, algoritmanın performansı incelendiğinde, çakışma olasılığını minimize etme konusundaki etkinliği gözlemlenmiştir.

A. Greedy Coloring Algoritmasının Sonuçları

Her bir düğümün (vertex) renk ataması, belirli bir gün ve saat kombinasyonunu temsil eder. Çakışmayan renkler, çakışmayan derslerin olduğunu ve algoritmanın başarıyla çalıştığını gösterir.

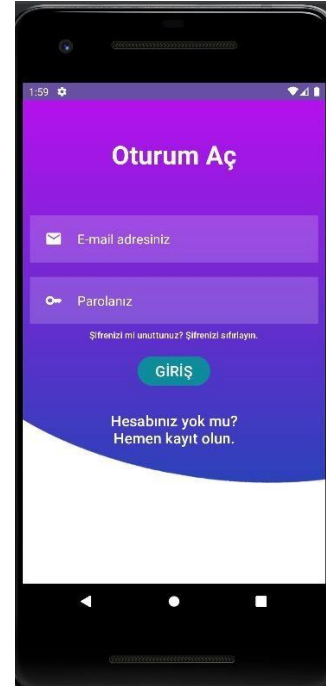
Örneğin, "vertex A has color 0" ifadesi, "A" dersinin renklendirilmiş olduğunu ve bu rengin gün ve saat kombinasyonunu belirttiğini ifade eder. "B" ve "D" dersleri farklı renklendirilmiş olduğu için aynı hocanın derslerinin çakışmadığı anlaşılır. Ayrıca, "C" dersi ve "E" dersi farklı renklerle etiketlenmiştir, bu da sınıf sayıları ile isimlerinin dikkate alındığını ve optimize edilmiş bir planlama elde edildiğini gösterir.

```
I Vertex A has color 0
I Vertex B has color 1
I Vertex C has color 2
I Vertex D has color 1
I Vertex E has color 0
```

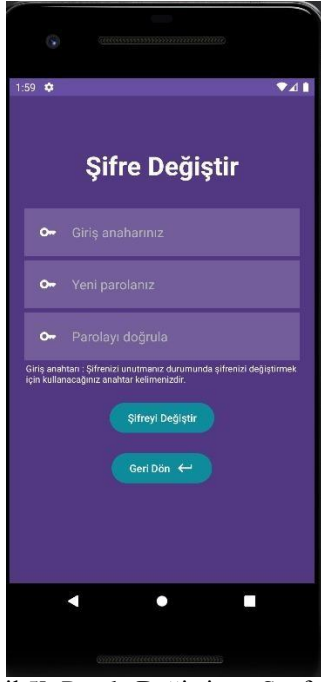
[Şekil 3]: Proje Akış Şeması görüntüsü.

Bu çıktılar, ders programının düzenli ve verimli olmasını sağlarken, algoritmanın çakışma olasılığını minimize etme konusundaki etkinliğini vurgular.

V. EKLENTİLER



[Şekil 4]: Öğretim Görevlisi Girişi Sayfasının Ekran Görüntüsü. [1,2,4 Numaralı kaynaklardan Yararlanılmıştır.]



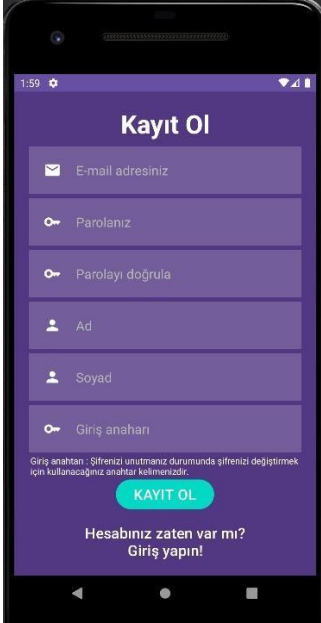
[Şekil 5]: Parola Değiştirme Sayfasının Ekran Görüntüsü.

[1 ve 3 numaralı kaynaklardan yararlanılmıştır.]



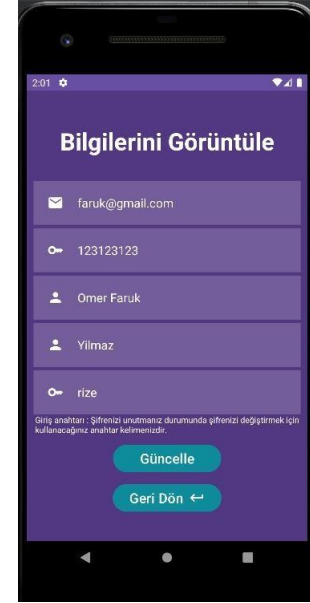
[Şekil 7]: Menü Sayfasının Ekran Görüntüsü.

[1 ve 3 numaralı kaynaklardan yararlanılmıştır.]



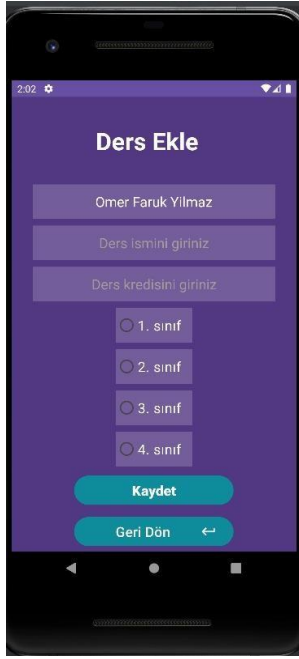
[Şekil 6]: Kayıt Ol Sayfasının Ekran Görüntüsü.

[1, 3 ve 7 numaralı kaynaklardan yararlanılmıştır.]



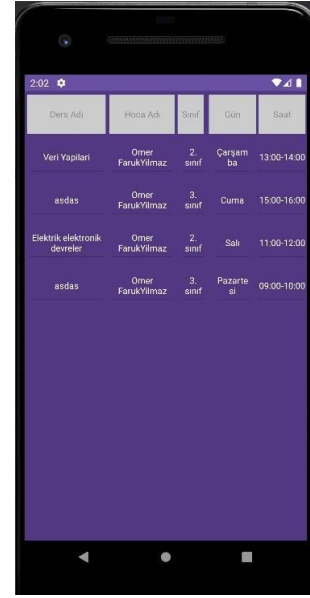
[Şekil 8]: Profil Düzenleme Sayfasının Ekran Görüntüsü.

[1, 3 ve 4 numaralı kaynaklardan yararlanılmıştır.]



[Şekil 9]: Ders Ekleme Sayfasının Ekran Görüntüsü.

[1, 3 ve 7 numaralı kaynaklardan yararlanılmıştır.]



[Şekil 11]: Ders Programı Görüntüleme Sayfasının Ekran Görüntüsü.

[1 ve 7 numaralı kaynaklardan yararlanılmıştır.]



[Şekil 10]: Ders Programı Düzenleme Sayfasının Ekran Görüntüsü.

[5,6,7 numaralı kaynaklardan yararlanılmıştır.]

KAYNAKÇALAR

- [1] YouTube Playlist-"Android Studio-Java Dersleri."Nexis.<https://youtube.com/playlist?list=PL20Zn-5nPIPHvLPq5xJTTImOd0qeNd9rW&si=3u9dLlf7NBTfB-mu>
- [2] Stack Overflow, "How to create ScrollView programmatically." [Online]. Available: <https://stackoverflow.com/questions/9887849/how-to-create-scrollview-programmatically>.
- [3] Stack Overflow, "Android Java onClick could not execute method of the activity." [Online]. Available: <https://stackoverflow.com/questions/29287152/android-java-onclick-could-not-execute-method-of-the-activity>.
- [4] Medium, "How to Make Custom Spinner Using PopupWindow in Android Studio." [Online]. Available: <https://medium.com/swlh/how-to-make-custom-spinner-using-popupwindow-in-android-studio-cc1c2971b814#:~:text=Create%20a%20new%20project%20using,display%20in%20your%20custom%20spinner>.
- [5] Stack Overflow, "Android Studio Resolving Duplicate Classes." [Online]. Available: <https://stackoverflow.com/questions/36990054/android-studio-resolving-duplicate-classes>.
- [6] Stack Overflow, "Programmatically Adding a Layout's Children." [Online]. Available: <https://stackoverflow.com/questions/55677068/programmatically-adding-a-layout-children>.
- [7] Stack Overflow, "How do I add a foreign key to an existing SQLite table?" [Online]. Available: https://stackoverflow.com/questions/1884818/how-do-i-add-a-foreign-key-to-an-existing-sqlite-table#:~:text=The%20correct%20way%20to%20add,then%20delete%20the%20Parent_Id%20column.

