

YAZILIM LABORATUVARI 2

2. PROJE

Muhammet Ali BAKINÇ
Abdulhalik SARIÇİÇEK
Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi

Özet

Bu doküman Yazılım Laboratuvarı 2 dersi 2. Projesi için çözümümü açıklamaya yönelik oluşturulmuştur. Dokümanda projenin tanımı, çözüme yönelik yapılan araştırmalar, kullanılan yöntemler, proje hazırlanırken kullanılan geliştirme ortamı ve kod bilgisi gibi programın oluşumunu açıklayan başlıklara yer verilmiştir. Doküman sonunda projemi hazırlarken kullandığım kaynaklar ve proje derlenirken dikkat edilmesi gereken hususlar bulunmaktadır.

1. Proje Tanımı

Projede bizden istenen Kocaeli'nin ilçelerinden Kocaeli Üniversitesi'ne gelen servis araçları için yolcu ve rota planlaması yapmamız beklenmektedir. Uygulamada iki farklı UI olması, bir ekranda yönetime ait işlemler, ikinci ekranda ise bir müşteri arayüzü olması istenmektedir.

Tasarlayacağımız programda kullanıcıların kayıt olabilmesi ve giriş yapabilmesi gerekmektedir.

Program işleyişi aşağıdaki gibi olmalıdır:

- Program giriş ekranıyla başlar. Kullanıcı giriş yap veya kayıt ol alanlarını kullanarak bilgilerini girmesi beklenmektedir. Veriler bulut veri tabanından kontrol edilerek admin ya da müşteri ana sayfasına geçilir.
- Admin ana sayfasında öncelikle duraklar ve bunlara kayıtlı yolcu

sayılarının kontrol edildiği sayfaya giren buton bulunur.

- Burada durak listesi ve karşılarında yolcu sayıları bulunur ve düzenlenip kaydedilebilir.
- İkinci butonda hesaplamalarda kullanılan parametreler güncellenebilmektedir.
- Rota görüntüle / oluştur butonu ile açılan sayfada sınırlı veya sınırsız servisler için rotalar hesaplanır ve gösterilir.
- Müşteri sayfasında bir başvuru ekranı ve rotanın görüntülendiği bir ekran bulunur.

2. Araştırmalar ve Yöntem

Projeye, en önemli kısım olan algoritmayı geliştirmek ile başladık.

Projede nasıl yapılması gerektiğini düşündüren en önemli nokta algoritmayı çözmekti. Bunun için çeşitli kaynaklardan dokümanlar, kodlar inceledik ve videolar izledik. Araştırmalarımız sonucu en uygun algoritmayı seçtik. Mobil geliştirme için *Flutter* ve *Dart* kullanmaya karar verdik. Tasarım esnasında zaman zaman kullanımında zorlandığım bazı araçlar için yine internette araştırmalar yaptık. Diğer bir kısım da bulut veri tabanı ve harita API'sini kullanmaktı. Bulut veri tabanında *Firestore* kullandık. Harita için de *Google Maps API*'yi kullandık.

3. Geliştirme Ortamı

Projemizi Windows sistemde, *Android Studio* üzerinde geliştirip derledik. *Dart*

dilini, *Firebase* bulut veri tabanını ve *Flutter* kütüphanelerini kullandık.

4. Kod Bilgisi

4.1. Akış Şeması

Son sayfadadır.

4.2. Algoritma

Bu kısımda projenin genel algoritmasına açıklık getireceğim.

4.2.1. Oluşturulan Sınıflar

Aşağıda belirtilen tüm model sınıflarda ortak olarak yapıcı (*constructor*) metotları, sınıflardaki tüm özellikler için *get*, *set* metotları tanımlanmıştır.

Home Kontrol sınıfı

Ekranın rotalar sayfasının kontrolünü sağlar. En üst sütunda bulunan harita rota bilgilerinin gelmesini bekler. Bir alt kısımda yüklenme ikonu algoritmanın tamamlanmasını bekler. Rota bilgileri gelince hem haritada hem de alt kısımda gösterilir. Sayfanın girişinde seçilen sınırlı veya sınırsız seçeneğine göre algoritma sınıfının ilgili metotları çağrılır.

Login Kontrol sınıfı

Programın başlangıç ekranı olan giriş yapma ekranını kontrol eder. E-posta ve şifre alanlarından alınan veriler veri tabanından sorgulanır. Sonuç doğru ise ilgili ana sayfaya yönlendirir. Kayıt Ol sayfasında da alınan bilgiler ile yeni kayıt oluşturulur. Aynı şekilde ana sayfaya yönlendirilir.

Map Servis sınıfı

Google Maps API ile bilgi akışını sağlayan metotlar bulunur. Algoritma sınıfından çağrılan metotlar aldığı bilgileri döndürür.

FirebaseCRUD sınıfı

Tüm bulut veri tabanı işlemlerinin yönetildiği sınıftır. Kullanıcı oluşturma, alma, konumları alma, yeni müşteri ekleme, silme gibi tüm işlemlerin *static* metotlar ile kullanımını sağlar.

Map Service sınıfı

Google Maps API'si ile iletişim sağlayan metotlar bulunduğu sınıftır. Gelen konum bilgileri *createMap* ile oluşturulan özel link ekranda gösterilmek üzere döndürülür. Aynı zamanda iki konum arasındaki mesafeyi hesaplayan *getDistance* metodu da API aracılığıyla işlemleri tamamlar.

Algorithm sınıfı

Rotaları hesaplayan algoritmaları kullanan metotları içerir. İlk olarak çağrılan *parametreleriAl* parametre olarak gelen konum listesini ve değişkenleri ilgili değerlere atar. Konumlar arasındaki mesafeleri hesaplayan *mesafeTabloOlustur* metodunu, ardından da *tasarrufTabloOlustur* ve *siraliTabloOlustur* metotlarını başlatır. Oluşturulan tabloların ardından sınırlı ve sınırsız servisler için rotalar oluşturan metotlardan isteneni başlatır ve arayüz sınıfına döndürür.

Model sınıfları

Veri tabanından alınan kullanıcı bilgileri için türetilen *User* sınıfı, rota bilgileri için *Rota* sınıfı, değişkenler için *Parametreler* sınıfı, konumlar için *Location* sınıfı oluşturulmuştur. *Email* ve *password*, *yolMaliyeti* veya *konumAdi* gibi birçok değişkenler bulunur. Buluttan gelen veriyi nesneye dönüştüren metotlar ve nesneyi veri tabanına yazmaya uygun hale getiren *toMap* metotları bulunur.

4.3. İstatistik

Program kodu boşluklar ve tasarım kodları dahil yaklaşık 4100 satırdan oluşmaktadır. Kullandığımız kütüphaneler ve ne için kullandığımız kabaca aşağıdaki gibidir:

http

İnternette veri almayı sağlamak için kullanılır.

cloud_firestore

Veri tabanı işlemleri için kullanıldı.

google_maps_flutter

Harita işlemleri için kullanıldı.

4.4. Programın Derlenmesi

Programın kaynak kodu 15 kontrol sınıfı, 5 model sınıfı, 11 veri tabanı ve servis

sınıfından oluşmaktadır. Bu projeyi *Android Studio* veya farklı bir Android IDE'si kullanarak derleyebilirsiniz. Derlerken dikkat edilmesi gereken mesele ise programa Flutter, Google Maps ve Firebase kütüphanelerinin eklenmiş olması gereklidir.

Kaynakça

1. Flutter kullanım rehberi,
<https://docs.flutter.dev>
2. Firebase kullanım rehberi,
<https://firebase.google.com/docs/firestore>
3. Çeşitli sorunları çözmek için,
<https://stackoverflow.com/>
4. Algoritma hakkında bilgi için,
<http://esjournal.cumhuriyet.edu.tr/tr/download/article-file/371156#:~:text=Tasarruf%20algoritması,nin%20kullanılma%20amacı%2C%20gerçek,matristen%20tasarruf%20miktarı%20değerleri%20hesaplandı.>

Akış Diyagramı



