

BOLUTAG - TAKSİ OTOMASYON SİSTEMİ

Hazırlayan: Muhammed Yusuf Gökyar **Numara:** 233405058 **Ders:** Nesne Tabanlı Programlama

PROJE GITHUB LİNKİ: <https://github.com/moskow34/BoluTag-Otomasyon>

1. PROJE NEDİR?

BoluTag, Bolu'daki taksi çağırma işini bilgisayar ortamına taşıyan bir masaüstü uygulamasıdır. Normalde taksi duraklarında telefonla yapılan işleri biz bu programla dijitalleştirdik.

Temel mantık şu: Yolcu olduğu yerden tek tıkla taksi çağırıyor, o mahalleden sorumlu olan şoförün ekranına bu çağrı düşüyor. Şoför kabul edince sistem eşleşmeyi sağlıyor. Ayrıca bir de "Admin" (Yönetici) panelimiz var, oradan da kim nereye gitmiş, sistemde kimler var hepsi kontrol edilebiliyor.

2. HANGİ TEKNOLOJİLERİ KULLANDIM?

Projeyi geliştirirken şu araçları kullandım:

- **Dil:** Java
- **Görünüm (Arayüz):** Java Swing
- **Veritabanı:** MySQL
- **Kodlama Ortamı:** Eclipse IDE

3. YAZILIM MANTIĞI (OOP KULLANIMI)

Kodları yazarken "kopyala-yapıştır" yapmaktan kaçındım ve dersin de amacı olan Nesne Yönelimli Programlama (OOP) mantığını kullandım.

- **Miras Alma (Inheritance):** Bütün kullanıcıların (Şoför, Yolcu, Admin) adı, şifresi ve TC'si ortak. O yüzden User diye bir ana sınıf oluşturdum. Diğerleri bu sınıftan türedi. Böylece aynı kodları tekrar tekrar yazmadım.
- **Kapsülleme (Encapsulation):** Şifre gibi önemli verilere dışarıdan kafasına göre kimse erişemesin diye "private" yaptım, sadece gerekli yerlerde "get/set" metodlarıyla açtım.

4. VERİTABANI YAPISI

MySQL üzerinde bolutagdb adında bir veritabanı kurdum. İki tane ana tablomuz var:

1. **Users (Kullanıcılar):** Yolcu, şoför ve adminlerin hepsi burada duruyor. Şoförler için ekstra olarak "Plaka" ve "Durak" sütunları var.
2. **Trips (Yolculuklar):** Kim, hangi şoförle, nereden nereye gitmiş bütün kayıtlar burada tutuluyor.

5. PROGRAM NELER YAPIYOR?

Giriş Sistemi: Kullanıcı TC ve Şifresini giriyor. Program arka planda bu kişinin Yolcu mu, Şoför mü olduğunu anlayıp ona göre doğru ekranı açıyor.

Yolcu Tarafı:

- Yolcu bulunduđu mahalleyi ve gideceđi yeri seçip "Şoför Çađır" diyor.
- **Önemli Özellik:** Bir yolcu, çađırdıđı taksi gelmeden veya işlem bitmeden ikinci bir taksi çađıramıyor. Böylece sistemi boş yere meşgul etmesi (spam yapması) engellenmiş oluyor.

Şoför Tarafı:

- Şoförler sadece kendi mahallelerinden gelen çađrıları görüyor.
- Gelen çađrı "BEKLIYOR" durumunda kırmızı gözüküyor. Şoför "Kabul Et" dediğinde işi üstüne alıyor.
- İş bitince "Tamamla" diyor ve yeni iş alabilir hale geliyor. Bir işi bitirmeden yenisini alamıyor.

Admin (Patron) Tarafı:

- Admin sisteme yeni şoför veya yolcu ekleyebiliyor.
- Canlı olarak o an kimin taksi beklediđini, hangi şoförün dolu olduđunu tek listede görebiliyor.

6. YAŞADIĞIM ZORLUKLAR VE ÇÖZÜMLERİ

Projeđi yaparken en çok verilerin anlık güncellenmesi kısmında zorlandım. Mesela yolcu taksi çađırınca şoförün ekranına hemen düşmesi gerekiyordu.

- **Çözüm:** Ekranlara "Yenile" butonu koydum ve Java'nın tablo yapısını (DefaultTableModel) her tıklamada veritabanından veri çekecek şekilde ayarladım.

7. SONUÇ

BoluTag projesi sayesinde hem Java'da arayüz tasarlamayı hem de veritabanı bağlamayı (JDBC) iyice pekiştirmiş oldum. Sadece kod yazmak deđil, "bir taksi sistemi gerçek hayatta nasıl çalışmalı?" diye düşünüp ona göre kısıtlamalar koymak (tek iş kuralı vb.) benim için güzel bir tecrübe oldu.