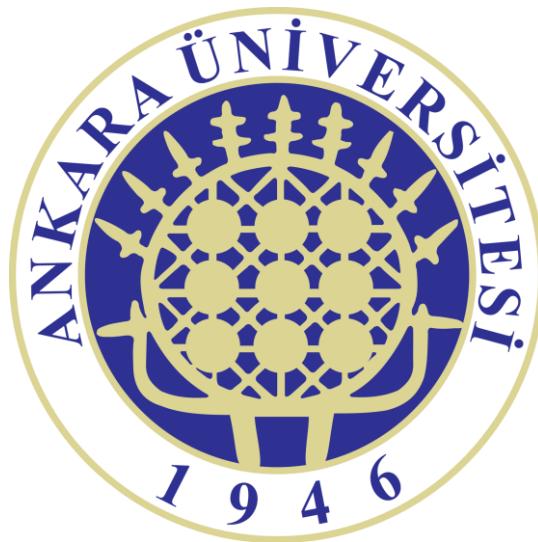


ANKARA ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BLM4537-IOS İLE MOBİL UYGULAMA
GELİŞTİRME



EZGİ SANKIR- 21290431

GitHub: <https://github.com/Ezgiis02/StuddyBuddyApp>

Video Bağlantısı: <https://youtu.be/P4YgL7PgBOU>

STUDYBUDDY PROJE TEKNİK RAPORU

1. PROJE ÖZETİ VE HEDEFLERİ

Bu proje, üniversite öğrencilerinin mobil cihazlar üzerinden her an ve her yerde ortak ders çalışma arkadaşı bulmalarını sağlayan, Flutter framework ile geliştirilmiş cross-platform bir mobil uygulamadır. Uygulama, backend tarafındaki ASP.NET Core API ile entegre çalışarak öğrencilere dinamik, hızlı ve konum tabanlı bir eşleşme deneyimi sunar.

1.1. Temel Hedefler

- Mobil Erişilebilirlik:** Öğrencilerin kampüs içinde veya dışında hızlıca çalışma arkadaşı bulmasını sağlamak.
- Anlık Etkileşim:** Etüt taleplerinin mobil arayüz üzerinden kolayca yönetilmesi (Kabul/Red).
- Veri Senkronizasyonu:** Web platformu ile tam uyumlu çalışan, merkezi bir veritabanı kullanımı.
- Modern Kullanıcı Deneyimi:** Kart tabanlı arayüz, BottomSheet kullanımı ve akıcı navigasyon.

2. VERİTABANI TASARIMI (MONGODB)

NoSQL yapısı sayesinde esnek ve hızlı bir veri modeli kurgulanmıştır.

2.1. Koleksiyon Yapısı (Collections)

- Users:** Kullanıcı temel bilgileri, konum verileri (City, District), bölüm ID'si, seçilen dersler ve arkadaş listesi (FavoriteUserIds).
- Courses:** Üniversite bünyesindeki ders havuzu (CourseCode, CourseName).
- Departments:** Bölüm bilgileri (DepartmentName, DepartmentCode).
- StudyRequests:** Kullanıcılar arasındaki iletişim talepleri (Durum: Pending, Accepted, Rejected).
- Cities:** Kayıt ve profil düzenleme ekranlarındaki şehir ve ilçe listelerini besleyen yardımcı koleksiyondur.

6. **StudyPreferences**: Kullanıcıların mekan bazlı tercihlerini ve güncellenme tarihlerini tutan koleksiyondur.

Collection name	Properties	Storage size	Documents	Avg. document size	Indexes	Total index size
cities	-	20.48 kB	10	140.00 B	1	20.48 kB
courses	-	40.96 kB	43	197.00 B	1	36.86 kB
departments	-	20.48 kB	16	66.00 B	1	20.48 kB
locations	-	36.86 kB	10	140.00 B	1	36.86 kB
studyPreferences	-	36.86 kB	3	106.00 B	1	36.86 kB
studyRequests	-	36.86 kB	17	121.00 B	1	36.86 kB
users	-	36.86 kB	3	581.00 B	1	36.86 kB

3. SİSTEMİN TEMEL MODÜLLERİ VE ALGORİTMALAR

3.1. Akıllı Eşleşme (Matching) Algoritması

Sistem, kullanıcının seçtiği ders ID'lerini kullanarak veritabanında bir kesişim kümesi sorusu çalıştırır. Kendisi olmayan ve mevcut arkadaş listesinde bulunmayan ortak ders sahibi kullanıcıları "Öneri" olarak dinamik kartlar halinde listeler.

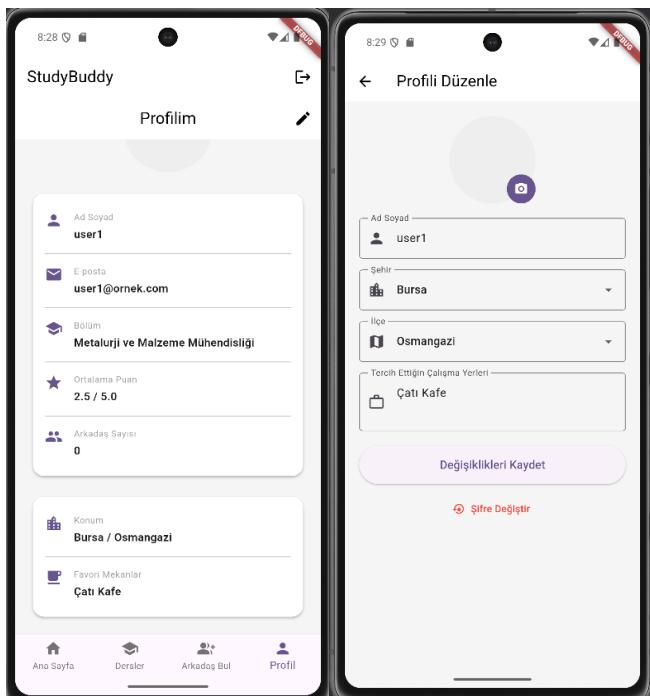
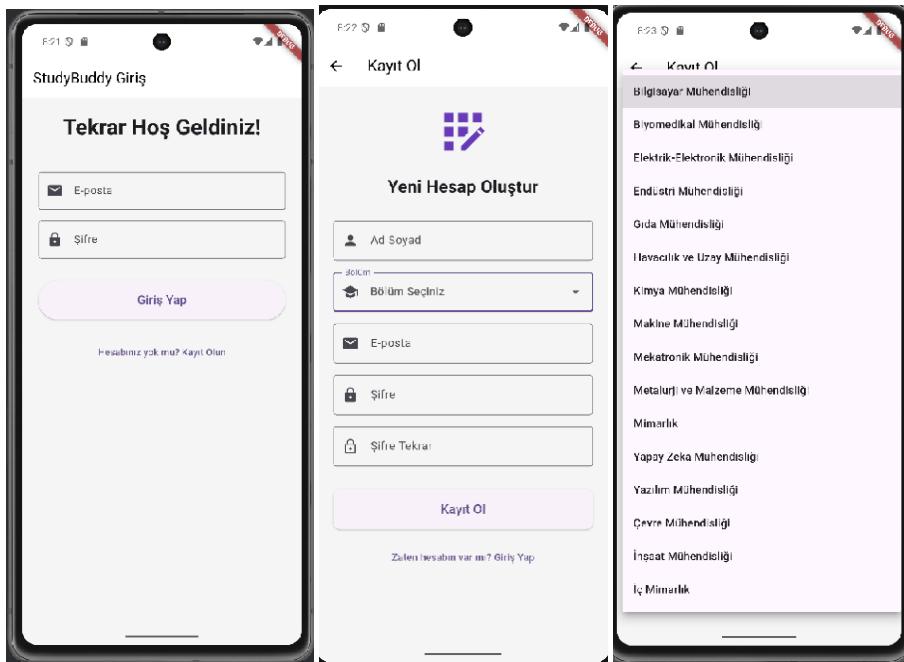
3.2. Dinamik Eşleşme ve Filtreleme Mantığı

MatchingScreen, API'den gelen ham JSON verisini Dart Map yapılarına dönüştürür. Ortak ders sayısı ve konum bilgisi gibi veriler, özel widget'lar aracılığıyla dinamik olarak render edilir.

4. UYGULAMA EKRANLARI VE FONKSİYONELLİK

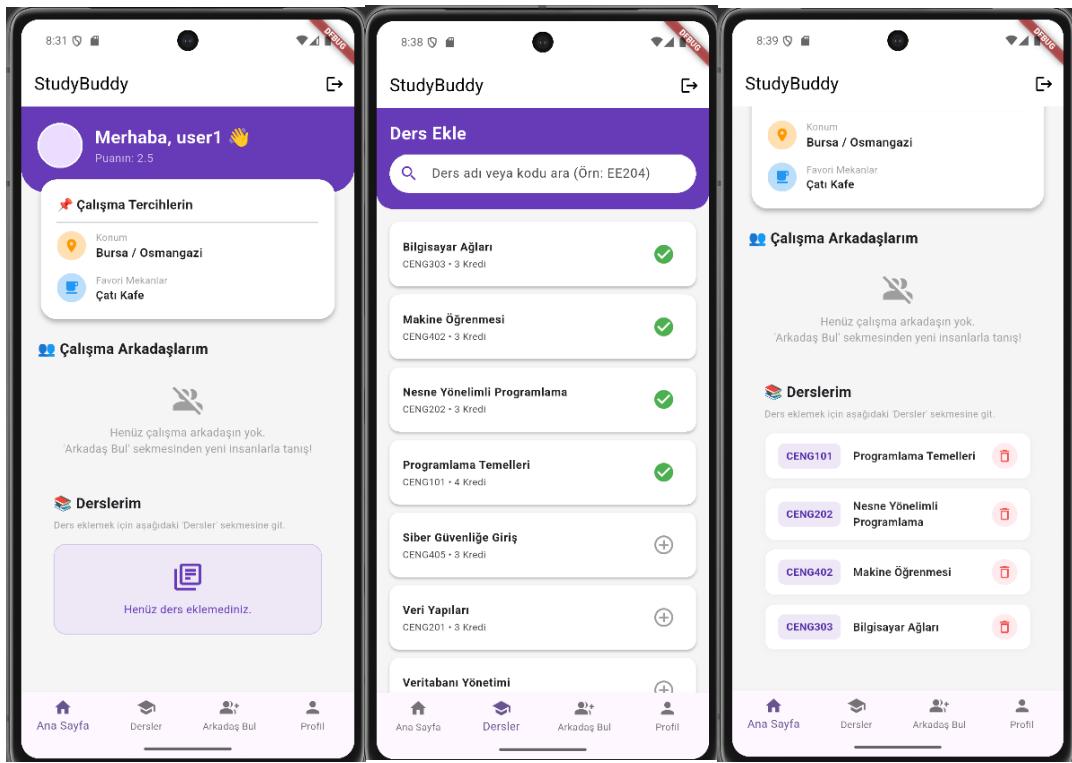
4.1. Hesap ve Profil Yönetimi

Kullanıcılar kayıt olurken departments koleksiyonundan çekilen verilerle bölüm seçimi yaparlar. Profil düzenleme sayfasında Şehir, İlçe Tercih Edilen Mekan bilgileri güncellenebilir. Şifre değiştirilebilir.



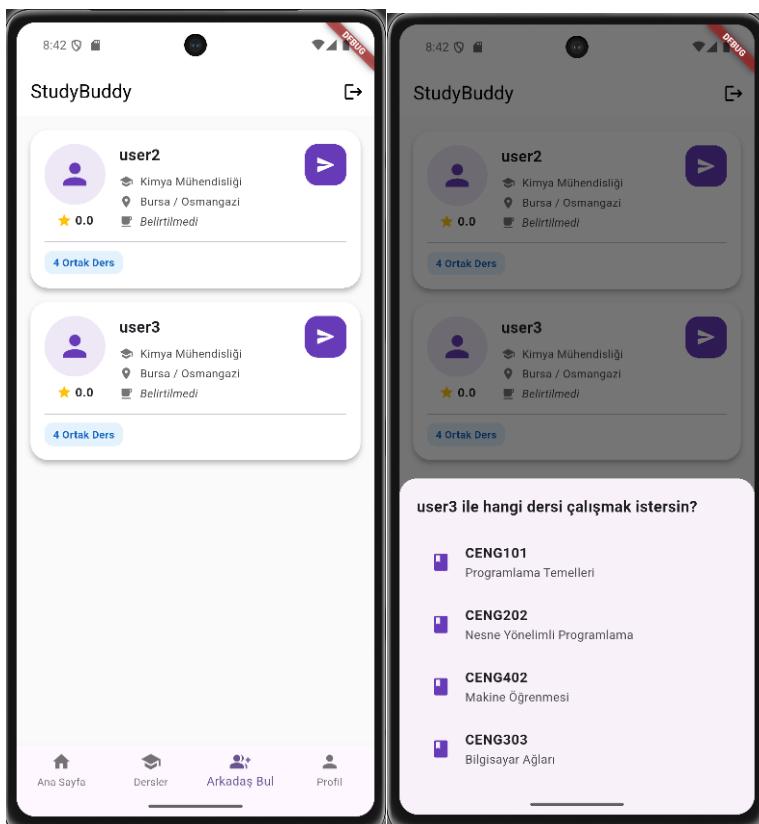
4.2. Ders Seçimi ve Profil Düzenleme

- Öğrenciler profil sayfalarından veritabanındaki ders havuzunu tarayarak Dersler kendi derslerini ekleyebilir veya silebilirler.



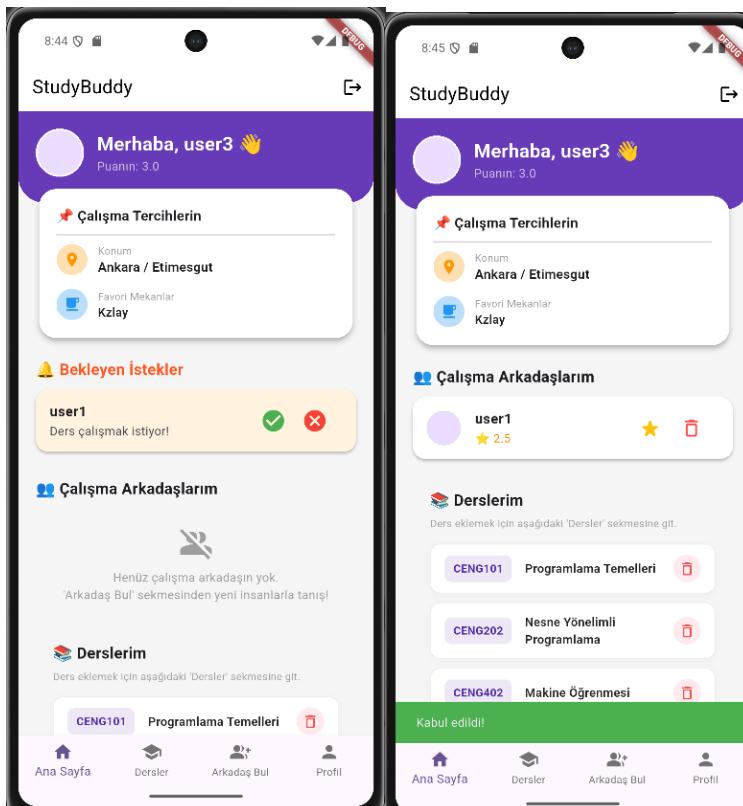
4.3. Dashboard ve Eşleşme Kartları

Kullanıcı giriş yaptığında, Arkadaş Bul sekmesinde kendisiyle en az bir ortak dersi olan öğrencileri "Kart" yapısında görür. Bu kartlarda; isim, şehir, ilçe, bölüm ve puan bilgileri yer alır. Öğrenciye seçtiği ders için talep gönderir.



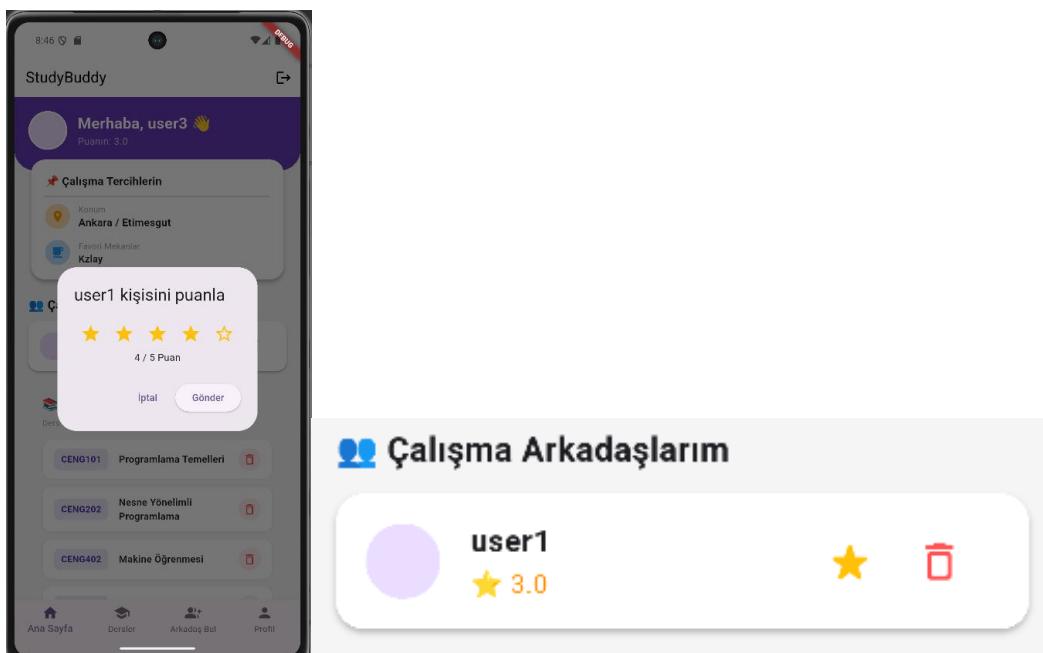
4.4. Gelen Kutusu ve Talep Yönetimi

Kullanıcılar kendilerine gelen çalışma taleplerini "Bekleyen İstekler" panelinde görürler. Bir talep kabul edildiğinde, sistem otomatik olarak her iki kullanıcıyı da birbirinin arkadaş listesine ekler.



4.5. Arkadaşlarım ve Puanlama Sistemi

Kullanıcılar, birlikte çalıştıkları arkadaşlarını 1-5 yıldız arasında puanlayabilirler. Bu işlem sonucunda karşı tarafın profilindeki puan anlık olarak güncellenir.



5. GENİŞLETİLMİŞ API DÖKÜMANTASYONU

Projede Etüt Arkadaşım Web Projesinde tasarlanmış kapsamlı bir REST API kullanılmıştır.

5.1. Önemli API Endpointleri

- POST /api/request/send/{receiver_id}: Etüt talebi gönderir.
- GET /api/departments: Tüm bölümleri dinamik olarak döndürür.
- DELETE /api/request/remove-friend/{id}: Arkadaşlığı sonlandırır.

Auth	
POST	/api/Auth/login
POST	/api/Auth/register
Courses	
GET	/api/Courses/getall
GET	/api/Courses/getmycourses
POST	/api/Courses/add/{courseId}
DELETE	/api/Courses/remove/{courseId}
Departments	
GET	/api/Departments
GET	/api/Departments/{id}
GET	/api/Departments/seed
DiscoverApi	
GET	/api/discover/search
Locations	
GET	/api/Locations/districts/{cityName}
Rating	
POST	/api/rating/rate
StudyRequest	
POST	/api/request/send/{receiverId}/{courseId}
GET	/api/request/pending
POST	/api/request/accept/{requestId}
POST	/api/request/reject/{requestId}
GET	/api/request/myfriends
DELETE	/api/request/removefriend/{friendId}
DELETE	/api/request/endmatch/{friendId}
UserApi	
GET	/api/user/profile

6. PROJE TAKVİMİ VE HAFTALIK GELİŞİM

Hafta Yapılan Çalışmalar

- 1-2 Flutter projesinin kurulumu, navigasyon yapısı ve Auth (Login/Register) modülleri.
- 3-4 ApiService entegrasyonu, ders seçimi ve listeleme ekranlarının tamamlanması.
- 5 MatchingScreen (Arkadaş Bul) tasarımı ve dinamik kart yapısının kurulması.
- 6 Talep gönderme/alma mantığı ve BottomSheet ile ders seçimi entegrasyonu.
- 7 DepartmentService ve CityService ile verilerin dinamikleştirilmesi; Puanlama sistemi.
- 8 Arkadaş silme özelliği, hata ayıklama (Debug) ve performans optimizasyonları.

7. SONUÇ

StudyBuddy Flutter uygulaması, öğrencilerin akademik iş birliğini kolaylaştıran, hızlı ve kullanıcı dostu bir platform olarak tamamlanmıştır. Backend ile kurulan asenkron yapı ve NoSQL mimarisi yapısı sayesinde genişletilebilir bir çözüm sunulmuştur.