

Reishi Projesi – Genel Bakış ve Yapı

Ekranlar (lib/presentation/screens)

Dosya Adı	Açıklama
splash_screen.dart	Kullanıcı karşılama ekranı
onboarding_screen.dart	Giriş yap / Kayıt ol sayfası
profile_onboarding_screen.dart	Kullanıcıdan ad, yaş, cinsiyet ve ADHD bilgisi alınır.
mood_selection_screen.dart	Kullanıcının duygu durumunu seçebileceği yüzler ekranı (depressed, meh, amazing)
daily_question_screen.dart	AI destekli sohbet ve görev yardımı ekranı
suggest_breakdown_screen.dart	AI ile incelenen mesajların görevlere ayrıldığı ekran
main_menu.dart	Günlük görevlerin genel görünümünü sağlayan, dinamik ilerleme ikonu içeren ana ekran
chat_screen.dart	AI destekli sohbet ekranı
task_list_screen.dart	Seçilen görevleri listeleme, tamamlama ve takip ekranı.
profile_screen.dart	Profil, rozetler, tamamlanan görev sayısı, daily streak, çıkış yapma ekranı

Hizmetler (lib/services)

Servis Adı	Açıklama
ai_service.dart	Gemini API
auth_service.dart	Firestore Authentication: Giriş, çıkış ve oturum yönetimi.
chat_example.dart	Gemini destekli diyalog altyapısı.
chat_service.dart	Gemini destekli diyalog altyapısı.
gamification_service.dart	Tamamlanan görevler, daily streak, rozetler
notification_service.dart	FCM ile kullanıcıya bildirim gönderme altyapısı.
task_service.dart	Görevleri veritabanından çekme, güncelleme ve silme işlemleri.

Veri Modelleri (lib/data/models)

- Firebase Realtime Database ile iletişimde kullanılan modeller burada yer alır.
- Her model toJson() ve fromJson() metodları içerir.
- Verilerin yapısal ve tür güvenli bir şekilde yönetilmesini sağlar.

Teknoloji Yığını (Tech Stack)

Katman	Teknoloji/Araçlar	Açıklama
Mobil	Flutter	Şu anlık sadece Android desteği. İleride birçok alana hızlıca entegre edebilmek için bu teknoloji tercih edildi.
Backend	Firebase Functions	Mobil uyumlu API yönetimi
Veri Tabanı	Firebase Realtime DB	Gerçek zamanlı veri eşleşmesi ve kolay yapılandırma
GenAI	Gemini API	Metin analizi, duygu sınıflandırması, görev parçalama
Bildirim Sistemi	Firebase Cloud Messaging (FCM)	Kişiselleştirilmiş bildirimler için mobil uyumlu altyapı
Kullanıcı Oturumu / Authorization	Firebase Authentication	Giriş sistemi (e-posta ve Google ile giriş)
Veri Analizi	Jupyter Notebook	ADHD hakkında toplanan gerçek/sentetik verilerin analizi için kullanıldı.

Achievement Sistemi

Achievement Adı	Açıklama	Kazanma Koşulu
Headstart	İlk görevi tamamladın!	completedTaskCount == 1 && isFirstTimeUser
Mushroom Madness	3 gün üst üste tüm görevleri tamamlama.	streak == 3 && streakCompletionRate
Third Time's the Charm	Bir günde tüm görevleri (3) tamamladın.	(int todayCount) => todayCount == 3
Early Bird	Sabah saatlerinde görev tamamladın.	Görev 06:00–12:00 arası tamamlandıysa
Tenacious Ten	10 görev başarıyla tamamladın.	completedTaskCount >= 10

GELECEK ADIMLAR

- Profil kısmında gelişimimizi görebileceğimiz bir duygusal gelişim grafiği eklenecektir.
- Speech-To-Text özelliği entegre edilecek. (İlk testler yapılmış olsa da, verimliliği artırılarak daha stabil bir versiyonu sisteme dahil edilecektir.)
- Gelecekte ChromaDB ile kullanıcının geçmiş verileriyle ilişkilendirme sunulması hedeflenmektedir..
- Türkçe dil desteği eklenecektir.
- Fine-tune edilmiş Gemini modelinin entegrasyonu sağlanacak.
- UI/UX kullanıcıdan gelen geri bildirimlere göre yeniden düzenlenebilir.

KULLANIM SENARYOSU

1. Uygulama başlatıldığında kullanıcı **splash_screen.dart** ekranını görür.
2. Uygulama, kullanıcı girişi yaptıysa hesap durumuna göre şu adımları izler:
 - Günlük ruh hali girişi yapılmamışsa → **mood_selection_screen.dart**
 - Ruh hali seçilmişse → **daily_question_screen.dart** (AI ile sohbet)
 - Daha önce görevler oluşturulmuşsa → **main_menu.dart**
3. **daily_question_screen.dart**'ta Kullanıcıdan şu soruya doğal bir yanıt istenir:
"What do you want to accomplish today?"
Örnek Yanıt:
"Bugün sunum hazırlamam, e-postaları kontrol etmem ve kitap okumam gerekiyor."
4. Kullanıcının girdiği metin, **ai_service.dart** tarafından **Gemini API**'ye gönderilir.
5. Gemini API bu metni analiz ederek görevleri küçük adımlara ayırır.
Örnek dönüş:
 - a. Sunum için konu araştırması yap (high)
 - b. Slaytları hazırla (medium)
 - c. Prova yap (low)
6. Kullanıcı bu görevleri **suggest_breakdown_screen.dart** ekranında görüntüler ve oluşan görevlere göre öncelik durumuna göre görevlerini seçip kaydeder.
7. Onaylanan görevler, **task_service.dart** aracılığıyla **Firestore Realtime DB**'ye kaydedilir.
8. Kullanıcı **main_menu.dart** ve ardından **task_list_screen.dart** ekranında bu görevleri görebilir.
9. Gün içinde kullanıcı görevleri tamamladıkça üzerlerine tıklar ve tamamlandıklarının üzeri çizilir.
10. Her tamamlanan görev, **gamification_service.dart** tarafından takip edilir.
 - Başarı takibi yapılır.
 - Günlük streak güncellenir.
 - Belirli şartlar/başarılar için rozetler kazanılır.
11. Kullanıcı profilde gelişimini ve başarılarını görebilir.