

To-Do List Uygulaması Dökümantasyonu

1. Proje Genel Bakışı

Projenin Amacı

Bu projenin amacı, Flutter dilini kullanarak geliştirilecek bir mobil uygulama ile kullanıcıların günlük görevlerini ve aktivitelerini yönetmelerini sağlamaktır. Ayrıca, takvim özellikleri eklenerek kullanıcıların planlarını organize etmeleri kolaylaştırılacak ve hava durumu bildirimleri ile dışarı aktiviteleri için ek bir özellik sunulacaktır.

Kullanılan Teknolojiler

- Flutter Framework: Kullanıcı arayüzü ve uygulama mantığını geliştirmek için kullanılacaktır.
- Firebase (Opsiyonel): Kullanıcı verilerini saklamak ve gerçek zamanlı senkronizasyon sağlamak için tercih edilebilir.
- Hava Durumu API'leri: Hava durumu bilgilerini almak için kullanılacaktır.

Kullanıcı Hedef Kitlesi

Bu uygulama, günlük işlerini ve planlarını organize etmek isteyen geniş bir kullanıcı kitlesine hitap edecektir. Özellikle, iş ve özel yaşam arasında denge sağlamak isteyen kişilerin kullanımını hedefler.

2. Proje Tanımı

Uygulama Adı ve Amaçları

Uygulama adı: "To-Do List App"

Amaçları:

- Günlük görevlerin yönetimi için to-do list fonksiyonu sunmak.
- Takvim özelliği ile planları organize etmeyi sağlamak.
- Kullanıcının aktivite türünü seçmesine olanak tanıyarak hava durumu bildirimleri sağlamak.

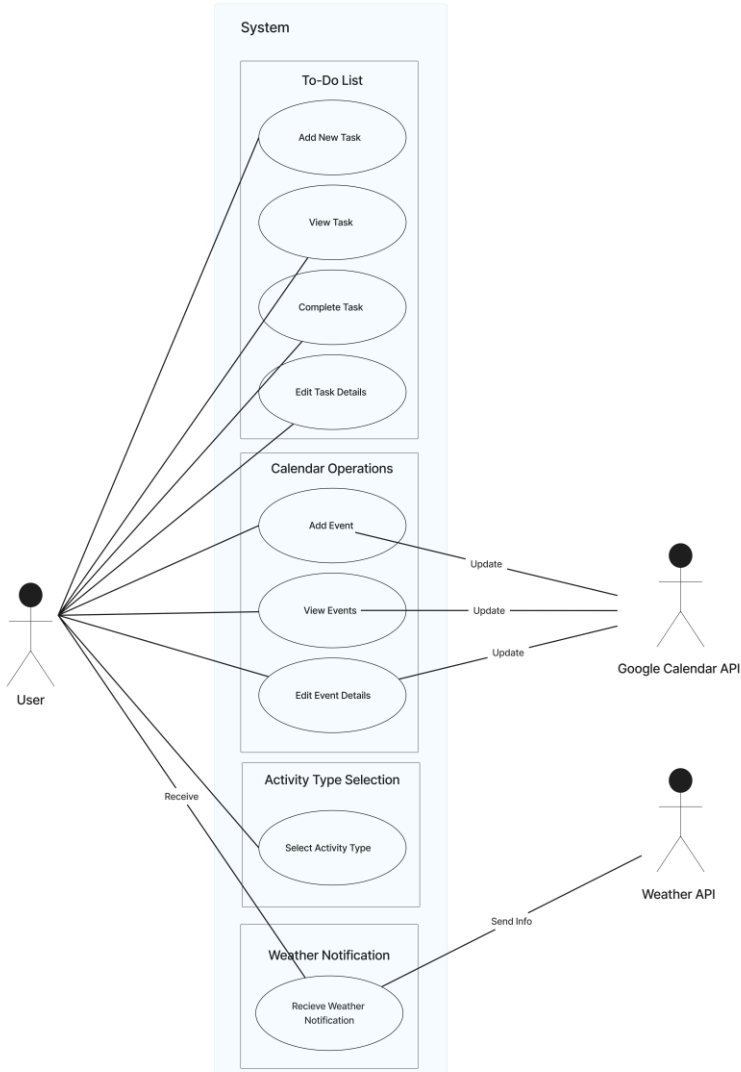
Özellikler

- To-Do List Fonksiyonu
- Takvim Özelliği
- Aktivite Türü Seçimi
- Hava Durumu Bildirimi

Kullanıcı Senaryoları

- Bir kullanıcı günlük görevlerini eklemek ve takvimde planlarını düzenlemek istiyor.
- Kullanıcı bir aktivite eklerken iç mekan mı dış mekan mı olduğunu seçmek istiyor ve eğer dış mekan aktivitesi ise hava durumu ile ilgili bildirim almak istiyor.

Proje Tanımı İçin Use Case Diyagramı



3. Kullanılan Teknolojiler

Flutter Framework

Flutter, Google tarafından geliştirilen ve hem iOS hem de Android için tek bir kod tabanıyla mobil uygulama geliştirmeyi sağlayan açık kaynaklı bir SDK'dir.

Firebase (Opsiyonel)

Firebase, mobil ve web uygulamaları için arka uç hizmetleri sağlayan bir platformdur. Kullanıcı kimlik doğrulaması, veritabanı, depolama ve gerçek zamanlı veritabanı gibi hizmetler sunar.

Google Calendar API (Opsiyonel)

Google Calendar API, kullanıcıların Google Takvim verilerine erişmelerini ve bu verileri uygulamalarında kullanmalarını sağlayan bir API'dir. Bu API, takvim özelliğini zenginleştirmek ve kullanıcıların Google Takvimlerini uygulamalarında yönetmelerini sağlamak için kullanılabilir.

Hava Durumu API'leri

Hava durumu API'leri, belirli bir konumun mevcut hava durumu verilerini sağlayan çevrimiçi hizmetlerdir.

4. Geliştirme Süreci

Proje Yapısı

Uygulama, genel olarak MVC (Model-View-Controller) veya benzeri bir mimariye dayalı olarak organize edilecektir. Bu, uygulamanın parçalarının ayrı tutularak geliştirilmesini ve bakımını kolaylaştıracaktır. Her bir özellik, uygun bir şekilde modüller halinde ayrılacak ve düzenli bir kodlama pratiği izlenecektir.

Veritabanı Yapısı (Opsiyonel)

Firebase kullanılıyorsa, Firebase Realtime Database veya Firebase Firestore gibi veritabanı çözümleri kullanılabilir. Veritabanı şeması, uygulamanın gereksinimlerine uygun olarak tasarlanacak ve kullanıcı verilerinin güvenli ve verimli bir şekilde saklanması sağlanacaktır.

UI/UX Tasarımı

Uygulamanın kullanıcı arayüzü, modern ve kullanıcı dostu bir tasarıma sahip olacaktır. Malzeme Tasarımı veya Cupertino stilinin Flutter widget'ları kullanılarak hem Android hem de iOS platformlarına uygun bir görünüm elde edilecektir. Kullanıcı deneyimi (UX), kolay kullanılabilirlik ve sezgisel gezinme üzerine odaklanacak şekilde optimize edilecektir.

Kodlama Standartları

Kodlama standartları, Flutter için tavsiye edilen pratiklere uygun olacaktır. Dart dili için standart kodlama kuralları ve Flutter için widget isimlendirme ve düzenleme standartları gibi konular üzerinde titizlikle durulacaktır.

5. Uygulama Akışı

Giriş Ekranı

Kullanıcının giriş yapabileceği veya kayıt olabileceği bir ekran olacak. Firebase kullanılıyorsa, Firebase Authentication entegrasyonu sağlanacaktır.

Ana Ekran

Ana ekran, kullanıcının mevcut görevleri ve takvim öğelerini görebileceği bir yer olacaktır. Görevler listesi ve takvim, kullanıcıların günlük planlarını görsel olarak takip etmelerini sağlayacaktır.

To-Do Ekleme Ekranı

Yeni görevlerin veya aktivitelerin eklenmesine izin veren bir form olacak. Kullanıcılar burada görevin adını, tarihini, saatini ve belki de açıklamasını girebilecekler.

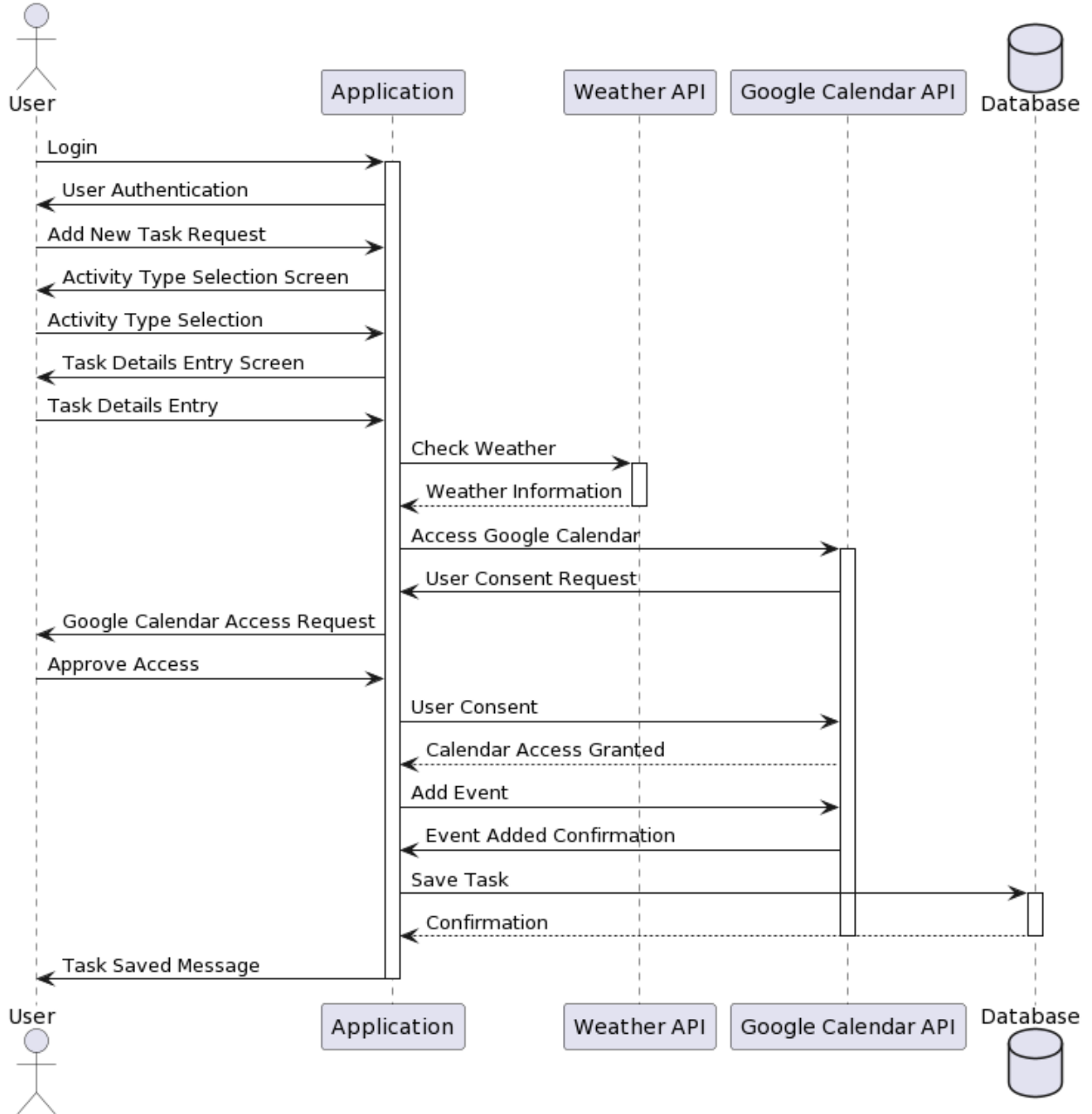
Aktivite Detay Ekranı

Kullanıcının bir aktiviteyi detaylı olarak görüntüleyebileceği bir ekran olacak. Bu ekran, aktivitenin ayrıntılarını, tarihini, saati ve diğer önemli bilgileri içerecek.

Ayarlar Ekranı

Kullanıcının uygulama ayarlarını yapılandırabileceği bir ekran olacak. Örneğin, bildirim tercihleri, tema seçimi gibi ayarlar burada bulunabilir.

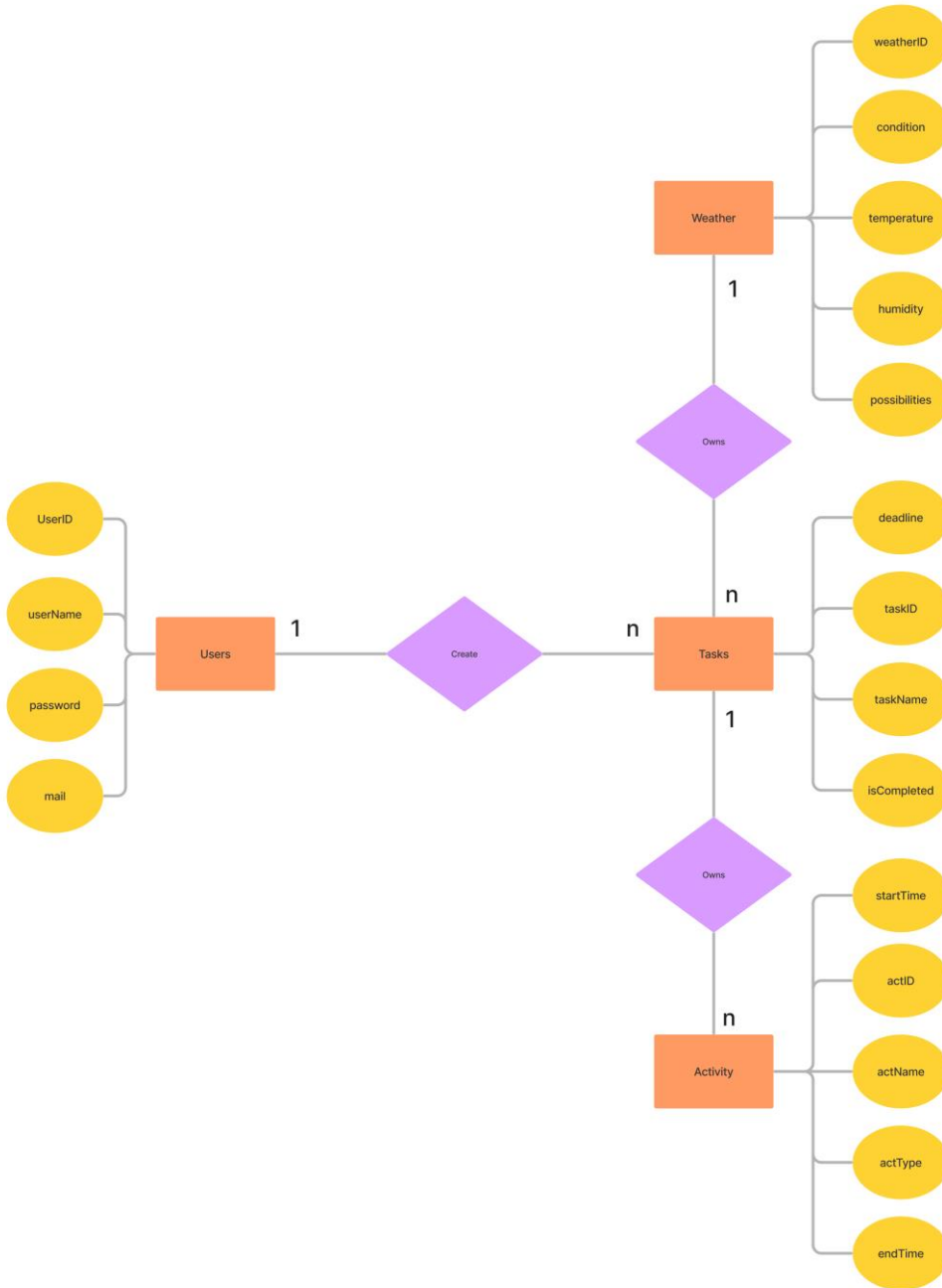
Uygulamanın Akışı İçin Hazırlanmış Sequence Diyagramı



6. Veritabanı Tasarımı (Opsiyonel)

Tablolar ve İlişkiler

Firebase Realtime Database veya Firebase Firestore kullanılarak veritabanı yönetilecek. Veritabanı ER diyagramı ile şöyle gösterilebilir :



7. Uygulama Güvenliđi

Kimlik Doğrulama

Firebase Authentication kullanılarak kullanıcı kimlik doğrulaması sağlanacaktır. Kullanıcıların giriş yapması veya kaydolması için e-posta/şifre, Google, Facebook gibi kimlik doğrulama yöntemleri sunulabilir.

Veri Güvenliđi

Firebase Security Rules kullanılarak veri tabanı güvenliđi sağlanacaktır. Sadece yetkilendirilmiş kullanıcılar belirli verilere erişebilecek ve bu veriler üzerinde değışiklik yapabilecektir.

8. Hata Takibi ve Test Süreci

Hata Raporlama Aracı

Firebase Crashlytics, uygulamanın çökme ve hata raporlarını toplamak ve analiz etmek için kullanılacaktır. Bu, uygulamanın kararlılıđını artırmak ve performansını optimize etmek için önemli bir araçtır.

Birim Testleri

Flutter'in birim test çerçeveleri kullanılarak uygulamanın farklı bileşenlerinin (fonksiyonlar, widget'lar vb.) doğru çalıştığı doğrulanacaktır. Bu, kod kalitesini artırmak ve uygulamanın beklenen şekilde çalışmasını sağlamak için önemlidir.

9. Dağıtım ve Yayınlama

Uygulama Mağazalarına Gönderme Süreci

Uygulama, Flutter'un sağladığı derleme ve yayınlama araçları kullanılarak iOS için App Store'a ve Android için Google Play'e gönderilecektir. Her platform için gereken adımların tamamlanması ve platform spesifik gereksinimlerin karşılanması sağlanacaktır.

Güncelleme Stratejisi

Uygulama güncellemeleri, kullanıcı geri bildirimleri, performans iyileştirmeleri ve yeni özellikler gibi faktörlere dayanarak düzenli olarak sunulacaktır. Kullanıcı deneyimini geliştirmek ve uygulamanın rekabet gücünü korumak için güncellemelerin düzenli olarak sağlanması önemlidir.

10. Gelecek Geliştirmeler

Ek Özellikler

- Kullanıcıların birbirleriyle görevleri paylaşmasını sağlayacak sosyal özellikler eklemek.
- Görevler arasında öncelik seviyeleri belirlemek için etiketleme özelliği eklemek.
- Kullanıcıların aktiviteleri için tekrarlayan hatırlatıcılar eklemek.

İyileştirme Fikirleri

- Kullanıcı arayüzünü daha da geliştirmek ve kullanıcı dostu bir deneyim sunmak için kullanıcı testleri yapmak.
- Uygulamanın performansını ve hızını artırmak için kod optimizasyonları yapmak.
- Kullanıcı geri bildirimlerini dikkate alarak mevcut özellikleri iyileştirmek ve hata gidermek.

11. İletişim Bilgileri

Proje Sahibi

İsim: Turhan Göksu

E-posta: turhangoksu3@gmail.com

Destek ve Geri Bildirim

İsim: Turhan Göksu

E-posta: turhangoksu3@gmail.com

KAYNAKLAR

- <https://www.figma.com/files/recents-and-sharing/recently-viewed?fuid=1346926376440796951>
- <https://www.wikipedia.org/>
- <https://plantuml.com/>
- <https://medium.com/@ckn.deesit/designing-a-to-do-list-application-ffcbe80f2fdf>
- <https://chat.openai.com/>