**AR PROJECT - BİTİRME RAPORU**

## ****İÇİNDEKİLER****

1. **GİRİŞ**
   * 1.1. Giriş
   * 1.2. Amaç
   * 1.3. Kapsam
   * 1.4. Tanımlar ve Kısaltmalar
2. **GENEL TANIM**
   * 2.1. Ürün Perspektifi
   * 2.2. Kullanıcı Özellikleri
   * 2.3. Kısıtlamalar
3. **GEREKSİNİMLER**
   * 3.1. Yazılım Gereksinimleri
   * 3.2. Donanım Gereksinimleri
4. **PROJE YAPISI**
   * 4.1. Klasör ve Dosya Yapısı
   * 4.2. Klasörlerin Açıklaması
   * 4.3. Proje Yapısının Avantajları
5. **UYGULAMA MODÜLLERİ**
   * 5.1. Ana Modüller
6. **TEST SÜREÇLERİ**
   * 6.1. Test Araçları
   * 6.2. Uygulanan Testler
7. **SONUÇ VE DEĞERLENDİRME**
   * 7.1. Gelecekte Yapılabilecek Geliştirmeler
   * 7.2. Sonuç

**1. GİRİŞ**

Bu rapor, AR Project adlı artırılmış gerçeklik (AR) tabanlı mobil uygulamanın geliştirme sürecini, kullanılan teknolojileri ve elde edilen sonuçları ayrıntılı bir şekilde açıklamaktadır. React Native ve Expo kullanılarak geliştirilen bu proje, mobil platformlarda AR deneyimi sağlamak ve kullanıcıların artırılmış gerçeklik teknolojisinden maksimum fayda elde etmelerini amaçlamaktadır.

Proje, modern mobil yazılım geliştirme yöntemleri kullanılarak hazırlanmış olup, modüler bir yapıya sahiptir. Ana bileşenler, App.js ve index.js dosyalarında tanımlanmış olup, uygulamanın farklı işlevleri ise components klasöründe yer alan bağımsız bileşenler aracılığıyla sağlanmaktadır. assets klasörü, uygulamanın simgeleri, splash ekranı ve diğer grafik ögelerini içermektedir. Uygulamanın konfigürasyon ve bağımlılık yönetimi ise app.json, babel.config.js ve metro.config.js dosyaları üzerinden gerçekleştirilmektedir.

Bu rapor kapsamında, projenin geliştirilme aşamaları, kod yapısı, modüllerin işlevleri, karşılaşılan zorluklar, test süreçleri ve gelecekte yapılması planlanan iyileştirmeler ayrıntılı bir şekilde ele alınmıştır. Kurulum ve kullanım talimatları, uygulamanın çalıştırılması ve hata giderme yöntemleri de raporda detaylandırılmıştır. Projenin sürdürülebilirliği için bakım süreçleri tanımlanmış, uygulamanın farklı cihazlarda nasıl optimize edileceği üzerinde durulmuştur. Ayrıca, test yöntemleri ve performans analizleri, sistemin güvenilirliğini ve doğruluğunu sağlamak amacıyla rapor içerisinde değerlendirilmiştir.

Bu proje, artırılmış gerçeklik teknolojisinin mobil uygulamalar ile nasıl entegre edilebileceğini araştırmak ve geliştiricilere rehber niteliğinde bir belge sunmak amacıyla hazırlanmıştır. Projenin başarılı bir şekilde tamamlanması, AR destekli mobil uygulamalar geliştirmek için Expo ve React Native'in verimli bir şekilde kullanılabileceğini göstermektedir.

**2. GENEL TANIM**

**2.1. Ürün Perspektifi**

Bu proje, artırılmış gerçeklik (AR) teknolojisini mobil platformlar üzerinden erişilebilir hale getirmek ve kullanıcıların gerçek dünya ile sanal dünyayı birleştirerek etkileşimli bir deneyim yaşamasını sağlamak amacıyla geliştirilmiştir. Kullanıcılar, AR destekli mobil cihazlar aracılığıyla sanal nesnelerle etkileşim kurabilir ve artırılmış gerçeklik deneyimini gerçek dünya ile entegre bir şekilde keşfedebilirler.

Proje, React Native ve Expo çerçeveleri kullanılarak geliştirilmiş olup, hem Android hem de iOS platformlarında çalışabilen çapraz platform bir uygulama sunmaktadır. Geliştirilen uygulama, eğlence, eğitim, e-ticaret, mühendislik, tıp ve interaktif pazarlama gibi birçok sektörde kullanım potansiyeline sahiptir. Gerçek zamanlı veri işleme, 3D model görüntüleme ve sanal öğelerle fiziksel ortamda etkileşim sağlama gibi işlevler, proje kapsamında geliştirilen temel özellikler arasındadır.

Bu proje aynı zamanda, kullanıcıların AR teknolojisini pratik bir şekilde deneyimleyebilmesine olanak tanıyacak şekilde optimize edilmiştir. Böylece, eğitim alanında artırılmış gerçeklik ders materyalleri oluşturulabilirken, mühendislik ve mimarlık alanlarında da dijital prototipler ile çalışma imkânı sağlanabilmektedir. Ayrıca, uygulamanın sahip olduğu çapraz platform desteği sayesinde, farklı işletim sistemlerine sahip mobil cihazlarda aynı işlevsellik korunabilmektedir.

Bu kapsamda, artırılmış gerçeklik tabanlı oyunlar, kullanıcı dostu bir arayüz ile entegre edilmiş ve mobil cihazlarda dokunmatik etkileşimler ile deneyimi zenginleştirmek için çeşitli modüller geliştirilmiştir. Uygulama, hem bireysel kullanıcılar hem de kurumsal firmalar için AR çözümleri sunmak amacıyla ölçeklenebilir bir altyapıya sahiptir.

**2.2. Kullanıcı Özellikleri**

Uygulama, geniş bir kullanıcı kitlesine hitap edecek şekilde tasarlanmıştır ve aşağıdaki kullanıcı özelliklerini desteklemektedir:

* **Mobil cihazlar aracılığıyla artırılmış gerçeklik öğeleriyle etkileşim kurma:** Kullanıcılar, uygulama aracılığıyla çevrelerine sanal nesneler yerleştirebilir ve bunlarla gerçek zamanlı etkileşim kurabilirler.
* **Oyun ve eğlence modülleri içeren bir arayüze sahip olma:** Uygulama, kullanıcıların eğlenmesini sağlamak için AR tabanlı mini oyunlar ve interaktif modüller içermektedir.
* **Kullanıcı profil yönetimi ve bildirim sistemi entegrasyonu:** Kullanıcılar, kendi profillerini oluşturabilir, başarımlarını takip edebilir ve bildirim sistemi aracılığıyla güncellemelerden haberdar olabilirler.
* **Kişiselleştirilebilir ayarlar menüsü:** Uygulama, kullanıcı deneyimini artırmak amacıyla görsel, ses ve arayüz ayarlarının özelleştirilmesine olanak tanımaktadır.
* **Gerçek zamanlı veri senkronizasyonu:** Kullanıcılar, uygulama içindeki ilerlemelerini bulut tabanlı sistemlerle senkronize edebilirler.
* **Çoklu cihaz desteği:** Aynı hesap üzerinden birden fazla cihazda oturum açma ve AR deneyimini farklı cihazlarda sürdürme imkânı sağlar.
* **Etkileşimli AR içerik desteği:** Kullanıcıların artırılmış gerçeklik öğelerini farklı şekillerde kullanabilmesine imkân tanır. Örneğin, sanal nesnelerle etkileşim, AR ortamında gezinme ve kullanıcı hareketlerine bağlı nesne yerleştirme gibi özellikler.
* **Offline kullanım modu:** İnternet bağlantısı olmadan da temel AR modüllerinin çalışmasını sağlar. Bu sayede, kullanıcılar her yerde AR deneyimini sürdürebilir.

**2.3. Kısıtlamalar**

Proje geliştirme sürecinde bazı teknik ve sistemsel kısıtlamalar belirlenmiştir. Bu kısıtlamalar aşağıdaki gibidir:

* **Uygulama yalnızca AR destekleyen cihazlarda çalışmaktadır:** Uygulamanın sunduğu özellikler, cihazın artırılmış gerçeklik desteği ve kamera erişimi gerektirmektedir. Bu nedenle ARCore veya ARKit desteği olmayan cihazlarda tam işlevsellik sağlanamayabilir.
* **Expo platformu üzerinden geliştirildiği için internet bağlantısı gerektirebilir:** Uygulama, Expo platformu kullanılarak geliştirildiği için, bağımlılıkların ve güncellemelerin yüklenmesi sırasında internet bağlantısı gerekmektedir. Bazı modüller çevrimdışı çalışabilse de, güncellemeler, veri senkronizasyonu ve gerçek zamanlı özellikler için internet bağlantısı zorunlu olabilir.
* **Donanım uyumluluğu:** Uygulamanın performansı, kullanılan mobil cihazın işlem gücüne ve grafik işlem birimine (GPU) bağlıdır. Düşük donanımlı cihazlarda görselleştirme gecikmeleri ve performans düşüşleri yaşanabilir.
* **Depolama ve RAM kullanımı:** Uygulamanın sunduğu AR deneyimi, yüksek çözünürlüklü modeller ve grafik öğeleri gerektirdiği için yeterli depolama alanı ve bellek tüketimi gerektirir.
* **Gerçek zamanlı işleme gereksinimi:** AR öğelerinin doğru çalışabilmesi için cihazın **kameradan gelen verileri anında işlemesi** gerekmektedir. Bu da eski nesil cihazlar için performans sorunları doğurabilir.
* **Pil tüketimi:** AR uygulamaları, yoğun grafik işlemeleri nedeniyle mobil cihazların batarya tüketimini artırabilir. Bu yüzden, kullanıcıların uzun süreli kullanımda güç yönetimine dikkat etmeleri önerilmektedir.
* **Bağlantılı cihaz gereksinimi:** Bazı gelişmiş özellikler (örneğin, harici sensörlerden veri alma veya karma gerçeklik destekleri) belirli ek donanımlar gerektirebilir.

Bu kısıtlamalar göz önüne alındığında, uygulamanın AR destekli cihazlar için optimize edilmiş bir deneyim sunması amaçlanmıştır. Kullanıcı geri bildirimleri ve sistem gereksinimleri doğrultusunda, gelecekte daha geniş donanım desteği sağlamak için ek geliştirmeler planlanmaktadır.

**3. GEREKSİNİMLER**

**3.1. Yazılım Gereksinimleri**

AR Project uygulamasının sorunsuz bir şekilde çalıştırılabilmesi için belirli yazılım gereksinimlerinin sağlanması gerekmektedir. Bu gereksinimler, geliştirme ortamının kurulumu, uygulamanın derlenmesi ve çalıştırılması için gerekli bağımlılıkları içermektedir. Proje React Native ve Expo çerçevesi kullanılarak geliştirildiğinden, aşağıdaki yazılımların sistemde kurulu olması gerekmektedir:

* **Node.js (LTS sürümü)**: Uygulamanın bağımlılıklarını yönetmek ve React Native ile Expo projelerini çalıştırmak için gereklidir. En güncel LTS (Long-Term Support) sürümünün yüklenmesi önerilir. Node.js'i yüklemek için:
* https://nodejs.org/ adresinden indirilebilir.

Node.js yüklendikten sonra aşağıdaki komutlarla kurulum doğrulanabilir:

node -v # Node.js sürümünü kontrol eder.

npm -v # NPM paket yöneticisinin kurulumunu doğrular.

* **Expo CLI**: React Native uygulamalarını Expo üzerinden geliştirmek ve test etmek için kullanılan araçtır. Kurulumu için aşağıdaki komut çalıştırılmalıdır:
* npm install -g expo-cli

Kurulumun başarılı olup olmadığını doğrulamak için şu komut çalıştırılabilir:

expo --version

* **Git (proje versiyon kontrolü için)**: Proje geliştirme sürecinde sürüm kontrolü ve iş birliği için Git gereklidir. Git yüklü değilse, aşağıdaki bağlantıdan indirilebilir:
* https://git-scm.com/downloads

Kurulum sonrası aşağıdaki komut ile doğrulama yapılabilir:

git --version

* **Android Studio ve iOS Simulator (Testler için önerilir)**: Geliştirilen uygulamanın test edilmesi için Android Studio ve iOS Simulator gereklidir.
  + Android Studio, Android cihazlar ve emülatörlerde test yapmak için gereklidir. AVD Manager (Android Virtual Device) üzerinden sanal cihaz oluşturulabilir.
  + iOS Simulator, yalnızca macOS sistemlerinde Xcode üzerinden çalıştırılabilir.

**Ekstra yazılım gereksinimleri:**

* **Metro Bundler**: React Native projelerini derlemek için kullanılır. Expo CLI ile birlikte gelir.
* **Babel**: JavaScript kodlarını dönüştürmek için kullanılan Babel konfigürasyonu babel.config.js dosyasında bulunur.
* **ESLint & Prettier**: Kodun temiz ve düzenli yazılmasını sağlayan geliştirme araçlarıdır.

**3.2. Donanım Gereksinimleri**

AR teknolojisinin gerektirdiği işlem gücü ve grafik performansı nedeniyle, uygulamanın çalışabilmesi için belirli donanım gereksinimlerinin karşılanması gerekmektedir:

* **AR destekli Android veya iOS cihazı**:
  + Android cihazlarda Google ARCore desteği gereklidir.
  + iOS cihazlarda Apple ARKit desteği olmalıdır.
* **Minimum 3 GB RAM ve 64-bit işlemci**:
  + Uygulama, artırılmış gerçeklik bileşenleri kullandığı için yüksek RAM tüketimi gerektirmektedir. Önerilen RAM miktarı 4 GB veya daha fazladır.
  + İşlemcinin ARM64 veya x86-64 mimarisine sahip olması gereklidir. Düşük performanslı cihazlarda gecikmeler yaşanabilir.
* **Yüksek çözünürlüklü kamera ve sensörler**:
  + AR deneyiminin daha iyi çalışabilmesi için minimum 1080p destekli bir kamera gereklidir.
  + Hareket algılayıcı sensörler (gyroscope, accelerometer, depth sensor) uygulamanın daha doğru çalışmasını sağlar.
* **Depolama alanı**:
  + Uygulamanın sorunsuz çalışabilmesi için en az 500 MB boş alan gerekmektedir.
  + AR içeriklerinin yüklenmesi için ek depolama alanı önerilir.
* **Grafik İşlemci Ünitesi (GPU) Desteği**:
  + OpenGL ES 3.0 veya üstü destekleyen GPU'lar gereklidir.
  + AR nesnelerinin daha akıcı görüntülenmesi için Adreno 600 serisi, Mali-G serisi veya Apple A12 Bionic ve üstü çipsetler önerilmektedir.
* **Pil Kullanımı**:
  + AR uygulamaları, yoğun grafik işleme ve kamera kullanımı nedeniyle yüksek pil tüketimi yapabilir. Uzun süreli kullanım için cihazın güç tasarrufu modları ve yüksek kapasiteli batarya önerilmektedir.
* **Bağlantı Gereksinimleri**:
  + Uygulama, bazı özellikleri için Wi-Fi veya mobil veri bağlantısı gerektirebilir.
  + AR nesnelerinin çevrimiçi olarak indirilebilmesi için stabil internet bağlantısı önerilir.

Bu gereksinimlerin karşılanması, uygulamanın daha akıcı ve stabil çalışmasını sağlamak için önemlidir. Geliştirme ve test sürecinde cihazın donanımına uygun yapılandırmalar yapılması önerilir.

**4. PROJE YAPISI**

Bu bölümde, AR Project uygulamasının dizin yapısı ve her bir dosyanın veya klasörün işlevi açıklanmaktadır. Proje, React Native ve Expo kullanılarak geliştirilmiş olup modüler ve ölçeklenebilir bir yapıya sahiptir. Aşağıda, projenin dizin yapısı ve her bir bileşenin işlevi ayrıntılı olarak verilmiştir.

**4.1. Klasör ve Dosya Yapısı**

Proje aşağıdaki klasör yapısına sahiptir:

AR-project-main/ (Ana dizin)

│── assets/ # Uygulama ikonları, splash ekranı ve grafik dosyaları

│── components/ # Uygulamanın farklı ekranlarını içeren React Native bileşenleri

│ │── GameScreen.js # Oyun modülü için ana bileşen

│ │── ProfileScreen.js # Kullanıcı profil yönetim ekranı

│ │── SettingsScreen.js # Uygulama ayarlarını yöneten bileşen

│ │── NotificationsScreen.js # Kullanıcı bildirimleri ekranı

│── navigation/ # Navigasyon işlemlerini yöneten dosyalar

│ │── AppNavigator.js # Uygulama genelinde sayfa geçişlerini yöneten bileşen

│── services/ # API istekleri ve veri yönetimini içeren servisler

│ │── api.js # Sunucuya istek gönderen yardımcı dosya

│── utils/ # Yardımcı işlevler ve genel fonksiyonlar

│ │── helpers.js # Küçük ölçekli yardımcı fonksiyonları içerir

│── App.js # Ana uygulama giriş noktası

│── index.js # Uygulamanın başlatıcı dosyası

│── app.json # Expo yapılandırma dosyası

│── metro.config.js # Metro Bundler yapılandırma dosyası

│── babel.config.js # Babel JavaScript derleyici yapılandırma dosyası

│── package.json # Proje bağımlılıklarını ve betikleri içeren dosya

│── README.md # Proje hakkında genel bilgi içeren dökümantasyon

**4.2. Klasörlerin Açıklaması**

* **assets/**: Uygulamanın görsel varlıklarını barındıran klasördür. Burada uygulama ikonları, splash ekranı ve diğer grafik dosyaları bulunmaktadır.
* **components/**: Uygulamanın farklı sayfalarını ve arayüz bileşenlerini içeren klasördür. Bileşenler modüler olarak oluşturulmuş olup oyun ekranı, profil yönetimi, ayarlar ekranı ve bildirim ekranı gibi farklı modüller burada yer almaktadır.
* **navigation/**: Uygulamanın farklı ekranlar arasında geçişini yöneten React Navigation konfigürasyonlarını içerir.
* **services/**: API çağrıları ve veri işleme işlevlerini içeren yardımcı dosyaları içerir. Sunucudan veri çekme, kullanıcı giriş işlemleri ve oyun içi etkileşimlerin yönetilmesi gibi işlevler bu klasörde yer alır.
* **utils/**: Uygulama içinde kullanılan yardımcı fonksiyonları ve genel işlevleri içeren klasördür. Tekrar eden işlemleri tek bir noktada tanımlamak ve kod tekrarını önlemek için kullanılır.
* **App.js**: React Native uygulamalarının giriş noktasıdır. Ana bileşenleri ve navigasyon sistemini içerir.
* **index.js**: Uygulamanın başlatıcı dosyasıdır. App.js bileşenini kök düzeyde render eder.
* **app.json**: Uygulamanın temel ayarlarının yapılandırıldığı dosyadır. Expo çerçevesi ile çalışan uygulamalar için gereklidir.
* **metro.config.js**: Metro Bundler yapılandırma dosyasıdır. React Native projelerinde kodun paketlenme şeklini belirleyen ayarları içerir.
* **babel.config.js**: JavaScript derleyicisi Babel'in yapılandırmalarını içeren dosyadır. ES6 ve üstü JavaScript kodlarının eski tarayıcılarda ve sistemlerde çalışabilmesi için dönüştürülmesini sağlar.
* **package.json**: Projede kullanılan bağımlılıkların ve npm betiklerinin (scripts) tanımlandığı dosyadır. Bağımlılıkların yönetimi ve projenin çalıştırılması için önemlidir.
* **README.md**: Proje hakkında genel bilgi içeren dökümantasyon dosyasıdır. Kurulum, kullanım ve geliştirme süreçlerine dair bilgiler içerir.

**4.3. Proje Yapısının Avantajları**

Bu dizin yapısı, modüler ve ölçeklenebilir bir sistem oluşturmayı sağlar. Geliştiriciler için şu avantajları sunar:

* **Kod Düzeni ve Yönetilebilirlik**: Farklı bileşenlerin ayrı klasörlerde tutulması, kodun düzenli ve okunabilir olmasını sağlar.
* **Kolay Genişletilebilirlik**: Yeni modüller ve bileşenler kolayca eklenebilir.
* **Performans ve Optimizasyon**: Dosya yapısı, React Native ve Expo projelerinde performansı artırmaya yönelik optimize edilmiştir.
* **Takım Çalışmasına Uygunluk**: Açık bir klasör yapısı, ekip içindeki geliştiricilerin farklı bölümler üzerinde çalışmasını kolaylaştırır.

Bu yapı, projenin sürdürülebilir olmasını ve uzun vadede geliştirilebilirliğini sağlamaktadır. Gelecekte yeni özellikler eklenirken mevcut modüler yapı korunarak, genişletilebilir bir sistem oluşturulması hedeflenmiştir.

**5. UYGULAMA MODÜLLERİ**

Bu bölümde, AR Project uygulamasında yer alan ana modüller açıklanmaktadır. Her bir modül, kullanıcı deneyimini artırmak ve uygulamanın temel işlevselliğini sağlamak amacıyla geliştirilmiştir.

**5.1. Ana Modüller**

* **Ana Sayfa:** Kullanıcının uygulamaya giriş yaptığı ana ekrandır. Burada, kullanıcı uygulama içindeki farklı bölümlere yönlendirilebilir. Ana sayfa bileşeni, navigation/AppNavigator.js dosyası üzerinden yönetilmektedir.
* **Mini Oyunlar:** Uygulamanın en önemli modüllerinden biridir. AR destekli oyunlar ve etkileşimli simülasyonlar içermektedir. Kullanıcılar, artırılmış gerçeklik teknolojisini kullanarak sanal nesnelerle oyun oynayabilir. Oyun yönetimi components/GameScreen.js dosyası üzerinden sağlanmaktadır.
* **Profil Yönetimi:** Kullanıcıların kişisel bilgilerini görüntüleyip düzenleyebildiği modüldür. Kullanıcı adı, profil fotoğrafı, başarımlar ve oyun içi ilerleme gibi bilgiler burada saklanmaktadır. components/ProfileScreen.js dosyasında bulunmaktadır.
* **Bildirimler:** Kullanıcının sistemden aldığı geri bildirimleri ve güncellemeleri görüntüleyebileceği ekrandır. Örneğin, oyun içi bildirimler, yeni özellik duyuruları ve hata mesajları burada gösterilir. components/NotificationsScreen.js bileşeni tarafından yönetilmektedir.
* **Ayarlar:** Kullanıcı tercihlerini yapılandırabileceği bir modüldür. Burada, görsel ayarlar, ses kontrolü, veri senkronizasyonu ve uygulama bildirimleri gibi seçenekler bulunmaktadır. components/SettingsScreen.js dosyasında yer almaktadır.
* **Gerçek Zamanlı Veri Senkronizasyonu:** Kullanıcıların profillerini bulut tabanlı veri tabanı ile senkronize etmesini sağlayan bir modüldür. Kullanıcılar oyun ilerlemelerini kaydedebilir ve farklı cihazlar arasında devam edebilir. services/api.js üzerinden yönetilmektedir.

Bu modüller, uygulamanın kullanıcı dostu bir deneyim sunmasını ve AR teknolojisinin en iyi şekilde kullanılmasını sağlamak amacıyla geliştirilmiştir.

**6. TEST SÜREÇLERİ**

Uygulamanın kararlılığı, performansı ve kullanıcı deneyimi açısından optimize edilmesi için test süreçleri uygulanmıştır. Birim testleri, entegrasyon testleri, performans testleri ve kullanıcı deneyimi testleri bu süreçlerin önemli bir parçasıdır.

**6.1. Test Araçları**

Uygulama testleri için aşağıdaki araçlar kullanılmıştır:

* **Jest & React Testing Library:** React Native bileşenlerinin doğruluğunu test etmek için kullanılır. Arayüz testleri ve modül bağımsızlığı testleri bu araçlarla gerçekleştirilmektedir.
* **Expo Go:** Uygulamanın farklı mobil cihazlarda test edilmesini sağlar. Gerçek donanım testleri ve performans ölçümleri için kullanılır.
* **Android Studio & Xcode Simulator:** Uygulamanın Android ve iOS platformlarında test edilmesini sağlar. Emulator ve fiziksel cihaz testleri yapılırken kullanılır.
* **Redux DevTools & Reactotron:** Durum yönetimi ve hata ayıklama süreçlerinde kullanılmaktadır.
* **Firebase Test Lab:**Farklı cihazlarda çapraz platform uyumluluğu testleri yapmak için kullanılır.

**6.2. Uygulanan Testler**

Uygulamanın farklı senaryolarda kararlı çalıştığını doğrulamak için çeşitli testler uygulanmıştır:

* **Birim Testleri:**
  + React Native bileşenlerinin bağımsız olarak test edilmesi sağlanmıştır.
  + Jest & React Testing Library kullanılarak her modülün bağımsız olarak çalıştığı doğrulanmıştır.
* **Entegrasyon Testleri:**
  + Farklı modüllerin birbiriyle olan iletişimi ve veri akışları test edilmiştir.
  + Bildirimlerin, oyun modüllerinin ve profil yönetiminin birbiriyle uyumlu çalıştığı doğrulanmıştır.
* **Performans Testleri:**
  + AR bileşenlerinin yükleme süreleri, hafıza kullanımı ve işlem gücü tüketimi test edilmiştir.
  + Özellikle düşük donanımlı cihazlarda uygulamanın akıcılığı ve çökme sorunları analiz edilmiştir.
* **Kullanıcı Deneyimi Testleri:**
  + Gerçek kullanıcılar üzerinde yapılan testlerle, arayüz tasarımı ve etkileşimli bileşenler test edilmiştir.
  + Navigasyon akışları, oyun oynanabilirliği ve kullanıcı geri bildirimleri analiz edilerek geliştirmeler yapılmıştır.

Bu test süreçleri, uygulamanın hatasız, hızlı ve kullanıcı dostu bir deneyim sunmasını sağlamak için geliştirme sürecinin önemli bir parçası olmuştur. Gelecekte yapılacak güncellemelerde test senaryoları daha da genişletilerek, yeni cihaz ve platform desteği için optimizasyonlar sağlanacaktır.

**7. SONUÇ VE DEĞERLENDİRME**

Bu proje, artırılmış gerçeklik (AR) tabanlı mobil uygulamalar geliştirmek için Expo ve React Native platformlarının güçlü bir altyapı sunduğunu ve bu teknolojilerin etkin bir şekilde kullanılabileceğini göstermiştir. Proje sürecinde React Native bileşenleri, Expo çerçevesi ve AR teknolojisi başarıyla entegre edilerek, kullanıcı dostu ve etkileşimli bir mobil deneyim oluşturulmuştur.

Geliştirme aşamalarında karşılaşılan donanım uyumluluğu, AR bileşenlerinin optimizasyonu ve performans testleri gibi teknik zorluklar analiz edilerek, uygulamanın farklı cihazlarda sorunsuz çalışması için çeşitli iyileştirmeler yapılmıştır. Yapılan testler sonucunda uygulamanın:

* Kullanıcı dostu bir arayüze sahip olduğu,
* AR nesneleriyle gerçek zamanlı ve akıcı etkileşim sunduğu,
* Cihaz performansına bağlı olarak optimize edilebileceği tespit edilmiştir.

Bununla birlikte, proje sırasında AR destekli cihazların gerekliliği ve donanım kısıtlamaları, uygulamanın belirli kullanıcı grupları için erişilebilirliğini etkileyen faktörler arasında yer almıştır. Uygulama geliştirilirken Expo Go, Jest, React Testing Library, Android Studio ve iOS Simulator gibi çeşitli test araçları kullanılarak yazılımın hatasız ve kararlı çalışması sağlanmıştır.

**Gelecekte Yapılabilecek Geliştirmeler**

Projenin daha geniş bir kullanıcı kitlesine hitap edebilmesi ve AR deneyiminin daha da iyileştirilmesi için aşağıdaki geliştirmeler önerilmektedir:

* **AR öğelerinin artırılması ve daha fazla etkileşimli içerik eklenmesi:**
  + 3D nesnelerle zenginleştirilmiş yeni AR modülleri eklenebilir.
  + Kullanıcıların sanal nesnelerle daha fazla etkileşime girmesine olanak tanıyan dokunma ve hareket algılama özellikleri geliştirilebilir.
* **Makine öğrenmesi ve yapay zeka desteğiyle gelişmiş görüntü işleme teknikleri eklenmesi:**
  + Kullanıcıların yüz ifadelerini ve el hareketlerini tanıyarak daha doğal bir etkileşim sunan AR modülleri geliştirilebilir.
  + Görüntü işleme algoritmaları ile artırılmış gerçeklik öğelerinin gerçek dünya ile daha iyi entegre edilmesi sağlanabilir.
* **Çevrimdışı mod desteği eklenerek internet bağlantısı olmadan çalışabilen bir yapı oluşturulması:**
  + Kullanıcıların önceden indirilmiş AR öğeleriyle çevrimdışı etkileşim kurabilmesi sağlanabilir.
  + Yerel veri depolama sistemleri kullanılarak, internet bağlantısı olmadan da oyunların ve AR öğelerinin çalışması mümkün hale getirilebilir.
* **Çoklu kullanıcı desteği:**
  + Aynı artırılmış gerçeklik ortamında birden fazla kullanıcının etkileşimli deneyimler yaşayabileceği bir yapı geliştirilebilir.
  + Kullanıcıların sanal nesneleri paylaşmasına ve gerçek zamanlı çok oyunculu AR deneyimleri yaşamasına olanak tanınabilir.
* **Platform bağımsız optimizasyonlar:**
  + Uygulamanın web tabanlı AR sistemleriyle uyumlu hale getirilmesi sağlanarak, mobil cihazlara ek olarak tarayıcı üzerinden de erişim sağlanabilir.
  + Düşük donanımlı cihazlar için hafif sürümler geliştirilerek, daha geniş bir kullanıcı kitlesine ulaşılması hedeflenebilir.

**Sonuç**

Bu rapor, AR Project uygulamasının geliştirme sürecini, karşılaşılan zorlukları ve elde edilen sonuçları ayrıntılı bir şekilde açıklamak amacıyla hazırlanmıştır. Proje sürecinde React Native, Expo ve artırılmış gerçeklik teknolojilerinin birlikte kullanılarak başarılı bir uygulama geliştirilmesi sağlanmıştır.

Gelecekte yapılacak iyileştirmeler ile daha geniş bir kullanıcı kitlesine hitap eden, daha güçlü ve gelişmiş bir artırılmış gerçeklik deneyimi sunulması hedeflenmektedir. Bu kapsamda, kullanıcı geri bildirimleri, yeni teknoloji trendleri ve makine öğrenmesi gibi gelişmiş teknikler kullanılarak projenin daha ileri seviyelere taşınması amaçlanmaktadır.