

Departman : Yazılım Mühend3slğ3  
Ders Adı : Yazılım İnşası YZM 201



# HIZLIQUIZ SİSTEMİ

## PROJE TASARIM RAPORU

Adlar ve Soyadlar :

- 1- MUHAMMED TÜRKOĞLU 2310238537
- 2- MUHAMMED ELHAZAR 2310238562
- 3- EMİR ELKASIM 2310238564

Asst.Prof.Dr. AYŞE NUR ALTINTAŞ TANKÜL

# İçindekiler

1 Sistemin Amacı.....	4
1.1 Tasarım Hedefleri.....	4
1.1.1 Kullanılabilirlik.....	4
1.1.2 Performans.....	4
1.1.3 Veri Güvenliği.....	4
1.1.4 Ölçeklenebilirlik.....	4
1.1.5 Taşınabilirlik.....	4
1.1.6 Genişletilebilirlik.....	5
1.2 Ödünler ve Kriterler .....	5
1.3 Kullanılabilirlik vs. Özellik Zenginliği.....	5
1.4 Performans vs. Güvenlik .....	5
1.5 Geliştirme Süresi vs. Genişletilebilirlik .....	5
1.6 Veri Güvenliği vs. Taşınabilirlik .....	5
1.7 Hızlı Erişim vs. Veritabanı Bütünlüğü .....	5
2 Sistem Mimarisi.....	6
2.1 Alt Sistem Ayırıştırma .....	6
2.1.1 Kullanıcı Arayüzü (UI) Alt Sistemi .....	6
2.1.2 İş Mantığı (Business Logic) Alt Sistemi.....	6
2.1.3 Veri Yönetimi Alt Sistemi .....	6
2.1.4 Geri bildirim ve performans Özeti Alt sistemi .....	7
2.2 Donanım/Yazılım Eşleme.....	7
2.2.1 Sunucu Tara^ .....	7
2.2.2 İstemci Tara^ .....	7
2.3 Kalıcı Veri Yönetimi .....	8
2.4 Erişim Kontrolü ve Güvenlik .....	8
2.4.1 Kullanıcı Kimlik Doğrulama .....	8
2.4.2 Veri Güvenliği.....	9
3 Alt Sistemi Hizmetleri .....	9
3.1 Kullanıcı Arayüzü Alt Sistemi Hizmetleri .....	9
3.1.1 Başlangıç Ekranı .....	9
3.1.2 Soru Görüntüleme.....	9
3.1.3 Kullanıcı İlerlemesi ve Skor takıby .....	9
3.1.4 Sonuç Ekranı .....	10
3.1.5 Hata ve Durum mesajları.....	10

3.1.6	Mobil Uyum ve Kullanılabilirlik.....	10
3.2	Soru Yönetim Alt Sistemi.....	10
3.2.1	Soru havuzu Yönetimi.....	10
3.2.2	Soru karıştırma ve gösterim .....	10
3.2.3	Cevap kontrolü .....	10
3.2.4	Performans ve Statistik Takıby .....	10
4	Düşük Seviyeli Tasarımı.....	11
4.1	Son Nesne tasarımı.....	11
4.1.1	Soru (Question)sını^.....	11
4.1.2	Quiz Yöneticisi Sını^.....	11
4.1.3	Geri Beldirim.....	12
4.1.4	Arayüz Yöneticisi .....	12
4.2	Tasarım Kararları ve Tasarım Desenleri .....	13
4.2.1	Tasarım Kararları.....	13
4.2.2	Tasarım Desenleri .....	13
4.2.3	Performans ve Ölçeklenebilirlik Kararları.....	14
4.2.4	Kullanıcı Dostu Arayüz Tasarımı .....	15

## **1. Sistemin Amacı**

Bu proje, kullanıcıların genel kültür alanında bilgi düzeylerini ölçmelerini ve eğlenirken öğrenmelerini sağlayan bir Çevrimiçi Quiz Sistemi geliştirmeyi hedeflemektedir. Sistem, kullanıcıların çeşitli konularla ilgili sorulara cevap vererek bilgi birikimlerini artırmalarına ve aynı zamanda interaktif bir deneyim yaşamalarına olanak tanır. Her oturumda kullanıcıya rastgele 12 soru sorulur ve doğru yanıtlarla belirli bir başarı seviyesine ulaşıldığında yeni sorularla başka bir oturum başlatılır.

### **1.1. Tasarım Hedefleri**

Çevrimiçi Quiz Sistemi tasarımında şu hedeflere odaklanılmıştır:

#### **1.1.1. Kullanılabilirlik**

Sistemin kullanıcı dostu bir arayüz sunması temel önceliktir. Sorulara hızlı erişim, yanıtların kolay seçimi ve sonuçların anlık geri bildirimle kullanıcıya sunulması sağlanmıştır. Quiz süreci akıcı bir deneyim sunacak şekilde tasarlanmıştır.

#### **1.1.2. Performans**

Soruların seçimi, yanıtların kontrolü ve skor hesaplamaları, gecikme yaşanmadan gerçekleşmelidir. Kullanıcının oturum sonu performans değerlendirmesi de hızla yapılır. Sistem, kullanıcıya kesintisiz bir deneyim sunmak için optimize edilmiştir.

#### **1.1.3. Veri Güvenliği**

Kullanıcıların verdiği yanıtlar ve elde ettiği skorlar yalnızca oturum süresince geçici olarak saklanır. Sistemin veri güvenliği ihtiyacı düşük seviyede olsa da tarayıcıda güvenli bir çalışma ortamı sunar.

#### **1.1.4. Ölçeklenebilirlik**

Sistem, aynı anda birden fazla kullanıcıya hizmet verecek şekilde yapılandırılmıştır. Soruların ve oturumların yönetimi, sistemin performansını etkilemeden sorunsuz şekilde sürdürülür.

#### **1.1.5. Taşınabilirlik**

Sistem, masaüstü, tablet ve mobil cihazlar gibi farklı platformlarda çalışacak şekilde responsive bir yapıda geliştirilmiştir. Her cihazda kullanıcıya kolay kullanım sunar.

### **1.1.6. Geniřletilebilirlik**

İleride sisteme yeni özellikler eklenebilmesi için modüler bir tasarım benimsenmiştir. Örneğin, farklı dillerde quizler, yeni soru türleri ya da kullanıcıya yönelik başarı grafiklerinin eklenmesi gibi özellikler kolayca entegre edilebilir.

### **1.2. Ödünler ve Kriterler**

Çevrimiçi Quiz Sistemi tasarımında bazı ödünleřmeler yapılmıştır. Bu ödünler, sistemin temel amacına odaklanarak önceliklerin belirlenmesiyle ilişkilidir.

### **1.3. Kullanılabilirlik vs Özellik Zenginlięi**

Sistem, kolay erişilebilir ve anlaşılır bir kullanıcı deneyimi sunmak amacıyla sade bir tasarım tercih edilmiştir. Karmařık özelliklerden kaçınılmış ve temel bir quiz deneyimi sunmaya odaklanılmıştır.

### **1.4. Performans vs Güvenlik**

Sistemin yüksek performansla çalışması sağlanmıştır. Verilerin tarayıcı tarafında geçici olarak saklanması gibi hafif güvenlik önlemleri, sistemin hızlı çalışmasını koruyacak şekilde uygulanmıştır.

### **1.5. Geliřtirme Süresi vs Geniřletilebilirlik**

Projenin temel işlevleri öncelikli olarak geliştirilmiş ve kısa sürede tamamlanmıştır. İleride eklenmesi planlanan özellikler için sistem modüler yapıda tasarlanmıştır.

### **1.6. Veri Güvenlięi vs Tařınabilirlik**

Sistem, çeřitli cihazlarda çalışacak şekilde tasarlanmıştır. Ancak taşınabilirlik ve veri güvenlięi dengesini sağlamak için oturum bazlı geçici veri kullanımı tercih edilmiştir.

### **1.7. Hızlı Eriřim vs Veritabanı Bütünlüęü**

Sistem, hızlı erişim sağlamak için soruları bellekte geçici olarak saklar ve karmařık sorgulamalardan kaçınır. Böylece kullanıcı deneyimi kesintisiz ve akıcı bir şekilde sunulur.

## 2. Sistem Mimarisi

Genel K lt r  evrimi i Quiz Sistemi, kullanicıların genel k lt r bilgilerini e lenceli ve interaktif bir  ekilde test etmelerini sa lamak i in tasarlanmı tır. Sistem, kullanicı aray z  (UI), i  mantı ı ve veri y netimi gibi temel katmanlardan olu ur. Mod ler yapısı sayesinde sistemin geni letilmesi ve bakımı kolaydır.

### 2.1. Alt Sistem Ayrıştırma

Sistem, kullanicı deneyimini optimize etmek ve i levselli i artırmak i in birkaç alt mod le ayrılmı tır:

#### 2.1.1. Kullanicı Aray z  (UI) Alt Sistemi

Kullanicıların sorulara eri imini ve cevaplamasını sa layan g rsel ve i levsel bir platform sunar.

- **Sade ve kullanicı dostu tasarım:** Sorular ve yanıt se enekleri a ık ve net bir  ekilde g sterilir.
- **Cihaz uyumlulu u:** Hem masa st  hem de mobil cihazlarda sorunsuz  alı an duyarlı bir aray z sunar.
- **Anlık geri bildirim:** Kullanicının do ru veya yanlı  cevaplarına g re anında geri bildirim sa lar.

#### 2.1.2. İ  Mantı ı Alt Sistemi

Sistemin temel kurallarını ve i levlerini y neten alt sistemdir.

- **Rastgele soru se imi:** Her oturumda tekrarlamayan soruların rastgele se ilmesini sa lar.
- **Skor takibi:** Kullanicının do ru cevap sayısını ve ba arı y zdesini hesaplar.
- **Dinamik oturum y netimi:** Kullanicının performansına ba lı olarak yeni sorular atar veya mevcut soruları tekrar sunar.

#### 2.1.3. Veri Y netimi Alt Sistemi

Quiz sistemindeki verilerin y netiminden sorumludur.

- **Ge ici veri y netimi:** Sorular ve oturum bilgileri tarayıcı belle inde saklanır.

- **Tekrar önleme:** Soruların bir kez kullanıldığını takip eden bir mekanizma ile aynı soruların tekrar sorulmasını engeller.
- **Kalıcı veritabanına ihtiyaç duymaz:** Sistem geçici olarak çalıştığı için karmaşık veri tabanı yapılarına gerek yoktur.

#### 2.1.4. Geri Bildirim ve Performans Özeti Alt Sistemi

Kullanıcının quiz deneyimini zenginleştirmek ve öğrenme sürecini desteklemek için tasarlanmıştır.

- Doğru/yanlış cevap geri bildirimleri: Kullanıcının cevaplarına bağlı olarak motive edici veya bilgilendirici yorumlar sağlar.
- Oturum özetleri: Kullanıcıya başarı oranı ve doğru cevap yüzdesi gibi performans bilgilerini sunar.

### 2.2. Donanım/Yazılım Eşleme

Genel Kültür Çevrimiçi Quiz Sistemi için gerekli donanım ve yazılım gereksinimleri, projenin ölçeği ve ihtiyaçlarına göre belirlenmiştir. Sistem, web tabanlı bir yapı kullanarak kullanıcıların herhangi bir cihazdan erişim sağlamasına olanak tanır.

#### 2.2.1. Sunucu Tarafı

**Not:** Proje basit bir çevrimiçi quiz sistemi olduğu için genellikle yerel sunucular ya da statik bir barındırma hizmeti kullanır. Kalıcı bir sunucu altyapısı gerekmez.

- **Donanım Gereksinimi:**  
Sunucu gerekliliği olmamakla birlikte, geliştirme sırasında bir yerel sunucunun çalıştırılması için minimum 2 GB RAM ve 1 GHz işlemci yeterlidir.
- **Yazılım Gereksinimi:**
  - **Yerel Sunucu:** Proje geliştirme aşamasında *Live Server* gibi basit yerel sunucular kullanılabilir.

#### 2.2.2. İstemci Tarafı

Quiz sistemine erişim için kullanıcı cihazlarında herhangi bir modern web tarayıcısının bulunması yeterlidir.

- **Donanım Gereksinimi:**

Bilgisayar, tablet veya akıllı telefon gibi bir cihaz; stabil bir internet bağlantısı gereklidir.

- **Yazılım Gereksinimi:**

- **Web Tarayıcısı:** Google Chrome, Mozilla Firefox, Safari veya Microsoft Edge gibi HTML5 uyumlu modern bir tarayıcı.
- **Kullanıcı Arayüzü:** HTML, CSS ve JavaScript ile geliştirilmiş bir web tabanlı kullanıcı arayüzü.

## 2.3. Kalıcı Veri Yönetimi

**Genel Kültür Çevrimiçi Quiz Sistemi**'nde kalıcı veri yönetimi, öncelikli olarak dinamik olmayan bir yapıya dayanır, çünkü bu sistemde veriler genellikle statik bir biçimde saklanır ve güncellenmez. Ancak gelecekte daha geniş kapsamlı bir veri yönetimi gereksinimi ortaya çıkarsa, aşağıdaki yöntemler değerlendirilebilir:

- **Veri Yapısı:**

- **Sorular:** Veriler JSON dosyalarında veya ilişkisel bir veritabanında saklanabilir. Bu yapı, soru metni, seçenekler, doğru cevap ve kategori gibi alanları içerir.

- **Veri Erişim Katmanı:** Sistem basit bir ön uç uygulaması olarak tasarlandığı için doğrudan tarayıcı üzerinden çalışır. Bir veritabanı bağlantısına ihtiyaç duyulmaz.

## 2.4. Erişim Kontrolü ve Güvenlik

Quiz sistemi temel bir kullanıcı doğrulama veya yetkilendirme içermemektedir. Sistem herkese açık olduğundan, güvenlik gereksinimleri aşağıdaki şekilde düzenlenmiştir:

### 2.4.1. Kullanıcı Kimlik Doğrulama

Şu anda kullanıcı doğrulaması uygulanmamaktadır. Ancak gelecekte kullanıcı hesabı yönetimi eklenecek olursa:

- Kullanıcı şifreleri güvenli bir şekilde saklanabilir .



- Oturum yönetimi için JSON Web Token (JWT) veya çerez tabanlı oturumlar tercih edilebilir.

### 2.4.2. Veri Güvenliği

- **HTTPS Kullanımı:** Veriler, istemci ve sunucu arasında güvenli bir şekilde iletilir.
- **Saldırı Koruması:** Kod, temel XSS (Cross-site Scripting) ve diğer kötüye kullanımları engellemek için kullanıcı girdilerini doğrular.

## 3. Alt Sistem Hizmetleri

**Genel Kültür Çevrimiçi Quiz Sistemi**, ana işlevleri desteklemek için bir dizi alt sistemden oluşur. Bu alt sistemler, sistemin modüler yapısını oluşturur ve kullanıcı deneyimini geliştirmek için belirli sorumluluklara sahiptir. Aşağıda Kullanıcı Arayüzü ve Soru Yönetimi alt sistemlerinin detayları sunulmuştur.

### 3.1. Kullanıcı Arayüzü Alt Sistemi Hizmetleri

Kullanıcı Arayüzü Alt Sistemi, kullanıcıların sistemle doğrudan etkileşimde bulunduğu katmandır. Bu katman, kullanıcıların quiz deneyimini kolaylaştırmak için çeşitli hizmetler sunar:

#### 3.1.1. Başlangıç Ekranı

- Kullanıcıların quiz başlatmasını sağlar.
- Ekranda sistemin temel açıklamaları (örneğin, "12 soruyu tamamlayın, doğru yanıtlarla ilerleyin!") sunulur.

#### 3.1.2. Soru Görüntüleme

- Kullanıcıya, her turda 12 soru sunar.
- Sorular, dinamik olarak yüklenir ve doğru/yanlış seçeneklerle görüntülenir.

#### 3.1.3. Kullanıcı İlerlemesi ve Skor Takibi

- Kullanıcının doğru cevaplarını sayar ve ilerleme çubuğu ile mevcut durumu görselleştirir.
- Eğer kullanıcı 6 veya daha fazla doğru cevap verirse, sonraki 12 soru otomatik olarak yüklenir.

### **3.1.4. Sonuç Ekranı**

- Quiz tamamlandığında kullanıcıya toplam skorunu ve performansını gösterir.
- Yeterli doğru cevap alınmadıysa "Tekrar Dene" seçeneği sunar.

### **3.1.5. Hata ve Durum Mesajları**

- Kullanıcı işlemleri başarılı olduğunda (örneğin, bir soruya doğru cevap verildiğinde) veya başarısız olduğunda (örneğin, bir seçenek seçilmediğinde) geri bildirim sağlar.
- Örnek mesajlar:
  - "Tebrikler, doğru cevap!"
  - "Maalesef, yanlış cevap."

### **3.1.6. Mobil Uyum ve Kullanılabilirlik**

- Arayüz, farklı cihazlara uygun olacak şekilde responsive tasarlanmıştır.
- Butonlar ve yazı tipi boyutları, dokunmatik ekranlara uygun olarak optimize edilmiştir.

## **3.2. Soru Yönetimi Alt Sistemi**

Soru Yönetimi Alt Sistemi, quiz sorularının doğru şekilde işlenmesini sağlar. Bu sistem, soruların yüklenmesi, karıştırılması ve yeniden kullanılmasını yönetir.

### **3.2.1. Soru Havuzu Yönetimi**

- Sorular bir JSON dosyasında veya statik bir kaynakta saklanır.
- Her oyun için rastgele 12 soru seçilir. Önceden sorulan sorular tekrar edilmez.

### **3.2.2. Soru Karıştırma ve Gösterim**

- Sorular, kullanıcıya gösterilmeden önce rastgele bir sıraya dizilir.
- Soru seçenekleri (doğru ve yanlış) da her gösterimde karıştırılarak kullanıcıya sunulur.

### **3.2.3. Cevap Kontrolü**

- Kullanıcının seçtiği seçenek, doğru cevapla karşılaştırılır.
- Doğru ise skor artırılır ve bir sonraki soru gösterilir. Yanlış ise kullanıcı bilgilendirilir.

### **3.2.4. Performans ve İstatistik Takibi**

- Kullanıcının cevaplama performansı izlenir ve ilerleyen aşamalar buna göre düzenlenir.

- Eğer kullanıcı bir turda yeterince doğru cevap verirse yeni bir tur başlatılır. Aksi takdirde kullanıcıya aynı sorular tekrar sunulur.

## 4. Düşük Seviyeli Tasarım

### 4.1. Son Nesne Tasarımı

Bu bölümde, nesne tabanlı modelin detaylarını açıklayarak sınıfları ve bunların sistemdeki etkileşimlerini ele alacağız. Bu sınıflar, sorular, quiz oturumları, kullanıcı puanları ve sonuçların görüntülenmesi gibi işlevleri yönetir.

#### 4.1.1. Soru (Question) Sınıfı

**Açıklama :** Soru sınıfı, quizdeki bir soruyu temsil eder. Sorunun metni, olası cevapları ve kullanıcı tarafından seçilen cevabı içerir.

**Özellikler :**

- `text: String` - Sorunun metni.
- `answers: List<Answer>` - Her biri metin ve doğruluğu belirten bir işaret içeren olası cevaplardan oluşan liste.
- `selectedAnswer: Answer | null` - Kullanıcı tarafından seçilen cevap. Başlangıçta null.

**Yöntemler :**

- `create(): void` - Soruyu ve cevaplarını etkileşimli HTML öğeleri olarak oluşturur.
- `disable(): void` - Kullanıcı bir cevap seçtikten sonra cevapları devre dışı bırakır.
- `update(): void` - Doğru/yanlış geri bildirimini göstererek arayüzü günceller.

**Olaylar :**

- `question-answered` - Bir cevap seçildiğinde tetiklenen olay.

#### 4.1.2. Quiz Yöneticisi (QuizManager) Sınıfı

**Açıklama :** QuizManager sınıfı, quizin ana mantığını yönetir. Puan takibi, oturum yönetimi ve sonuçların görüntülenmesi bu sınıfın sorumluluğundadır.

**Özellikler :**

- `questionsData: List<Question>` - Quiz için mevcut olan tüm soruların listesi.
- `currentSessionQuestions: List<Question>` - O anki quiz oturumunda gösterilen sorular.
- `score: int` - Kullanıcının mevcut oturumdaki puanı.
- `answeredCount: int` - Oturumda yanıtlanan soru sayısı.
- `completedSessions: int` - Tamamlanan quiz oturumlarının sayısı.

- `totalSessions: int` - Planlanan toplam quiz oturumu sayısı.

#### **Yöntemler :**

- `shuffle(array: List<any>): void` - Bir dizinin elemanlarını karıştırır. (Sorular için kullanılır.)
- `getRandomQuestions(): List<Question>` - Tekrarlanmayan, rastgele seçilmiş bir soru alt kümesi döndürür.
- `loadQuestions(retry: boolean): void` - Yeni bir oturum için soruları yükler veya mevcut oturumu tekrar başlatır.
- `evaluatePerformance(): void` - Bir oturumun sonunda kullanıcının performansını değerlendirir ve sonrasındaki aksiyonları belirler (devam et, tekrar oyna, sonlandır).
- `displayFinalCongratulations(): void` - Tüm oturumlar tamamlandığında bir tebrik mesajı görüntüler.
- `endQuiz(): void` - Quiz'i sıfırlayarak sayfayı yeniden yükler.

#### **4.1.3. Geri Bildirim (Feedback) Sınıfı**

**Açıklama :** Feedback sınıfı, her cevaptan sonra kullanıcıya gösterilen geri bildirimi yönetir.

#### **Özellikler :**

- `questionFeedback: HTML_Element` - Kullanıcıya doğru/yanlış cevabı gösteren HTML ögesi.

#### **İşleyiş :**

- Feedback sınıfı, her soruya entegre edilmiştir ve kullanıcıya cevap seçildikten sonra anında geri bildirim sağlar.

#### **4.1.4. Arayüz Yöneticisi (UI Manager) Sınıfı**

**Açıklama :** UI Manager sınıfı, oturum sonlarında gösterilen modaller ve etkileşimli mesajlar gibi arayüz elemanlarını yönetir.

#### **Ana Elemanlar :**

- `customAlert: HTML_Element` - Bir oturumun sonunda kullanıcıya yeniden oynama veya devam etme seçeneklerini sunan modal.
- `replayButton: HTML_Element` - Kullanıcının mevcut oturumu yeniden başlatmasını sağlayan buton.
- `endButton: HTML_Element` - Kullanıcının quiz'i sonlandırmasını sağlayan buton.

## 4.2. Tasarım Kararları ve Tasarım Desenleri

Bu bölümde, Soru-Cevap Sistemi için alınan tasarım kararları ve kullanılan tasarım desenleri ele alınacaktır. Tasarım kararları, sistemin esnekliğini, sürdürülebilirliğini ve kullanıcı dostu olmasını sağlamak için stratejik bir yaklaşımdır. Tasarım desenleri ise, tekrarlanan problemleri çözmek için uygulanan, iyi tanınmış yazılım mühendisliği yöntemleridir.

### 4.2.1. Tasarım Kararları

#### 1. Modülerlik ve Ayrılabilirlik:

Sistemin her bir bileşeni, bağımsız çalışabilmesi ve gerektiğinde kolayca güncellenebilmesi için modüler olarak tasarlanmıştır. Örneğin, soruları yönetmek için Question sınıfı, kullanıcı geri bildirimlerini sağlamak için Feedback sınıfı kullanılır. Bu modüler yapı, sistemin bakımını ve genişletilmesini kolaylaştırır.

#### 2. Soru ve Cevapların Yönetimi:

Sorular ve cevaplar, sistemdeki ana veri elemanlarıdır. Her soru, olası cevaplar ve kullanıcının seçtiği cevapla birlikte Question sınıfında temsil edilir. Soruların rastgele seçilmesi ve tekrarlanmaması için QuizManager sınıfı, soruları yönetir ve her oturumda farklı soru setleri oluşturur. Bu tasarım kararı, kullanıcı deneyimini zenginleştirir ve monotonluğu önler.

#### 3. Kullanıcı Etkileşimi ve Geri Bildirim:

Kullanıcıların quiz deneyiminde anlık geri bildirim alması, etkileşimi artırır ve öğrenmeyi destekler. Feedbacksınıfı, doğru/yanlış bilgilerini kullanıcıya anında sunar. Ayrıca, her oturum sonunda başarılı bir performans gösterildiğinde kullanıcıyı tebrik etmek için UIManager sınıfı ile etkileşimli mesajlar gösterilir.

#### 4. Durum Yönetimi:

Her bir sorunun durumu (selectedAnswer ve status) ayrı ayrı izlenir ve kullanıcı cevabını verdikçe güncellenir. Bu, doğru geri bildirimi sağlamak ve soruların yalnızca bir kez cevaplanmasını sağlamak için gereklidir. Ayrıca, quiz oturumlarının başarılı bir şekilde takip edilmesi için QuizManager sınıfında kullanıcı puanları ve cevaplanan sorular takip edilir.

### 4.2.2. Tasarım Desenleri

1. **Factory Deseni:** Question ve Answer sınıfları, Factory Design Pattern kullanılarak oluşturulmuştur. Bu desen, belirli kriterlere göre soruları ve cevapları dinamik bir

şekilde oluşturmayı sağlar. Özellikle quiz sistemlerinde, farklı türde soruların (çoktan seçmeli, doğru/yanlış vb.) hızlıca eklenebilmesi için bu desen kullanılır.

2. **Observer Deseni:** Feedback sınıfı, Question sınıfına entegre edilmiş olup, kullanıcı bir cevap seçtiğinde geri bildirim sağlamak için Observer Design Pattern kullanılır. Her soru cevaplandığında, ilgili geri bildirim otomatik olarak güncellenir ve kullanıcıya bildirilir. Bu, kullanıcı etkileşimlerinin anında ve doğru bir şekilde sisteme yansıtılmasını sağlar.
3. **Singleton Deseni:** QuizManager sınıfı, her quiz oturumunda yalnızca bir kez yaratılır ve oturum boyunca merkezi bir yönetici olarak görev alır. Bu amaçla, Singleton Design Pattern kullanılmıştır. Bu tasarım, sistemde birden fazla QuizManager örneğinin yaratılmasını engeller ve kaynak yönetimini optimize eder.
4. **Strategy Deseni:** Soruların rastgele seçilmesi ve quiz oturumlarının yönetilmesi için QuizManager sınıfında Strategy Design Pattern kullanılmıştır. Kullanıcıya gösterilecek soruları seçme ve düzenleme stratejileri, runtime (çalışma zamanı) sırasında değiştirilebilir. Örneğin, testlerin zorluk seviyelerine göre soru setleri oluşturulabilir veya soruların sırası rastgele değiştirilebilir.
5. **State Deseni:** Quiz oturumlarının durumu, QuizManager sınıfında State Design Pattern ile yönetilir. Kullanıcının quiz oturumunun başlatılmasından tamamlanmasına kadar her adımda sistemin durumu güncellenir. Bu desen, sistemin çeşitli durumlarda doğru tepkiler vermesini sağlar ve uygulamanın akışını yönetir.

#### 4.2.3 Performans ve Ölçeklenebilirlik Kararları

- **Veri Yapıları:**

Kullanıcılar için soruların ve cevapların hızlı bir şekilde yönetilmesi amacıyla listeler ve diziler kullanılır. QuizManager sınıfı, soruların şuffling (karıştırma) işlemini hızlı ve verimli bir şekilde gerçekleştirebilmek için optimizasyon yapmıştır.

- **Asenkron İşlemler:**

Kullanıcıların soruları yanıtladıktan sonra anlık geri bildirim alabilmesi için arka planda asenkron işlemler kullanılmıştır. Bu, kullanıcı deneyimini iyileştirir ve sayfa yenilemeleri gereksiz hale gelir.

#### **4.2.4 Kullanıcı Dostu Arayüz Tasarımı**

Arayüz tasarımında, kullanıcıların quiz deneyimini kolayca takip edebilmesi için basit ve anlaşılır bir kullanıcı arayüzü hedeflenmiştir. Kullanıcıların doğru/yanlış cevaba dair geri bildirim alması ve performanslarını takip etmesi için görsel geri bildirimler ve modaller kullanılır. Bu arayüz tasarım kararları, kullanıcı etkileşimini artırmak ve eğitimsel verimliliği sağlamak amacıyla optimize edilmiştir.

