

Departman: Yazılım Mühendisliği
Ders Adı: Yazılım İnşası YZM 201



HIZLIQUIZ SİSTEMİ

PROJE ANALİZ RAPORU

Adlar ve Soyadlar:

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

Asst.Prof.Dr. AYŞE NUR ALTINTAŞ TANKÜL

İçindekiler

HızlıQuiz Sistemi(QuickQuiz System)	3
Proje Özeti	3
1. Gereksinimler	3
1.1. İşlevsel Gereksinimler.....	3
1.1.1. Başlangıç ve Soru Görüntüleme	3
1.1.2. İlerleme Mekanizması.....	3
1.1.3. Tekrar Kullanılmayan Sorular	3
1.1.4. Tebrik Mesajları ve Bildirimler.....	3
1.1.5. Sonuçların Görüntülenmesi	3
1.2. İşlevsel Olmayan Gereksinimler.....	4
1.2.1. Kullanılabilirlik.....	4
1.2.2. Performans	4
1.2.3. Güvenlik.....	4
1.2.4. Taşınabilirlik.....	4
1.2.5. Uyumluluk.....	4
1.2.6. Genişletilebilirlik	4
1.2.7. Bakım Kolaylığı	4
1.2.8. Ölçeklenebilirlik	4
2. Kullanım Durumu Diyagramı	5
2.1. Aktörler.....	5
2.2. Kullanım Durumları.....	5
2.2.1. Quiz Başlatma ve Çözme	5
2.2.2. Soruların Tekrarlanmaması.....	5
2.2.3. Quiz Sonuçlarını Görüntüleme	5
2.2.4. Quiz Tekrarı	5
3. Dinamik Modeli (Sequence Diagram).....	7
3.1. Quiz Başlatma ve Çözme Dinamik Modeli	7
3.2. Soruların Tekrarlanmaması Dinamik Modeli	7
3.3. Quiz Sonuçlarının Gösterimi Dinamik Modeli	7
3.4. Quiz Tekrarı Dinamik Modeli	8
3.5. Quiz Tamamlanınca Tebrik Mesajı Dinamik Modeli	8
4. Soru-Cevap Sistemi Nesne Tabanlı Modeli (Class Diagram)	8
4.1. Sınıflar ve Açıklamalar	8
4.1.1. Soru(Question).....	9
4.1.2. Quiz Yöneticisi (Quiz Manager).....	9

4.1.3. Geri bildirim (Feedback)	9
4.1.4. Arayüz Yöneticisi (UI Manager).....	10
4.1.5. Veri ve Depolama (Data and Storage)	10
4.2. Sınıflar Arası İlişkiler	10
4.2.1. Quiz - QuizManager	10
4.3. Class Diagram Özeti	10
5. Quiz sistemi Kullanıcı Arayüzü	11
5.1. Tasarım ve Yapı.....	12
5.2. Puan yönetimi	12
5.3. Etkileşimler ve Geri bildirim	12
5.4. Erişebilirlik ve Duyarlılık.....	13
5.5. Basıtlık ve Verimlilik.....	13

Hızlı Quiz Sistemi (Quick Quiz System)

Proje Özeti:

Bu projede, genel kültür hakkında interaktif bir sınav web sitesi geliştirilecektir. Bu sistem, kullanıcıların performanslarına dayalı bir ilerleme mekanizması ile 12 rastgele sorudan oluşan serilere cevap vermelerini sağlayacaktır. Daha önce sorulan sorular bir oturum sırasında asla tekrarlanmayacak ve başarısızlık durumunda aynı sorularla bir tekrar oynama seçeneği sunulacaktır.

1. Gereksinimler

1.1. İşlevsel Gereksinimler

İşlevsel gereksinimler, sistemin yerine getirmesi gereken spesifik özellikleri ve kullanıcı etkileşimlerini kapsar.

1.1.1. Başlangıç ve Soru Görüntüleme

- Kullanıcı, siteye girdiğinde 12 rastgele soru görür ve hemen cevaplamaya başlayabilir.

1.1.2. İlerleme Mekanizması

- Kullanıcı en az 6 soruya doğru cevap verirse, sistem 12 yeni soru daha görüntüler.
- Tüm sorular doğru cevaplanana kadar süreç bu şekilde devam eder.

1.1.3. Tekrar Kullanılmayan Sorular

- Bir oturum sırasında kullanıcıya daha önce sorulmuş bir soru tekrar gösterilmez.

1.1.4. Tebrik Mesajları ve Bildirimler

- Kullanıcı tüm soruları başarıyla tamamladığında, ekranda büyük bir tebrik mesajı görüntülenir.
- Başarısızlık durumunda (6'dan az doğru cevap), kullanıcıya aynı sorularla tekrar deneme seçeneği sunulur.

1.1.5. Sonuçların Görüntülenmesi

- Kullanıcı, doğru cevapladığı soru sayısını ve genel başarı durumunu oturum sonunda görebilir.

1.2. İşlevsel Olmayan Gereksinimler

1.2.1. Kullanılabilirlik

- Arayüz basit, kullanıcı dostu ve mobil uyumlu olacak şekilde tasarlanmalıdır.

1.2.2. Performans

- Sorular hızlı bir şekilde yüklenmeli ve aralarında geçiş yaparken gecikme yaşanmamalıdır.

1.2.3. Güvenlik

- Sistem, kullanıcı verilerinin güvenliğini sağlamalıdır. Kullanıcı sorulara sadece cevap verebilir; içerik ekleme veya düzenleme yetkisine sahip değildir.

1.2.4. Taşınabilirlik

- Uygulama, tüm modern tarayıcılarda sorunsuz çalışmalıdır.

1.2.5. Uyumluluk

- Proje, W3C standartlarına uygun şekilde HTML, CSS ve JavaScript kullanılarak geliştirilmelidir.

1.2.6. Genişletilebilirlik

- Sistem, ileride daha fazla kategori veya soru eklenebilecek şekilde tasarlanmalıdır.

1.2.7. Bakım Kolaylığı

- Kod yapısı düzenli, okunabilir ve iyi belgelenmiş olmalıdır.

1.2.8. Ölçeklenebilirlik

- Gelecekte daha fazla kullanıcıya hizmet verebilmek için sistem ölçeklenebilir şekilde tasarlanmalıdır.

2. Kullanım Durumu Diyagramı

2.1. Aktörler

- **Kullanıcı (User) :**
 - Siteye giriş yapan ve quizleri çözüp sonuçlarını görüntüleyebilen kişi.
- **Yönetici (Admin) :**
 - Quizleri ve soruları ekleyen , düzenleyen , silen ve yönetim işlemlerini gerçekleştiren yetkili.

2.2. Kullanım Durumları

2.2.1. Quiz Başlatma ve Çözme

- **Açıklama :** Kullanıcı, sistemin sunduğu rastgele 12 sorudan oluşan bir quiz başlatır ve soruları cevaplar.
- **Ön Koşul :** Kullanıcının siteye erişim sağlaması.
- **Adımlar :**
 - Kullanıcı siteye girer ve sistem otomatik olarak 12 soruyu görüntüler.
 - Kullanıcı, her soru için bir cevap seçer.
 - Sistem her cevabı değerlendirir ve toplam doğru sayısını hesaplar.
 - Eğer kullanıcı en az 6 soruya doğru cevap verirse, sistem 12 yeni soru daha görüntüler.
 - Tüm sorular tamamlandığında kullanıcı büyük bir tebrik mesajı alır.
- **Sonuç :** Kullanıcı quiz sonucunu ve doğru cevap oranını görür.

2.2.2. Soruların Tekrarlanmaması

- **Açıklama :** Bir oturumda sorulan soruların tekrar gösterilmemesi sağlanır.
- **Ön Koşul:** Kullanıcının bir quiz başlatmış olması.
- **Adımlar :**
 - Sistem, daha önce sorulan soruları kontrol eder.
 - Yeni seçilen 12 soru, daha önce sorulmamış sorulardan oluşur.
- **Sonuç:** Kullanıcı, her yeni bölümde farklı sorularla karşılaşır.

2.2.3. Quiz Sonuçlarını Görüntüleme

- **Açıklama:** Kullanıcı, quiz tamamlandıktan sonra performansını görüntüler.
- **Ön Koşul:** Kullanıcının quizde tüm soruları cevaplamış olması.
- **Adımlar :**
 - Sistem, verilen cevapları doğru ve yanlış olarak değerlendirir.
 - Kullanıcı, doğru cevap sayısını ve genel performansını içeren bir sonuç ekranı görür.
- **Sonuç:** Kullanıcı, başarısını ve doğru/yanlış oranını görüntüler.

2.2.4. Quiz Tekrarı

- **Açıklama:** Kullanıcı başarısız olduğunda aynı sorularla quiz tekrarına yönlendirilir.
- **Ön Koşul:** Kullanıcının 6'dan az doğru cevap vermiş olması.
- **Adımlar:**
 - Sistem, kullanıcıya aynı sorularla bir tekrar seçeneği sunar.
 - Kullanıcı tekrar başlatmayı kabul ederse, sorular aynı şekilde görüntülenir.
- **Sonuç:** Kullanıcı aynı sorularla quiz tekrarını tamamlar.

3. Dinamik Modeli

3.1. Quiz Başlatma ve Çözme Dinamik Modeli

Senaryo : Kullanıcı, sisteme giriş yapıp quiz başlatmak istiyor.

Aktörler ve Nesneler :

- Kullanıcı (User)
- Kullanıcı Arayüzü (UI)

Akış:

1. **Kullanıcı:** Siteye giriş yapar ve quiz başlatır.
2. **UI:** 12 rastgele soru seçer ve ekranda gösterir (JavaScript üzerinden).
3. **Kullanıcı:** Sorulara tek tek cevap verir.
4. **UI:** Kullanıcının her cevabını kontrol eder ve doğru/yanlış sonuçları kaydeder.
5. **UI:** Kullanıcıya doğru cevap sayısını gösterir.
6. **UI:** Eğer kullanıcı 6 veya daha fazla doğru yaptıysa, yeni bir 12 soru seti hazırlar. Aksi takdirde, tekrar deneme seçeneği sunar.

3.2. Soruların Tekrarlanmaması Dinamik Modeli

Senaryo: Kullanıcıya aynı oturum içinde bir soru tekrar gösterilmemeli.

Aktörler ve Nesneler:

- Kullanıcı (User)
- Kullanıcı Arayüzü (UI)

Akış:

1. **UI:** Kullanıcının oturumunda sorulan soruları kaydeder.
2. **UI:** Yeni soru seçerken, önceki soruları kontrol ederek tekrar etmeyen bir soru seçer.
3. **UI:** Kullanıcıya bu yeni soruyu göster.

3.3. Quiz Sonuçlarının Gösterimi Dinamik Modeli

Senaryo: Kullanıcı, tamamladığı quizin sonuçlarını görmek istiyor.

Aktörler ve Nesneler:

- Kullanıcı (User)
- Kullanıcı Arayüzü (UI)

Akış:

1. **Kullanıcı:** Quiz sonuçlarını görüntüleme isteğinde bulunur.
2. **UI:** Kullanıcının verdiği cevapları kontrol ederek doğru ve yanlışları hesaplar.
3. **UI:** Toplam doğru sayısını ve başarı oranını kullanıcıya gösterir.

3.4. Quiz Tekrarı Dinamik Modeli

Senaryo: Kullanıcı başarısız olduğunda (6'dan az doğru cevap) tekrar seçeneği sunuluyor.

Aktörler ve Nesneler:

- Kullanıcı (User)
- Kullanıcı Arayüzü (UI)

Akış:

1. **UI:** Kullanıcının doğru cevap sayısını kontrol eder.
2. **UI:** Eğer kullanıcı başarısız olmuşsa, tekrar seçeneği sunar.
3. **Kullanıcı:** "Tekrar dene" seçeneğini tıklar.
4. **UI:** Aynı soruları tekrar yükler ve kullanıcıya gösterir.

3.5. Quiz Tamamlanınca Tebrik Mesajı Dinamik Modeli

Senaryo: Kullanıcı tüm soruları başarıyla tamamladığında tebrik mesajı alıyor.

Aktörler ve Nesneler:

- Kullanıcı (User)
- Kullanıcı Arayüzü (UI)

Akış:

1. **UI:** Kullanıcının tüm sorularını doğru cevapladığını kontrol eder.
2. **UI:** Kullanıcıya büyük bir tebrik mesajı (örneğin: "Tebrikler! Tüm soruları tamamladınız.") gösterir.
3. **Kullanıcı:** Sonuçları görüntüler ve başarıyla oturumu tamamlar.

4. Soru-Cevap Sistemi İçin Nesne Tabanlı Model (Sınıf Diyagramı)

4.1. Sınıflar ve Açıklamalar

4.1.1. Soru (Question)

- **Açıklama:** Bir soruyu, onun metnini, olası cevaplarını ve durumunu (seçilen cevap olup olmadığını) temsil eder.
- **Özellikler:**
 - `Text`: Sorunun metni.
 - `answers`: Olası cevapların listesi, her cevap bir metin (text) ve doğru olup olmadığını gösteren bir işaret (isCorrect) içerir.
 - `selectedAnswer`: Kullanıcı tarafından seçilen cevap (başlangıçta null).
- **Yöntemler:**
 - `create()`: Sorunun HTML ögesini, etkileşimli cevaplarla birlikte oluşturur.
 - `disable()`: Bir soru cevaplandıktan sonra cevapları devre dışı bırakır.
 - `update()`: Doğru/yanlış geri bildirimi göstermek için arayüzü günceller.
- **Olaylar:**
 - `question-answered`: Bir cevap seçildiğinde tetiklenir.

4.1.2. Quiz Yöneticisi (QuizManager)

- **Açıklama:** Puan takibi, oturumların yönetimi ve sonuçların görüntülenmesi gibi quizin ana mantığını yönetir.
- **Özellikler:**
 - `questionsData`: Quiz için mevcut olan soruların listesi.
 - `currentSessionQuestions`: Mevcut oturumda görüntülenen sorular.
 - `score`: Kullanıcının mevcut oturumdaki puanı.
 - `answeredCount`: Oturumda cevaplanan soru sayısı.
 - `completedSessions`: Tamamlanan oturumların sayısı.
 - `totalSessions`: Planlanan toplam oturum sayısı.
- **Yöntemler:**
 - `shuffle(array)`: Bir dizinin elemanlarını karıştırır (sorular için kullanılır).
 - `getRandomQuestions()`: Tekrarlanmayan rastgele bir soru alt kümesini seçer.
 - `loadQuestions(retry)`: Yeni bir oturum sorularını yükler veya mevcut oturumu yeniden oynatır.
 - `evaluatePerformance()`: Bir oturumun sonunda kullanıcının performansını değerlendirir ve sonraki eylemleri belirler (devam et, yeniden oyna veya bitir).
 - `displayFinalCongratulations()`: Tüm oturumlar tamamlandığında bir final mesajı görüntüler.
 - `endQuiz()`: Sayfayı yeniden yükleyerek quiz'i sıfırlar.

4.1.3. Geri Bildirim (Feedback)

- **Açıklama:** Her cevap sonrası kullanıcıya gösterilen etkileşimli geri bildirimi yönetir.
- **Özellikler:**
 - `questionFeedback`: Geri bildirimi (doğru/yanlış) gösteren HTML ögesi.

- **İşleyiş** : Kullanıcıya anında geri bildirim sağlamak için `Question` sınıfına entegre edilmiştir.

4.1.4. Arayüz Yöneticisi (UI Manager)

- **Açıklama:** Oturum sonlarında gösterilen modaller ve etkileşimli mesajlar gibi arayüz elemanlarını yönetir.
- **Ana Elemanlar:**
 - `customAlert` : Bir oturumun sonunda yeniden oynama veya devam etme seçeneklerini sunan modal.
 - `replayButton` : Bir oturumu yeniden oynatmaya yarayan düğme.
 - `endButton` : Quiz'i sonlandırmaya yarayan düğme.

4.1.5. Veri ve Depolama (Données et Stockage)

- **Açıklama:** Quiz verileri, yani sorular ve cevaplar, net ve tutarlı bir yapıda organize edilmiştir.
- **Veri Organizasyonu:**
 - Her soru, bir metin ve ilgili cevapların listesiyle temsil edilir.
 - Cevaplar, doğru olup olmadığını belirten bir gösterge içerir.
- **Veri İşleme:**
 - Veriler, oyun başında yüklenir ve tekrarları önlemek için dinamik olarak filtrelendir.
 - Quiz sırasında, her oturum için rastgele bir soru alt kümesi seçilir.

4.2. Sınıflar Arası İlişkiler

4.2.1. Question – QuestionManager

- **Açıklama:** `Question` sınıfı, `QuestionManager` tarafından yönetilir. `QuestionManager` sınıfı, birden fazla `Question` nesnesini içerir ve bu soruları rastgele seçer veya tekrar eder.
- **İlişki Türü:** Bileşim (Composition)

4.3. Class Diagram Özeti

`Question` sınıfı, quiz sisteminin temel bileşenidir. Bu sınıf, her bir soruyu temsil eder ve soruyla ilişkili çeşitli işlemleri yönetir. İşte bu sınıfın ana özellikleri ve işlevleri :

```

+-----+
|      Question      |
+-----+
| - text              |
| - answers           |
| - selectAnswer     |
| - feedback         |
+-----+
| + updateFeedback() |
| + disable()        |
+-----+

```

Question Sınıfının Açıklaması

1. Özellikler :

- `text` : Sorunun metnini tutar.
- `answers` : Sorunun doğru cevapları ve seçeneklerini içeren bir dizi.
- `selectedAnswer` : Kullanıcının seçtiği cevabı tutan nesne.
- `feedback` : Kullanıcının cevabını verdikten sonra gösterilecek geri bildirimi içeren mesaj.

2. Metodlar :

- `updateFeedback()` : Kullanıcı bir cevap seçtikten sonra bu metod, geri bildirimi günceller. Cevabın doğru ya da yanlış olduğuna göre bir mesaj görüntüler.
- `disable()` : Kullanıcı cevabını seçtikten sonra soruyu devre dışı bırakır. Bu metod, kullanıcının cevabını değiştirmesini engeller ve yanıt butonlarını devre dışı bırakır.

Sistemdeki Rolü

Question sınıfı, quizdeki her bir soruyu yönetmek için kritik öneme sahiptir. Bu sınıfın her örneği, bir soruyu ve ona ait birkaç cevabı temsil eder. Kullanıcı bir soruya cevap verdiğinde, Question sınıfı, cevabın seçilmesini yönetir, geri bildirimi günceller ve cevabı seçtikten sonra soruyu devre dışı bırakır.

Özetle, Question sınıfı, her sorunun oluşturulması, görüntülenmesi ve kullanıcının etkileşimleri ile yönetilmesini sağlayarak, kullanıcının cevaplarına göre anında geri bildirim almasını sağlar.

5. Quiz sistemi Kullanıcı Arayüzü

Quiz sisteminin kullanıcı arayüzü (UI), kullanıcı ihtiyaçlarını ve kullanıcı deneyimini göz önünde bulundurarak basit, sezgisel ve keyifli bir şekilde tasarlanmıştır. Ana hedef, kullanıcıların sistemi kolayca etkileşime geçebilmesi ve kesintisiz bir deneyim yaşaması için tasarımda karmaşıklıktan kaçınılmasıydı.

5.1. Tasarım ve Yapı

Arayüz minimal bir tasarıma sahip olup, modern görsel öğelerle sadeleştirilmiştir. Ekranda, kullanıcı hemen quiz'e dahil edilir; sorular, yanıt seçenekleri ve puan açıkça sunulur. Tasarım, kullanıcıların kolayca ve hızlıca gezinebileceği şekilde düşünülmüştür. Kullanıcı, her bir soruyu sırayla görür ve yanıt seçenekleri düğmeler olarak sunulur, böylece yanıt vermek yalnızca bir tıklama ile mümkündür.

Sorular okunabilir bir şekilde yerleştirilmiştir ve her yanıt seçeneği net bir şekilde ayrılmıştır, bu da herhangi bir kafa karışıklığını engeller. Arayüz öğelerinin yeterince büyük ve görünür olması, bilişsel stresin azalmasını sağlar.

5.2. Puan Yönetimi

Puan, arayüzün üst kısmında görüntülenir, bu sayede kullanıcı, gerçek zamanlı olarak ilerlemesini izleyebilir. Her sorudan sonra puan hemen güncellenir, böylece geri bildirim anında sağlanır. Bu, kullanıcıyı kesintisiz bir şekilde ilerlemeye teşvik eder.

Kullanıcı, bir soru setini (veya bir oturumu) tamamladığında, performansının bir değerlendirmesi bağlamsal mesajlar olarak görüntülenir: doğru cevap sayısına göre bir tebrik mesajı veya düşük puan durumunda soruları tekrar oynama önerisi. Bu, kullanıcıyı sürekli olarak gelişmeye teşvik eder.

5.3. Etkileşimler ve Geri Bildirim

Arayüzün belirgin özelliklerinden biri, her yanıt sonrası anında geri bildirim verilmesidir. Yanıt verildikten sonra, yanıtın doğru ya da yanlış olduğuna dair bir geri bildirim mesajı hemen sorunun yanında görünür, doğru yanıtlar için yeşil bir arka plan

ve yanlış yanıtlar için kırmızı bir arka plan ile görsel bir farkındalık sağlanır. Bu, her adımda kullanıcının performansına dair açık bir gösterge sunar.

Ayrıca, her oturum sonrasında, kullanıcıya yeni sorularla devam etmek isteyip istemediği ya da aynı soruları tekrar oynamak isteyip istemediği sorulan basit, anlaşılır pencereler gösterilir. Bu modallar, kolayca etkileşime geçilebilen, açıkça etiketlenmiş aksiyon düğmeleri içerir, böylece kullanıcı deneyimi sorunsuz ve keyifli olur.

5.4. Erişilebilirlik ve Duyarlılık

Tasarım, tamamen duyarlı (responsive) olacak şekilde optimize edilmiştir, yani kullanıcılar, masaüstü bilgisayardan mobil cihazlara kadar her tür cihazda, deneyimi bozmadan quiz'e erişebilirler. Kullanıcılar, herhangi bir cihazdan arayüze sorunsuz bir şekilde erişebilir.

Renk kontrastları, tüm ekran türlerinde iyi bir okunabilirlik sağlamak için dikkatlice seçilmiş olup, yazı tipi boyutları ekran boyutuna göre ayarlanabilir, böylece görünürlük problemi yaşanmaz.

5.5. Basitlik ve Verimlilik

Kullanıcı deneyimini maksimum düzeyde basitleştirmek, aynı zamanda quiz'in düzgün bir şekilde ilerlemesi için gerekli olan tüm işlevsellikleri korumak amaçlanmıştır. Düğmeler büyük, tıklanması kolaydır ve önemli bilgiler (sorular, puan, geri bildirim) açık bir şekilde sunulur. Kullanıcı hiçbir zaman kaybolmaz ya da kafa karışıklığına uğramaz, sadece quiz'e odaklanabilir.

Arayüz, kullanıcının quiz'de nerede olduğunu net bir şekilde bilmesini sağlar: kaç sorunun kaldığı, mevcut puanı ve sonraki adımların ne olduğunu (devam et veya bitir). Bu da deneyimi daha ilgi çekici ve stresiz hale getirir.