



ZAR OYUNU

PROJE ANALİZ RAPORU

AYŞE NUR ALTINTAŞ TANKÜL

İÇİNDEKİLER

1. GİRİŞ

- 1.1. Projenin Amacı ve Kapsamı
- 1.2. Teknolojik Yaklaşım ve Metodoloji
- 1.3. Analiz Raporunun Yapısı ve Hedefleri

2. SİSTEM GEREKSİNİMLERİ

- 2.1. Fonksiyonel Gereksinim
- 2.2. Fonksiyonel Olmayan Gereksinim
- 2.3. Sistem Kısıtları

3. AKTÖRLER

- 3.1. Aktör Etkileşim Detayları

4. SİSTEM MODELLERİ

- 4.1. Kullanım Durumu Modeli (Use Case)
- 4.2. Dinamik Modeller (Sequence & Activity)
- 4.3. Sınıf Modelleri (Class Diagram)

5. ARAYÜZ TASARIMI

- 5.1. Görsel Kimlik ve Tasarım Tercihleri
- 5.2. Kullanıcı Arayüzü Ekran Analizi
- 5.3. Cihaz Uyumluluğu ve Kullanım Kolaylığı

6.SONUÇ

1. GİRİŞ

1.1. Projenin Amacı ve Kapsamı

Bu analiz raporu, Yazılım İnşası prensipleri doğrultusunda geliştirilen "Web Tabanlı Dinamik Zar Oyunu Sistemi"nin tasarım, modelleme ve uygulama süreçlerini kapsamaktadır. Projenin temel amacı; geleneksel bir şans oyununu modern web teknolojileri (HTML5, CSS3, JavaScript) ile dijital ortama aktarırken, yazılım geliştirme yaşam döngüsünün analiz ve tasarım evrelerini akademik standartlarda uygulamaktır.

Uygulama, iki kullanıcının rekabetine dayalı, sıra tabanlı (turn-based) bir oyun motoru üzerine inşa edilmiştir. Sadece bir görsel simülasyon olmanın ötesinde; rastgele sayı üretim algoritmaları, asenkron olay yönetimi (event-driven programming), DOM (Document Object Model) manipülasyonu ve hata toleransı gibi kritik yazılım disiplinlerinin bütüncül bir uygulamasını temsil etmektedir.

1.2. Teknolojik Yaklaşım ve Metodoloji

Sistemin mimari yapısı oluşturulurken modülerlik ve sürdürülebilirlik ilkeleri ön planda tutulmuştur. Bu doğrultuda seçilen teknolojilerin projedeki stratejik rolleri şöyledir:

- HTML5:** Uygulamanın anlamsal iskeletini oluşturmak ve erişilebilirlik standartlarını sağlamak amacıyla kullanılmıştır.
- CSS3:** Kullanıcı deneyimini optimize eden görsel hiyerarşinin kurulması, :root değişkenleri ile merkezi tema yönetimi ve zar etkileşimlerini fizikal dünyaya yaklaşırıtan gelişmiş animasyonların modellenmesi için tercih edilmiştir.
- JavaScript (ES6+):** Sistemin mantıksal katmanını oluşturan JavaScript; veri doğrulama, oyun durum yönetimi (state management) ve ses motoru entegrasyonu gibi işlevleri yürüterek sistemin dinamik yapısını garanti altına almaktadır.

1.3. Analiz Raporunun Yapısı ve Hedefleri

Bu doküman, Karabük Üniversitesi Yazılım Mühendisliği standartları gözetilerek hazırlanmış olup; sistemin işlevsel (functional) ve işlevsel olmayan (non-functional) gereksinimlerini, UML tabanlı dinamik modellerini (Sequence ve Activity diyagramları) ve nesne yönelimli analiz çıktılarını detaylandırmaktadır. Hazırlanan bu analiz, yazılımın inşasından önce sistemin tüm olası senaryolarının (use cases) ve kısıtlarının teorik olarak doğrulanmasını hedeflemektedir.

2. SİSTEM GEREKSİNİMLERİ

2.1. Fonksiyonel Gereksinim

FR1 – Oyuncu ve Oturum Yönetimi

- Oyuncu Kaydı:** Sistem, açılış ekranında iki oyuncudan isim bilgilerini almalı ve bu bilgileri oyun süresince label bileşenlerinde saklamalıdır.
- Varsayılan Değer Atama:** İsim alanlarının boş bırakılması durumunda, sistem hata vermeden "Player 1" ve "Player 2" isimlerini otomatik olarak tanımlamalıdır.
- Asenkron Ekran Geçisi:** İsimler onaylandığında, giriş ekranı (Name Screen) gizlenmeli ve oyun alanı (Game Container) kullanıcıya sunulmalıdır.

FR2 – Zar Mekanizması ve Algoritmik Süreç

- Rastgele Değer Üretimi:** Her "Roll" tetiklendiğinde, JavaScript Math.random() kütüphanesi aracılığıyla 1 ile 6 arasında tamsayı üretilmelidir.
- Görsel Geri Bildirim:** Zar atma esnasında 650ms süren bir sallanma (shake) animasyonu oynatılmalı, ardından üretilen sayıya uygun görsel ekrana basılmalıdır.
- İşitsel Senkronizasyon:** Zar durduğunda ve kazanan belirlendiğinde ilgili ses dosyaları oynatılmalıdır.

FR3 – Oyun Mantığı ve Karar Mekanizması

- **Sıra Kontrolü (Turn-Based):** Karışıklığı önlemek için sistem, sırası olmayan oyuncunun butonunu pasif (disabled) hale getirerek oyun disiplinini sağlamalıdır.
- **Skor Karşılaştırması:** Her iki oyuncu zar attığında değerler kıyaslanmalı; yüksek atan oyuncunun skoru 1 artırılmalıdır.
- **Galibiyet Denetimi:** Her tur sonunda skorlar kontrol edilmeli, 3 puana ulaşan ilk oyuncu modal bir pencere ile "Şampiyon" ilan edilmelidir.

FR4 – Sistem Sıfırlama ve Kontrol

- **Oyun Yenileme:** "Play Again" veya "Refresh" komutu ile tüm skorlar, isimler ve görseller ilk haline döndürülmelidir.
- **Hata Yönetimi:** Dosya yükleme veya kullanıcı giriş hatalarında sistemin kararlılığı korunmalı ve kullanıcı bilgilendirilmelidir.

2.2. Fonksiyonel Olmayan Gereksinimler

NFR1 – Kullanılabilirlik (Usability)

- **Arayüz Tasarımı:** Kullanıcı dostu, sade ve sezgisel bir arayüz tasarımı hedeflenmiştir.
- **Tipografi ve Görsel Hiyerarşi:** "Indie Flower" ve "Lobster" fontları ile oyunun eğlence teması desteklenmiş; --accent rengi ile önemli butonlar vurgulanmıştır.
- **Kullanıcı Yönergeleri:** Oyunun her aşamasında (status paneli aracılığıyla) kullanıcıya sıradaki eylem hakkında net bilgi verilmektedir.

NFR2 – Performans (Performance)

- **Yanıt Süresi:** Zar atma sonrası skor hesaplama ve görsel güncelleme işlemleri 1 saniyenin altında tamamlanarak akıcı bir deneyim sağlanmaktadır.
- **Hafıza Kullanımı:** Uygulama, tarayıcı kaynaklarını minimum düzeyde tüketerek düşük donanımlı cihazlarda dahi performans kaybı yaşamadan çalışmaktadır.

NFR3 – Güvenilirlik ve Hata Toleransı (Reliability)

- Hata Yakalama:** Ses dosyalarının yüklenememesi veya oynatılamaması durumunda sistemin çökmesini engellemek için JavaScript try-catch blokları entegre edilmiştir.
- Veri Tutarlılığı:** Oyun sırasında sayfanın yenilenmesi veya kapatılması durumunda sistemin kendini güvenli bir şekilde sıfırlaması sağlanmıştır.

NFR4 – Taşınabilirlik ve Uyumluluk (Portability & Compatibility)

- Tarayıcı Uyumluluğu:** Uygulama; Chrome, Firefox ve Safari gibi modern web standartlarını destekleyen tüm güncel tarayıcılarda sorunsuz çalışmaktadır.
- Responsive Yapı:** CSS Flexbox ve max-width özelliklerini kullanılarak, oyunun hem masaüstü hem de mobil cihaz ekranlarında doğru ölçeklenmesi sağlanmıştır.

NFR5 – Bakım Kolaylığı (Maintainability)

- Modüler Kod Yapısı:** CSS tarafında :root değişkenleri, JavaScript tarafında ise bağımsız fonksiyonlar kullanılarak kodun gelecekte geliştirilmesi ve güncellenmesi kolaylaştırılmıştır.

3. AKTÖRLER

Sistemin işleyisi ve kullanıcı etkileşimleri temel olarak tek bir ana aktör üzerinden kurgulanmıştır.

Aktör	Rol	Açıklama
Oyuncu (Player)	Baş Aktör	Sistemin tüm fonksiyonlarını (isim girişi, zar atma, oyunu yeniden başlatma) aktif olarak kullanan kişidir.
Sistem	İkincil Aktör	Oyuncunun komutlarına yanıt veren; rastgele sayı üretimi, skor karşılaşması ve kazananın ilanı gibi otomatik süreçleri yürüten mekanizmadır.

3.1. Aktör Etkileşim Detayları

Sistem, yerel bir web uygulaması olarak tasarlandığı için tüm operasyonlar oyuncu tarafından manuel olarak başlatılır. Oyuncu, sistemle şu temel noktalarda etkileşime girer:

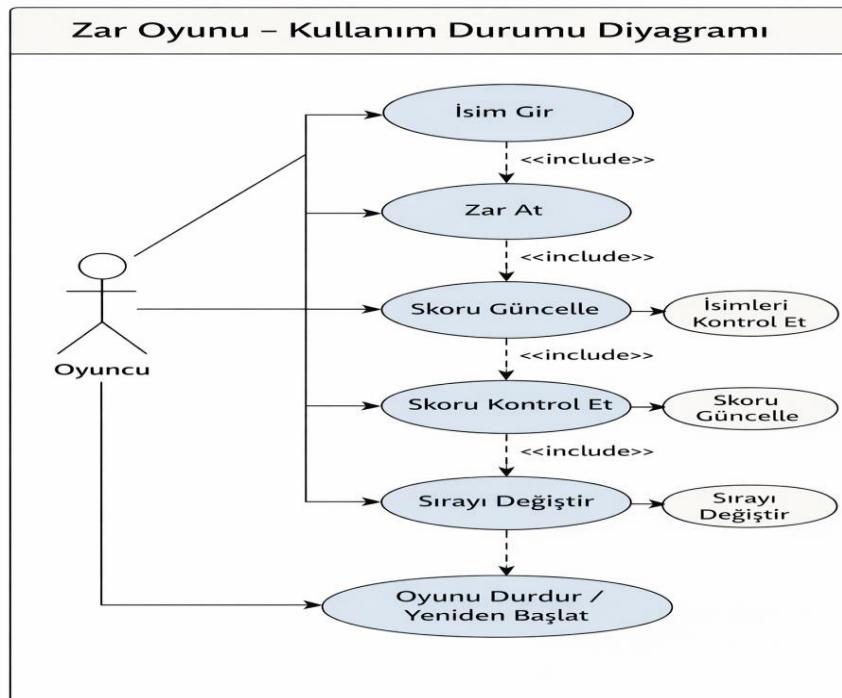
- **Veri Girişi:** Oyuncu, oyunun başlangıcında kimlik bilgilerini (isim) sisteme tanımlar.
- **Komut Tetikleme:** Zar atma eylemini butona basarak başlatır ve sistemden görsel/işitsel bir yanıt bekler.
- **Durum Kontrolü:** Oyun sonunda kazanan bilgisini doğrular ve sistemi istege bağlı olarak sıfırlar.

4. SİSTEM MODELLERİ

Bu bölüm, sistemin işleyişini, bileşenler arasındaki etkileşimi ve verinin işlenme sürecini görsel ve mantıksal modellerle açıklamaktadır.

4.1. Kullanım Durumu Modeli (Use Case Model)

Kullanım durumu modelleri, aktörün sistemle olan etkileşimlerini fonksiyonel bazda tanımlar.



4.1.1. Kullanım Durumu Tanımları

- **UC1 - İsim Kaydı ve Giriş:** Oyuncu, isimlerini girerek oyun oturumunu başlatır. Sistem, girilen veriyi label nesnelerine aktarır.
- **UC2 - Zar Atma (Roll Dice):** Oyuncu sırası geldiğinde butona basar. Sistem, rastgele sayı üretir ve animasyonu tetikler.
- **UC3 - Skor Takibi:** Sistem, üretilen sayıları karşılaştırır ve ilgili oyuncunun skorunu günceller.
- **UC4 - Kazananın İlanı:** Hedef skora ulaşıldığında sistem oyun akışını durdurur ve sonucu raporlar.

4.2. Dinamik Modeller

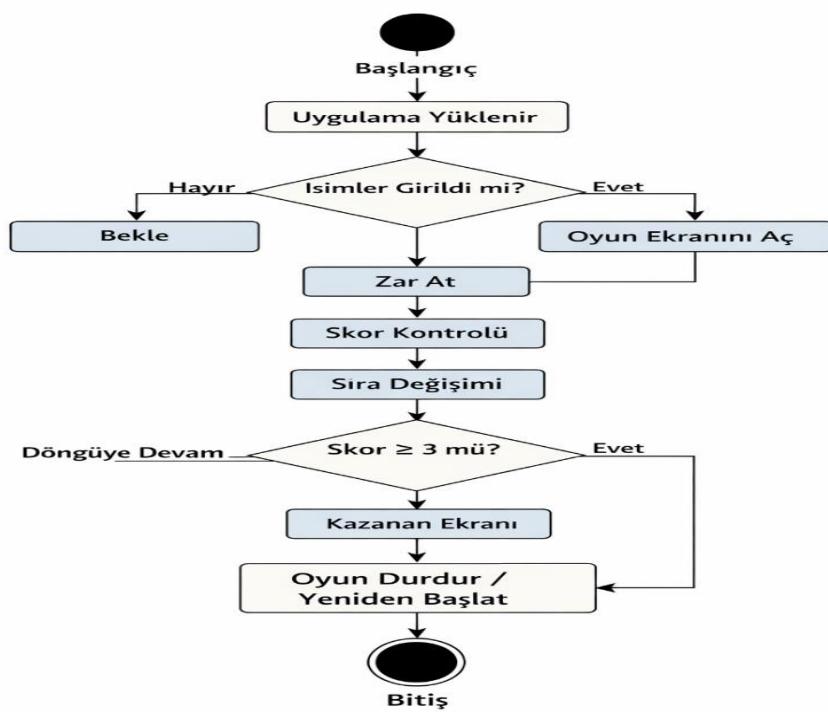
Dinamik modeller, sistemin zaman içerisindeki davranışını ve mesaj akışını temsil eder.

4.2.1. Sequence (Ardışılı) Diyagramı Açıklaması

Zar atma eylemi sırasında nesneler arasındaki mesaj trafiği şu sırayı izler:

1. **Oyuncu:** rollBtn nesnesine tıklama mesajı gönderir.
2. **UIController:** shake animasyon sınıfını diceImage nesnesine ekler ve diceSound nesnesini oynatır.
3. **GameLogic:** Math.random() fonksiyonunu çağırarak 1-6 arası değer üretir.
4. **Timer:** 650ms gecikme (asenkron bekleme) sonrası animasyon sınıfını kaldırır.
5. **DOMManager:** Üretilen değere göre zarın src bilgisini ve oyuncu puanını (textContent) günceller.

4.2.2. Aktivite Diyagramı (Activity Diagram) Akışı



Sistemin genel iş akışı şöyledir:

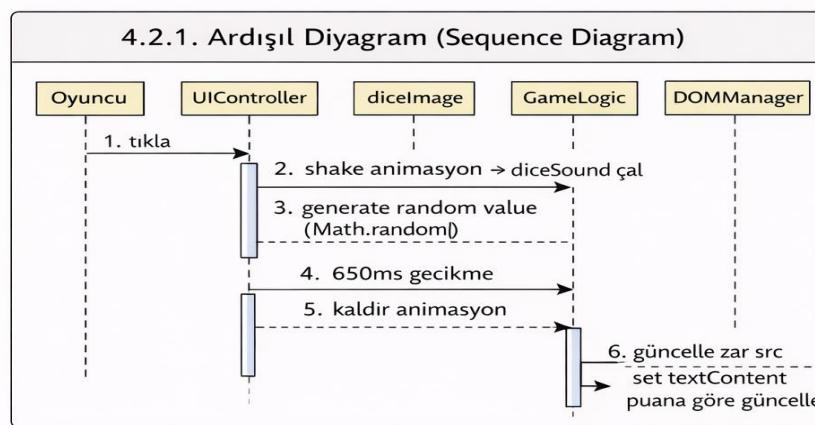
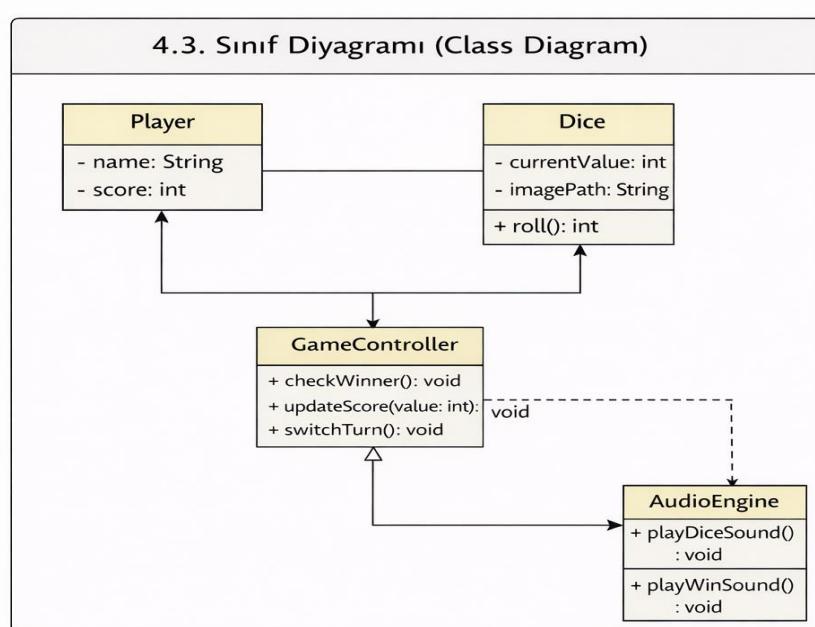
- Başlangıç:** Uygulama yüklenir.
- Karar:** İsimler girildi mi? (Hayır -> Bekle / Evet -> Oyun Ekranını Açı).
- Döngü:** Zar Atma -> Skor Kontrolü -> Şıra Değişimi.
- Karar:** Skor ≥ 3 mü? (Hayır -> Döngüye Devam / Evet -> Kazanan Ekranını Göster).
- Bitiş:** Oyun durdurulur veya yeniden başlatılır.

4.3. Sınıf Modelleri (Class Models)

Sistem, nesne yönelimli programlama (OOP) prensiplerine uygun olarak şu mantıksal sınıflardan oluşur:

- Player (Oyuncu Sınıfı):** name ve score özniteliklerini (attribute) barındırır.

- **Dice (Zar Sınıfı)**: currentValue ve imagePath özniteliklerine sahiptir; roll() metodunu içerir.
- **GameController (Oyun Yöneticisi)**: checkWinner(), updateScore() ve switchTurn() metodları ile oyunun merkezi mantığını yönetir.
- **AudioEngine**: playDiceSound() ve playWinSound() metodları ile multimedya yönetimini sağlar.



5. ARAYÜZ TASARIMI (INTERFACE DESIGN)

Bu bölümde, oyunun dış görünüşü, kullanıcıyı yönlendirme biçimini ve ekranlar arasındaki geçiş mantığı incelenmektedir. Tasarımda sadelik ve akıcılık temel

alınarak, oyuncunun dikkatinin doğrudan oyun sürecine odaklanması hedeflenmiştir.

5.1. Görsel Kimlik ve Tasarım Tercihleri

Oyunun genel atmosferi, kullanıcıya modern ve dijital bir oyun salonu hissi verecek şekilde kurgulanmıştır:

- Renk Seçimi:** Arka planda koyu lacivert tonları tercih edilerek, oyunun ana öğeleri olan açık renkli zarların ve yazıların belirginleşmesi sağlanmıştır. Bu sayede görsel bir derinlik ve odak noktası oluşturulmuştur.
- Yazı Tipleri:** Başlıklarda kullanılan estetik yazı stili oyunun eğlenceli tarafını vurgularken, skor tablosundaki el yazısı tarzındaki fontlar ise samimi bir oyun ortamı yaratmaktadır.
- Buton Tasarımı:** Etkileşimli öğeler, üzerlerine gelindiğinde hafifçe yükselerek tepki vermektedir, bu da kullanıcıya "tıklanabilirlik" hissini fiziksel bir geri bildirimle hissettirmektedir.

5.2. Kullanıcı Arayüzü Ekran Analizi

5.2.1. Karşılama ve Kayıt Ekranı

ZAR OYUNU
Oyuncu 1 Adı : _____
Oyuncu 2 Adı : _____
[Oyunu Başlat]

Uygulama başlatıldığında oyuncuyu karşılayan ilk bölümdür. Bu ekranın tek bir görevi vardır: Oyuncuları oyuna hazırlamak.

- Giriş Alanları:** İki oyuncu için ayrılmış bölgelere isimler yazılır. Tasarım, isim yazılacak kutuları sayfanın merkezine alarak kullanıcının kafasının karışmasını önler.
- Başlatma Butonu:** İsimler girildikten sonra basılan bu buton, bir kapı görevi görerek giriş ekranını kapatır ve ana oyun alanını saniyeler içinde açar.

5.2.2. Oyun Sahnesi ve Zar Alanı

Oyunun asıl heyecanının yaşandığı, zarların ve puanların yer aldığı ana bölümdür.

Oyuncu 1		Oyuncu 2
Zar: []		Zar: []
Skor: 1		Skor: 2
[Zar At]		
Durum: Sıra Oyuncu 1'de		

- İki Sütunlu Yapı:** Ekranın solu birinci oyuncuya, sağ ise ikinci oyuncuya ayrılmıştır. Bu simetrik yapı, her iki tarafın da puanını ve zarlarını aynı anda görmesini sağlayarak adil bir izleme imkanı sunar.
- Zar Görselleri:** Zarlar, arka plandan ayrılacak şekilde yumuşak fildiği tonlarında tasarlanmıştır. Zar atıldığından gerçekteşen sarsılma hareketi, kullanıcıya gerçek bir zar atma eyleminin enerjisini hissettirir.
- Durum Bildirimi:** Ekranın alt kısmında bulunan bilgi kutusu, sürekli olarak "Sıra Veli'de" veya "Ali 6 attı" gibi mesajlar vererek oyuncuları yönlendirir.

Durum: Ali 6 attı
Durum: Sıra Veli'de

5.2.3. Kazananı Kutlama Paneli

Oyun sona erdiğinde (bir oyuncu 3 puana ulaştığında) devreye giren özel bildirim ekranıdır.

 KAZANAN: ALİ 
[Tekrar Oyna]
[Çıkış]

- Odaklıma Etkisi:** Arka plan hafifçe karartılarak, sadece kazananın isminin yazdığı kutlama penceresi öne çıkarılır.

- **Tekrar Oyna Seçeneği:** Bu pencere üzerinde yer alan buton, oyuncuların sayfayı yenilemeye uğraşmadan tek tıkla yeni bir maça başlamasını sağlar.

5.3. Cihaz Uyumluluğu ve Kullanım Kolaylığı

Tasarım, sadece bilgisayar ekranlarında değil, telefon ve tabletlerde de aynı akıcılıkla çalışacak şekilde esnek bir yapıdadır. Ekran daraldığında bileşenler birbirinin üzerine binmez, aksine cihazın boyutuna göre kendiliğinden hizalanır. Bu durum, kullanıcının cihaz fark etmeksızın oyunu her yerde aynı kalitede oynamasını garanti altına alır.

6. SONUÇ

Bu çalışma kapsamında gerçekleştirilen **Dijital Zar Oyunu Sistemi**, Yazılım İnşası prensipleri ve Nesne Yönetimli Analiz ve Tasarım teknikleri temel alınarak kurgulanmış kapsamlı bir web uygulamasıdır. Projenin her aşamasında; gereksinim analizi, kullanıcı senaryoları, dinamik modelleme ve kullanıcı arayüzü tasarımları bir bütün olarak ele alınmış; sistemin hem işlevsel hem de yapısal açıdan tutarlı bir biçimde modellenmesi sağlanmıştır.

Sistemin analiz aşamasında oluşturulan **Kullanım Senaryoları (Use Cases)**, oyuncu ile sistem arasındaki tüm etkileşimleri açık ve hiyerarşik bir yapı içinde tanımlamıştır. Bu sayede projenin kapsamı netleşmiş, oyuncu isimlerinin alınmasından kazananın ilan edilmesine kadar olan tüm süreçlerin işlevsel çerçevesi çizilmiştir.

Ardından hazırlanan **Sınıf Modelleri ve Nesne Yapıları**, sistemde yer alan varlıkların (Oyuncu, Zar, Oyun Kontrolcüsü) özelliklerini ve birbirleriyle olan ilişkilerini somutlaşтыrmış; böylece yazılımın modüler mimarisinin temelleri atılmıştır. Bu yapı, uygulamanın gelecekte "çoklu zar" veya "online mod" gibi özelliklerle genişletilmesine imkan tanıyan sürdürülebilir bir altyapı sunmaktadır.

Süreçlerin dinamik akışını anlatan **Sequence (Ardışılı) Diyagramları**, zar atma eyleminden skor güncellemeye kadar olan operasyonların adım adım nasıl işlediğini ortaya koymustur. Özellikle JavaScript tarafından rastgele sayı üretim mantığı ile CSS animasyonlarının senkronizasyonu, iş kurallarının eksiksiz

uygulanması açısından kritik öneme sahip kontrol noktalarını modellemiştir. Bu süreçlere ek olarak hazırlanan **Activity (Aktivite) Diyagramları**, sistemin genel faaliyet yapısını başlangıçtan bitiş düğümüne kadar mantıksal bir bütünlük içinde açıklamıştır.

Projenin **Arayüz Tasarımı**, kullanıcı deneyimini (UX) ön planda tutan, sade ve işlevsel bir yapı sunmaktadır. Çok aşamalı giriş süreci ve sıra tabanlı oyun akışı, kullanıcıyı doğru yönlendirecek şekilde tasarlanmıştır; sistem genelinde tutarlı bir etkileşim modeli sağlanmıştır.

Sonuç olarak, ortaya çıkan bu sistem; yazılım geliştirme yaşam döngüsüne uygun olarak analiz ve tasarım aşamalarının tümünden başarıyla geçmiş; teknik açıdan doğru ve mantıksal olarak tutarlı bir mimari sunmaktadır. Tüm UML diyagramları, iş kuralları ve kullanıcı arayüzü tasarımlarıyla birlikte düşünüldüğünde bu proje, yazılım mühendisliğinin temel ilkelerini uygulamalı olarak ortaya koyan bütüncül bir çalışma niteliği taşımaktadır.

