

GAZİ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ



BM495/ BM496 BİLGİSAYAR
PROJESİ

SOFTWARE DESIGN DOCUMENT (SDD)
Sihirli Ayna

Doç Dr. Murat Yılmaz

191180758-Berat Berkay Erken
191180086-Ahmed Senih Yıldırım
191180008-Tuğba Akın

Kelime Sayısı:2838

Aralık 2023

İNTİHAL BEYANI

Bu çalışmadaki tüm bilgilerin akademik kurallara ve etik davranışa uygun olarak alındığını ve sunulduğunu ve bu belgede alıntı yaptığımı belirttiğim yerler dışında sunduğum çalışmanın kendi çalışmam olduğunu, Yükseköğretim Kurumları Bilimsel Araştırma ve Yayın Etiği Yönergesinde belirtilen bilimsel araştırma ve yayın etiği ilkelerine uygun olduğunu beyan ederim.

Numara: 191180758, 191180086, 191180008

Ad Soyad: Berat Berkay Erken, Ahmed Senih Yıldırım, Tuğba Akın

Tarih: 06/11/2023

Ahmed Senih Yıldırım

Tuğba Akın

Berat Berkay Erken


İmza:



İmza:



İmza:



İÇİNDEKİLER

1. GİRİŞ

1.1. Amaç

1.2. Hedef Kitle

1.3. Kısaltmalar ve Tanımlar

1.4. Referanslar

1.5. Genel Bakış

2. İLGİLİ DOKÜMANLAR

3. KAVRAMSAL MODEL

3.1. Bağlam Üzerinde Yazılım Tasarımı

3.2. Yazılım Yaşam Döngüsü İçerisinde SDD

3.2.1. SDD Hazırlamadaki Etkiler

3.2.2. Yazılım Yaşam Döngüsü Ürünleri Üzerindeki Etkiler

3.1. Gerekli Durum ve Modlar

3.2. Fonksiyonel Gereksinimler

3.3. Dış Arayüz Gereksinimleri

3.4. Dahili Arayüz Gereksinimleri

3.5. Dahili Veri Gereksinimleri

3.6. Performans Gereksinimleri

3.7. Diğer Gereksinimler

3.8. Gereksinimlerin Önceliği ve Kritikliği

4. YAPISAL TASARIM

4.1. Tasarımın Paydaşları

4.2. Tasarımın Bakış Açıları

4.3. Tasarım Öğeleri

4.4. Tasarım Gerekçeleri

4.5. Tasarım Dilleri

5. TASARIM BAKIŞ AÇILARI

5.1. Bağlam Bakış Açısı

5.1.1. Tasarım Öğeleri

5.1.2. Örnek Diller

5.2. Kompozisyon Bakış Açısı

5.2.1. Tasarım Öğeleri

5.2.2. Örnek Diller

5.3. Etkileşim Bakış Açısı

5.3.1. Tasarım Öğeleri

5.3.2. Örnek Diller

5.4. Mantıksal Bakış Açısı

5.4.1. Tasarım Öğeleri

5.4.2. Örnek Diller

6. SDD DETAYLI PLANI

6.1. Giriş ve Hedef Belirleme

6.2. Kavramsal Model Oluşturma

6.3. Yapısal Tasarım Belirlenmesi

6.4. Tasarım Bakış Açıları Oluşturma

6.5. Gereksinim Analizi ve Önceliklendirme

6.6. Dahili ve Dış Arayüzlerin Tanımlanması

6.7. Dahili Veri Yapılarının ve Performans Gereksinimlerinin Belirlenmesi

6.8. Güvenlik ve Gizlilik Politikalarının Oluşturulması

6.9. Detaylı Planın Dokümantasyonu

6.10. Ekler ve İnceleme

7. EKLER

1. GİRİŞ

1.1. Amaç

Bu belgenin temel amacı, Sihirli Ayna projesinin tasarımını ve mimarisini çeşitli bakış açılarıyla açıklamak ve bu açıklamaları görsel olarak sunmaktır. Yazılım Gereksinimleri Belgesi'nde belirtilen gereksinimleri karşılamak üzere yapılandırılmış bir yazılım sistemini tanımlama hedeflemektedir. Bu doküman, uygulama aşamasında başvurulacak temel kaynak olacak ve projenin nihai durumu hakkında detaylı bilgiler içerecektir.

1.2. Hedef Kitle

Bu proje, genellikle bilgiye dayalı eğlence ve analitik düşünmeyi seven ve herhangi bir yaş aralığına bakmaksızın, sohbet etmeyi seven bütün oyunseverler için tasarlanmıştır.

1.3. Kısaltmalar ve Tanımlar

API	Application Programming Interface - Yazılım Uygulama Arayüzü
GUI	Graphical User Interface - Grafiksel Kullanıcı Arayüzü
Turing Testi	Bir yapay zekanın insan benzeri davranışları sergileyip sergilemediğini değerlendiren bir test
ChatGPT	OpenAI tarafından geliştirilen doğal dil işleme modeli

1.4. Referanslar

Bu projenin geliştirilmesi sırasında aşağıdaki kaynaklardan yararlanılmıştır:

Bu proje için önceden hazırlanmış olan Software Requirement Specification Dokümanına(SRS v1.0) atıfta bulunulmuştur. Ayrıca Bu dokümanın hazırlanmasında IEEE Software Design Document Template baz alınmıştır. Ulaşım adresi ise <https://ieeexplore.ieee.org/document/278258> "IEEE Guide to Software Design Descriptions," in IEEE Std 1016.1-1993, vol., no., pp.1-32, 25 May 1993, doi: 10.1109/IEEESTD.1993.119206.

1.5. Genel Bakış

Bu rapor, "Sihirli Ayna" projesinin Yazılım Tasarım Dokümanıdır. Projeyi temel alan bu belge, projenin amacını, bileşenlerini, tasarım bakış açılarını ve detaylı planını içermektedir.

Projenin amacı, kullanıcılara tarihi karakterlerle etkileşim ve analitik düşünme becerilerini geliştirme imkânı sunan "Sihirli Ayna" adlı yazılımı oluşturmaktır. ChatGPT API'si ve kullanıcı ara yüzü, projenin ana bileşenlerini oluşturur. Bu doküman, yazılım yaşam döngüsündeki rolü, gereksinimleri, tasarım bakış açılarını ve detaylı planı detaylı bir şekilde açıklamaktadır. Ayrıca, tasarım bakış açıları, tasarım öğeleri ve örnek diller ile projenin yapısı detaylı bir şekilde açıklanmıştır. Bu doküman, proje ekibinin tasarım sürecini anlamak ve diğer ilgili taraflara projeyi açıklamak için kullanılacak temel bir kaynaktır.

2. İLGİLİ DOKÜMANLAR

Bu proje dokümanında daha önceden hazırlanmış olan Software Requirements Specification (SRS v1.0) dokümanlarına atıfta bulunulmuştur.

3. KAVRAMSAL MODEL

Bu bölümde Software Design Document için kavramsal model sunulacaktır. Aynı zamanda, SDD’de bulunan kavramlar ve bağlam hakkında bilgiler yer alacaktır.

3.1. Bağlam Üzerinde Yazılım Tasarımı

Bu projenin yazılım tasarımı, iki ana bileşenden oluşmaktadır. Bunlardan ilki, ChatGPT API'sini entegre ederek ses giriş ve çıkış işlemlerini yönetme üzerine kurulmuştur. Kullanıcının sesli komutlarını anlamak, ChatGPT API'sini kullanarak bu komutlara cevap üretmek ve cevabı diğer bileşene iletmekle görevlidir. Uygulamanın diğer bir ana bileşeni ise, kullanıcı arayüzünü içerir ve tarihi karakterlerle etkileşimde bulunmayı sağlar. Kullanıcı, arayüz üzerinden bir tarihi karakter seçer, karakterin kendini tanıtmalarını dinler ve ardından sesli olarak karaktere yanıt verir. Bileşenler arasındaki etkileşim, kullanıcının sesli girişini almak, işlemek ve cevap almak için sürekli bir bağlam oluşturur. Projede kullanılan ChatGPT API'si, OpenAI tarafından sağlanmaktadır, bu da projenin dış bağlamını oluşturur ve yapay zeka ile etkileşim sağlayan temel bir bileşendir. Sesli giriş ve çıkışın yanı sıra, tarihi karakterlerle etkileşime geçme ve analitik düşünme yeteneklerini geliştirme hedefleri, tasarımın odak noktalarını oluşturur.

3.2. Yazılım Yaşam Döngüsü İçerisinde SDD

Bu bölüm projenin yazılım geliştirme süreçlerini kapsayan yazılım tasarımının detaylarını içermektedir. Yazılım Yaşam Döngüsü içerisinde önemli bir yeri olan SDD, projenin genel yapısı ve işleyişi ve içerisinde bulunan bileşenler arasındaki ilişkileri ayrıntılı bir şekilde işlemektedir. Ayrıca, yazılımın işlevselliğini performans gereksinimlerini güvenlik önlemlerini modüler yapısını ve diğer önemli tasarım unsurlarını içermektedir.

3.2.1. SDD Hazırlamadaki Etkiler

SDD Hazırlamadaki Etkiler, tasarım kararlarının ve tasarım sürecinin genel etkilerini değerlendirme ve takibini kolaylaştırma gibi nedenlerden dolayı önemlidir. Entegrasyon bağlantıları, kullanılabilirlik ve arayüz tasarımı, performans etkileri gibi konular hakkında fikir sahibi olunmasına yardımcı olur.

3.2.2. Yazılım Yaşam Döngüsü Ürünleri Üzerindeki Etkiler

SDD, yazılımın genel yapısını, işleyişini ve bileşenler arasındaki ilişkileri detaylı bir şekilde inceleyerek, yazılım yaşam döngüsü içinde kritik bir rol oynamaktadır. Bu çerçevede, SDD'nin hazırlanma sürecindeki etkiler, entegrasyon bağlantıları, kullanılabilirlik, arayüz tasarımı, performans ve modüler yapı gibi konulara odaklanarak, yazılımın diğer yaşam döngü ürünleriyle uyumlu ve etkili bir biçimde entegre edilmesine katkı sağlamaktadır. Bu değerlendirme, SDD'nin proje süreçlerine olan katkılarını ve diğer yazılım yaşam döngü ürünleri üzerindeki etkilerini bütünlük içinde ele almaktadır.

3.1. Gerekli Durum ve Modlar

Proje, karakter seçim ekranı, sohbet ekranı ve ayarlar ekranı olmak üzere temel üç durumda çalışacaktır. Kullanıcı karakter seçim ekranında seçim yaparak sohbet ekranına geçiş yapabilecektir.

3.2. Fonksiyonel Gereksinimler

Kullanıcılar, tarihi karakterlerle metin tabanlı bir arayüz üzerinden iletişim kurabilecektir.

Kullanıcılar, karakterlerin sorduğu bilmeceleri çözmeye çalışması istenecektir.

Kullanıcılar, sesli giriş ve sesli çıkış özelliklerini kullanarak karakterlerle iletişimde bulunabilecektir.

3.3. Dış Arayüz Gereksinimleri

ChatGPT API, projenin temel iletişim aracı olacaktır. API'nin doğru şekilde çalışabilmesi için internet bağlantısı gereklidir.

3.4. Dahili Arayüz Gereksinimleri

Kullanıcı Arayüzü: Unity veya Unreal Engine tabanlı grafik arayüzü, kullanıcıların oyun içi etkileşimde bulunmasını sağlayacaktır.

Oyun Motoru: Kullanıcı arayüzü ile entegre çalışarak karakterlerin hareketini ve iletişimini sağlayacaktır.

3.5. Dahili Veri Gereksinimleri

Kullanıcıların isim ve soy isim bilgileri veri tabanında tutulacaktır.

3.6. Performans Gereksinimleri

Oyunda karakterlerin yanıt süresi, kullanıcı eylemlerine hızlı bir şekilde cevap vermelidir (örneğin, 2 saniye içinde).

Oyunda kullanıcıların karakterlere ileteceği yanıtlar hızlı bir şekilde gerçekleşmelidir (örneğin, 2 saniye içinde).

Oyunda kullanıcıların karakterleri seçmesi ardından karakterlerin kendileri hakkında bilgi verme işlemin başlaması kısa bir sürede olmalıdır (örneğin, 4 saniye içinde).

3.7. Diğer Gereksinimler

Güvenlik: Kullanıcı bilgilerinin güvenliği için gerekli önlemler alınmalıdır.

Gizlilik: Kullanıcı verileri, gizlilik politikalarına uygun bir şekilde işlenmelidir.

3.8. Gereksinimlerin Önceliği ve Kritikliği

- Temel iletişim işlevselliği (yüksek öncelikli)
- Bilmecelerin doğru işlemesi (yüksek öncelikli)
- Sesli giriş ve çıkışın doğru çalışması (yüksek öncelikli)
- Güvenlik önlemlerinin uygulanması (orta öncelikli)
- Performans optimizasyonları (orta öncelikli)
- Gizlilik politikalarının uygulanması (düşük öncelikli)

4. YAPISAL TASARIM

Bu bölümde, tasarım ve geliştirme sürecini desteklemek için bir dizi tasarım görünümü sunulacaktır. Bu proje iki kısım halinde gerçekleştirilecektir. Biri, ChatGPT API'sinin ve audio input alma işlemi ve audio output verme işlemlerinin gerçekleşeceği backend. İkincisi, karakter

görüntülerinin olduğu arayüz ve bu karakterlerin konuşma animasyonlarının gerçekleşeceği frontend. Tasarım görünümüleri ve ilgili görünümeler bölüm 5’te ayrıntılı olarak açıklanmaktadır.

4.1. Tasarımın Paydaşları

Bu projenin paydaşları arasında oyunun geliştiricileri, kullanıcıları ve ChatGPT API’sini sağlayan OpenAI bulunmaktadır. Geliştiriciler, oyunun tasarım ve geliştirme sürecinden sorumludur. Kullanıcılar, oyunun son kullanıcılarıdır ve onların deneyimleri, oyunun başarısını belirler. OpenAI, ChatGPT API’sini sağlar ve bu, oyunun yapay zeka ile iletişim kurmasını sağlar.

4.2. Tasarımın Bakış Açıları

Tasarım, kullanıcı dostu bir arayüz oluşturmayı ve kullanıcıların tarihi karakterlerle etkileşimde bulunabileceği bir ortam sağlamayı hedeflemektedir. Ayrıca, kullanıcıların analitik düşünme yeteneklerini geliştirmek için bilmece sunmayı amaçlamaktadır. Bu bakış açısı, kullanıcıların hem eğlenceli hem de öğretici bir deneyim yaşamasını sağlamayı hedefler.

4.3. Tasarım Öğeleri

Tasarımın temel öğeleri arasında kullanıcı arayüzü, tarihi karakterler, bilmece, bulmacalar ve sesli giriş seçeneği bulunmaktadır. Kullanıcı arayüzü, kullanıcıların oyunla etkileşim kurmasını sağlar. Tarihi karakterler, kullanıcıların sohbet edebileceği ve etkileşimde bulunabileceği karakterlerdir. Sesli giriş seçeneği, kullanıcıların sesli olarak yanıt vermesini sağlar. Uygulama kullanıcısı ilk önce arayüzden bir tarihi karakter seçer ve daha sonrasında bu karakterin kendini tanıtmayı beklenir sonrasında kullanıcıdan bir input beklenir böylece sohbet başlar. Eğer çok fazla input gecikmesi varsa arayüze tekrar geçiş yapılır.

4.4. Tasarım Gereksinimleri

Tasarımın amacı, kullanıcıların hem eğlenceli hem de zihin açıcı bir deneyim yaşamasını sağlamaktır. Bu nedenle, tasarım, kullanıcıların tarihi karakterlerle etkileşimde bulunabileceği ve aynı zamanda analitik düşünme yeteneklerini geliştirebileceği bir ortam oluşturmayı hedeflemektedir. Bu, kullanıcıların oyunu oynarken aynı zamanda öğrenmelerini sağlar.

4.5. Tasarım Dilleri

Sihirli Ayna projesinin akış diyagramı EK-1’de yer almaktadır

5. TASARIM BAKIŞ AÇILARI

5.1. Bağlam Bakış Açısı

Sistemin bağlam bakış açısı, sistem ve çevresi (kullanıcılar, sistemler vb.) arasındaki ilişkileri, bağımlılıkları ve etkileşimleri tanımlar. Sistemin ne yaptığını, kendisi ile dış dünya arasındaki sınırların nerede olduğunu ve sistemin bu sınırlar boyunca diğer sistemler, organizasyonlar ve insanlarla nasıl etkileşime girdiğini tanımlar. Sistemin bağlamı Şekil 1’de verilmiştir.

5.1.1. Tasarım Öğeleri

Uygulamamızda, uygulamanın kullanıcısı olan bir aktör vardır. Önce sisteme giriş yapılmalı ve ardından ilk adım olan karakter seçim ekranından sonra Ek-1’de yer alan kullanım durumlarından geçer.

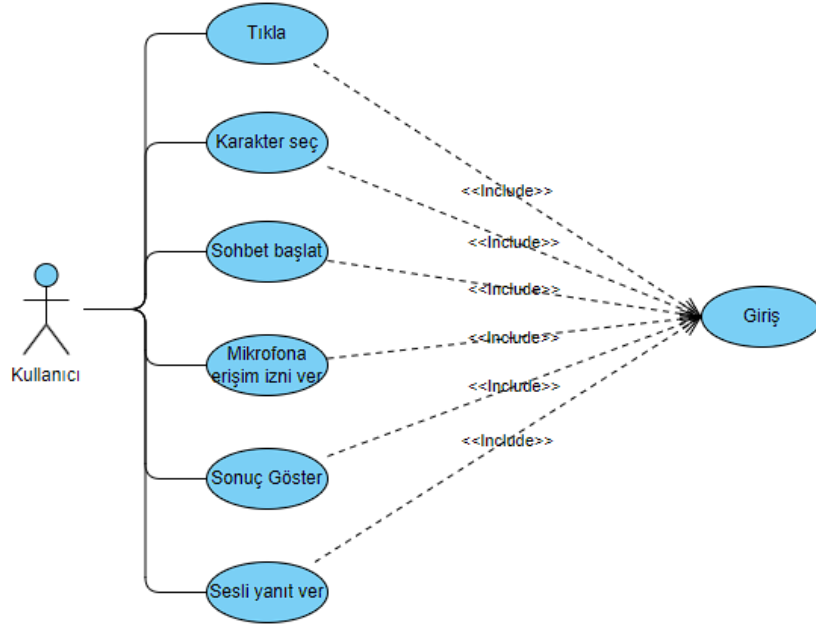
Senaryo:

1. Tıkla: Kullanıcı ana ekranındaki ‘tarihi karakter seç’ butonuna tıklar.

2. Karakter Seç: Kullanıcı, sunulan tarihi karakterler arasından birini seçer.
3. Sohbet Başlat: Karakterin kendisini tanıtmasıyla sohbet başlar.
4. Mikrofona Erişim İzni Ver: Kullanıcının ilk kullanımda sesli giriş yapabilmesi için mikrofona erişim izni vermesi gerekir.
5. Sesli Yanıt Ver: Kullanıcı, mikrofon arayüzünü kullanarak sesli yanıt verir.
6. Sonuç Göster: Sistem, kullanıcının yanıtını ChatGPT API'na atarak dönen sonucu gösterir.

5.1.2. Örnek Diller

Bu bölümde kullanım durumu (use case) diyagramı ve açıklamalı tablosu bulunmaktadır.



Şekil 1: Sihirli Ayna: ChatGPT Entegreli Sohbet Oyunu Sistem Use Case Diyagramı

Use Case ID	1
Use Case	Tıkla
Aktör	Kullanıcı
Açıklama	Arayüzden karakter seçim ekranına geçilir.
Tetikleyici (Trigger)	Kullanıcı butona tıklar.
Birincil Senaryo	-Kullanıcı karşısına gelen butondan karakter seçim ekranına geçer.
Olağanüstü Senaryo	-Kullanıcı butona tıklamadoğı sürece diğer arayüzlere geçemez.

Tablo 1: Tıkla Use Case

Use Case ID	2
Use Case	Karakter Seç
Aktör	Kullanıcı
Açıklama	Karakter seçme ekranındaki karakterlerden biri seçilir.
Tetikleyici (Trigger)	Kullanıcı karakterin üstüne tıklar.
Birincil Senaryo	-Kullanıcı karakter seçme arayüzündeki karakterleri inceledikten sonra sohbet etmek istediği karakterin üzerine tıklar.
Olağanüstü Senaryo	-Kullanıcı karakterlerden birini seçmedikçe sohbet ekranına geçilmeyecektir.

Tablo 2: Karakter Seç Use Case

Use Case ID	3
Use Case	Sohbet Başlat
Aktör	Kullanıcı
Açıklama	Kullanıcı sohbet etmek istediği karakteri seçtikten sonra, karakterin kendisini tanıtmasını bekler ve sonrasında sohbeti başlatmak için mikrofon butonuna basılır.
Tetikleyici (Trigger)	Kullanıcı mikrofon butonuna tıklar.
Birincil Senaryo	-Kullanıcı sohbet ekranında karakterin kendisini tanıtmayı bittikten sonra karşısına gelen mikrofon butonunun aktif olması ile butona basarak sesli input verecektir.
Olağanüstü Senaryo	-Kullanıcı mikrofon butonuna basmadıkça sohbet devam etmeyecektir ve arayüz ekranına bir süre sonra geçiş yapacaktır.

Tablo 3: Sohbet Başlat Use Case

Use Case ID	4
Use Case	Mikrofona Erişim İzni Ver
Aktör	Kullanıcı
Açıklama	Kullanıcı mikrofon için erişim izni verir.
Tetikleyici (Trigger)	Kullanıcı pop-up olarak çıkan mikrofon erişimine izin ver butonuna tıklar.
Birincil Senaryo	-Kullanıcı mikrofona erişim izni verdikten sonra sesli olarak input verebilecektir. -Sesli input verdikten sonra ChatGPT API'sına istek gönderilir.
Olağanüstü Senaryo	-Kullanıcı mikrofona erişim izni vermediği sürece sesli input gönderemeyecektir. -Sesli input text olarak çevrilemeyecektir. -Text ChatGPT API'sine istek olarak gönderilemeyecektir.

Tablo 4: Mikrofona Erişim İzni Ver Use Case

Use Case ID	5
Use Case	Sonuç Göster
Aktör	Sistem
Açıklama	Sistem ChatGPT'den gelen sonucu gösterir.
Tetikleyici (Trigger)	Kullanıcın sesli input göndermesi.
Birincil Senaryo	-Sistem ChatGPT'den gelen texti ekranda gösterir. -Text aynı zamanda Google text-to-speech API'sına gider.
Olağanüstü Senaryo	-İnternet bağlantısı stabil olmadığında sonuç geç döner. -Sesli input yanlış bir şekilde çevrilmiş olabilir. Bunun sonucunda sonuç anlamsız olabilir. -API'da sorun olabilir. Bu yüzden sonuç çıkmayabilir.

Tablo 5: Sonuç Göster Use Case

Use Case ID	6
Use Case	Sesli Yanıt Ver
Aktör	Sistem
Açıklama	Sonucu Google text-to-speech API sayesinde sesli outputa çevirir.
Tetikleyici (Trigger)	ChatGPT çıktısı.
Birincil Senaryo	-ChatGPT API'sından çıkan sonuç Google text-to-speech API'sına gönderilir. Sesli output oluşturulur. -Çıkan sonuç ekranda gösterilirken aynı zamanda sesli bir şekilde output edilir.
Olağanüstü Senaryo	-ChatGPT API'sından çıkan sonuç Google text-to-speech API'sına gönderilirken bir hata oluşur. Bu durumda, sistem hata mesajını kullanıcıya gösterir ve sesli çıktı oluşturulamaz. -Sesli çıktı oluşturulur ancak kullanıcıya iletilemez. Bu durumda, sistem bir hata mesajı gösterir ve kullanıcıya sesli çıktının neden sunulmadığını açıklar. Örneğin, kullanıcının sesli çıktıyı alabilmesi için gerekli donanım (hoparlörler, kulaklıklar vb.) olmayabilir veya düzgün çalışmıyor olabilir.

Tablo 6: Sesli Yanıt Ver Use Case

Use Case ID	7
Use Case	Giriş
Aktör	Kullanıcı
Açıklama	Kullanıcı uygulamayı açar.
Tetikleyici (Trigger)	Kullanıcı uygulamayı çalıştırır.
Birincil Senaryo	-Kullanıcı uygulamayı yükler.
Olağanüstü Senaryo	-Uygulamanın yüklenmemesi durumudur.

Tablo 7: Giriş Use Case

5.2. Kompozisyon Bakış Açısı

Bu bakış açısı, oyunun farklı bileşenlerinin nasıl bir araya geldiğini inceler. Bileşenlerin amacı, kullanımı ve aralarındaki ilişki hakkında kapsamlı bir açıklama yaparak sistem bileşenlerini ve alt sistemleri ayrıntılı olarak açıklamaktadır.

5.2.1. Tasarım Öğeleri

Uygulama sistemin tek ana öğesidir. Kullanıcı arayüzü, sohbet penceresi, mikrofon(donanım), ChatGPT API ve Google text-to-speech API modüllerini kapsamaktadır.

5.2.2. Örnek Diller

Kullanıcı Arayüzü: Oyun içinde karakter seçimini yönetir.

Sohbet Penceresi: Kullanıcının karakteri görmesini ve iletişim kurmasını sağlar.

Mikrofon (Donanım): Sesli komutları alır ve karakterler arasında sesli etkileşimi mümkün kılar.

ChatGPT API: Karakterlerin doğal dil işleme yetenekleriyle oyun senaryosunu zenginleştirir.

Google Text-to-Speech API: Yazılı metinleri sesli hale getirerek daha özgün bir deneyim sunar. Google text-to-speech API, yazılı metinleri sesli hale getirerek daha benzersiz bir deneyim sunar.

5.3. Etkileşim Bakış Açısı

Sistemin etkileşim bakış açısı, her bir nesnenin birbirini nasıl çalıştırdığını temsil eder.

5.3.1. Tasarım Öğeleri

Etkileşim bakış açısı altında, uygulamanın farklı bileşenleri arasındaki etkileşimleri anlamak için önemli tasarım öğeleri bulunmaktadır. Bu öğeler arasında kullanıcı arayüzü, karakter seçimi, mikrofon erişimi ve ChatGPT API ile iletişim ve metni sesli bir şekilde kullanıcıya vermek gibi aşamalardan oluşur.

5.3.2. Örnek Diller

Kullanıcı, uygulamaya giriş yaptıktan sonra ana menüden karakter seçimine yönlendirilir. Bu noktada, kullanıcı arayüzü, kullanıcıya farklı tarihi karakterleri gösterir ve bir seçim yapması için etkileşimli bir deneyim sunar.

Kullanıcı, seçtiği karakterle etkileşimde bulunmak için mikrofon erişimine ihtiyaç duyar. Mikrofon kullanılarak gerçekleştirilen sesli etkileşim, kullanıcının karaktere sorular sorması ve karakterin sesli yanıtlarını algılaması şeklindedir. Kullanıcının tarihi bir karaktere sorular sorması ve karakterin canlı gibi sesli yanıtlar vermesi etkileşimi güçlendirebilir.

ChatGPT API, kullanıcı ve karakter arasında metin tabanlı etkileşimi destekler. Kullanıcının sorduğu sorular yazı olarak ChatGPT tarafına iletilir ve yapay zekâ tarafından üretilen cevap alınır. Bu etkileşim, karakterle derinlemesine ve gerçekçi diyaloglar kurma imkanı sağlar.

5.4. Mantıksal Bakış Açısı

5.4.1. Tasarım Öğeleri

Mantıksal bakış açısı altında, uygulamanın içsel yapısını anlamak için temel tasarım öğeleri önemlidir. Bu öğeler arasında istek ve cevapların mantığı, işlem mantığı, ve animasyon mekanizmaları gibi unsurlar yer alır. Her bir birleşen kendi içinde çalışmaktadır ancak projeyi bir bütünü olarak birbirlerine parametre değerlerini sağlamaktadırlar.

5.4.2. Örnek Diller

Projede kullanılan programlama dilleri ve teknolojiler, sistemdeki işlevselliği destekler. Oyun kısmı için Unreal Engine ve C++ kullanılması planlanmıştır. Ayrıca, sunucu tarafı işlemler için Python tercih edilecektir. Kullanıcıya verilecek cevapların oluşturulması için de ChatGPT API'ı kullanılacaktır.

Bu diller, sistemin temel yapı taşlarını oluşturan algoritmaları ve işlevleri desteklemek için seçilmiştir. Unreal Engine'in kullanımı, oyun mekaniğini ve görselliği optimize etmek için tasarlanmıştır. Python, sunucu tarafında veri işleme ve kontrol işlemleri için uygun bir dil olarak tercih edilmiştir. ChatGPT API ise kullanıcı ile etkileşimi daha zengin ve gerçekçi hale getirmek amacıyla entegre edilmiştir.

Bu diller, projenin genel tasarım hedeflerine uygun olarak seçilmiş ve bir araya getirilmiştir. Sistem, bu dillerin sağladığı güçlü özellikleri kullanarak etkileşimli ve dinamik bir deneyim sunmayı amaçlamaktadır.

6. SDD DETAYLI PLANI

6.1. Giriş ve Hedef Belirleme

Proje ekibi, proje hedeflerini ve kapsamını gözden geçirir.

Geliştirilecek yazılımın temel özellikleri ve beklenen çıktılar belirlenir.

6.2. Kavramsal Model Oluşturma

Kavramsal model, yazılımın temel bileşenlerini ve bunlar arasındaki ilişkileri içerir.

Bağlam bakış açısı altında yazılım tasarımının ana hatları belirlenir.

6.3. Yapısal Tasarım Belirlenmesi

Tasarımın paydaşları göz önüne alınarak, tasarımın genel yapısı belirlenir.

ChatGPT API entegrasyonu, ses giriş ve çıkış işlemleri ve karakter arayüzü, kullanıcı etkileşimi tasarımları detaylı olarak planlanır.

6.4. Tasarım Bakış Açıları Oluşturma

Bağlam, kompozisyon, etkileşim ve mantıksal bakış açıları altında detaylı tasarım belgeleri hazırlanır.

Örnek dillerle desteklenen bu bakış açıları, yazılımın farklı yönlerini kapsar.

6.5. Gereksinim Analizi ve Önceliklendirme

Fonksiyonel ve performans gereksinimleri detaylı bir şekilde analiz edilir.

Gereksinimler önceliklendirilir; kritik ve temel özellikler belirlenir.

6.6. Dahili ve Dış Arayüzlerin Tanımlanması

ChatGPT API, Google text-to-speech API gibi dış arayüzlerin entegrasyonu planlanır.

Dahili arayüzler (kullanıcı arayüzü, mikrofon) detaylı olarak incelenir.

6.7. Dahili Veri Yapılarının ve Performans Gereksinimlerinin Belirlenmesi

Kullanıcı bilgileri için veri tabanı yapısı belirlenir.

Performans gereksinimleri, yanıt süreleri ve sistem optimizasyonları planlanır.

6.8. Güvenlik ve Gizlilik Politikalarının Oluşturulması

Kullanıcı verilerinin güvenliği için alınacak önlemler belirlenir.

Gizlilik politikaları oluşturularak uygulanacak kurallar planlanır.

6.9. Detaylı Planın Dokümantasyonu

Her adım detaylı bir şekilde belgelenir.

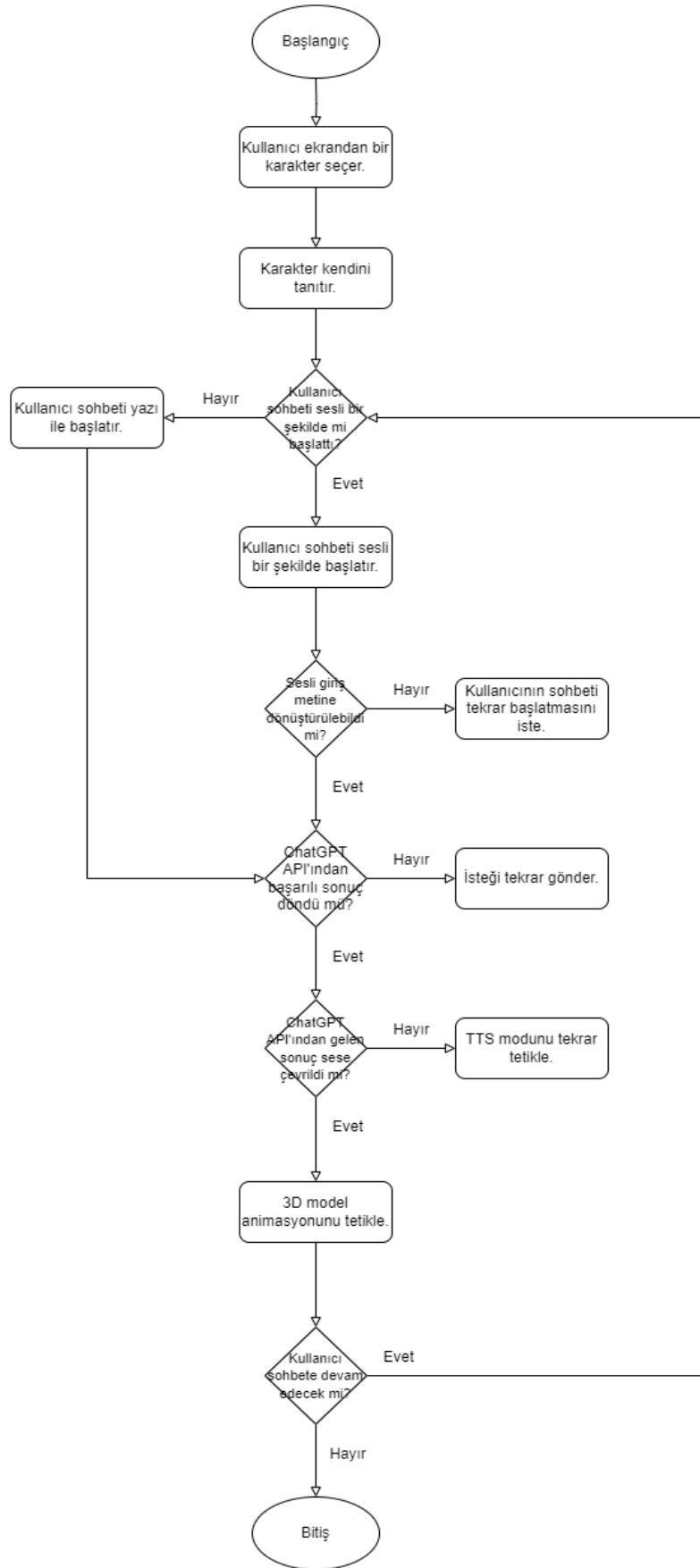
Ek açıklamalar, şematik görseller ve gerektiğinde örnek kodlar içeren bir dokümantasyon hazırlanır.

6.10. Ekler ve İnceleme

Ek-1'de yer alan Sihirli Ayna: ChatGPT Entegreli Sohbet Oyunu Akış Diyagramı, detaylı planın bir parçası olarak gözden geçirilir ve gerektiğinde güncellenir.

Bu detaylı plan, proje ekibinin süreci adım adım takip etmesini ve yazılım tasarımının her yönünü kapsamasını sağlamak amacıyla oluşturulmuştur. Her adım, proje hedeflerine ve kullanıcı beklentilerine uygunluğu göz önünde bulundurarak titizlikle yürütülmelidir.

7. EKLER



Ek-1: Sihirli Ayna: ChatGPT Entegreli Sohbet Oyunu Akış Diyagramı