

**Bölüm:**Yazılım Mühendisliği

**Ders Adı:** Yazılım İnşası

**Asst.Prof.Dr.** AYŞE NUR ALTINTAŞ TANKÜL

**teslim tarihi :** 12/24

## **YAPAY ZEKA DESTEKLİ SOHBET SİSTEMİ (AI CHAT BOT)**

### **PROJE ANALİZ RAPORU**

|   |                    |            |
|---|--------------------|------------|
| 1 | Abdulmouen Kanatri | 2310238519 |
| 2 | MOHAMED ABIYA      | 2410238560 |

|  |   |
|--|---|
| PROJE ANALİZ RAPORU .....  | 1 |
| 1. İçindekiler .....   | 1 |
| Yapay Zeka Destekli Sohbet Sistemi (AI Chat Bot).....                | 1 |
| 1. Gereksinimler .....   | 1 |
| 1.1. İşlevsel Gereksinimler .....                                    | 2 |
| 1.2. İşlevsel Olmayan Gereksinimler .....                            | 2 |
| 2. Kullanım Durumu Diyagramı (Use Case Diagram) .....                | 3 |
| 2.1. Aktörler .....  | 3 |
| 2.2. Kullanım Durumları .....  | 3 |
| 3. Dinamik Model (Sequence Diagram) .....                            | 3 |
| 3.1. Mesaj Gönderme Senaryosu .....                                  | 4 |
| 4. Soru-Cevap Sistemi İçin Nesne Tabanlı Model (Class Diagram) ..... | 4 |
| 4.1. Sınıflar ve Açıklamalar .....                                   | 5 |
| 4.2. Sınıf Diyagramı Özeti (İlişkiler) .....                         | 5 |
| 5. Kullanıcı Arayüzü (User Interface) .....                          | 6 |
| 5.1. Tasarım ve Yapı .....   | 6 |
| 5.2. Etkileşimler .....  | 7 |
| 5.3. Kod Blokları .....  | 7 |

# Yapay Zeka Destekli Sohbet Sistemi (AI Chat Bot)

**Proje Özeti:** Bu projede, modern web teknolojileri (Next.js, TypeScript) ve Büyük Dil Modelleri (LLM) kullanılarak geliştirilmiş, insan benzeri etkileşim sunan bir sohbet botu sistemi tasarlanmıştır. Sistem, kullanıcıların yapay zeka (Google Gemini) ile gerçek zamanlı (streaming) sohbet etmelerini, kod yazmalarını ve karmaşık sorulara yanıt almalarını sağlar. Ayrıca, kullanıcıların geçmiş sohbetleri veritabanında (PostgreSQL) saklanır ve yönetilebilir bir arayüz ile sunulur.

## 1. Gereksinimler

### 1.1. İşlevsel Gereksinimler

İşlevsel gereksinimler, sohbet sisteminin kullanıcıya sunduğu temel özellikleri kapsar.

#### 1.1.2. Gerçek Zamanlı Sohbet (Streaming)

- Kullanıcı bir mesaj gönderdiğinde, yapay zekanın cevabı tamamlanmasını beklemeden kelime kelime ekranda belirtmelidir (Vercel AI SDK Streaming).
- Sistem, konuşma bağlamını (context) koruyarak önceki mesajları hatırlamalıdır.

#### 1.1.3. Sohbet Geçmişi Yönetimi

- Her yeni sohbet oturumu otomatik olarak veritabanına kaydedilmelidir.
- Kullanıcı, sol menüden (Sidebar) eski sohbetlerine tıklayarak geçmiş konuşmalarını görüntüleyebilmelidir.

#### 1.1.4. Kod ve Format Desteği

- Sistem, kullanıcının veya yapay zekanın yazdığı kod bloklarını (Python, JS vb.) renklendirilmiş formatta (Syntax Highlighting) göstermelidir.
- Matematiksel ifadeler ve tablolar düzgün bir şekilde render edilmelidir (Markdown desteği).

#### 1.1.5. Sohbet Yönetimi

- Kullanıcı, istediği zaman bir sohbeti silebilir veya başlığını değiştirebilir.

## 1.2. İşlevsel Olmayan Gereksinimler

### 1.2.1. Performans

- Yapay zeka yanıtları, kullanıcı mesajı gönderdikten sonra 1 saniye içinde akmaya başlamalıdır (Düşük Gecikme / Low Latency).

### 1.2.2. Güvenlik

- Kullanıcı verileri ve sohbet geçmişi şifrelenmiş bağlantılar üzerinden iletilmelidir. API anahtarları (API Keys) sunucu tarafında saklanmalıdır.

### 1.2.3. Kullanılabilirlik

- Arayüz, mobil cihazlar ve masaüstü tarayıcılarla tam uyumlu (Responsive) olmalıdır.

### 1.2.4. Ölçeklenebilirlik

- Sistem, PostgreSQL veritabanı sayesinde artan kullanıcı ve mesaj sayısını performans kaybı olmadan yönetebilmelidir.

## 2. Kullanım Durumu Diyagramı (Use Case Diagram)

### 2.1. Aktörler

- **Kullanıcı (User):** Sisteme giriş yapan, soru soran ve geçmişini yöneten kişi.
- **Yapay Zeka Servisi (AI Provider):** Mesajları işleyen ve cevap üreten dış servis (Google Gemini API).

### 2.2. Kullanım Durumları

#### 2.2.1. Sohbet Başlatma ve Mesajlaşma

- **Açıklama:** Kullanıcı yeni bir sohbet oluşturur ve yapay zekaya bir soru sorar.
- **Adımlar:**
  - Kullanıcı arayüzdeki mesaj kutusuna metni girer ve gönderir.
  - Sistem mesajı API'ye iletir.
  - Cevap anlık olarak (stream) kullanıcı ekranına yansıtılır.
  - Mesaj ve cevap veritabanına kaydedilir.

#### 2.2.2. Geçmiş Sohbetleri Görüntüleme

- **Açıklama:** Kullanıcı eski bir konuşmaya geri döner.

- **Adımlar:**
  - Kullanıcı yan menüden bir başlığa tıklar.
  - Sistem ilgili sohbetin tüm mesajlarını veritabanından çeker.
  - Sohbet ekranı eski mesajlarla güncellenir.

### 2.2.3. Oturum Yönetimi (Login/Logout)

- **Açıklama:** Kullanıcının sisteme güvenli girişi.
- **Adımlar:**
  - Kullanıcı "Giriş Yap" butonuna tıklar.
  - Sağlayıcı (Google/Email) üzerinden kimlik doğrulanır.
  - Başarılı girişte kullanıcı ana ekrana yönlendirilir.

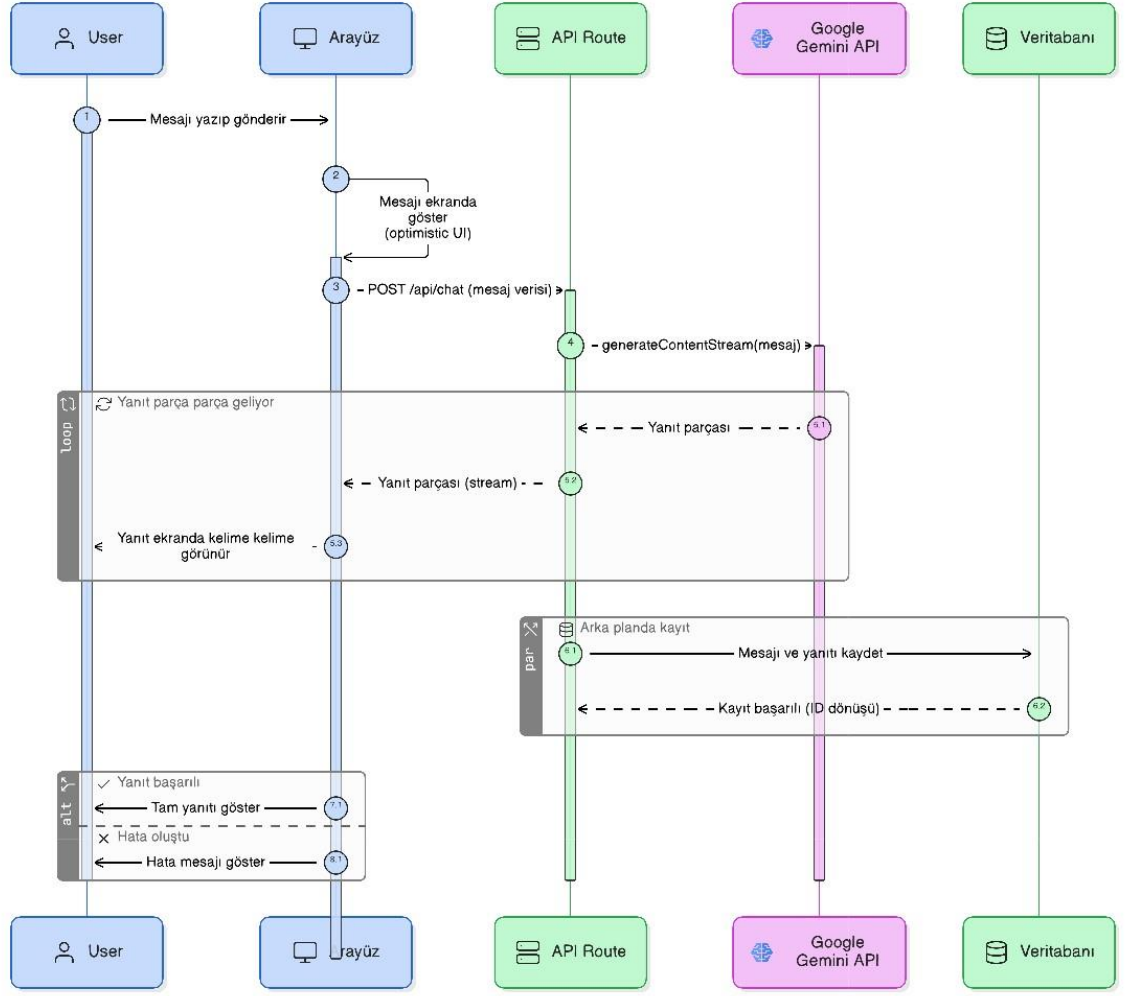
## 3. Dinamik Model (Sequence Diagram)

### 3.1. Mesaj Gönderme Senaryosu

**Senaryo:** Kullanıcı "Merhaba" yazar ve yapay zeka cevap verir.

**Akış:**

1. **Kullanıcı (User):** Arayüzde mesajı yazar ve "Gönder"e basar.
2. **Arayüz (UI):** Mesajı hemen sohbet penceresine ekler ve /api/chat uç noktasına (Endpoint) istek atar.
3. **Sunucu (Server API):** İsteği doğrular ve Google Gemini API'sine iletir.
4. **Google Gemini:** Metni işler ve cevabı parça parça (chunk) üretir.
5. **Sunucu (Server API):** Gelen parçaları anında Arayüz'e (UI) iletir (Streaming).
6. **Veritabanı (Prisma/DB):** Sohbet tamamlandığında mesajı ve cevabı Message tablosuna kaydeder.
7. **Arayüz (UI):** Cevap tamamlandığında kullanıcıya hazır olduğunu gösterir.
8. **Dinamik Model (Sequence Diagram)**



## 4. Soru-Cevap Sistemi İçin Nesne Tabanlı Model (Class Diagram)

Projenin Prisma şemasına dayalı sınıf yapısı aşağıdadır.

### 4.1. Sınıflar ve Açıklamalar

#### 4.1.1. Kullanıcı (User)

- **Açıklama:** Sistemi kullanan kayıtlı kişidir.
- **Özellikler:**
  - id: Benzersiz kullanıcı kimliği (String).
  - email: Kullanıcı e-posta adresi.
  - image: Profil fotoğrafı URL'si.
- **İlişkiler:** Bir Kullanıcı, birden fazla Sohbet'e (Conversation) sahiptir.

#### 4.1.2. Sohbet (Conversation)

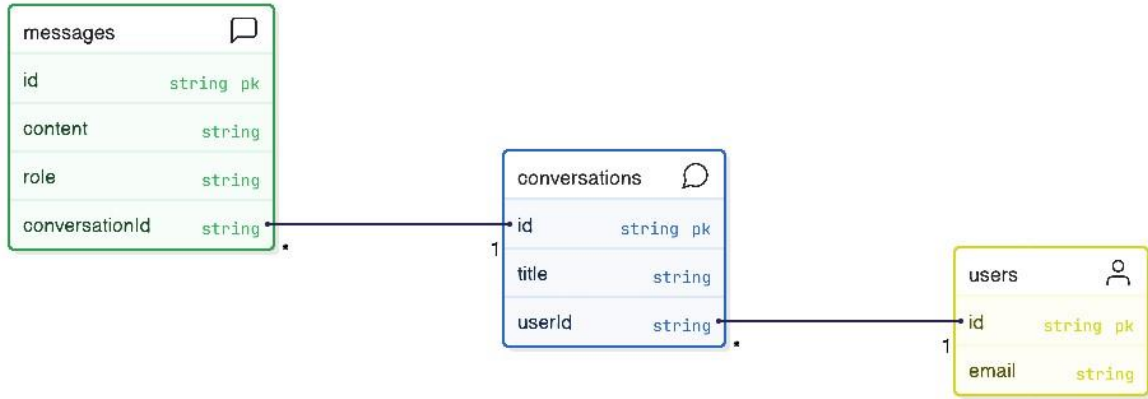
- **Açıklama:** Bir oturumdaki karşılıklı konuşmaların bütünüdür.
- **Özellikler:**
  - **id:** Sohbet kimliği.
  - **title:** Sohbetin başlığı (örn: "React Hakkında Sorular").
  - **createdAt:** Oluşturulma tarihi.
  - **userId:** Sohbeti başlatan kullanıcının ID'si.
- **İlişkiler:** Bir Sohbet, birden fazla Mesaj'a (Message) sahiptir.

#### 4.1.3. Mesaj (Message)

- **Açıklama:** Sohbet içindeki tekil bir soru veya cevaptır.
- **Özellikler:**
  - **id:** Mesaj kimliği.
  - **content:** Mesajın metin içeriği (Text).
  - **role:** Gönderen rolü (user veya assistant).
  - **conversationId:** Bağlı olduğu sohbetin ID'si.

#### 4.2. Sınıf Diyagramı Özeti (İlişkiler)

- **User (1) ----> (N) Conversation:** Bir kullanıcı birçok sohbet oluşturabilir.
- **Conversation (1) ----> (N) Message:** Bir sohbet birçok mesaj içerir.



|               |           |
|---------------|-----------|
| class Message |           |
| +String       | id        |
| +String       | content   |
| +String       | role      |
| +DateTime     | createdAt |

|                     |               |
|---------------------|---------------|
| class User          |               |
| +String             | id            |
| +String             | email         |
| +List~Conversation~ | conversations |

|                    |           |
|--------------------|-----------|
| class Conversation |           |
| +String            | id        |
| +String            | title     |
| +DateTime          | createdAt |
| +addMessage()      |           |
| +remove()          |           |

## 5. Kullanıcı Arayüzü (User Interface)

Sistemin arayüzü, odaklanmayı kolaylaştıran "Clean UI" prensiplerine göre tasarlanmıştır.

### 5.1. Tasarım ve Yapı

- **Ana Ekran:** Ekranın ortasında geniş bir sohbet alanı ve en altta mesaj giriş kutusu bulunur. Bu, kullanıcının içeriğe odaklanmasını sağlar.

- **Sidebar (Yan Menü):** Sol tarafta, geçmiş sohbetlerin listelendiği bir menü bulunur. Kullanıcı buradan eski sohbetlere geçiş yapabilir veya yeni sohbet başlatabilir.

## 5.2. Etkileşimler

- **Yükleniyor Durumu (Loading State):** Yapay zeka düşünürken kullanıcıya görsel bir indikatör (örneğin 3 nokta animasyonu) gösterilir.
- **Karanlık Mod (Dark Mode):** Sistem, kullanıcının cihaz tercihine göre otomatik olarak koyu veya açık temaya geçer.

## 5.3. Kod Blokları

- Yazılım geliştiriciler için özel olarak, kod blokları gri bir arka plan üzerinde, sözdizimi renklendirmesi (Syntax Highlighting) ve "Kodu Kopyala" butonu ile birlikte sunulur.