# **Smart Doc Insight - Yazılım Gereksinim Analizi Raporu**

Bu doküman, "Smart Doc Insight" projesinin yazılım gereksinim analizi sürecinde oluşturulan tanımlamaları, modelleri ve diyagramları içermektedir.

### **Adım 1: Proje Amacı, Sorun ve Hedef Analizi**

Proje Amacı

Bu projenin temel amacı, eğitmenlerin proje yönetimi ve değerlendirme süreçlerini otomatize ederek daha verimli hale getirmektir. Geliştirilecek olan "Smart Doc Insight" web uygulaması, eğitmenlerin sunduğu PDF formatındaki proje belgelerini yapay zeka destekli dil modelleri kullanarak analiz etmelerini sağlayacak, böylece manuel belge inceleme yükünü ortadan kaldıracaktır.

Sorun Analizi

Mevcut durumda eğitmenler, özellikle çok sayıda öğrenci projesini değerlendirirken, her bir belgeyi manuel olarak açıp belirli kriterlerin varlığını tek tek kontrol etmek zorundadır. Bu süreç, ciddi anlamda zaman alıcı, verimsiz ve hata yapma olasılığı yüksek bir süreçtir. Bu durum, eğitmenlerin projelere daha hızlı ve etkili bir şekilde geri bildirim vermesini engellemektedir.

Hedef Analizi

Bu sorundan yola çıkarak projenin temel hedefleri şunlardır:

* **Verimliliği Artırmak:** Eğitmenlerin PDF dosyalarını otomatik olarak analiz ederek değerlendirme süreçlerini hızlandırmak.
* **Zaman Tasarrufu Sağlamak:** Geniş kapsamlı doküman incelemelerinde harcanan zamanı önemli ölçüde azaltmak.
* **Anlamlı Bilgi Çıkarmak:** Yapay zeka modelleri kullanarak belgelerden sadece anahtar kelimeleri değil, aynı zamanda özetler ve anlamlı raporlar oluşturmak.
* **Kolay Erişim ve Yönetim Sağlamak:** Kullanıcıların analiz ettikleri belgelere ve sonuçlarına daha sonra kolayca erişebilmelerini ve yönetebilmelerini sağlamak.
* **Akıllı Analiz Sunmak:** Yerel veya API tabanlı dil modelleri aracılığıyla metinlerdeki belirli bilgileri tespit etmek ve sorulara yanıt bulmak.

### **Adım 2: Detaylı İhtiyaç Belirtimi**

#### **Fonksiyonel Gereksinimler**

**Genel Kullanıcı Gereksinimleri (Öğrenci ve Öğretmen için Ortak)**

* **FR-G-01:** Sistem, kullanıcıların e-posta ve şifre ile kayıt olabilmesini sağlamalıdır.
* **FR-G-02:** Sistem, kayıtlı kullanıcıların e-posta ve şifre ile giriş yapabilmesini sağlamalıdır.
* **FR-G-03:** Sistem, kullanıcıların kendi yükledikleri dosyaları silebilmesini sağlamalıdır.

**Öğrenci Rolüne Özgü Gereksinimler**

* **FR-Ogrenci-01:** Öğrenci, yeni bir proje oluşturabilmelidir.
* **FR-Ogrenci-02:** Öğrenci, oluşturduğu projeye diğer kayıtlı öğrencileri üye olarak ekleyebilmelidir.
* **FR-Ogrenci-03:** Öğrenci, projesine PDF formatında dokümanlar yükleyebilmelidir.

**Öğretmen Rolüne Özgü Gereksinimler**

* **FR-Ogretmen-01:** Öğretmen, sistemdeki tüm projeleri ve detaylarını görüntüleyebilmelidir.
* **FR-Ogretmen-02:** Öğretmen, doküman analizi için yerel veya harici bir dil modeli (LLM) seçebilmelidir.
* **FR-Ogretmen-03:** Öğretmen, harici bir API sağlayıcısı seçtiyse, kendi API anahtarını sisteme girebilmeli ve kaydedebilmelidir.
* **FR-Ogretmen-04:** Öğretmen, bir dokümanı varsayılan veya kendi belirlediği özel başlıklarla analiz ettirebilmelidir.
* **FR-Ogretmen-05:** Sistem, analiz sonuçlarını öğretmene sunmalı ve veritabanına kaydetmelidir.
* **FR-Ogretmen-06:** Öğretmen, hatalı analiz sonuçlarını sistemden temizleyebilmelidir.

#### **Fonksiyonel Olmayan Gereksinimler**

* **Performans:** Sistem, metin analizi işlemlerini makul bir sürede tamamlamalıdır.
* **Güvenlik:** Kullanıcı şifreleri ve API anahtarları gibi hassas veriler güvenli bir şekilde (şifrelenerek) saklanmalıdır.
* **Kullanılabilirlik:** Kullanıcı arayüzü basit, anlaşılır ve sezgisel olmalıdır.
* **Güvenilirlik:** Sistem, analiz sürecinde oluşabilecek hataları yöneterek çökmek yerine kullanıcıya bilgilendirme mesajı göstermelidir.
* **Uyumluluk:** Web uygulaması, modern web tarayıcılarında (Chrome, Firefox vb.) sorunsuz çalışmalıdır.

### **Adım 3: User Story'ler (Kullanıcı Hikayeleri)**

**Öğrenci Rolü İçin:**

* **Bir öğrenci olarak,** yeni bir proje oluşturabilmek istiyorum, böylece ödevimi ve ilgili tüm dokümanları tek bir yerde toplayabilirim.
* **Bir öğrenci olarak,** projeme takım arkadaşlarımı ekleyebilmek istiyorum, böylece proje üzerinde birlikte çalışabiliriz.
* **Bir öğrenci olarak,** proje dokümanlarımı PDF formatında sisteme yükleyebilmek istiyorum, böylece öğretmenim tarafından incelenebilirler.

**Öğretmen Rolü İçin:**

* **Bir öğretmen olarak,** sistemdeki tüm projeleri tek bir ekranda görebilmek istiyorum, böylece değerlendireceğim projeyi kolayca seçebilirim.
* **Bir öğretmen olarak,** bir dokümanı analiz etmek için yerel veya harici bir yapay zeka modeli seçebilmek istiyorum, böylece değerlendirme sürecinde istediğim teknolojiyi kullanabilirim.
* **Bir öğretmen olarak,** bir dokümandaki belirli bilgilerin varlığını otomatik olarak kontrol ettirebilmek istiyorum, böylece değerlendirme sürecini önemli ölçüde hızlandırabilirim.
* **Bir öğretmen olarak,** analiz sonuçlarını özet halinde ve anlaşılır bir şekilde görebilmek istiyorum, böylece projenin istenen kriterleri karşılayıp karşılamadığını hızla anlayabilirim.

### **Adım 4: Use Case Tanımları ve Diyagramı**

#### **Use Case Diyagramı Açıklaması**

Bu diyagram, bir çizim aracı kullanılarak aşağıdaki bileşenlerle oluşturulmalıdır:

* **Aktörler:** Öğrenci, Öğretmen
* **Use Case'ler:** Kayıt Ol, Giriş Yap, Proje Yönet, Doküman Yükle, Projeleri Görüntüle, Dokümanı Analiz Et, Doküman Sil.
* **İlişkiler:** Aktörler ile ilgili Use Case'ler arasında düz çizgiler. Fonksiyonların çalışması için "Giriş Yap" gerekliliği olan diğer tüm Use Case'lerden "Giriş Yap"a doğru <<include>> ilişkisi kurulmalıdır.

#### **Use Case Tanım Tablosu Örneği: Dokümanı Analiz Et**

| **Use Case Adı:** | **Dokümanı Analiz Et** |
| --- | --- |
| **ID:** | UC-01 |
| **Aktör(ler):** | Öğretmen |
| **Amaç:** | Bir öğrenci projesine ait PDF dokümanını, seçilen LLM sağlayıcısı ile analiz ederek sonuçları elde etmek. |
| **Ön Koşullar:** | Öğretmen sisteme giriş yapmıştır ve sistemde analiz edilecek bir doküman mevcuttur. |
| **Ana Akış:** | 1. Öğretmen analiz edilecek dokümanı seçer.  2. Analiz ayarlarını (LLM Sağlayıcı, başlıklar) yapar.  3. "Analizi Başlat" butonuna tıklar.  4. Sistem, PDF'i metne dönüştürür ve LLM sağlayıcıya istek gönderir.  5. LLM'den gelen sonuç alınır.  6. Sonuçlar ekranda gösterilir ve veritabanına kaydedilir.  7. Dokümanın durumu "Analiz Edildi" olarak güncellenir. |
| **Alternatif Akış:** | **4a.** LLM sağlayıcısı ile iletişimde hata oluşursa, sistem kullanıcıya hata mesajı gösterir ve dokümanın durumunu "Hata" olarak günceller. |

### **Adım 5: Zengin Görsel Gösterimler**

#### **Rich Picture (Zengin Resim) Açıklaması**

Serbest bir çizimle aşağıdaki öğeler bir araya getirilmelidir:

* **Merkezde:** "Smart Doc Insight Sistemi"
* **Paydaşlar:** "Öğrenci" ve "Öğretmen" figürleri ve onların endişelerini yansıtan düşünce balonları ("Değerlendirmeler çok zaman alıyor!").
* **Sorunlar:** "Manuel incelemenin verimsizliği", "API maliyetleri" gibi ifadeler.
* **Süreçler:** Aktörler, sistem ve dış servisler (LLM API'leri) arasındaki veri akışını gösteren oklar.

#### **Context Diagram (Bağlam Diyagramı) Açıklaması**

Sistemin sınırlarını ve dış varlıklarla veri alışverişini gösteren yapısal bir diyagramdır.

* **Merkezde Süreç:** "Smart Doc Insight Sistemi"
* **Dış Varlıklar:** Öğrenci, Öğretmen, LLM API Sağlayıcısı.
* **Veri Akışları:** Bu varlıklar ile sistem arasındaki veri alışverişini gösteren etiketli oklar (Örn: Öğrenci'den sisteme PDF Dokümanı, Sistem'den Öğretmen'e Analiz Sonuçları).

### **Adım 6: Sistem Mimarisi Tanımı**

* **Genel Mimari:** İstemci-Sunucu (Client-Server) modeline dayalı Katmanlı Mimari (Layered Architecture).
* **İstemci (Client):** Kullanıcıların etkileşimde bulunduğu web arayüzü (HTML/CSS/JS).
* **Sunucu (Server):** İş mantığını yürüten, veritabanı ve LLM servisleri ile iletişim kuran Node.js (Express) uygulaması.
* **Veri Depolama (Data Storage):** MongoDB veritabanı.
* **Teknoloji Yığını:**
  + **Backend:** Node.js, Express
  + **Veritabanı:** MongoDB, Mongoose
  + **PDF İşleme:** pdf-parse
  + **LLM Entegrasyonu:** Ollama, LM Studio, Gemini API, OpenAI API

### **Adım 7: ERD ve Veri Sözlüğü**

#### **ERD (Varlık-İlişki Diyagramı) Açıklaması**

NoSQL veritabanı için mantıksal varlıklar ve ilişkileri aşağıdaki gibidir:

* **Varlıklar:** Users, Projects, Documents, AnalysisResults, InstructorSettings.
* **İlişkiler:** Bir User birden çok Project oluşturabilir. Bir Project birden çok Document içerebilir. Bir Document birden çok AnalysisResult'a sahip olabilir.

#### **Veri Sözlüğü (Örnek: Documents Koleksiyonu)**

| **Alan Adı (Field Name)** | **Veri Tipi (Data Type)** | **Açıklama** |
| --- | --- | --- |
| \_id | ObjectId | Otomatik oluşan eşsiz kimlik numarası. |
| filename | String | Yüklenen dosyanın orijinal adı. |
| filepath | String | Dosyanın sunucudaki saklama yolu. |
| project | ObjectId | Dokümanın ait olduğu projenin kimliği. |
| status | String | Dokümanın analiz durumu ('awaiting\_analysis', 'analyzed', 'error'). |
| uploadedAt | Date | Dokümanın yüklenme tarihi. |

### **Adım 8: UML Diyagramları**

Aşağıda, çizilmesi gereken temel UML diyagramlarının açıklamaları bulunmaktadır.

* **Sınıf Diyagramı:** Sistemdeki User, Project, Document gibi ana sınıfları, niteliklerini, metotlarını ve aralarındaki ilişkileri (1'e N, Composition vb.) gösterir.
* **Aktivite Diyagramı:** "Doküman Analizi" gibi bir iş akışının adımlarını, karar noktalarını ([Evet]/[Hayır] dallanmaları) ve sonuçlarını başından sonuna kadar gösterir.
* **Sıralama (Sequence) Diyagramı:** "Doküman Analizi" sürecinde Öğretmen, Arayüz, Sistem ve LLMSağlayıcı nesneleri arasında gönderilen mesajları ve bunların zaman içindeki sırasını gösterir.
* **Durum (State) Diyagramı:** Document nesnesinin yaşam döngüsünü, yani Yüklendi, Analiz Bekliyor, Analiz Edildi ve Hata Oluştu gibi durumlar arasındaki geçişleri gösterir.
* **Bileşen (Component) Diyagramı:** Sistemin ana modüllerini (Kullanıcı Arayüzü, SmartDoc API, Veritabanı) ve aralarındaki arayüzleri gösterir.
* **Deployment Diyagramı:** Sistemin fiziksel olarak nasıl dağıtıldığını (Kullanıcı Bilgisayarı, Uygulama Sunucusu, Veritabanı Sunucusu) ve donanım bileşenlerinin nasıl bağlandığını gösterir.