



**ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ**  
**MÜHENDİSLİK MİMARLIK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
**VERİ TABANI YÖNETİM SİSTEMLERİ (B)**  
**2024-2025 EĞİTİM-ÖĞRETİM YILI**  
**GÜZ DÖNEMİ**  
**PROJE GEREKSİNİM ANALİZİ RAPORU**

**MR GÖRÜNTÜLERİ İÇİN VEKTÖR VERİ TABANI ÇÖZÜMÜ**

**Projeyi Hazırlayanlar – Grup No: 13**

152120211057 Atakan Berber (Grup Koordinatörü)

152120211052 Umut Öztürk

152120211105 Eren Eroğlu

**Proje Yetkilisi:** Sergen Aşık    [sergen.asik@ogu.edu.tr](mailto:sergen.asik@ogu.edu.tr)

**Versiyon 2.0 / Kasım 2024**

# İçindekiler

1. Giriş .....	3
1.1 Amaç.....	3
1.2 Kapsam .....	3
1.3 Projenin Hedefi ve Başarı Kriterleri .....	3
1.4 Tanımlar ve Kısaltmalar.....	3
1.5 Dokümana Genel Bakış .....	3
2. Genel Tanım.....	4
2.1 Ürüne Bakış .....	4
2.1.1 Sistem Arayüzleri .....	4
2.1.2 Kullanıcı Arayüzleri.....	4
2.1.3 Donanım Arayüzleri.....	4
2.1.4 Yazılım Arayüzleri.....	4
2.2 Ürün İşlevleri .....	4
2.3 Kullanıcı Özellikleri .....	5
2.4 Kısıtlar .....	5
2.4.1 Uygulanan Kısıtlar .....	5
2.4.2 Uygulanması Düşünülmeyen Kısıtlar .....	5
3. Özel Gereksinimler.....	5
3.1 Fonksiyonel Gereksinimler.....	5
3.1.1 İşlevsel Gereksinimler .....	5
3.1.2 Veri Yapısı ve Vektör Veri Tabanı Gereksinimleri.....	6
3.2 Fonksiyonel Olmayan Gereksinimler.....	10
3.2.1 Kalite Özellikleri .....	10
3.3 Arayüz Gereksinimleri.....	12
3.3.1 Kullanıcı Arayüz Gereksinimleri .....	12
4. Gereksinimlerin Önceliği ve Kritikliği .....	13
5. Veri Tabanı ve Veri Yapısı İle İlgili Tüm Gereksinimler .....	14
6. Proje Görev Dağılımı ve Adam-Gün Hesabı .....	16

# 1.Giriş

## 1.1 Amaç

Bu belgenin amacı, Vector Database teknolojisi kullanılarak geliştirilecek olan **MR Yönetim Sistemi** için yazılım gereksinimlerini detaylandırmaktır. Bu sistem, DICOM dosya formatında yer alan metadataları ve MR(MRI), X-Ray, CT gibi taramaları yönetmek, depolamak, analiz etmek ve ilgili sağlık personeline kolayca sunmak üzere tasarlanmıştır. Hedef kitlesi, yazılım geliştiriciler, sistem yöneticileri, sağlık kuruluşlarının IT birimi ve ilgili sağlık profesyonelleridir.

## 1.2 Kapsam

Bu belge, MR Yönetim Sistemi'nin tüm yazılım gereksinimlerini içermektedir. Sistem, MR taramalarını kaydeder, görüntüler, düzenler ve raporlar üretir. Ayrıca, veri güvenliği gibi kritik konuları da kapsamaktadır.

## 1.3 Projenin Hedefi ve Başarı Kriterleri

Programın temel hedefi ve başarı kriteri, uzman doktorun program içerisinde dosya seç bölümünden seçtiği görüntüleri inceleyip görüntünün işlevselliğine göre karar verebilmesidir. Program, doktorun seçtiği görüntüyü ve ilgili verileri ekranda görmesini sağlayacaktır. Program, doktorun veritabanından hasta arayarak o hastaya ait tüm verileri görüntüleyebilmesini de sağlamayı hedeflemektedir. Bu süreç, doktorun tanı sürecinde kullanabileceği işlevsel görüntü ve hasta bilgilerini veritabanında toplayarak gereksiz olanları hariç tutmayı amaçlamaktadır.

## 1.4 Tanımlar ve Kısaltmalar

**MR (Maynetik Rezonans):** Tıbbi görüntüleme yöntemi.

**DBMS (Database Management System):** Veritabanı Yönetim Sistemi.

**CT (Computed Tomography):** Bilgisayarlı Tomografi.

## 1.5 Dokümana Genel Bakış

Bu belgede, sistemin genel tanımı, işlevsel ve performans gereksinimleri ile yazılım ve donanım arayüz gereksinimleri anlatılmaktadır. Kullanıcı arayüzleri ve veri yönetimi gibi temel konular detaylandırılacak ve son bölümde gereksinimlerin izlenebilirliği sağlanacaktır.

## 2. Genel Tanım

### 2.1 Ürüne Bakış

Bu proje kapsamında geliştirilecek olan **MR Yönetim Sistemi**, ilgili doktora ait bilgisayar üzerinde lokal veritabanı ile entegre çalışacak yazılım ürünüdür. Sistem, hastaların MR taramalarını güvenli bir şekilde depolar ve bu taramaları doktorlar ve isteğe bağlı olarak diğer yetkililer için erişilebilir hale getirir. Sistem, veritabanı olarak Qdrant Vector Database kullanarak hızlı ve esnek veri sorgulama olanakları sunar.

#### 2.1.1 Sistem Arayüzleri

MR cihazlarından gelen görüntülerin otomatik olarak sisteme yüklenmesi ve kullanılabilir görüntülerin kaydedilmesi için bir arayüz bulunacaktır.

#### 2.1.2 Kullanıcı Arayüzleri

Kullanıcı arayüzü için PyQt5 kütüphanesinden yararlanılacaktır. Qt Designer üzerinden ilgili kullanıcı arayüzü tasarlanıp kullanıcı arayüzü oluşturulacaktır.

#### 2.1.3 Donanım Arayüzleri

Sistem, yerel veri depolama ve yedekleme sunucuları ile entegre olacaktır.

#### 2.1.4 Yazılım Arayüzleri

Hastalara ait bilgilerin veritabanına kaydedilmesi ve veri tabanından okunması Python aracılığıyla gerçekleştirilecektir. Hastaya ait MR görüntüsü ve radyoloji raporları, veri tabanlarına kaydedilmeden önce ilgili embedding modeller ile vektörlere dönüştürülerek vektör veri tabanında depolanması planlanmaktadır.

### 2.2 Ürün İşlevleri

MR taramalarının depolanması ve arşivlenmesi.

Tarama sonuçlarının kullanıcıya sunulması.

Uygun görüntülerin seçilebilmesi

## 2.3 Kullanıcı Özellikleri

- Doktorlar, teknisyenler ve yöneticiler sisteme erişim sağlayacaktır.
- Kullanıcıların temel bilgisayar kullanım becerilerine sahip olmaları yeterlidir.

## 2.4 Kısıtlar

### 2.4.1 Uygulanan Kısıtlar

- Veri tabanında koleksiyona ait her bir point in eşsiz ve boş olmayan değeri olmalı.(Primary Key Constraint)
- Vektör boyutları önceden belirlenmiş ve sabit olmalı.(Domain Constraint)
- Her bir point'e ait image,report vektörü ve annotation bilgisi olmalı.(Domain Constraint)
- Annotation bilgisi JSON formatında veri tabanında tutulmalı.(Domain Constraint)
- Annotation bilgilerinin hepsi saklanmayacak. (Örneğin: Pixel Data)

### 2.4.2 Uygulanması Düşünülmeyen Kısıtlar

Problem kapsamında düşünülmeyen kısıt bulunmamaktadır.

## 3. Özel Gereksinimler

Bu bölümde, sistemin geliştirilmesi ve test edilmesi için gerekli tüm işlevsel ve işlevsel olmayan gereksinimler detaylandırılacaktır. Her gereksinim, sistemdeki belirli bir işlevi veya özellik setini tanımlayacaktır.

### 3.1 Fonksiyonel Gereksinimler

#### 3.1.1 İşlevsel Gereksinimler

**Gereksinim:** Kullanıcılar sisteme giriş yaptıktan sonra MR tarama sonuçlarını ve ilgili hasta bilgilerini görüntüleyebilecek.

**Gereksinim:** Sistem, tarama sonuçlarını ilgili hastaya ait dosyalarla ilişkilendirerek saklayacak.

**Gereksinim:** Sistem, tarama sonuçlarını tarih ve hasta adına göre arama yapma imkanı sunacak.

**Gereksinim:** Sistem, kullanıcıların tarama sonuçlarını indirmesine ve rapor oluşturmaya izin verecek.

**Gereksinim:** Sistem, bir hastaya ait görüntülerin iligli doktor(lar) tarafından incelenerek olası rahatsızlığın teşhis edilebilirliğini kontrol ederek görüntünün uygunluğuna bağlı olarak ilgili

görüntüyü “Kabul Et” ve “Reddet” butonları aracılığıyla veritabanına kayıt edebilme yetkisine sahip olacak.

### 3.1.2 Veri Yapısı ve Vektör Veri Tabanı Gereksinimleri

#### 3.1.2.1 Veri Tabanı Oluşturma

```
qdrant.recreate_collection(  
    collection_name="dicom_files",  
    vectors_config={  
        "image_vector": VectorParams(size=1000, distance=Distance.COSINE),  
        "report_vector": VectorParams(size=256, distance=Distance.DOT),  
    }  
)
```

Görsel 1. Verilerin vektör veri tabanında saklanması için oluşturulan koleksiyon

**Gereksinim 3.1.2.1.1:** Vektör veri tabanına ilgili MR görüntülerini, radyoloji rapor bilgilerini ve hasta bilgilerini saklamak için koleksiyon oluşturulmalı.

**Gereksinim 3.1.2.1.2:** Oluşturulan vektör veri tabanında vektör boyutu ve vektörler arası uzaklığın nasıl yapılacağı belirtilmeli.

**Gereksinim 3.1.2.1.3:** Oluşturulan koleksiyon veri tabanında tek ve eşsiz olmalı, olası diğer koleksiyonlarla karıştırılmamalı.

**Gereksinim 3.1.2.1.4:** Koleksiyonda MR görüntüsü ve radyoloji raporu embedding modeller ile vektörlere dönüştürülerek veri tabanında saklanacaktır. Bu yüzden bu iki veriye ait vector\_config bilgileri girilmelidir.

```
qdrant.create_payload_index(  
    collection_name="dicom_files",  
    field_name="name",  
    field_schema=PayloadSchemaType.KEYWORD,  
)  
  
qdrant.create_payload_index(  
    collection_name="dicom_files",  
    field_name="data",  
    field_schema=PayloadSchemaType.TEXT,  
)
```

Görsel 2. DICOM dosyasına ait annotation ların saklanması için oluşturulan payload fieldları

**Gereksinim 3.1.2.1.5:** Koleksiyonda DICOM dosyasında yer alan annotationlar payload(metadata) kısmında saklanacaktır.

**Gereksinim 3.1.2.1.6:** Veri tabanında arama yapılacağı zaman kolaylık sağlaması adına payload field da isim bilgisi ayrıca tutulacaktır. Böylelikle arama yaparken daha verimli sonuç elde edilmesi amaçlanmaktadır.

### 3.1.2.2 Veri Tabanına Veri Ekleme

```
ds = pydicom.dcmread(dicom_file)
image_array = ds.pixel_array
im = Image.fromarray(image_array).convert("RGB")

image_tensor = transform(im).unsqueeze(0)

encoded_image = model.encoder(image_tensor)

decoded_image = model.decoder(encoded_image)
```

Görsel 3. Vektör veri tabanında hasta ait MR görüntüsünün saklanması için ilgili kod

**Gereksinim 3.1.2.2.1:** DICOM görüntüsünü Pydicom kütüphanesi yardımıyla okuyup görüntü pixel array'e dönüştürülmeli.

**Gereksinim 3.1.2.2.2:** Pixel array'e dönüştürülen görüntü önceden eğitilmiş modellerle encode ederek image embedding elde edilmeli.

**Gereksinim 3.1.2.2.3:** Elde edilen embedding ise gerektiğinde decode edilerek tekrardan görüntü elde edilme işlemi gerçekleştirilmeli.

```
dicom_file = 'test.dcm'
dicom_data = pydicom.dcmread(dicom_file)
payload_dicts = dicom_data.to_json_dict()
```

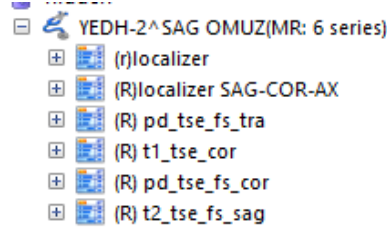
Görsel 4. Vektör veri tabanında hasta ait annotationların elde edilmesi için gereken kod

**Gereksinim 3.1.2.2.3:** İlgili DICOM dosyalarından annotation bilgileri JSON formatında elde edilmeli ve bu şekilde veri tabanında tutulmalıdır.

```
info2 = {"name": "veli", "annotations": payload_dicts}
qdrant.upsert([
    collection_name="dicom_files",
    points=[
        PointStruct(
            id=6,
            vector={
                "image_vector": encoded_image,
                "report_vector": encoded_report,
            },
            payload = info2
        ),
    ],
])
```

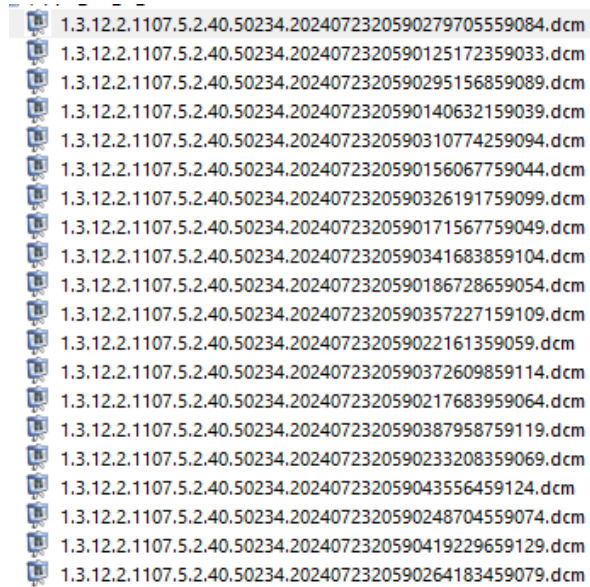
Görsel 5. Veri tabanına DICOM dosyasına ait bilgilerin kaydını yapan ilgili kod

**Gereksinim 3.1.2.2.3:** Her bir görüntü veri tabanına kaydedilirken point olarak koleksiyona eklenecektir. Her bir pointin eşsiz ve boş olmayan “id” değeri, vektör değerleri olmalıdır.



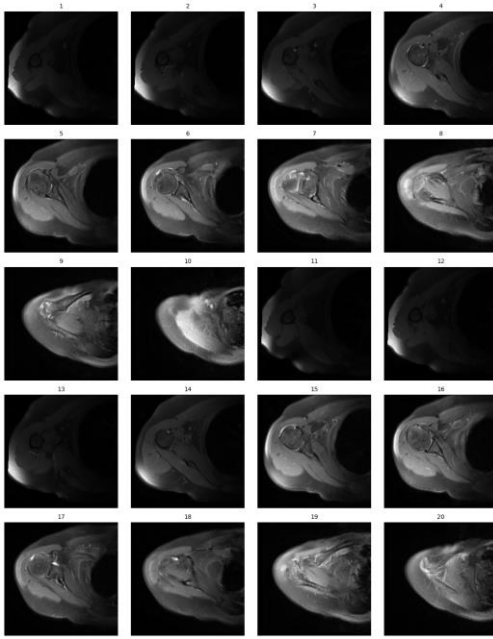
Görsel 6. Örnek bir hastaya ait DICOM dosyalarının yer aldığı klasörler

**Gereksinim 3.1.2.2.4:** Bir hastaya ait olan tüm DICOM klasörleri kaydedilmemelidir. Yukarıdaki örnekte “localizer” görüntülerinin kaydedilmesine gerek yoktur.

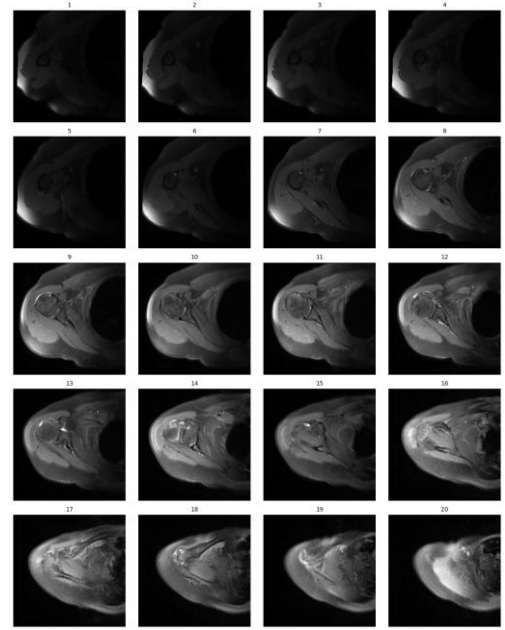


Görsel 7. Örnek bir hastaya ait bir DICOM klasörünün içinde yer alan DICOM dosyaları





Görsel 8. Timestamp'e göre sıralama



Görsel 9. Instance Number'a göre sıralama

**Gereksinim 3.1.2.2.5:** Bir hastanın belirli bölgesine ait DICOM görüntülerini kullanıcı arayüzünde gösterme esnasında görüntülerin sırası “Timestamp” değerlerine göre sıralama yapıldığında doğru sıralama oluşmamaktadır. Bu yüzden annotation kısmında yer alan “Instance Number” değerlerine göre yapıpıp ekranda gösterilmelidir.

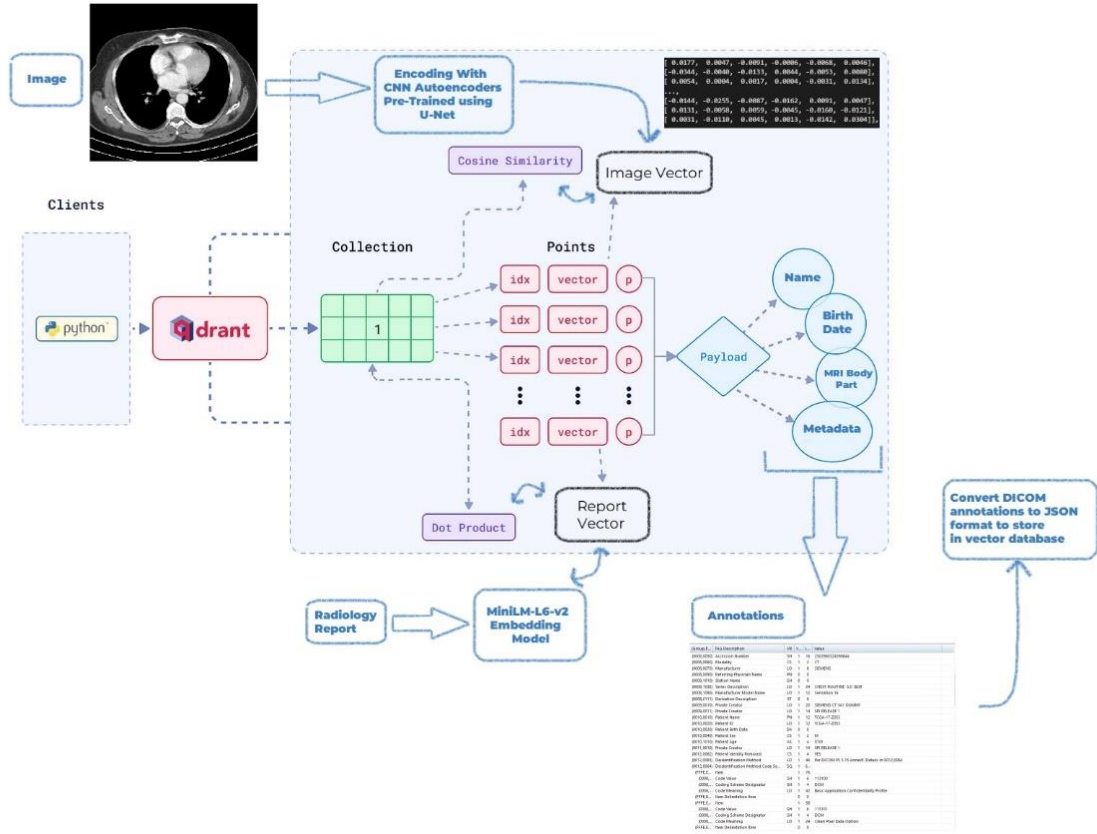
### 3.1.2.3 Veri Tabanına Veri Arama

```
qdrant.scroll(
    collection_name="dicom_files",
    scroll_filter=models.Filter(
        must=[
            models.FieldCondition(key="name", match=models.MatchValue(value="veli")),
        ]
    ),
    limit=1,
    with_payload=False,
    with_vectors=True,
)
```

Görsel 10. Vektör veri tabanında hasta ait MR görüntüsünün saklanması için ilgili kod

**Gereksinim 3.1.2.3.1:** Veri tabanından hastaya ait görüntüyü arama işlemini gerçekleştirirken payload kısmında yer alan isim bilgisi ile arama işlemi yapılmalı.

**Gereksinim 3.1.2.3.2:** Kullanıcı arayüzünde aranan isme göre o kişiye ait tüm görüntüye ait pointler veri tabanından çekilmeli, sonrasında işlenerek kullanıcıya sunulmalı.



Görsel 11. Vektör veri yapısını açıklayan şema

## 3.2 Fonksiyonel Olmayan Gereksinimler

### 3.2.1 Kalite Özellikleri

Kalite özellikleri, yazılımın güvenilir, kullanılabilir ve sürdürülebilir olmasını sağlamak gibi amaçlarla belirlenen kriterleri içerir.

#### 3.2.1.1 Kullanılabilirlik

**Gereksinim 3.2.1.1.1:** Yazılım, mevcut hastane bilgi sistemleri ile uyumlu olmalıdır.

**Gereksinim 3.2.1.1.2:** Veri tabanı tasarımında kullanılan yapıların performans optimizasyonunu engellemeyecek şekilde modüler ve esnek olması gerekmektedir.

**Gereksinim 3.2.1.1.3:** Yazılım, kullanıcı dostu ve kolay anlaşılır bir arayüze sahip olmalıdır. Eğitim gereksinimlerini minimuma indirmek için sezgisel tasarım tercih edilecektir.

**Gereksinim 3.2.1.1.4:** Kullanıcılar, sisteme erişim sağladıktan sonra hızlıca hasta bilgilerine ve MR sonuçlarına ulaşabilmelidir.

**Gereksinim 3.2.1.1.5:** Sistem, kullanıcı hatalarını minimuma indirmek için geri bildirim mekanizmaları ve kullanıcı hatalarını tespit edip çözüm önerileri sunan kılavuzlar içermelidir.

**Gereksinim 3.2.1.1.6:** Kullanıcıların yazılımı etkin kullanabilmesi için yardım dokümanları, kılavuzlar sağlanacaktır.

### *3.2.1.2 Sürdürülebilirlik*

**Gereksinim 3.2.1.2.1:** Yazılım, gelecekteki teknolojik gelişmelere ve sektörel değişikliklere uyum sağlayacak esnek bir mimari ile tasarlanmalıdır. Bu, yeni teknolojilerin entegrasyonunu kolaylaştıracak şekilde yapılandırılmalıdır.

**Gereksinim 3.2.1.2.2:** Yazılım güncellemeleri, minimum kesintiyle gerçekleştirilmelidir. Bakım ve güncellemeler sırasında sistemin kritik fonksiyonlarının sürekli çalışabilirliği sağlanmalıdır.

**Gereksinim 3.2.1.2.3:** Sistem, kullanıcı talepleri ve ihtiyaçlarına göre kolayca geliştirilebilir ve genişletilebilir olmalıdır. Yazılımın modüler yapısı, yeni özelliklerin eklenmesini ve mevcut özelliklerin iyileştirilmesini kolaylaştırmalıdır.

**Gereksinim 3.2.1.2.4:** Yazılım, sağlık sektörü ile ilgili düzenlemelerdeki değişikliklere uyum sağlayabilecek şekilde yapılandırılmalıdır. Mevzuat değişikliklerine hızlı yanıt vererek, hukuki ve sektörel gereksinimlerin karşılanması garanti altına alınmalıdır.

### *3.2.1.3 Güvenlik*

**Gereksinim 3.2.1.3.1:** Güvenlik ihlallerini tespit etmek ve izlemek için sistem logları sürekli olarak kayıt altına alınmalı ve düzenli olarak analiz edilmelidir.

**Gereksinim 3.2.1.3.2:** Hasta bilgilerine ve MR verilerine erişim, kullanıcı kimlik doğrulama ile güvence altına alınacaktır.

**Gereksinim 3.2.1.3.3:** Sistem, hastane ağları gibi kapalı ve kontrollü ağlarda kullanılacağından, dış sistemlerle entegrasyon sınırlı olacak ve güvenlik politikaları çerçevesinde ele alınacaktır.

### *3.2.1.4 Bakım-yapılabilirlik*

**Gereksinim 3.2.2.4.1:** Yeni özellikler ve güncellemeler, sistemin mevcut işlevselliğini etkilemeden modüler olarak eklenebilmelidir.

### 3.2.1.5 Veri tabanı Mimarisi ve Sistem Gereksinimleri

**Gereksinim 3.2.1.5.1:** Qdrant vektör veri tabanı kurulumu yapılarak veriler bu veri tabanında saklanacaktır.

**Gereksinim 3.2.1.5.2:** Qdrant vektör veri tabanı için uygun donanım ve performans gereksinimleri sağlanacaktır.

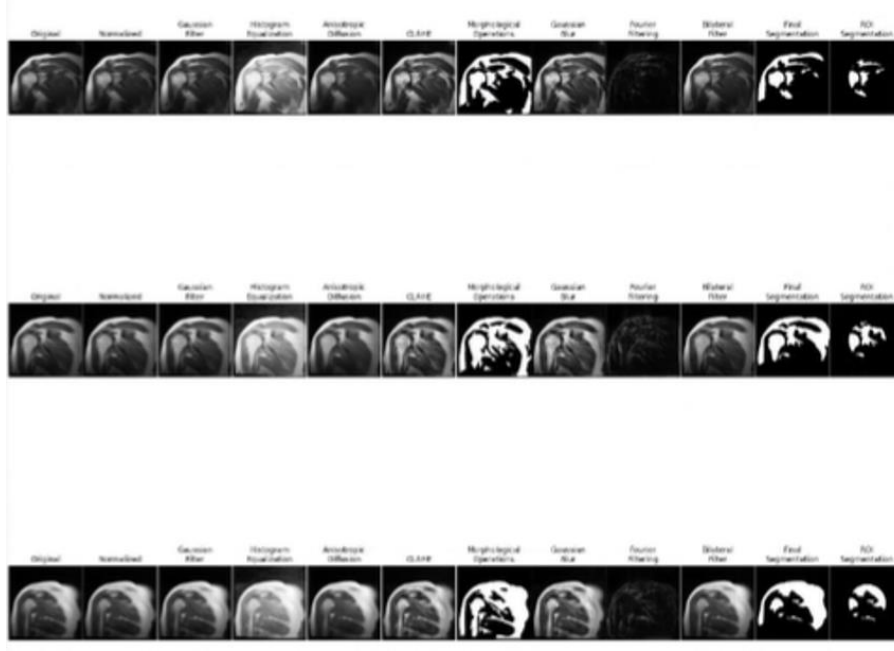
**Gereksinim 3.2.1.5.3:** Yazılım, masaüstü cihazlarla uyumlu olmalıdır. Bu sebeple, tasarımda farklı ekran çözünürlüklerine ve işletim sistemlerine uyum sağlayacak bir mimari tercih edilmelidir.

## 3.3 Arayüz Gereksinimleri

### 3.3.1 Kullanıcı Arayüz Gereksinimleri

- **3.3.1.1 Arayüz Tasarımı**
  - Kullanıcı arayüzü, PyQt5 kütüphanesi ve Qt Designer araçları kullanılarak geliştirilecektir.
  - Arayüz, sezgisel bir yapıya sahip olacak ve kullanıcıların kolayca hasta bilgilerine, MR taramalarına ve raporlara erişmesini sağlayacaktır.
  - Renk temaları ve yazı tipleri, sağlık çalışanlarının uzun süreli kullanımı göz önüne alınarak ergonomik ve sade bir şekilde tasarlanacaktır.
- **3.3.1.2 Giriş Ekranı**
  - Uygulama açıldığı esnada veri tabanına bağlı olunması gerekmektedir. Bu durum uygulama açıldığı anda kontrol edilmelidir.
- **3.3.1.3 Ana Ekran**
  - Hastaya ait ilk MR görüntülerinin görüntülenmesi için kullanıcının bilgisayardan DICOM dosyayı veya dosyaları seçebileceği “DICOM Dosya Seç” ve “DICOM Klasör Seç” şeklinde seçenekler olmalı.
  - Seçilen dosyaları veri tabanına kaydetmek için “Veri Tabanına Kaydet” şeklinde başka bir seçenek daha olmalı
  - Veri tabanı üzerinden hasta arama ve filtreleme seçenekleri sunulacaktır.
  - Seçilen hastanın geçmiş MR taramaları, tarih sırasına göre listelenecektir.
- **3.3.1.4 Görüntüleme Ekranı**
  - MR, X-Ray ve CT taramaları uygun çözünürlükte ve etkileşimli bir görüntüleyici ile sunulacaktır.
  - MR görüntüsüne ait temel bilgiler(hasta bilgisi, MR ın çekildiği bölge bilgisi vs.) MR görüntüsünün etrafında gösterilmeli.

- Aynı şekilde radyoloji raporu da MR görüntüsüyle birlikte gösterilmeli.
- Kullanıcı hastaya ait diğer annotation bilgilerine toolbar üzerinden “Annotations” kısmını seçip görüntüleyebilmeli.
- Sistem, 1920x1080 çözünürlük başta olmak üzere, farklı ekran boyutlarında düzgün çalışacak şekilde tasarlanacaktır.



Görsel 12. Hastaya ait MR görüntüsünün farklı filtreleme yöntemleriyle gösterimi

- Ekranda yer alan MR görüntüsü isteğe göre çeşitli filtreleme yöntemlerinden geçirilip kullanıcıya sunulabilmelidir.

## 4. Gereksinimlerin Önceliği ve Kritikliği

Gereksinimlerin önceliklendirilmesi, yazılımın işlevselliği, güvenliği ve kullanıcı deneyimi açısından kritik olan özellikleri daha erken aşamalarda gerçekleştirmeye odaklanacaktır.

- **Yüksek Öncelikli Gereksinimler:** Güvenlik, veritabanı yönetimi, performans ve kullanıcı arayüzü ile ilgili gereksinimler yüksek öncelikli kabul edilecektir. Özellikle hasta bilgileri ve MR sonuçlarının gizliliği ve güvenliği kritik önem taşımaktadır.
- **Orta Öncelikli Gereksinimler:** Sürdürülebilirlik ve bakım yapılabilirlik gibi özellikler, sistemin uzun vadede işletimini desteklemek için orta öncelikli olacaktır.
- **Düşük Öncelikli Gereksinimler:** Kapsamlı kişiselleştirme seçenekleri ve gelişmiş kullanıcı arayüzü özellikleri gibi gereksinimler, temel işlevlerin tamamlanmasından sonra uygulanabilir.

## 5. Veri Tabanı ve Veri Yapısı İle İlgili Tüm Gereksinimler

1. Vektör veri tabanında MR görüntüleri, radyoloji rapor bilgileri ve hasta bilgilerini saklamak için koleksiyon oluşturulmalı.
2. Vektör boyutu ve vektörler arası uzaklık ölçütleri tanımlanmalı.
3. Koleksiyon tek ve eşsiz olmalı, diğer koleksiyonlarla karıştırılmamalı.
4. MR görüntüleri ve radyoloji raporları embedding modeller ile vektörlere dönüştürülmeli ve veri tabanında saklanmalı.
5. DICOM dosyasında bulunan annotation bilgileri payload (metadata) kısmında tutulmalı.
6. Veri tabanında arama yapılırken kolaylık sağlamak amacıyla payload alanında isim bilgisi tutulmalı.
7. Pydicom kütüphanesi kullanılarak DICOM görüntüleri okunup pixel array'e dönüştürülmeli.
8. Pixel array'ler, önceden eğitilmiş modellerle encode edilerek embedding verisi oluşturulmalı.
9. Elde edilen embedding gerektiğinde decode edilerek tekrar görüntü elde edilebilmeli.
10. DICOM dosyalarındaki annotation bilgileri JSON formatında elde edilmeli ve veri tabanında bu şekilde saklanmalı.
11. Her bir görüntü veri tabanına "point" olarak kaydedilmeli.
12. Her point eşsiz ve boş olmayan bir "id" değeri içermeli.
13. Pointler vektör değerleri içermeli.
14. Hastaya ait görüntü aramalarında payload kısmındaki isim bilgisi kullanılmalı.
15. Kullanıcı arayüzünde, aranan isime göre tüm görüntü point'leri veri tabanından alınarak işlenmeli ve kullanıcıya sunulmalı.
16. Yazılım, mevcut hastane bilgi sistemleriyle uyumlu olmalı.
17. Veri tabanı yapıları performansı engellemeyecek şekilde modüler ve esnek tasarlanmalı.
18. Kullanıcı dostu ve sezgisel bir arayüze sahip olmalı, minimum eğitim gereksinimiyle kullanılabilir olmalı.
19. Kullanıcılar hasta bilgilerine ve MR sonuçlarına hızlıca erişebilmeli.
20. Yazılım, teknolojik gelişmelere ve sektörel değişikliklere uyum sağlayacak şekilde tasarlanmalı.
21. Güncellemeler sırasında sistemin kritik işlevleri çalışmaya devam etmeli.
22. Sistem, kullanıcı ihtiyaçlarına göre kolayca genişletilebilir olmalı.
23. Sağlık sektörü düzenlemelerine uyum sağlamak için yapılandırılmalı.
24. Hasta bilgileri ve MR verilerine erişim, kullanıcı kimlik doğrulaması ile sağlanmalı.
25. Sistem, güvenlik politikaları çerçevesinde dış sistemlerle sınırlı entegrasyona sahip olmalı.
26. Yeni özellikler ve güncellemeler, mevcut işlevselliği bozmadan modüler olarak eklenebilmelidir.
27. Qdrant vektör veri tabanı kurulumu yapılmalı.
28. Qdrant veri tabanı için uygun donanım ve performans gereksinimleri sağlanmalı.
29. Yazılım, masaüstü cihazlarla uyumlu olmalı ve farklı ekran çözünürlüklerini desteklemeli.
30. PyQt5 ve Qt Designer kullanılarak sezgisel bir arayüz tasarlanmalı.

31. Renk temaları ve yazı tipleri ergonomik ve sade olacak şekilde ayarlanmalı.
32. Uygulama açılırken veri tabanı bağlantısı kontrol edilmeli.
33. Kullanıcı DICOM dosya veya klasörlerini seçebilmeli.
34. Seçilen dosyalar, "Veri Tabanına Kaydet" seçeneği ile veri tabanına aktarılabilir.
35. Hasta arama ve filtreleme özellikleri sunulmalı.
36. Hastanın geçmiş MR taramaları, tarih sırasına göre listelenmeli.
37. MR, X-Ray ve CT taramaları uygun çözünürlükte ve etkileşimli bir görüntüleyici ile sunulmalı.
38. MR görüntülerine ait temel bilgiler ve radyoloji raporu aynı ekranda gösterilmeli.
39. Annotation bilgileri, "Annotations" sekmesinden erişilebilir olmalı.
40. Çeşitli ekran çözünürlüklerine uygun şekilde tasarlanmalı.
41. Güvenlik, veri tabanı yönetimi, performans ve kullanıcı arayüzü ile ilgili gereksinimler yüksek öncelikli kabul edilmelidir.
42. Sürdürülebilirlik ve bakım yapılabilirlik gibi özellikler, sistemin uzun vadede işletimini desteklemek için orta öncelikli olacaktır.
43. Yazılım, kullanıcıların sisteme erişim sağladıktan sonra hızlıca MR tarama sonuçlarına ve ilgili hasta bilgilerine ulaşmasını sağlamalıdır.
44. Tarama sonuçları, ilgili hastaya ait dosyalarla ilişkilendirilerek saklanmalıdır.
45. Sistem, kullanıcıların tarama sonuçlarını tarih ve hasta adına göre aramasını sağlamalıdır.
46. Kullanıcıların tarama sonuçlarını indirip rapor oluşturabilmelerine izin verilmelidir.
47. Sistem, veri tabanında koleksiyonların eşsiz olmasını sağlamalıdır (Primary Key Constraint).
48. Veri tabanındaki vektör boyutları önceden belirlenmiş ve sabit olmalıdır (Domain Constraint).
49. Her bir görüntü ve radyoloji raporu veritabanına kaydedilirken vektör verileri ve annotation bilgileri bulunmalıdır (Domain Constraint).
50. Annotation bilgileri JSON formatında veri tabanında tutulmalıdır.
51. Veri tabanı tasarımında performansı optimize etmek için modüler bir yapı tercih edilmelidir.
52. Kullanıcılar, hasta isim bilgisiyle veri tabanında arama yaparak daha verimli sonuçlara ulaşabilmelidir.
53. Veri tabanı, Qdrant vektör veri tabanı kullanılarak yapılandırılmalı ve vektör veri saklama kapasitesine uygun donanım sağlanmalıdır.
54. Yazılım güncellemeleri sırasında minimum kesintiyle sürdürülebilir olmalıdır.
55. Mevcut sistem mimarisi, yeni özelliklerin kolayca entegre edilmesine olanak tanımalıdır.
56. Sistem, düzenli analiz ve güncellemeler ile veri güvenliğini sağlamalıdır.
57. Kullanıcı dostu bir yardım kılavuzu ve eğitim materyalleri sağlanmalıdır.
58. Arayüz tasarımı sağlık çalışanlarının uzun süreli kullanımına uygun ergonomik prensipler doğrultusunda yapılmalıdır.
59. Hasta bilgileri, MR sonuçları ve annotation bilgileri, kullanıcıların erişebileceği sezgisel bir arayüzle sunulmalıdır.
60. Uygulamanın giriş ekranında veri tabanı bağlantısı kontrol edilmeli ve kullanıcıya durum bilgisi verilmelidir.
61. Kullanıcı arayüzünde geçmiş MR taramaları tarih sırasına göre listelenmelidir.



62. Yazılımın görüntüleme ekranında MR görüntüleri, temel hasta bilgileri ve radyoloji raporu aynı ekranda gösterilmelidir.
63. MR, X-Ray ve CT taramalarını etkileşimli bir görüntüleyici ile sunacak sistem tasarlanmalıdır.
64. Annotation bilgileri için kullanıcı arayüzünde özel bir erişim aracı ("Annotations" seçeneği) sunulmalıdır.
65. Yazılımın farklı ekran çözünürlüklerinde sorunsuz çalışması sağlanmalıdır.

## 6. Proje Görev Dağılımı ve Adam-Gün Hesabı

TARİH	ÖĞRENCİ	YAPILAN İŞ	BAŞLANGIÇ ZAMANI	BİTİŞ ZAMANI
15.10.2024	Atakan Berber	Toplantı	19:15	21:23
15.10.2024	Eren Eroğlu	Toplantı	19:15	21:23
15.10.2024	Umut Öztürk	Toplantı	19:15	21:23
20.10.2024	Atakan Berber	qdrant veritabanına verilerin eklenmesi hakkında araştırma	11:15	14:27
20.10.2024	Eren Eroğlu	qdrant veritabanına verilerin eklenmesi hakkında araştırma	13:00	15:44
19.10.2024	Umut Öztürk	qdrant veritabanına verilerin eklenmesi hakkında araştırma	17:00	18:55
26.10.2024	Atakan Berber	Kullanıcı arayüzü çalışmaları	20:00	22:23
26.10.2024	Eren Eroğlu	Kullanıcı arayüzü çalışmaları	20:00	22:23
26.10.2024	Umut Öztürk	Kullanıcı arayüzü çalışmaları	20:00	22:23
10.11.2024	Atakan Berber	MR görüntülerinden image embedding elde edilmesi için uygun autoencoderların araştırılması	10:10	12:50
10.11.2024	Eren Eroğlu	dicom dosyalarının veritabanına kaydedilmesi ve arayüzde görüntülenmesi için incelenmesi	09:45	13:00
10.11.2024	Umut Öztürk	MR görüntülerinden image embedding elde edilmesi için uygun autoencoderların araştırılması	10:10	14:34
11.11.2024	Atakan Berber	dicom dosyalarının veritabanına kaydedilmesi ve arayüzde görüntülenmesi için incelenmesi	13:15	15:00
14.11.2024	Atakan Berber	Toplantı	16:15	17:45
14.11.2024	Eren Eroğlu	Toplantı	16:15	18:00
14.11.2024	Umut Öztürk	Toplantı	16:15	18:00
19.11.2024	Atakan Berber	Rapor	20:00	22:15
19.11.2024	Eren Eroğlu	Rapor	20:00	22:15
19.11.2024	Umut Öztürk	Rapor	20:00	22:15
25.11.2024	Atakan Berber	Rapor	21:30	23:40
25.11.2024	Eren Eroğlu	Rapor	21:30	23:40
25.11.2024	Umut Öztürk	Rapor	20:30	22:40
25.11.2024	Atakan Berber	Sunum Hazırlama	22:40	23:45
25.11.2024	Eren Eroğlu	Sunum Hazırlama	22:40	23:45
25.11.2024	Umut Öztürk	Sunum Hazırlama	22:40	23:45
26.11.2024	Atakan Berber	Sunum Hazırlama	19:45	23:00
26.11.2024	Eren Eroğlu	Sunum Hazırlama	19:45	23:00
26.11.2024	Umut Öztürk	Sunum Hazırlama	19:45	23:00



