

KOCAELİ ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ

TBL331: VERİTABANI YÖNETİM SİSTEMLERİ

Proje Raporu

Proje Adı

HASTANE OTOMASYONU

17. Proje Grubu

İrem Karayel – 231307101

Gizemnur Arslan – 231307024

Umut Şahin – 231307091

Proje Yürütücüleri

Dr. Öğr.Gör. Samet DİRİ

Arş.Gör. M.M.Enes YURTSEVER

Arş.Gör. Zeynep SARI

İÇİNDEKİLER

1. Özet
2. Problem Tanımı
 - 2.1. Sağlık Sistemindeki Problemler
 - 2.2. Projenin Hedefleri ve Çözüm Teklifi
3. Yapılan Araştırmalar
 - 3.1. Önceki Çalışmalar
 - 3.2. Teknolojik Araştırmalar
4. Akış Şeması
 - 4.1. İş Akışı
 - 4.2. Kullanıcı Akışları
5. Yazılım Mimarisi
 - 5.1. Mimari Seçimler
 - 5.2. Katmanlı Mimari
 - 5.3. Kullanılan Teknolojiler
6. Veri Tabanı Diyagramı
 - 6.1. Veritabanı Yapısı
 - 6.2. Tablolar Arası İlişkiler
 - 6.3. ER Diyagramı
7. Genel Yapı
 - 7.1. Proje Genel Yapısı
 - 7.2. Arayüz Tasarımı ve Kullanıcı Etkileşimi
8. Referanslar

1. ÖZET

Bu rapor, hastane otomasyon sisteminin geliştirilmesi ve uygulanması sürecini kapsamaktadır. Sağlık hizmetlerinde, özellikle hastane yönetimlerinde birçok manuel işlem ve hata riski bulunmaktadır. Projemiz, bu manuel süreçleri dijitalleştirerek sağlık hizmetlerini hızlandırmayı, verimli hale getirmeyi ve hata oranlarını azaltmayı amaçlamaktadır. Sistem, kullanıcıların kolayca randevu almasını, ilaç reçetelerini yazmasını, hastalık geçmişi görmesini ve sağlık hizmetlerine hızlı erişimini sağlar. Ayrıca, sistemdeki veriler güvenli bir şekilde saklanır ve her an erişilebilir durumdadır.

Proje; kullanıcı dostu arayüz tasarımı, güçlü güvenlik özellikleri ve ölçeklenebilir veritabanı yapısı ile sağlık sektörüne katkı sağlamayı hedeflemektedir.

2. PROBLEM TANIMI

2.1. Sağlık Sistemindeki Problemler

Sağlık sektörü, dünyadaki en önemli ve en hassas sektörlerden biridir. Ancak, bu sektördeki hizmetlerin büyük bir kısmı hâlâ verimsiz yöntemlerle ve eski sistemlerle yapılmaktadır. Çoğu hastane, sağlık verilerini dijital ortamda saklamak yerine hâlâ eski yazılımlar veya karmaşık sistemler kullanmaktadır. Bu durum, zaman kaybı, veri kayıpları, hata oranlarının artması ve verimsiz çalışma koşullarına yol açmaktadır. Özellikle, hasta bilgileri ve doktor reçeteleri gibi kritik verilerin doğru bir şekilde kaydedilmesi ve erişilmesi büyük bir zorluk teşkil etmektedir. Ayrıca, hastalar arasında yapılan randevu takibi, ilaç yazımı ve hasta geçmişi gibi işlemler çoğunlukla veritabanı tabanlı yazılımlar yerine eski sistemlerle yapılmaktadır.

2.2. Projenin Hedefleri ve Çözüm Teklifi

Bu proje, sağlık sektöründe dijitalleşme sağlayarak, veri yönetimi ve hasta hizmetleri süreçlerini hızlandırmayı amaçlamaktadır. Projenin ana hedefleri:

Veri güvenliği: Hastaların ve doktorların verilerinin güvenli bir şekilde saklanması.

İşlem hızını artırma: Randevu alımı, ilaç yazımı ve hasta takip işlemlerinin dijital ortamda yapılması.

Kullanıcı dostu arayüz: Sistemin kolayca kullanılabilir ve erişilebilir olması.

Verimlilik: Sağlık hizmetlerinin daha hızlı, doğru ve verimli sunulması.

Bu hedeflere ulaşmak için, bir üç katmanlı mimari kullanılarak, kullanıcı arayüzü, backend (sunucu) ve veritabanı arasında sağlam bir entegrasyon sağlanacaktır.

3. YAPILAN ARAŞTIRMALAR

3.1. Önceki Çalışmalar

Hastane otomasyon sistemlerine dair birçok çalışma yapılmıştır. Ancak mevcut sistemler genellikle kapsamlı değildir ve çoğu zaman yalnızca belirli bir alanda çalışmaktadır. Bazı sistemler yalnızca randevu takibi sunarken, diğerleri ilaç yazımı veya hastalık geçmişi yönetimi gibi tek bir alanla sınırlıdır. (Örneğin: MHRS, E-nabız)

Bu projede, tam entegre bir sistem geliştirilmesi amaçlanmıştır. Yani, hasta randevusu alabilmeli, ilaçları yazılabilmeli, geçmiş sağlık verilerine erişebilmelidir.

3.2. Teknolojik Araştırmalar

Proje için yapılan araştırmalarda, en uygun teknolojiler belirlenmiştir. Node.js ve React.js gibi modern teknolojiler, hem performans hem de kullanıcı deneyimi açısından en iyi sonuçları verecek şekilde seçilmiştir. Ayrıca, veritabanı yönetimi için PostgreSQL tercih edilmiştir çünkü ilişkisel veritabanları yüksek güvenlik ve veri doğruluğu sunmaktadır.

4. AKIŞ ŞEMASI

4.1. İş Akışı

Proje, belirli bir sırayla gerçekleştirilen iş akışları üzerinden işlemektedir. Kullanıcılar önce sisteme kayıt olur, giriş yapar, ardından randevu alır ve doktorların hastalık geçmişi hakkında bilgi edinir. Bu akış aşağıdaki gibi özetlenebilir:

- Kullanıcı giriş yapar. Randevu alır veya geçmiş sağlık bilgilerini görüntüler.
- Doktor, hasta randevularını onaylar ve ilaç reçetesi yazar.
- Hasta, yazılan ilaçları takip eder.

4.2. Kullanıcı Akışları

Kullanıcılar üç ana rol ile sisteme erişir: Admin, Doktor ve Hasta. Her rol için özel kullanıcı akışları oluşturulmuştur:

Admin Akışı: Admin, sisteme yeni kullanıcı ekler, genel yönetimi gerçekleştirir.

Doktor Akışı: Doktor, hasta bilgilerini görüntüler, randevu onaylar veya iptal eder, tedavi sürecini takip eder ve ilaç reçeteleri yazar.

Hasta Akışı: Hasta, randevu alır, ilaç reçetelerini takip eder ve tıbbi geçmişini görür.

5. YAZILIM MİMARİSİ

5.1. Mimari Seçimler

Proje, üç katmanlı mimari kullanılarak tasarlanmıştır. Bu mimari, sunum, işlem ve veri katmanlarının birbirinden bağımsız çalışmasını sağlar. Bu sayede, her katman kendi işlevine odaklanarak sistemin daha verimli çalışmasına olanak tanır.

5.2. Katmanlı Mimari

Sunum Katmanı (Frontend): Kullanıcıların etkileşime gireceği kısmı temsil eder. React.js ile geliştirilen bu katman, kullanıcı dostu ve dinamik bir arayüz sunmaktadır.

İşlem Katmanı (Backend): Sunucu tarafındaki tüm işlemleri yönetir. Node.js ve Express.js kullanılarak geliştirilmiştir.

Veri Katmanı (Database): PostgreSQL veritabanı, hasta ve doktor bilgileri ile randevu ve ilaç yazımları gibi kritik verileri saklar.

5.3. Kullanılan Teknolojiler

Frontend: React.js, Material UI

Backend: Node.js, Express.js

Veritabanı: PostgreSQL

Ana kod Editörü: Visual Studio Code

Versiyon Kontrolü: Git ve GitHub

6. VERİ TABANI DİYAGRAMI

6.1. Veritabanı Yapısı

Bu projede kullanılan veritabanı PostgreSQL tabanlıdır ve temel olarak **beş ana tablo** üzerine kuruludur: **users**, **appointments**, **prescriptions**, **medical_history** ve **departments**. Bu tablolar, sistemdeki kullanıcılar (hasta, doktor, yönetici), randevular, reçeteler, tıbbi geçmiş kayıtları ve departman bilgilerini yönetir.

Users Tablosu: Kullanıcı bilgilerini barındırır. Tek bir tabloda hem doktorlar, hem hastalar hem de sistem yöneticileri tanımlanır.

Alanlar:

- **id:** Birincil anahtar (otomatik artan)
- **name:** Kullanıcının tam adı
- **email:** Kullanıcının e-posta adresi (benzersiz)
- **password:** Giriş işlemleri için gerekli şifre
- **role:** Kullanıcının rolü (admin, doctor, patient)
- **department:** Kullanıcının bağlı olduğu birim (doktorlar için geçerlidir)
- **created_at:** Kayıt oluşturulma tarihi

Appointments (Randevular) Tablosu: Hastaların doktorlarla yaptıkları randevuların detaylarını içerir.

Alanlar:

- **id:** Birincil anahtar
- **patient_id:** Hasta kullanıcı ID'si (users tablosuna bağlı yabancı anahtar)
- **doctor_id:** Doktor kullanıcı ID'si (users tablosuna bağlı yabancı anahtar)
- **date:** Randevu tarihi ve saati
- **description:** Randevuya dair açıklama

- **status:** Randevunun durumu (pending, confirmed, cancelled)
- **created_at:** Kayıt oluşturulma tarihi

Prescriptions (Reçeteler) Tablosu: Hastalara yazılan ilaç reçeteleri ve talimatlarını içerir.

Alanlar:

- **id:** Birincil anahtar
- **patient_id:** Reçete alan hastanın kullanıcı ID'si (users tablosuna bağlı yabancı anahtar)
- **doctor_id:** Reçeteyi yazan doktorun kullanıcı ID'si (users tablosuna bağlı yabancı anahtar)
- **medications:** Reçetede yer alan ilaçlar
- **instructions:** Kullanım talimatları
- **created_at:** Kayıt oluşturulma tarihi

Medical History (Tıbbi Geçmiş) Tablosu: Hastaların geçmiş tıbbi kayıtlarını detaylı şekilde tutar.

Alanlar:

- **id:** Birincil anahtar
- **patient_id:** Hasta kullanıcı ID'si (users tablosuna bağlı yabancı anahtar)
- **doctor_id:** Kayıt yapan doktorun kullanıcı ID'si (users tablosuna bağlı yabancı anahtar)
- **diagnosis:** Teşhis bilgisi (zorunlu)
- **treatment:** Uygulanan veya önerilen tedavi
- **allergies:** Hastanın bilinen alerjileri
- **chronic_conditions:** Mevcut kronik hastalıklar
- **last_updated:** Son güncelleme zamanı

Departments (Departmanlar) Tablosu: Hastanedeki bölümlerin bilgilerini tutar.

Alanlar:

- **id:** Birincil anahtar
- **name:** Departman adı
- **head_id:** Departman başkanı olan doktorun kullanıcı ID'si (users tablosuna bağlı yabancı anahtar)
- **created_at:** Kayıt oluşturulma tarihi

6.2 Tablolar Arası İlişkiler

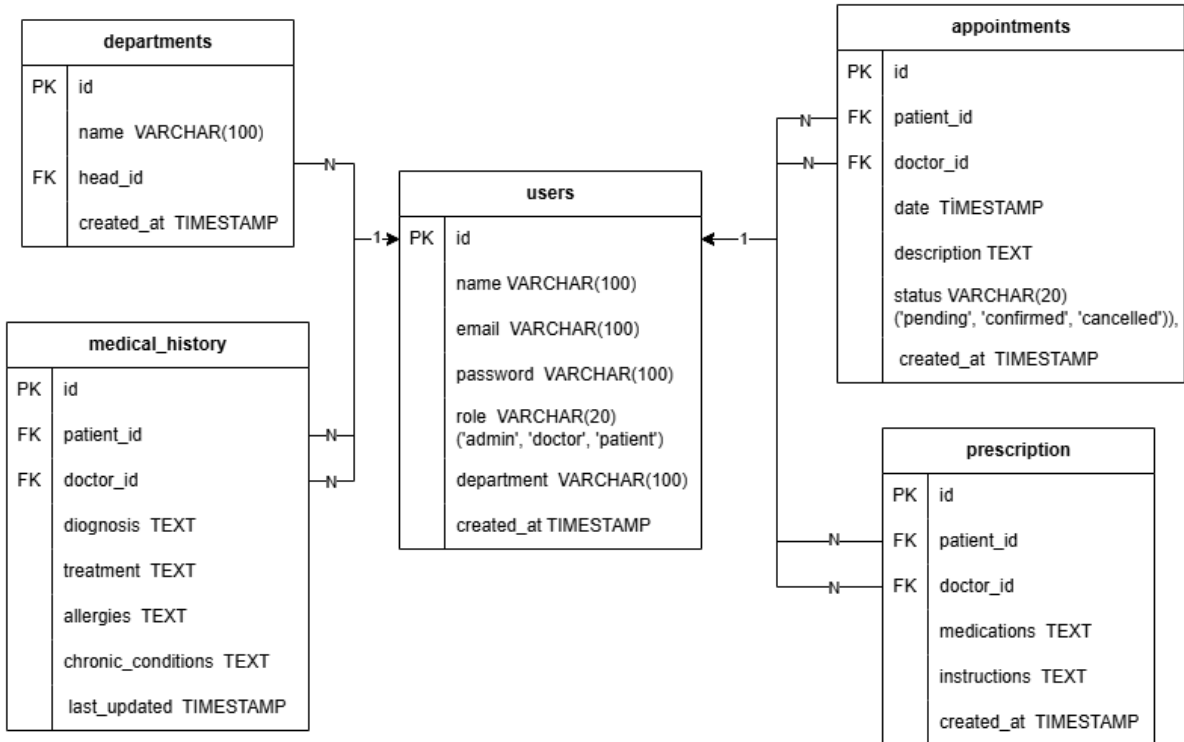
Aşağıda tablolar arasındaki ilişkiler açıklanmıştır:

- appointments.patient_id → users.id
- appointments.doctor_id → users.id
- prescriptions.patient_id → users.id
- prescriptions.doctor_id → users.id
- medical_history.patient_id → users.id
- medical_history.doctor_id → users.id
- departments.head_id → users.id

Bu ilişkiler sayesinde her randevu, reçete ve tıbbi kayıt ilgili doktor ve hasta ile ilişkilendirilir. Ayrıca, departmanlar ve bu departmanların başkanları da tanımlanır.

6.3 ER Diyagramı

Aşağıda proje kapsamında oluşturulan ilişkisel veri tabanı modelini gösteren ER diyagramı yer almaktadır.



Şekil 6.1. ER Diyagramı – Kullanıcılar, Randevular, Reçeteler, Tıbbi Geçmiş ve Departmanlar Arasındaki İlişkiler

7. GENEL YAPI

7.1. Proje Genel Yapısı

Bu proje, sađlık sektörünün dijital dönüşümüne katkı sağlamak amacıyla geliştirilmiş kapsamlı bir **Hastane Otomasyon Sistemi**dir. Temel hedef, hastanelerdeki süreçlerin daha verimli, hızlı ve güvenli bir şekilde yönetilmesini sağlamaktır. Sistem; **hasta, doktor ve yönetici** olmak üzere üç farklı kullanıcı rolü üzerinden hizmet verir ve her role özel, sade ve kullanıcı dostu arayüzler sunar.

Kullanıcılar, sistem üzerinden kolayca **randevu** alabilir, **tıbbi geçmiş bilgilerini** görüntüleyebilir, doktorlar ise **reçete** yazabilir, **hasta tıbbi kayıtlarını (medical_history)** güncelleyebilir ve randevularını yönetebilirler. Tüm işlemler, **modern web teknolojileri** kullanılarak geliştirilmiş, güvenli ve performanslı bir altyapı üzerinde gerçekleştirilir.

Veri yönetimi, merkezi bir veritabanı sistemi üzerinden yapılmakta olup, kullanıcı bilgileri (users) ve tüm medikal veriler (appointments, prescriptions, medical_history, departments) yüksek güvenlik standartları ile korunmaktadır. Projede kullanılan arayüzler, kullanıcı alışkanlıkları gözeticiler sade ve sezgisel bir şekilde tasarlanmıştır. Böylece kullanıcıların sisteme hızlı uyum sağlaması ve işlemleri kolayca gerçekleştirmesi mümkün hale gelmiştir. Hem estetik hem de işlevsellik açısından dengeli bir yapı ortaya çıkmıştır.

7.2 Arayüz Tasarımı ve Kullanıcı Etkileşimi

Sistemin kullanıcı arayüzü, her seviyeden kullanıcı için anlaşılır ve etkileşimli olacak şekilde planlanmıştır. Özellikle kullanıcıların sistem içerisinde kaybolmasını engellemek adına; açık menüler, ikon destekli butonlar, açıklayıcı uyarılar ve yönlendirici tasarım öğeleri eklenmiştir.

Her rol için özel olarak tasarlanan panel görünümü sayesinde; doktorlar kendi hastalarını ve randevularını (appointments), reçetelerini (prescriptions) yönetebilirken, hastalar da sadece kendileriyle ilgili bilgilere erişim sağlar. Bu rol tabanlı yapı, veri güvenliği ve kullanım kolaylığı açısından önemli bir avantaj sunmaktadır.

Mockup görselleri ve prototip örnekleri, sistemin ön geliştirme sürecinde kullanıcı deneyimini öngörmeye ve hataları minimuma indirmeye olanak tanımıştır. Bu sayede proje, sadece teknik olarak değil, **kullanıcı odaklılık** açısından da yüksek standartlara ulaşmıştır.

8. REFERANSLAR

- [1] T. Connolly and C. Begg, *Database Systems: A Practical Approach to Design, Implementation, and Management*, 6th ed. Pearson Education, 2014.
- [2] A. Silberschatz, H. Korth, and S. Sudarshan, *Database System Concepts*, 7th ed., McGraw-Hill Education, 2019.
- [3] Microsoft, "Getting Started with Visual Studio Code," [Online]. Available: <https://code.visualstudio.com/docs>. [Accessed: 12 May 2025].
- [4] PostgreSQL Global Development Group, "PostgreSQL Documentation," [Online]. Available: <https://www.postgresql.org/docs/>. [Accessed: May 16, 2025].
- [5] "SQL Tutorial," W3Schools, [Online]. Available: <https://www.w3schools.com/sql/>. [Accessed: 16 May 2025].
- [6] Material UI, "Material UI: React components for faster and easier web development," [Online]. Available: <https://mui.com>. [Accessed: 12 May 2025].
- [7] GitHub Inc., "GitHub Documentation," [Online]. Available: <https://docs.github.com/en>. [Accessed: 12 May 2025].
- [8] OpenAI, "ChatGPT – AI-powered assistant," [Online]. Available: <https://chat.openai.com/>. [Accessed: 16 May 2025].
- [9] diagrams.net, "draw.io – Diagramming Tool," [Online]. Available: <https://www.diagrams.net/>. [Accessed: 16 May 2025].
- [10] World Health Organization (WHO), "Digital Health," [Online]. Available: <https://www.who.int/health-topics/digital-health>. [Accessed: 12 May 2025].