

Programlama Laboratuvarı II, Proje III

1. Furkan Can İşçi
Bilgisayar Mühendisliği
Kocaeli Üniversitesi

Kocaeli, Türkiye
furkancanisci@outlook.com

2. Fevzi İsmail Şahin
Bilgisayar Mühendisliği
Kocaeli Üniversitesi

Kocaeli, Türkiye
sahinismail050@gmail.com

I. ÖZET

Bu proje, Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümü'nde Programlama Lab. II dersi kapsamında gerçekleştirilen bir proje olan "Hasta Takip ve Yönetim Sistemi" içermektedir. Proje, bir hastane yönetim sistemi olarak tasarlanmıştır ve PHP programlama dili ile PostgreSQL veritabanı kullanılarak geliştirilmiştir. Temel amacı, hastaların kayıt oluşturabileceği, doktorlarla randevu alabileceği, tıbbi raporları saklayabileceği ve genel olarak sağlıkla ilgili işlemleri yönetebileceği bir platform sunmaktır. Projede, nesne yönelimli programlama prensipleri kullanılarak her bir bileşen (hasta, doktor, randevu, tıbbi rapor vb.) bir sınıf olarak modellenmiş ve ilgili metotlar tanımlanmıştır. Projenin tamamlanması için, belirtilen istekler doğrultusunda hasta ekleme/silme, doktor ekleme/silme, randevu alma iptal etme, tıbbi rapor ekleme ve bilgi güncelleme gibi işlevler sağlanmıştır. Bu işlevlerin gerçekleşmesi durumunda, arayüzde bir onay mesajı gösterilmekte ve veri tabanı değişiklikleri anlık olarak raporlanmaktadır. Proje, nesne yönelimli programlama, veri tabanı yönetimi ve web tabanlı uygulama geliştirme becerilerini pekiştirmeyi amaçlamaktadır.

Anahtar Kelimeler: Hasta Takip ve Yönetim Sistemi, PHP, PostgreSQL, Nesne Yönelimli Programlama, Web Uygulaması.

II. GİRİŞ

Günümüzde sağlık sektöründe bilgi teknolojilerinin kullanımı, hasta bakımı ve yönetimi süreçlerini iyileştirmek adına önemli bir rol oynamaktadır. Bu bağlamda, hasta takip ve yönetim sistemleri, hastanelerde verimliliği artırmak ve hastalara daha iyi hizmet sunmak için hayati öneme sahiptir. Bu projenin amacı da, bu gereksinimlere yönelik bir çözüm sunmaktır. "Hasta Takip ve Yönetim Sistemi" projesi, Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümü'nde Programlama Lab. II dersi kapsamında gerçekleştirilmiştir. Proje, bir hastane yönetim sistemi olarak tasarlanmıştır ve PHP programlama dili ile PostgreSQL veritabanı kullanılarak geliştirilmiştir.

Projenin temel amacı, hastaların kayıt oluşturabileceği, doktorlarla randevu alabileceği, tıbbi raporları saklayabileceği ve genel olarak sağlıkla ilgili işlemleri yönetebileceği bir platform sunmaktır. Proje, nesne yönelimli programlama prensipleri kullanılarak her bir bileşenin (hasta, doktor, randevu, tıbbi rapor vb.) bir sınıf olarak modellenmesi ve ilgili metotların tanımlanması ile şekillendirilmiştir.

Veri tabanı tasarımı, hastalar, doktorlar, yöneticiler, randevular ve tıbbi raporlar gibi temel tabloları içermekte olup, ilişkisel veri tabanı yönetimi ve normalizasyon kurallarına uygun bir şekilde oluşturulmuştur. Güvenlik önlemleri olarak, hassas verilerin korunması için HTTPS protokolleri ve uygun şifreleme yöntemleri kullanılmıştır. Arayüz geliştirmeleri kapsamında, kullanıcıların veri tabanında yapılan tüm değişiklikleri takip edebilmesi sağlanmıştır. Kullanıcı arayüzünün yenilenmesi, dinamik bileşenlerin kullanımı ve bir bildirim sisteminin kurulumu gibi geliştirmeler yapılmıştır. Ayrıca, çeşitli sorguların yapılabilmesi ve sonuçlarının görüntülenebilmesi için bir sorgu sistemi de sağlanmıştır.

Projemizin amacı, programlama dersinde edindiğimiz bilgi ve becerileri kullanarak, karmaşık bir mühendislik sorununu çözmek ve pratik bir yazılım geliştirmektir. Bu projenin başarıyla tamamlanması, nesne yönelimli programlama, veri tabanı yönetimi ve web tabanlı uygulama geliştirme veri işleme konularındaki yeteneklerimizi geliştirmemize katkı sağlayacaktır.

III. YÖNTEM

Bu projede, Hasta Takip ve Yönetim Sistemi'nin geliştirilmesi için PHP, PostgreSQL, HTML, CSS ve JavaScript teknolojileri kullanılmıştır. Geliştirme süreci aşağıdaki adımları içermektedir:

Web Sunucusu Kurulumu:

Projede XAMPP 8.2.12 kullanılarak bir web sunucusu ortamı oluşturulmuştur. Web geliştirme ortamı sağlamaktadır.

Veritabanı Tasarımı:

İlk adım olarak, proje gereksinimleri doğrultusunda PostgreSQL 16 veritabanı için bir ER (Entity-Relationship) diyagramı oluşturulmuştur. Bu ER diyagramı, hastalar, doktorlar, randevular, tıbbi raporlar ve diğer ilgili tablolar arasındaki ilişkileri ve tabloların yapısını görsel olarak tanımlamaktadır.

Veritabanı Uygulaması:

Oluşturulan ER diyagramına göre, PostgreSQL veritabanı üzerinde ilgili tablolar ve ilişkiler oluşturulmuştur. Normalizasyon kurallarına uygun bir şekilde tablolar tasarlanmış ve ilişkilendirilmiştir.

Güvenlik Önlemleri:

Veritabanı erişimi ve kullanıcı girişleri HTTPS protokolü kullanılarak güvenli hale getirilmiştir. Bu, hassas verilerin korunması ve yetkisiz erişimlerin engellenmesi için önemli bir adımdır.

Kullanıcı Arayüzü Geliştirme:

HTML, CSS ve JavaScript kullanılarak, hastaların ve doktorların kullanabileceği kullanıcı arayüzü tasarlanmıştır. Bu arayüz, kullanıcıların sisteme giriş yapabileceği, randevu alabileceği, tıbbi raporlarını görüntüleyebileceği ve diğer sağlıkla ilgili işlemleri gerçekleştirebileceği bir platform sunmaktadır.

Bildirim Sistemi Kurulumu:

Doktorlar tıbbi rapor eklediğinde, ilgili hastalara bildirim gönderilmesi için bir bildirim sistemi entegre edilmiştir. Bu, hastaların tıbbi raporlarını zamanında ve doğru bir şekilde takip etmelerini sağlamaktadır.

Test ve Doğrulama:

Proje süresince, yapılan her adımın doğruluğunu ve işlevselliğini sağlamak için kapsamlı testler yapılmıştır. Kullanıcı senaryoları üzerinde testler gerçekleştirilerek, hata ayıklama süreci yürütülmüş ve herhangi bir hata veya eksiklik giderilmiştir.

sağlanmıştır. Bu da kullanıcıların verilerinin güvenliğini sağlamak için önemli bir adımdır.

Karşılaşılan Zorluklar:

Proje sürecinde, farklı teknolojilerin entegrasyonu ve uyumluluğu konusunda bazı zorluklar yaşanmıştır. Ancak, bu zorluklar detaylı testler ve çözüm odaklı yaklaşımlar ile başarıyla aşılmıştır. Özellikle, tıbbi raporların saklanması ve güvenliği gibi hassas konuların ele alınması gerekliliği bazı teknik zorluklar doğurmuştur. Ancak, bu zorluklar proje ekibimiz tarafından başarıyla çözülmüştür.

Gelecekteki Geliştirme Önerileri:

Projede sağlanan işlevlerin yanı sıra, sisteme daha fazla detay eklenerek kullanıcı deneyimi daha da geliştirilebilir. Örneğin, kullanıcıların sağlık geçmişlerini daha ayrıntılı bir şekilde yönetmelerine olanak tanıyan özellikler eklemek gelecekteki bir geliştirme alanı olabilir. Ayrıca, sistemdeki bildirim sisteminin daha etkili bir şekilde çalışması için iyileştirmeler yapılabilir. Özellikle, doktorlar tarafından eklenen tıbbi raporlarda önemli bir değişiklik olduğunda hastalara bildirim gönderilmesi gibi özellikler eklenerek kullanıcıların daha bilgilendirici bir deneyim yaşamaları sağlanabilir. Bu proje, hastane yönetim sistemlerinin modern bir şekilde tasarlanması ve geliştirilmesi konusunda önemli bir adım olmuştur. Kullanılan teknolojiler ve sağlanan işlevler, sağlık sektöründe daha etkili ve verimli bir yönetim sağlamak için önemli bir katkı sağlamaktadır. Gelecekte yapılacak geliştirmelerle birlikte, sistem daha da kullanıcı dostu ve işlevsel hale getirilebilir.

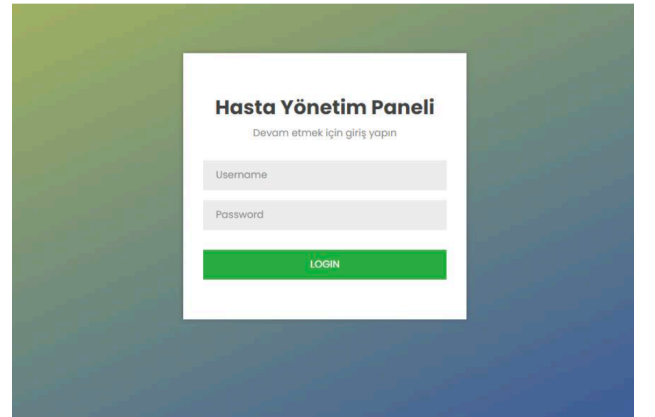
IV. SONUÇ

Bu proje kapsamında, Hasta Takip ve Yönetim Sistemi'nin geliştirilmesi için PHP, PostgreSQL, HTML, CSS ve JavaScript teknolojileri kullanılarak bir web uygulaması oluşturulmuştur. Proje, hastaların kayıt oluşturabilmesi, doktorlarla randevu alabilmesi, tıbbi raporları saklayabilmesi ve genel olarak sağlıkla ilgili işlemleri yönetebilmesi için bir platform sunmayı amaçlamaktadır.

Projenin Başarıları:

Geliştirilen sistem, hastaların ve doktorların istenen tüm işlevleri başarıyla yerine getirmesini sağlamıştır. Bu işlevler arasında hasta kaydı oluşturma, randevu alma, tıbbi raporları saklama ve yönetme gibi önemli özellikler bulunmaktadır. Kullanıcı arayüzü, kullanıcıların sistemle etkileşimini kolaylaştırmak ve sağlık işlemlerini daha erişilebilir hale getirmek için tasarlanmıştır. Bu sayede, kullanıcılar sistemi kolayca kullanabilir ve istedikleri bilgilere hızlıca erişebilirler. Güvenlik önlemleri olarak HTTPS protokolleri ve uygun şifreleme yöntemleri kullanılarak hassas verilerin korunması

V. DENEYSEL SONUÇLAR



Resim 1: Giriş Ekranı

CompanyLogo Profile Göt Çıkış Yap

Hastalar

Hasta Ekle &

Ad	Soyad	Doğum Tarihi	Cinsiyet	Telefon Numarası	Adres	
Ali	Çalçıkan	2003-01-05	Erkek	0560511654	Test adresi	Sil
Ahmet	Seçkin	2003-01-05	Erkek	0560511654	Test adresi	Sil

Doktorlar

Doktor Ekle &

Ad	Soyad	Uzmanlık Alanı	Çalıştığı Hastane	
Ali	Akkuş	Kardiyoloji	EMSAK HASTANESİ	Sil

Resim 2: Yönetici Paneli

CompanyLogo Profile Göt Çıkış Yap

Raporlar

Rapor Tarihi	Rapor İçeriği	Randevu Tarihi
2024-05-08	Randevu 02	2024-05-31
2024-05-07	Randevu 03	2024-05-31

Randevular

Yeni Randevu &

Randevu Tarihi	Randevu Saati	Doktor	Durum
2024-05-31	02:04:00	Hasan	İşletilebilir
2024-05-31	00:04:00	Hasan	İşletilebilir

Resim 6: Hasta Ekranı

Yönetici Profili

Ad:

Soyad:

Geri Gel Kaydet

Resim 3: Yönetici Profili

Rapor Ekle

Rapor İçeriği:

Randevular:

Rapor Tarihi:

Close Rapor Ekle

Resim 4: Rapor Ekleme

Randevu Tarihi	Randevu Saati	Doktor
2024-05-31	02:04:00	Ahmet

Randevular

Randevu Tarihi	Randevu Saati
2024-05-31	03:03:00
2024-05-31	02:04:00
2024-05-29	00:04:00

Hastalar

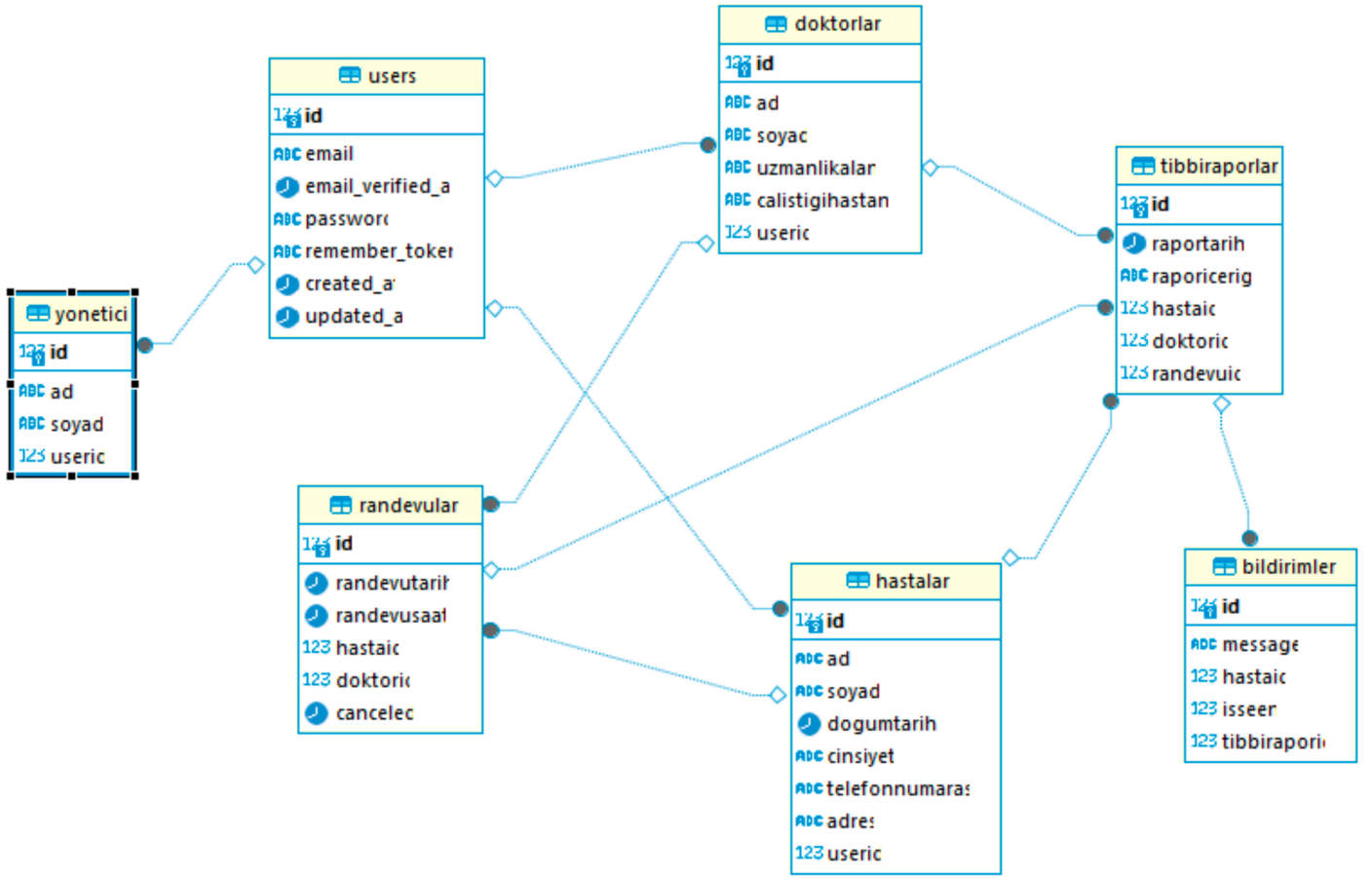
Rapor Tarihi	Rapor İçeriği	Randevu Tarihi	Randevu Saati
Kamil Çalçıkan	2003-01-05	Erkek	0560511654
Ahmet Seçkin	2003-01-05	Erkek	0560511654

Resim 5: Doktor Ekranı

VI. ER DİYAGRAMI & YALANCI KOD

```
<?php
class Hasta {
public $HastaID;
public $Ad;
public $Soyad;
// Diğer özellikler...
function __construct($HastaID, $Ad, $Soyad) {
$this->HastaID = $HastaID;
$this->Ad = $Ad;
$this->Soyad = $Soyad;
}
// Diğer metotlar...
}
class Doktor {
public $DoktorID;
public $Ad;
public $Soyad;
// Diğer özellikler...
function __construct($DoktorID, $Ad, $Soyad) {
$this->DoktorID = $DoktorID;
$this->Ad = $Ad;
$this->Soyad = $Soyad;
}
// Diğer metotlar...
}
class Randevu {
public $RandevuID;public $RandevuTarihi;
public $RandevuSaati;
// Diğer özellikler...
function __construct($RandevuID, $RandevuTarihi, $RandevuSaati) {
$this->RandevuID = $RandevuID;
$this->RandevuTarihi = $RandevuTarihi;
$this->RandevuSaati = $RandevuSaati;
}
// Diğer metotlar...
}
// Hasta, Doktor ve Randevu nesneleri oluşturma
$hasta = new Hasta(1, 'Ali', 'Veli');
$doktor = new Doktor(1, 'Ahmet', 'Mehmet');
$randevu = new Randevu(1, '2024-05-20', '14:00');
// Nesnelerin özelliklerini yazdırma
echo $hasta->Ad; // Ali
echo $doktor->Ad; // Ahmet
echo $randevu->RandevuTarihi; // 2024-05-20
?>
```

Resim 7: Yalancı Kod



Resim 8: Entity Relationship Model

KAYNAKLAR

- <https://www.php.net/manual/en/>
- <https://www.postgresql.org/docs/16/index.html>
- Prog. Lab. II - 3. Proje Forum
- <https://html.spec.whatwg.org/>
- <https://getbootstrap.com/docs/5.3/customize/overview/>
- <https://gioele-bencivenga.github.io/>
- <https://www.lucidchart.com> HYPERLINK "<http://www.lucidchart.com/>"