

Hasta Takip ve Yönetim Sistemi Projesi

Kocaeli Üniversitesi

1. Cihangir Emre Er
Bilgisayar Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
220202036@kocaeli.edu.tr

2. Berker Yiğit
Bilgisayar Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
220202046@kocaeli.edu.tr

I. Giriş (HEADING I)

Hastane yönetim sistemleri, sağlık hizmetlerinin etkin bir şekilde yönetilmesini sağlayan önemli araçlardır. Bu proje, modern teknolojiler kullanılarak hastane yönetim süreçlerini dijitalleştirerek hedeflemektedir. Veri tabanı tasarımları, nesne yönelimli programlama, güvenlik önlemleri ve arayüz geliştirmeleri gibi çeşitli bileşenlerin bir araya getirilmesiyle birlikte, kullanıcıların sağıyla ilgili işlemleri kolayca gerçekleştirebilecekleri bir platform oluşturulmuştur.

Özet

Bu rapor, bir hastane yönetim sistemi olarak tasarlanmış olan "Hasta Takip ve Yönetim Sistemi" projesini tanıtmaktadır. Projenin amacı, hastaların kayıt oluşturabileceği, doktorlarla randevu alabileceği, tıbbi raporları saklayabileceği ve genel olarak sağıyla ilgili işlemleri yönetebileceğii bir web uygulaması geliştirmektir. Rapor, projenin amaçlarını, tasarımlını, geliştirme sürecini ve sonuçlarını detaylı olarak açıklamaktadır.

II. YÖNTEM

Proje geliştirme süreci aşağıdaki adımları içermektedir:

Veri Tabanı Tasarımı: İlk olarak, projenin gereksinimlerine göre veri tabanı tasarlanmıştır. Temel tablolar belirlenmiş ve ilişkiler kurulmuştur. Veri tabanı normalizasyon kurallarına uygun olarak oluşturulmuştur.

Nesne Yönelimli Programlama: Her bileşen için uygun sınıflar oluşturulmuş ve gerekli metodlar tanımlanmıştır. Trigger fonksiyonları yazılarak veri tabanındaki güncellemelerin yönetimi sağlanmıştır.

Güvenlik Önlemleri: Hassas verilerin güvenliği için HTTPS protokolleri ve şifreleme yöntemleri kullanılmıştır.

Arayüz Geliştirmeleri: Kullanıcıların kolayca kullanabilecekleri bir arayüz tasarlanmıştır. AJAX çağrıları kullanılarak dinamik bileşenler eklenmiş ve bildirim sistemi

kurulmuştur..

```
<html lang="tr">
<head>
</head>
<body>
<header>
<nav class="navbar navbar-expand-lg navbar-light bg-light">
<a class="navbar-brand" href="#"> Hospital Hasta Arayüzü</a>
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

Randevularım</a>
</li>
<li class="nav-item"><a class="nav-link" href="http://127.0.0.1:8000/profilim">Profilim</a>
</li>
<li class="nav-item"><a class="nav-link" href="http://127.0.0.1:8000/search/doctor">Doktorlar</a>
</li>
</ul>
</div>
</header>
<main class="container mt-4">
<h1 class="text-center">Hos Geldiniz!</h1>
```

```
def login_view(request):
    if request.method == 'POST':
        user_type = request.POST.get('user_type')
        username = request.POST.get('username')
        password = request.POST.get('password')

        conn = sqlite3.connect('hospital.db')
        cursor = conn.cursor()

        # Determine the appropriate SQL query based on the user type
        if user_type == 'patient':
            cursor.execute('SELECT hastaSifre FROM Hastalar WHERE HastaID=?', [username])
        elif user_type == 'doctor':
            cursor.execute('SELECT doktorSifre FROM Doktorlar WHERE DoktorID=?', [username])
        elif user_type == 'm' (module):
            cursor.execute('SELECT yoneticiSifre FROM Yoneticiler WHERE YoneticiID=?', [username])
        else:
            messages.error(request, 'Geçersiz kullanıcı türü.')
            return render(request, "hospital/homePage.html")

        # Fetch the result from the executed query
        row = cursor.fetchone()

        # check if a result was found and if the password matches
        if row is not None and row[0] == password:
            # Redirect to the appropriate user interface
            if user_type == 'patient':
                return render(request, "hospital/hastaArayuz.html")
```

```
from django.urls import path
from . import views
# http://127.0.0.1:8000/
```

```
urlpatterns=[

    path("",views.index),
    path("homePage",views.index),
    path("SearchDoctor",views.SearchDoctor),
    path("SearchPatient",views.SearchPatient),
    path("profilim",views.profilim),
    path("hastaArayuz",views.hastaArayuz),
    path("randevuArayuz",views.randevuArayuz),
    path("Randevularim",views.randevularim),
    path("Doctors",views.Doctors),
    path("Manager",views.Manager),
    path("Patient",views.Patient),
]
```

```
def index(request):
    return render(request,"hospital/homePage.html")
def Doctors(request):
    return render(request,"hospital/doctors.html")
def SearchDoctor (function) method: Any
    if request.method == 'POST':
        query = request.POST.get('query', '')
        # Veritabanı bağlantısı
        conn = sqlite3.connect('hospital.db')
        cursor = conn.cursor()

        # Doktorları adı veya soyadı göre arama
        cursor.execute("SELECT * FROM doktorlar WHERE Ad LIKE ? OR Soyad LIKE ? ", ('%' + query + '%', '%' + query + '%'))
        results = cursor.fetchall()

        # Veritabanı bağlantısını kapat
        conn.close()

    return render(request, 'hospital/SearchDoctor.html', {'query': query, 'results': results})
```

```
<div>
<div class="card-deck">
<ol style="list-style-type: none;" type="1">
    <li class="list-group-item">Randevu 1 <button type="button" class="btn btn-danger btn-sm float-end">İptal </button></li>
    <li class="list-group-item">Randevu 2 <button type="button" class="btn btn-danger btn-sm float-end">İptal </button></li>
    <li class="list-group-item">Randevu 3 <button type="button" class="btn btn-danger btn-sm float-end">İptal </button></li>
</ol>
<button id="notification-btn" type="button" class="btn btn-outline-info mt-3" style="position: fixed; right: 20px; top: 70px; @Eldirime/But</button>
<a href="#">Profil </a>
<a href="#">Profil url'ü </a> <span style="float: right; margin-right: 10px; color: #ccc; font-size: 0.8em; border: 1px solid #ccc; padding: 2px 5px; border-radius: 10px; position: absolute; right: -10px; top: 50%;>Profil</span>
<div id="notification-popup" style="display: none; position: fixed; top: 150px; right: 20px; background-color: #fff; border: 1px solid #ccc; padding: 10px; width: fit-content; z-index: 1000; border-radius: 10px; font-size: 0.9em; font-weight: bold; text-align: center; opacity: 0.9; transition: all 0.3s ease; ">
    <span>Sizin randevularınız</span>
    <span>Bildirimi</span>
    <span>Görsel ve herhangi bir bildiriminiz bulunamadı!</span>
</div>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js"></script>
<script>
    const notificationContainer = document.getElementById('notification-html');
    const notificationPopup = document.getElementById('notification-popup');
</script>
```

```
<h2 class="text-center mt-4">Search Results:</h2>
<div class="table-responsive mt-3">
    <table class="table table-striped">
        <thead>
            <tr>
                <th scope="col">ID</th>
                <th scope="col">Ad</th>
                <th scope="col">Soyad</th>
                <th scope="col">DG</th>
                <th scope="col">Cinsiyet</th>
                <th scope="col">Telefon</th>
                <th scope="col">Adres</th>
            </tr>
        </thead>
        <tbody>
            {% for hastalar in results %}
                <tr>
                    <td>{{ hastalar.0 }}</td>
                    <td>{{ hastalar.2 }}</td>
                    <td>{{ hastalar.3 }}</td>
                    <td>{{ hastalar.4 }}</td>
                    <td>{{ hastalar.5 }}</td>
                    <td>{{ hastalar.6 }}</td>
                    <td>{{ hastalar.7 }}</td>
                </tr>
            {% endfor %}
        </tbody>
    </table>
</div>
```

```
# Password validation
# https://docs.djangoproject.com/en/5.0/ref/settings/#auth-password-validation

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]
```

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [
            BASE_DIR / "templates"
        ],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

```
</head>
<body>
<div class="container">
    <h1 class="mb-3">Açılış Hastanesi</h1>
    <form id="loginForm" method="post" class="mt-4 form-box">
        {% csrf_token %}
        <div class="mb-3">
            <label for="user_type" class="form-label">Giriş yapmak istediğiniz kullanıcı&lt;input type="radio" name="user_type" value="doctor"&gt;</label>
            <div class="form-check">
                <input class="form-check-input" type="radio" name="user_type" id="doctor" value="doctor">
                <small>The input element represents a typed data field, usually with a form control to allow the user to edit the data.</small>
            </div>
            <div>MDN Reference
                <input class="form-check-input" type="radio" name="user_type" id="patient" value="patient">
                <label class="form-check-label" for="patient">Hasta</label>
            </div>
            <div class="form-check">
                <input class="form-check-input" type="radio" name="user_type" id="manager" value="manager">
                <label class="form-check-label" for="manager">Yönetici</label>
            </div>
        </div>
        <div class="mb-3">
            <label for="username" class="form-label">ID:</label>
            <input type="text" class="form-control" id="username" name="username">
        </div>
        <div class="mb-3">
```

III. DENEYSEL SONUÇLAR

Projenin geliştirme süreci başarıyla tamamlanmıştır. Arayüz üzerinden yapılan işlemler veri tabanında doğru bir şekilde yansıtılmakta ve kullanıcılarla geri bildirimler verilmektedir. Güvenlik önlemleri sayesinde kullanıcı verileri korumakta ve gizlilik sağlanmaktadır..

```
<html lang="tr">
  <body>
    <header>
      </nav>
    </header>
    <main class="container mt-4">
      <h1 class="text-center">Hoşgeldiniz!</h1>
      <!-- Ana sayfa içeriği buraya gelecek -->
    </main>
    <footer class="footer mt-auto py-3 bg-light">
      <div class="container text-center">
        <!-- Footer içeriği buraya gelecek -->
      </div>
    </footer>
  </body>
</html>
```

```
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Profilim</title>
</head>
<body>
  <h1>Hoşgeldiniz!</h1>
  <form method="post" action="<% url 'profilim' %>">
    <% csrf %> The id of a label&lt;label&gt; form-related element is the same document as the <label> element. The first element in the document with an <label id="id"> matching the value of the for attribute is the labeled control for this label element. If it is a labelable element, then the for attribute has no effect. If there are other elements which also match the id value, later in the document, they are not considered.
    <label> Note A <label> element can have both a for attribute and a contained control element, as long as the for attribute points to the contained control element.
    <label for="ad">Ad</label>
    <input type="text" id="ad" name="ad" value="<% ad %>"><br><br>
    <label for="soyad">Soyad</label>
    <input type="text" id="soyad" name="soyad" value="<% soyad %>"><br><br>
    <label for="dogum_tarihi">Doğum Tarihi:</label>
    <input type="date" id="dogum_tarihi" name="dogum_tarihi" value="<% dogum_tarihi %>"><br><br>
    <label for="cinsiyet">Cinsiyet:</label>
    <select id="cinsiyet" name="cinsiyet">
      <option value="erkek" <% if cinsiyet == 'erkek' %> selected <% endif %>>Erkek</option>
      <option value="kadin" <% if cinsiyet == 'kadin' %> selected <% endif %>>Kadın</option>
    </select>
  </form>
</body>
```

IV. SONUÇ

"Hasta Takip ve Yönetim Sistemi" projesi, hastane yönetim süreçlerini dijitalleştirme ve kullanıcı deneyimini geliştirme amacıyla başarıyla tamamlanmıştır. Projede kullanılan yöntemler ve teknolojiler, benzer uygulamaların geliştirilmesinde de kullanılabilir.

V. Psuedo Kod

Fonksiyon: login_view(request)

Amaç: Kullanıcının giriş yapmasını sağlar ve kullanıcı türüne göre doğru arayüze yönlendirir.

Parametreler:

request: İstek nesnesi.

İşlemler:

Eğer istek yöntemi POST ise:

Kullanıcı türünü, kullanıcı adını ve şifreyi al.

Veritabanına bağlan.

Kullanıcı türüne göre uygun SQL sorgusunu oluştur ve kullanıcı adıyla birlikte çalıştır.

Sorgu sonucunu al.

Eğer sonuç varsa ve şifre eşleşiyorsa:

Kullanıcı türüne göre uygun arayüzü göster.

Eğer sonuç yoksa veya şifre eşleşmiyorsa:

Hata mesajı göster ve ana sayfaya yönlendir.

Eğer istek yöntemi GET ise:

Ana sayfayı göster.

Fonksiyon: index(request)

Amaç: Ana sayfayı gösterir.

Parametreler:

request: İstek nesnesi.

İşlemler:

Ana sayfayı göster.

Fonksiyon: Doctors(request)

Amaç: Doktorlar sayfasını gösterir.

Parametreler:

request: İstek nesnesi.

İşlemler:

Doktorlar sayfasını göster.

Fonksiyon: SearchDoctor(request)

Amaç: Verilen soruya göre doktor arar ve sonuçları gösterir.

Parametreler:

request: İstek nesnesi.

İşlemler:

Eğer istek yöntemi POST ise:

Arama sorgusunu al.

Veritabanına bağlan.

Doktorları ad veya soyada göre ara.

Sonuçları al.

Sonuçları göster.

Eğer istek yöntemi GET ise:

Boş arama formunu göster.

Fonksiyon: SearchPatient(request)

Amaç: Verilen soruya göre hastaları arar ve sonuçları gösterir.

Parametreler:

request: İstek nesnesi.

İşlemler:

Eğer istek yöntemi POST ise:

Arama sorgusunu al.

Veritabanına bağlan.

Hastaları ad veya soyada göre ara.

Sonuçları al.
 Sonuçları göster.
 Eğer istek yöntemi GET ise:
 Boş arama formunu göster.
 Fonksiyon: Manager(request)

Amaç: Yönetici panelini yönetir.

Parametreler:

request: İstek nesnesi.

İşlemler:

Eğer istek yöntemi POST ise:

Eğer 'sil_doktor' butonuna tıklanmışsa:

Silinecek doktorun kimliğini al.

Veritabanından doktoru sil.

Başarılı silme mesajını göster.

Eğer 'sil_hasta' butonuna tıklanmışsa:

Silinecek hastanın kimliğini al.

Veritabanından hastayı sil.

Başarılı silme mesajını göster.

Eğer 'DoktorID' formu doluyaşa:

Yeni doktor bilgilerini al.

Veritabanına yeni doktoru ekle.

Eğer 'HastaID' formu doluyaşa:

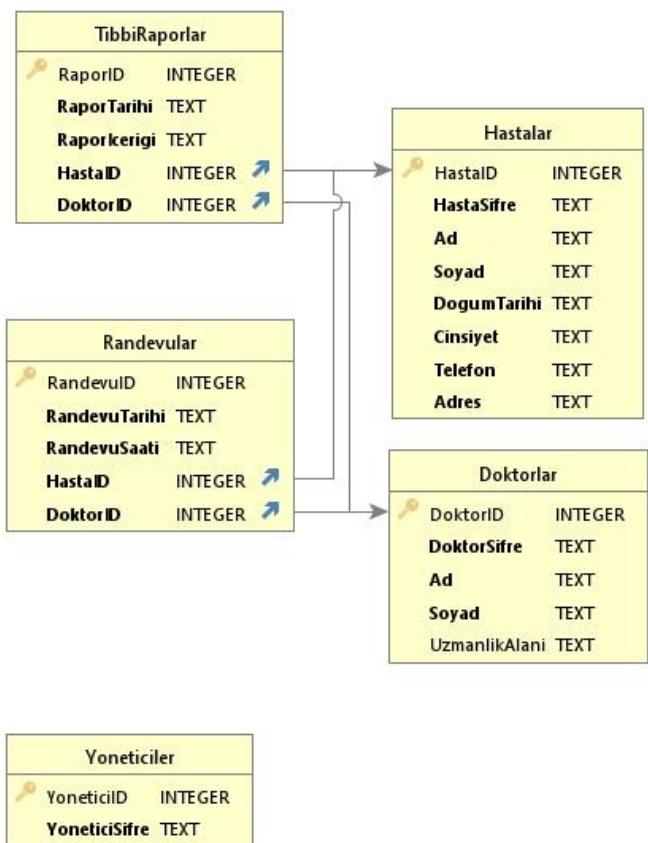
Yeni hasta bilgilerini al.

Veritabanına yeni hastayı ekle.

Veritabanından mevcut doktor ve hasta bilgilerini al.

Doktor ve hasta bilgilerini göster.

VI. ER DİAGRAMI



KAYNAKLAR

- [1] <https://www.dbvis.com>
- [2] <https://www.youtube.com/watch?v=nwZLJiZ7-oQ>
- [3] https://home.bigid.com/demo-governance?utm_campaign=EUROPE+-+Data+Governance&utm_source=google&utm_medium=cpc&utm_term=data%20mapping%20tool&hsa_ver=3&hsa_cam=16774635055&hsa_grp=138179567227&hsa_net=adwords&hsa_ad=591155408608&hsa_kw=data%20mapping%20tool&hsa_mt=b&hsa_acc=9924524058&hsa_src=g&hsa_tgt=kwd-10541561&gad_source=l&gclid=Cj0KCQjwxqayBhDFARIsAANWRnRgHswz2wSUmCjhRsqPBwvifjwmJwHUvsmAIneTBHqKD-jJ4Tyrq5QaAvPhEALw_wcB
- [4] * <https://www.bilisimkonulari.com/proteus-isis-ve-ares-kullanimi-resimli-anlatim.html>
- [5] * <https://www.circuitstoday.com/proteus-software-introduction>
- [6]
- [7] *<https://leetcode.com>

```

<html lang="en">
<body>
  <div class="container">
    <div class="row mt-4">
      <div class="col">
        <table border="1">
          <tr>
            <td>Doktor ID</td>
            <td>Doktor Sifre</td>
            <td>Ad</td>
            <td>Soyad</td>
            <td>Uzmanlık Alanı</td>
          </tr>
          <% for x in range(3)>
            <tr>
              <td>{{x}}</td>
              <td>{{x}}</td>
              <td>{{x}}</td>
              <td>{{x}}</td>
              <td>{{x}}</td>
            </tr>
          <% endfor %>
        </table>
        <form method="post">
          [<x> csrf token </x>]
          <div class="form-group">
            <label for="DoktorID">Doktor ID:</label>
            <input type="text" id="DoktorID" name="DoktorID" class="form-control">
          </div>
          <div class="form-group">
            <label for="DoktorSifre">Doktor Sifre:</label>
            <input type="password" id="DoktorSifre" name="DoktorSifre" class="form-control">
          </div>
          <div class="form-group">
            <label for="Ad">Ad:</label>
            <input type="text" id="Ad" name="Ad" class="form-control">
          </div>
          <div class="form-group">
            <label for="Soyad">Soyad:</label>
            <input type="text" id="Soyad" name="Soyad" class="form-control">
          </div>
          <div class="form-group">
            <label for="UzmanlikAlanı">Uzmanlık Alanı:</label>
            <input type="text" id="UzmanlikAlanı" name="UzmanlikAlanı" class="form-control">
          </div>
          <button type="submit" class="btn btn-primary">Doktor Ekle</button>
        </form>
      </div>
    </div>
  </div>
</body>
</html>
  
```