Departman: Yazılım Mühendisliği
Ders Adı: Yazılım İnşası YZM 201

TO DO LIST UYGULAMASI

PROJE ANALİZ RAPORU

Adlar ve Soyadlar:

1. BEİBARYS GALYMZHAN [2310238543]
2. ALISHO ODZHIEV [2410238528]

Danışman Öğretim Üyesi: Dr. Öğr. Üyesi Ayşe Nur ALTINTAŞ TANKÜL

Tarih: aral**ık** 2025 Karabük

İÇİNDEKİLER

1. PROJE ÖZETİ

To Do List Uygulaması, kullanıcıların günlük görevlerini yönetebileceği bir web tabanlı uygulamadır. Sistem, kullanıcıların kayıt olup giriş yapmasını, kendi görevlerini oluşturmasını, düzenlemesini, silmesini ve tarih atamasını sağlar.

Uygulama client-server mimarisi üzerine kurulmuştur. Backend tarafında Go programlama dili ve Gin framework kullanılmış, veritabanı olarak PostgreSQL tercih edilmiştir. Frontend tarafında ise modern ve responsive bir arayüz oluşturmak için Angular framework kullanılmıştır. Kimlik doğrulama işlemleri JWT (JSON Web Token) ile gerçekleştirilmiştir.

Proje, Yazılım İnşası dersi kapsamında geliştirilmiş olup, gereksinim analizi, tasarım ve uygulama aşamalarını içermektedir.

2. GEREKSİNİMLER

2.1. İşlevsel Gereksinimler

- Kullanıcı kayıt olabilmeli (isim, e-posta, şifre ile).
- Kullanıcı giriş yapabilmeli.
- Giriş yapmış kullanıcı kendi görevlerini listeleyebilmeli.
- Kullanıcı yeni görev oluşturabilmeli (başlık, açıklama, son tarih).
- Kullanıcı mevcut görevleri düzenleyebilmeli.
- Kullanıcı görevleri silebilmeli.
- Sistem, kullanıcıya ait olmayan görevleri göstermemeli (güvenlik).
- Çıkış yapma işlemi bulunmalı.

2.2. İşlevsel Olmayan Gereksinimler

- **Kullanılabilirlik**: Arayüz basit, anlaşılır ve responsive olmalı (mobil uyumlu).
- **Performans**: Görev listeleme ve oluşturma işlemleri 2 saniye içinde tamamlanmalı.
- **Güvenlik**: Şifreler hash'lenmeli, JWT ile yetkilendirme yapılmalı, CORS ayarları doğru olmalı.
- **Taşınabilirlik**: Modern tarayıcılarda (Chrome, Firefox, Safari) sorunsuz çalışmalı.
- **Genişletilebilirlik**: İleride kategori, öncelik gibi özellikler kolayca eklenebilmeli.
- **Bakım Kolaylığı**: Kod clean architecture prensiplerine uygun, modüler olmalı.

3. USE CASE DİYAGRAMI

Aktörler:

- Kullanıcı (User)

Use Case'ler:

1. Kayıt Ol (Register)
2. Giriş Yap (Login)
3. Görev Oluştur (Create Task)
4. Görevleri Listele (List Tasks)
5. Görev Düzenle (Edit Task)
6. Görev Sil (Delete Task)

7. Çıkış Yap (Logout)

Use Case Açıklamaları (örnek birkaç tane):

- **Kayıt Ol** Açıklama: Yeni kullanıcı sisteme kayıt olur. Ön Koşul: Kullanıcı giriş yapmamış olmalı. Akış: Kullanıcı kayıt formunu doldurur → Sistem verileri doğrular → Kullanıcı veritabanına eklenir → Başarılı mesajı gösterilir.
- **Görev Oluştur** Açıklama: Kullanıcı yeni bir görev ekler. Ön Koşul: Kullanıcı giriş yapmış olmalı. Akış: Kullanıcı başlık, açıklama ve tarih girer → Sistem veriyi kaydeder → Görev listeye eklenir.

4. DİNAMİK MODEL (SEQUENCE DİYAGRAMLARI)

Senaryo 1: Kullanıcı Girişi Aktör/Nesneler: User → Frontend → Backend → Database
Akış:

1. User login formunu doldurur
2. Frontend /login endpoint'ine POST yapar
3. Backend şifreyi kontrol eder, JWT üretir
4. Database'den kullanıcı doğrulanır
5. Başarılı yanıt döner → Frontend ana sayfaya yönlendirir

Senaryo 2: Görev Oluşturma Aktör/Nesneler: User → Frontend → Backend → Database
Akış:

1. User yeni görev formunu doldurur
2. Frontend /api/v1/events endpoint'ine POST yapar (JWT header'da)
3. Backend yetkilendirmeyi kontrol eder
4. Yeni görev veritabanına kaydedilir
5. Başarılı yanıt döner → Liste yenilenir

Senaryo 3: Görev Silme Akış: User sil butonuna tıklar → Frontend DELETE isteği yollar → Backend kontrol eder → Database'den siler → Liste güncellenir

5. CLASS DİYAGRAMI

Ana Sınıflar:

- **User** Özellikler: ID, Name, Email, PasswordHash Yöntemler: Register(), Login()

- **Task (Event)** Özellikler: ID, UserID, Title, Description, DueDate, CreatedAt, UpdatedAt Yöntemler: Create(), Update(), Delete()
- **UserRepository** & **TaskRepository** GORM ile veritabanı işlemleri
- **UserService** & **TaskService** İş mantığı (JWT üretimi, yetkilendirme)
- **Handler** (Gin) HTTP isteklerini karşılar

İlişkiler: User 1 --- * Task (bir kullanıcıya birden fazla görev)

6. KULLANICI ARAYÜZÜ (UI) TASARIMI

- **Genel Tasarım**: Minimalist, açık mavi gradyan arka plan, Raleway fontu kullanılmıştır.
- **Giriş ve Kayıt Sayfaları**: Ortalanmış formlar, responsive tasarım.
- **Ana Sayfa (Görev Listesi)**: Görev kartları, "Create Task" butonu, "Refresh List" butonu.
- **Görev Kartı**: Başlık, açıklama, tarih, Edit/Delete butonları.
- **Etkileşim**: Modal pencerelerle görev oluşturma/düzenleme.
- **Erişilebilirlik**: Yüksek kontrast, mobil uyumlu, keyboard navigation destekli.

KAYNAKÇA

1. Gin Web Framework – https://gin-gonic.com
2. GORM – The fantastic ORM library for Golang – https://gorm.io
3. Angular Documentation – https://angular.io
4. PostgreSQL Official Documentation – https://www.postgresql.org
5. JWT – https://jwt.io

EKLER

- Uygulama ekran görüntüleri
- GitHub reposu: https://github.com/ufoinz/yaz_ins_project