GAZİ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ

BM495 BİLGİSAYAR MÜHENDİSLİĞİ PROJESİ I

SOFTWARE DESIGN DESCRIPTION (SDD)

Ayşe ÖZ-21118080055

Yusuf Emre BAYRAKCI-22118080006

Yazılım Projesi Yönetimi Yazılımı

AKADEMİK DANIŞMAN

Prof. Dr. HACER KARACAN

28.11.2025

7031 Kelime

İÇİNDEKİLER

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

KAPSAM

1.1 Tanım

Bu doküman, yazılım geliştirme ekiplerinin proje süreçlerini merkezi bir platform üzerinden planlayabilmesi, görev atayabilmesi, döngü süreçlerini yönetebilmesi, bildirim alabilmesi ve proje performansını analiz edebilmesini sağlayan Yazılım Projesi Yönetim Sistemi (SPMS) yazılımının tasarım detaylarını içermektedir.

Bu doküman SPMS sürüm 1.0 için hazırlanmıştır ve yayın kodu SPMS-SDD-2025-01 olarak belirlenmiştir.

SPMS, web tabanlı bir mimaride çalışan; React (frontend), FastAPI (backend) ve PostgreSQL (veritabanı) teknolojileri üzerine inşa edilmiş bir proje yönetim yazılımıdır. Sistem; çok kullanıcılı iş birliği, rol tabanlı yetkilendirme, gerçek zamanlı bildirim akışı ve kapsamlı raporlama bileşenleri ile yazılım ekiplerinin çalışma verimliliğini artırmayı hedefler.

1.2 Sisteme Genel Bakış

SPMS; yazılım geliştirme ekiplerinin proje yönetimi işlevlerini dijital bir ortamda gerçekleştirmesine olanak tanıyan, modüler bir yapıya sahip web tabanlı bir sistemdir. Sistem aşağıdaki beş ana bileşenden oluşur:

- **Kullanıcı ve Yetkilendirme Modülü**

  Sisteme giriş, kullanıcı kaydı, rol tabanlı erişim kontrolü, oturum yönetimi ve güvenlik politikalarından sorumludur.

- **Proje ve Görev Yönetimi Modülü**

  Projelerin oluşturulması, proje üyelerinin belirlenmesi, görevlerin atanması, görev bağımlılıkları, durum yönetimi, döngü planlama, Kanban görünümü ve görev güncellemelerini kapsar.

- **Bildirim ve Mesajlaşma Modülü**

  Görev güncellemeleri, proje değişiklikleri ve kullanıcı etkileşimleri gibi olaylar için gerçek zamanlı bildirim üretir.

  Kullanıcılar arası mesajlaşmayı ve görev/yorum bildirim akışını yönetir.

- **Raporlama ve Analitik Modülü**

  Proje performansı, ekip verimliliği, döngü ilerlemesi, görev dağılımı, tamamlanma oranları gibi metriklere dayalı analizleri grafiksel ve tablo formatlarında sunar.

  PDF/Excel çıktıları sağlar.

- **Süreç Modeli Seçimi ve Özelleştirme Modülü**

  Kullanıcının proje için bir süreç modeli seçmesine (Scrum, Kanban, Waterfall vb.) ve bu modele uygun yapıların (sprintler, panolar, şablonlar, görev akışları) otomatik oluşturulmasına olanak tanır.

  Kullanıcı süreç modelini daha sonra değiştirebilir veya özelleştirebilir.

  Sistemin hedef kullanıcıları:

- Yazılım geliştiricileri
- Proje yöneticileri
- Ekip liderleri
- Analistler
- Teknik yöneticiler
- Danışman ve dış katılımcılar (sınırlı erişim)

1.2 Dokümana Genel Bakış

Bu doküman SPMS yazılımının mimarisini, bileşen yapılarını, dinamik tasarım planlarını, arayüzlerini ve veri modellerini ayrıntılı biçimde sunar.

Doküman aşağıdaki amaçlara hizmet eder:

- Yazılım geliştiricilerin sistemin iç yapısını anlaması
- Test mühendislerinin modülleri gereksinimlere göre sınaması
- Bakım ekibinin sistemi sürdürülebilir şekilde yönetebilmesi
- Gereksinim → tasarım → uygulama izlenebilirliğinin sağlanması

Bu SDD dokümanı; modüllerin tanımları, algoritmaları, arayüzleri, veri akışları, UML yapıları ve gereksinim eşlemeleri ile kapsamlı bir teknik rehber niteliğindedir.

2. İLGİLİ DOKÜMANLAR

Bu bölüm, SPMS projesinin analiz, planlama ve tasarım aşamalarında oluşturulan ve bu Yazılım Tasarım Dokümanı (SDD) ile doğrudan ilişkili olan dokümanları içermektedir. Söz konusu dokümanlar, gereksinimlerin doğruluğunu, tasarımın tutarlılığını ve yazılım bileşenleri arasındaki izlenebilirliği sağlamaktadır.

Doküman No	**Doküman Başlığı**	**Yayın No**	**Sürüm**	**Tarih**
SPMS-SPMP-2025-01	Yazılım Proje Yönetim Planı (SPMP)	001	v1.0	Ekim 2025
SPMS-SRS-2025-01	Yazılım Gereksinim Şartnamesi (SRS)	002	v1.0	Kasım 2025
SPMS-SDD-2025-01	Yazılım Tasarım Dokümanı (SDD) [Bu belge]	003	v1.0	Kasım 2025
PostgreSQL-REF-2025	PostgreSQL Resmi Dokümantasyonu	-	Güncel	2025
FastAPI-REF-2025	FastAPI Geliştirici Kılavuzu	-	Güncel	2025

React-REF-2025	React Resmi Dokümantasyonu	-	Güncel	2025

Bu dokümanlar; SPMS'nin gereksinim analizinden kullanıcı arayüzü tasarımına, veri tabanı yapısından API tasarımına kadar tüm süreçle ilgili referans oluşturmaktadır.

2.1 Dokümanların Kaynakları

SPMS projesine ait tüm proje belgeleri ve teknik dokümanlar, ekip tarafından yönetilen Git tabanlı sürüm kontrol deposunda saklanmaktadır. Her doküman, yayın numarası ve sürüm bilgisiyle birlikte zaman damgalı revizyon geçmişi içerir.

Bu depoda:

- SRS, SPMP, SDD, STD gibi ana proje dokümanları
- UML diyagramları
- API tasarım dosyaları
- Veri tabanı model çizimleri
- Sprint planları ve backlog listeleri
- Yazılım yapılandırma dosyaları

versiyon kontrollü biçimde tutulur.

Geliştirme sürecinde kullanılan dış kaynak dokümantasyonları (PostgreSQL, FastAPI, React, WebSocket teknolojileri vb.) resmi geliştirici sitelerinden alınmakta olup güncel sürüm uyumluluğu düzenli olarak kontrol edilmektedir.

Bu yapı sayesinde SPMS geliştirme süreci boyunca:

- belgelere tek noktadan erişim,
- dokümanların tutarlı şekilde güncellenmesi,
- versiyonlamanın kaybolmaması,
- gereksinim–tasarım–uygulama ilişkilerinin korunması

sağlanmıştır.

3. YKE ÇAPINDA TASARIM KARARLARI

SPMS, yazılım geliştirme ekiplerinin proje yönetim süreçlerini tek bir platformda yürütebilmesi için tasarlanmış bir web uygulamasıdır. Bu bölümde, sistemin genel davranışını belirleyen, tüm modülleri etkileyen ve yazılım bileşenlerinin seçiminde rol oynayan sistem çapındaki tasarım kararları açıklanmaktadır.

Tasarımlar; kullanıcı deneyimi, performans, güvenlik, veri bütünlüğü, sürdürülebilirlik, esneklik ve erişilebilirlik gibi temel yazılım kalitesi öznitelikleri dikkate alınarak belirlenmiştir.

3.1. Girdi ve Çıktılarla İlgili Tasarım Kararları

Girdi Kararları

SPMS; proje yönetimi süreçlerinde kullanılan verilerin doğru, tutarlı ve güvenli şekilde işlenebilmesi için aşağıdaki girdi türlerini kabul edecek şekilde tasarlanmıştır:

- Kullanıcı giriş bilgileri (e-posta/parola)
- Proje oluşturma ve düzenleme verileri
- Görev tanımları: başlık, açıklama, öncelik, durum, atanan kullanıcı
- Sprint tanımları (Scrum seçildiğinde)
- Kanban panosunda drag-and-drop görev yer değiştirmeleri
- Mesaj ve yorum içerikleri
- Rapor oluşturmak için filtre kriterleri

- Süreç modeli seçimi ve modelin özelleştirme parametreleri (örn. sprint süresi, WIP limitleri)

Tüm girdiler hem istemci hem de sunucu tarafında doğrulanır. Yetkisiz veya eksik veri geldiğinde sistem işlem yapmaz ve kullanıcıya nedenini iletir.

Çıktı Kararları

SPMS'nin üreteceği çıktıların ana amacı, ekiplerin proje durumunu net biçimde izleyebilmesini sağlamaktır:

- Görev ve proje durum bilgileri
- Kullanıcı rolüne göre filtrelenmiş JSON API yanıtları
- Gerçek zamanlı bildirimler (task_updated, new_message, project_changed)
- Kanban pano verileri
- Sprint ilerleme metrikleri
- Zaman çizelgesi (timeline)
- Proje performans raporları (grafikler + PDF/Excel çıktıları)
- Süreç modeline göre üretilen otomatik proje yapısı (ör. Scrum → sprint klasörleri)

Bu çıktılar kullanıcı rolüne göre sınırlandırılır ve yetkisiz erişimler engellenir.

3.2 Girdi/Koşul Bazlı Sistem Davranışları

SPMS; kullanıcı rolü, seçilen süreç modeli ve sistemde meydana gelen olaylara göre dinamik davranış sergiler.

Rol Bazlı Davranış

- Yönetici, tüm projeleri görebilir, kullanıcıları yönetebilir.
- Proje Yöneticisi, kendi projelerinde görev oluşturabilir, döngü tanımlayabilir, süreç modelini değiştirebilir.

- Ekip Üyesi, sadece kendine atanmış görevlerde işlem yapabilir.

Süreç Modeli Bazlı Davranış

 Scrum seçildiğinde:

- döngü oluşturma ekranı açılır.
- Görevler döngülere atanır.
- Sprint süresi dolunca sprint otomatik olarak "tamamlandı" olur.

 Kanban seçildiğinde:

- Görevler sütunlar arasında sürükle-bırak ile ilerletilir.
- WIP limitleri uygulanabilir.

 Waterfall seçildiğinde:

- Görev bağımlılıkları zorunludur.
- Adım geçişleri sırayla yapılır.

Koşullara Göre Sistem Davranışı

- Görev durumu değiştiğinde ilgili tüm proje üyelerine anlık bildirim gönderilir.
- Bir sprint süresi dolmuşsa görev hareketi kısıtlanır.
- Kullanıcı yetkisi yetersizse işlem başlamadan reddedilir.
- Veri bütünlüğünü bozan bir hareket (örn. bağımlılığı bitmeden görevin tamamlanması) engellenir.

3.3 Veri Tabanı Görünürlüğüne İlişkin Tasarım Kararları

SPMS veritabanı doğrudan kullanıcıya açılmaz; tüm erişimler API katmanı aracılığıyla yapılır.

Temel prensipler:

- Kullanıcı yalnızca dahil olduğu projelere ait veri setlerini görebilir.
- Görev, proje, sprint, mesaj ve bildirim tabloları ilişkisel yapıda saklanır.
- Kritik alanlar (parola hash, token, iç operasyon alanları) hiçbir zaman istemciye gönderilmez.
- Soft-delete yapısı kullanılır; veriler doğrudan silinmez, işaretlenir.

- Sorgu performansını artırmak için durum, kullanıcı, proje ve tarih alanlarına indeks uygulanır.
- Raporlama amaçlı bazı sorgular için salt-okunur (read-only) bağlantı kullanılır.

3.4 Emniyet, Güvenlik ve Gizlilik Yaklaşımı

SPMS, proje verilerinin hassasiyeti nedeniyle güçlü güvenlik mekanizmaları ile tasarlanmıştır:

- Kimlik doğrulama JWT ile yapılır.
- Parolalar bcrypt/argon2 ile şifrelenir.
- API çağrılarında rol tabanlı erişim kontrolü uygulanır.
- Mesaj ve bildirim içerikleri yalnızca yetkili kullanıcılara gösterilir.
- Tüm iletişim HTTPS üzerinden sağlanır.
- Yanlış giriş denemeleri sonrası geçici engelleme uygulanır (rate-limiting).
- Veritabanı yedekleme ve geri yükleme mekanizması sistem kararlılığı için aktiftir.
- Projeye dahil olmayan bir kullanıcı hiçbir veri setine erişemez.

3.5 Esneklik, Elde Edilebilirlik ve İdame Ettirilebilirlik Yaklaşımları

SPMS uzun vadeli sürdürülebilirliği hedefleyen modern bir mimariyle tasarlanmıştır:

- Modüller gevşek bağlı (loosely-coupled) yapıdadır.
- Gelecekte yeni süreç modelleri veya modüller eklenebilir.
- Raporlama altyapısı yeni grafik türleri eklemeye açıktır.
- Sistem çoklu sunucu dağıtımına uygundur (horizontal scaling).
- Gerçek zamanlı bildirimler mesaj kuyruğu üzerinden yönetilir (Redis/WebSocket).
- Sunucu çökse bile veriler kaybolmaz (transaction + backup).

- Kod yapısı controller → service → repository şeklinde ayrıştırılmıştır.
- Her modül bağımsız test edilebilir.
- Hata ve olay loglamaları merkezi bir yapıdan izlenir.
- Geliştirme Git akışı ve versiyonlama prensiplerine uygundur.

4. YKE'NİN YAPISAL TASARIMI

Bu bölümde SPMS sistemini oluşturan yazılım konfigürasyon elemanları (YKE) tanımlanmakta, modüllerin yapısı, amaçları, geliştirilebilirlik durumu, donanım kaynağı kullanımı ve birbirleriyle ilişkileri açıklanmaktadır.

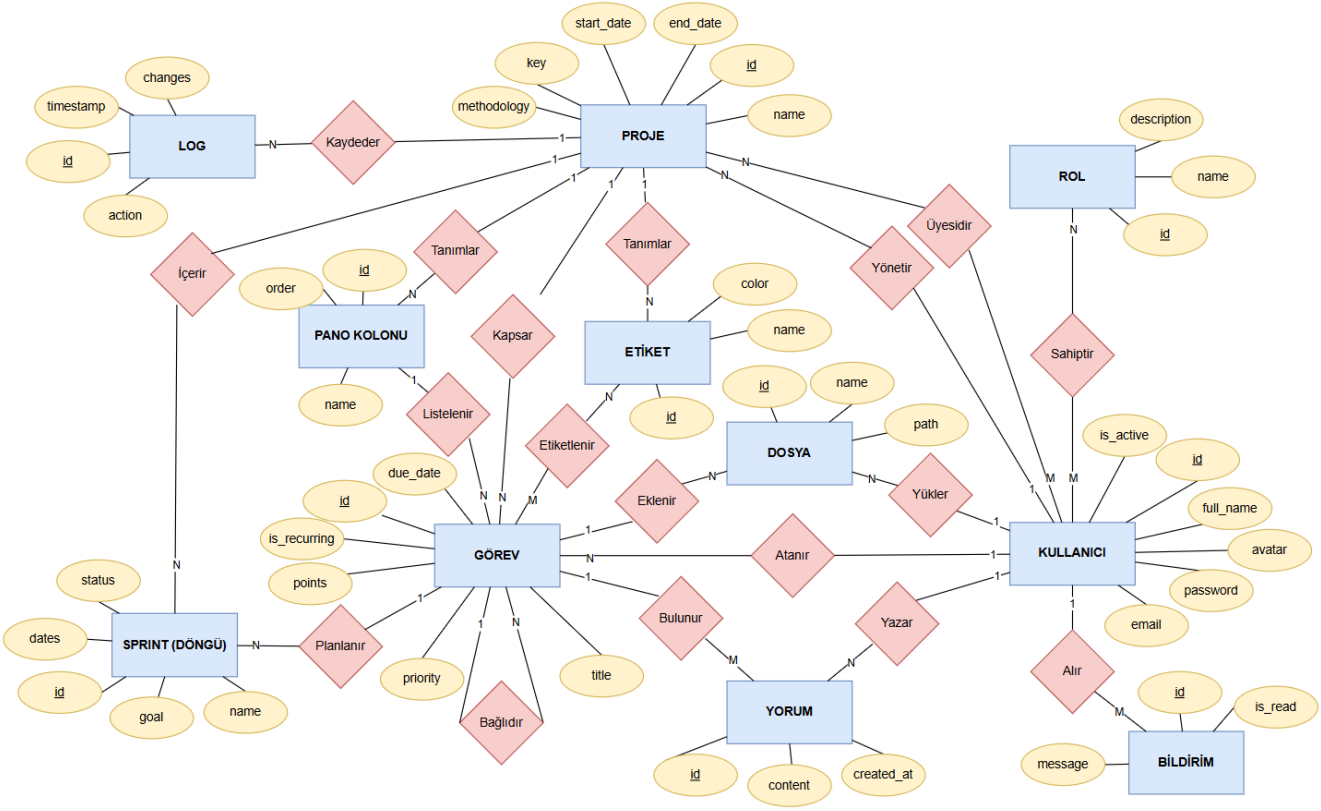
SPMS beş temel yazılım modülünden oluşur:

1. **SPMS-MOD-AUTH** — Kullanıcı ve Yetkilendirme Modülü
2. **SPMS-MOD-TASK** — Proje ve Görev Yönetimi Modülü
3. **SPMS-MOD-NOTIF** — Bildirim ve Mesajlaşma Modülü
4. **SPMS-MOD-REPORT** — Raporlama ve Analitik Modülü
5. **SPMS-MOD-PROCESS** — Süreç Modeli Seçimi ve Özelleştirme Modülü

Her modül, sistemin fonksiyonel gereksinimlerini gerçekleştiren bağımsız fakat birbiriyle etkileşimli yazılım birimlerinden oluşmaktadır.

4.1 YKE Bileşenleri

Bu bölümde SPMS yazılımını oluşturan beş temel modül, kapsamları, sağladıkları işlevler ve sistem içindeki konumlarıyla birlikte tanımlanmaktadır. Modüller birbirinden bağımsız çalışabilecek şekilde tasarlanmış ancak proje yönetimi sürecinin bütünsel akışını sağlamak için birbiriyle uyumlu bir biçimde entegre edilmiştir. Projenin isterlerine uygun olarak tasarlanmış varlık nesne ilişkisi diyagramı şekil 4.1.'de görüldüğü gibidir.



Şekil 4.1. Varlık Nesne İlişkisi (ER) Diyagramı

4.1.1 Kullanıcı ve Yetkilendirme Modülü (SPMS-MOD-AUTH)

Bu modül, SPMS'nin güvenlik, kimlik doğrulama ve erişim kontrolü altyapısını oluşturur. Kullanıcıların sisteme giriş yapması, oturum bilgilerinin tutulması, ekip rolleri arasındaki yetki farklarının uygulanması ve kullanıcı bilgilerinin doğrulanması bu modül tarafından yönetilir. Sistemin geri kalanı bu modül tarafından sağlanan kimlik ve rol doğrulama bilgisine dayanarak çalışır.

Modülün başlıca sorumlulukları:

- Kullanıcı kaydı, giriş ve çıkış işlemleri
- JWT tabanlı oturum yönetimi
- Rol tabanlı erişim kontrolü (Admin / Proje Yöneticisi / Ekip Üyesi)
- Profil bilgisi düzenleme
- Güvenlik ihlallerine karşı temel koruma mekanizmaları

SPMS'de yer alan diğer tüm modüller (görev yönetimi, bildirim, raporlama vb.) bu modülden gelen kullanıcı kimliği ve yetki bilgisine göre davranış belirler. Bu nedenle sistemin çekirdeğini oluşturan kritik bileşenlerden biridir.

4.1.2 Proje ve Görev Yönetimi Modülü (SPMS-MOD-TASK)

SPMS'nin ana fonksiyonlarını sağlayan bu modül, bir projenin oluşturulmasından görevlerin tamamlanmasına kadar geçen tüm süreci yönetir. Proje yöneticileri ekip atayabilir, görev oluşturabilir, görevleri sprintlere veya Kanban panolarına yerleştirebilir, ekip üyeleri kendi görevlerini güncelleyebilir.

Modül; görev bağımlılıklarını, sprint tarihlerini, durum değişimlerini ve proje içindeki tüm hareketleri yönetir.

Bu modülün işlevleri doğal olarak iki ana eksende toplanır:

Proje Yönetimi Fonksiyonları

- Yeni proje oluşturma
- Proje bilgilerini güncelleme
- Ekip üyeleri ekleme / çıkarma
- Süreç modeliyle uyumlu proje yapısı hazırlama

Görev Yönetimi Fonksiyonları

- Görev oluşturma, düzenleme, silme
- Görev durumu (todo–in progress–done) güncelleme
- Sprintlere görev atama (Scrum)
- Kanban pano hareketleri (drag-and-drop)
- Görev bağımlılıklarını kontrol etme
- Göreve yorum ekleme

Bu modül ayrıca görev hareketlerini SPMS-MOD-NOTIF ile paylaşarak bildirim tetiklenmesini sağlar; analitik raporlar için SPMS-MOD-REPORT'a veri kaynağı oluşturur.

4.1.3 Bildirim ve Mesajlaşma Modülü (SPMS-MOD-NOTIF)

Bu modül, SPMS içindeki tüm olayların kullanıcılara gerçek zamanlı iletilmesini sağlayan iletişim altyapısını oluşturur. Görev güncellemeleri, yeni mesajlar, proje değişiklikleri gibi tüm etkileşimler bu modül üzerinden yayınlanır. Ek olarak, proje üyeleri arasında kısa mesajlaşma özelliği sunarak ekip içi iletişimi güçlendirir.

Modülün sağladığı başlıca özellikler:

- WebSocket tabanlı anlık bildirim sistemi
- Görev güncellemesi, yorum, yeni görev, proje güncellemesi gibi tüm olayların iletimi
- Kullanıcılar arası mesajlaşma ekranı
- Bildirim okundu/okunmadı takibi
- Mobil veya web istemci için farklı bildirim seviyeleri

SPMS'nin dinamik yapısını tamamlayan bu modül, kullanıcıların gerçek zamanlı olarak güncel duruma hâkim olmasını sağlar.

4.1.4 Raporlama ve Analitik Modülü (SPMS-MOD-REPORT)

Bu modül, projede gerçekleşen tüm görev hareketlerini, sprint ilerlemelerini ve kullanıcı aktivitelerini analiz ederek raporlara dönüştürür. Hesaplamalar backend tarafında yapılır; kullanıcıya grafiksel veya dosya çıktıları şeklinde sunulur.

Modülün ürettiği rapor türleri:

- Görev tamamlama oranı
- Sprint hız grafiği (Scrum için)
- Kanban akış verimi (cycle time / lead time)
- Kullanıcı aktivite raporları
- Proje seviyesinde genel durum raporları
- PDF ve Excel çıktıları

Bu modül, proje yöneticileri için karar destek aracı işlevi görür ve SPMS-MOD-TASK modülünden aldığı verileri analiz ederek sistemin genel görünümünü çıkarır.

4.1.5 Süreç Modeli Seçimi ve Özelleştirme Modülü (SPMS-MOD-PROCESS)

Bu modül, bir projenin çalışma şeklini belirleyen süreç modelini yönetir. Kullanıcı proje kurulum aşamasında Scrum, Kanban, Waterfall gibi modellerden birini seçebilir; sistem proje yapısını buna göre otomatik olarak kurar.

Modül tarafından sağlanan başlıca yetenekler şunlardır:

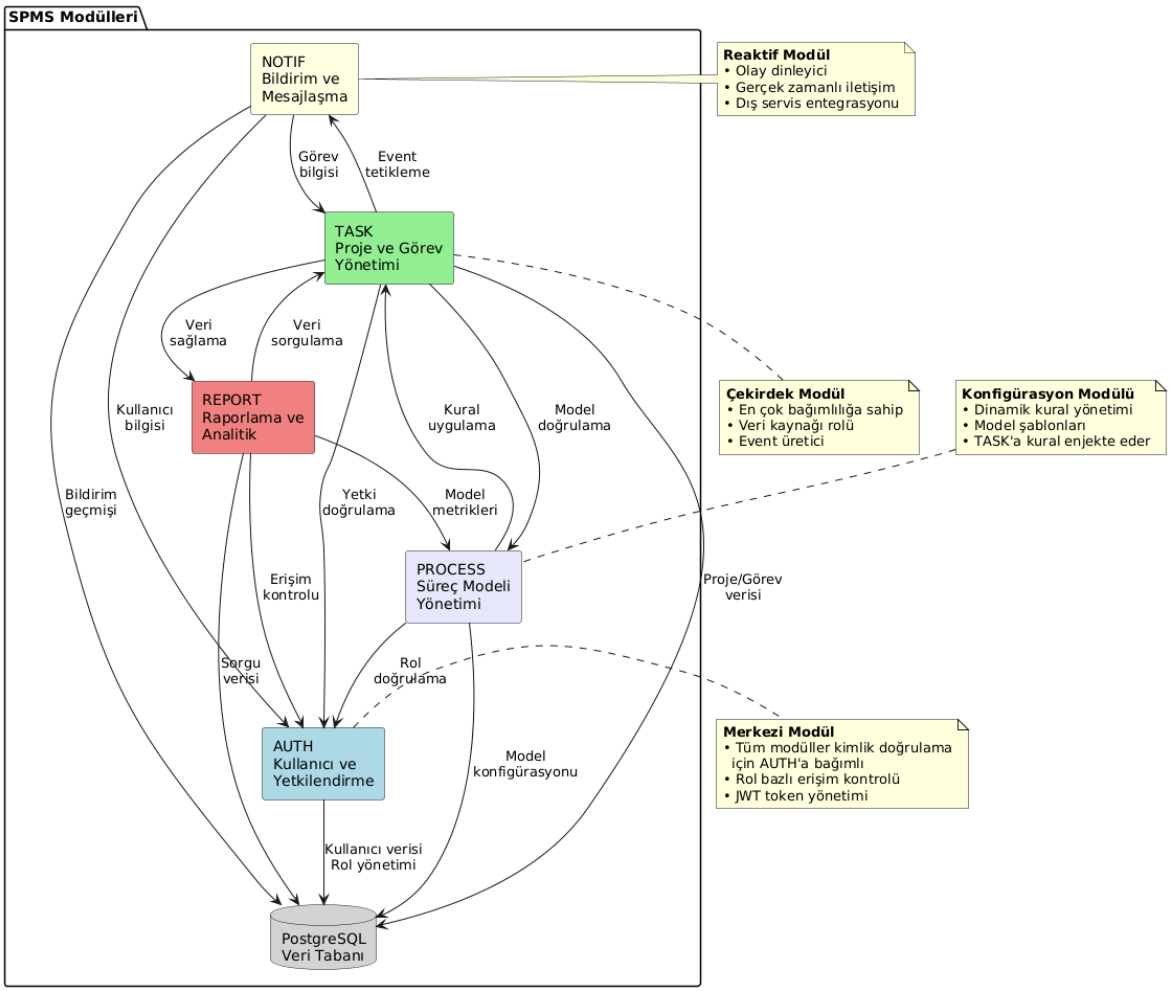
Süreç Modeli Yükleme ve Uygulama

- Projenin başlangıç modeli seçilir
- Model değiştirildiğinde proje verilerinin uyumu kontrol edilir
- Modelin gerektirdiği yapı otomatik oluşturulur

Model Bazlı Özelleştirmeler

- Scrum: sprint süreleri, sprint sayısı, backlog yapısı
- Kanban: WIP limitleri, pano sütun yapısı
- Waterfall: ardışık görev akışları, zorunlu bağımlılıklar

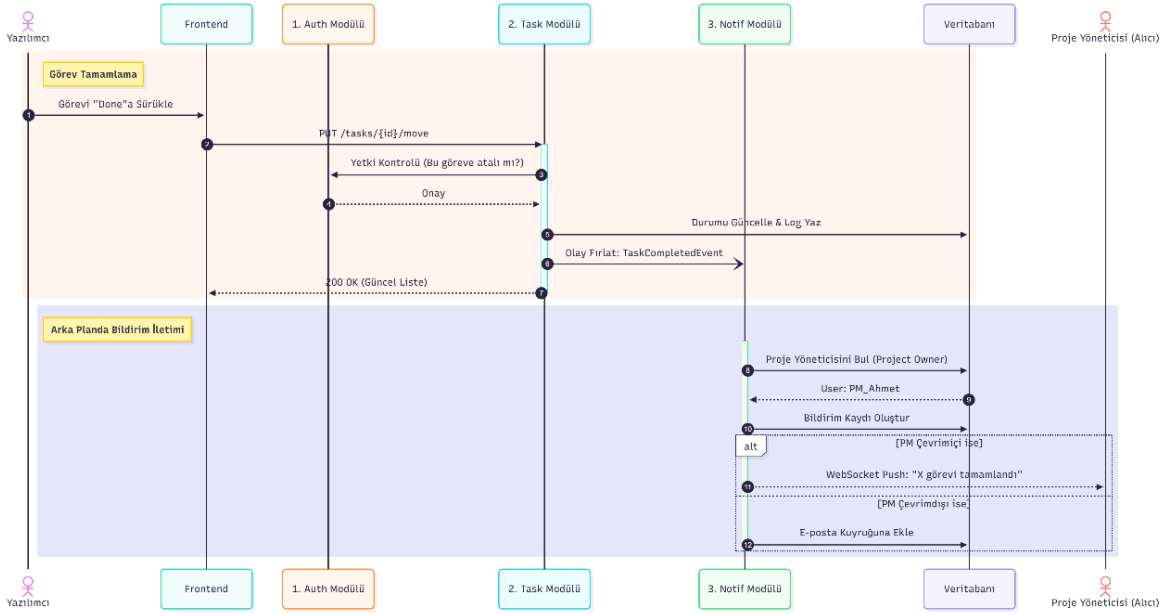
Bu modül SPMS'ye esneklik kazandırır ve farklı çalışma disiplinlerine sahip ekiplerin sistemi kendi yöntemlerine göre kullanabilmesini sağlar.



Şekil 4.2. Modül Bağımlılık Diyagramı

4.2 Genel Çalıştırma (Execution) Kavramı

SPMS, modüler ve olay odaklı bir mimari üzerine inşa edilmiştir. Uygulama çalışma zamanında, kullanıcıdan gelen talebe göre gerekli modülleri devreye alır ve süreç boyunca diğer modüllerle koordineli şekilde etkileşime girer. Sistemin çalışması, kimlik doğrulama ile başlayan, proje ve görev verilerinin yüklenmesiyle devam eden, olay üreten her işlemin bildirim mekanizmasını tetiklediği ve tüm bu süreçlerin geri planda veri tabanı işlemleriyle bütünleşik olarak yürütüldüğü bir akışla yönetilir.



Şekil 4.3. Yazılımcı Ardışıklık Diyagramı

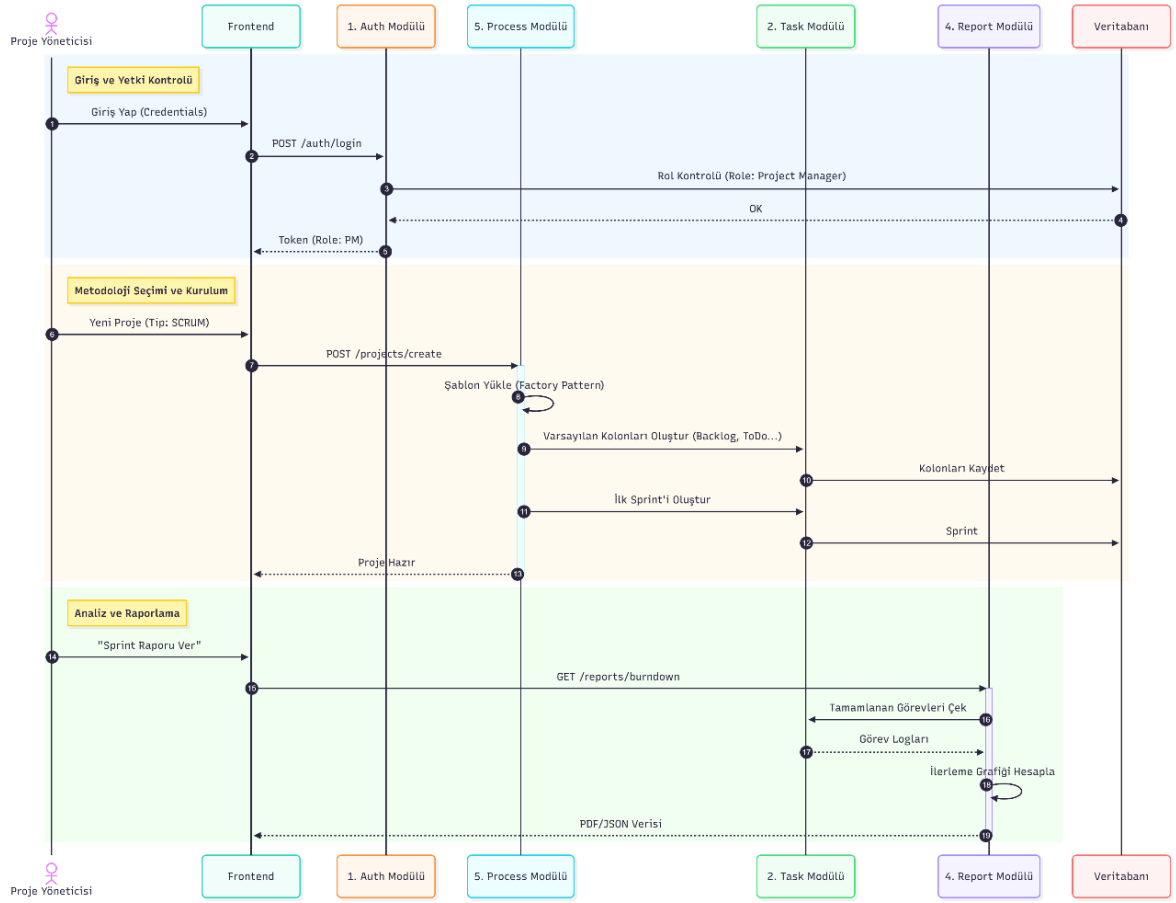
Kullanıcı uygulamaya giriş yaptığında Kimlik ve Yetkilendirme Modülü devreye girer ve kullanıcının rolü doğrulanır. Yetkisi doğrulanan kullanıcıya projeleri sunulur ve seçilen proje için gerekli tüm veriler Proje ve Görev Yönetimi Modülü tarafından yüklenir. Bu aşamada sistem, proje yapısını ve seçilen süreç modelini (Scrum, Kanban, Waterfall) dikkate alarak ilgili iş kurallarını otomatik olarak etkinleştirir. Örneğin Scrum modelinde sprint yapısı ve sprint tarihleri devreye girerken; Kanban modelinde pano sütunları ve görev akış sınırları (WIP limitleri) uygulanır. Bu seçim, görev durumlarının ne şekilde ilerleyeceğini, bağımlılık kurallarının nasıl doğrulanacağını ve kullanıcıya sunulan arayüzün nasıl şekilleneceğini belirler.

Kullanıcı herhangi bir görev üzerinde işlem yaptığında — yeni görev ekleme, görev durumu değiştirme, sprint atama, yorum ekleme gibi — bu işlem önce görev modülü tarafından doğrulanır ve veritabanında güncellenir. Güncelleme işlemi tamamlandığı anda, sistemdeki diğer modüller tetiklenir: Bildirim ve Mesajlaşma Modülü, ilgili proje üyelerine gerçek zamanlı bildirim gönderir; Raporlama Modülü ise değişen veriyi analiz sürecine dahil eder ve ileride kullanılacak istatistiksel çıktıları günceller. Böylece SPMS içinde her görev değişikliği yalnızca veritabanını etkilemekle kalmaz, görev akışı, kullanıcı iletişimi ve raporlama süreçlerinin tamamına etki eder.

Gerçek zamanlı iletişim, sistemin çalışma anındaki davranışının önemli bir kısmını oluşturur. Görev güncellemeleri, mesajlaşmalar, proje ayarlarında yapılan değişiklikler veya

süreç modelinin değiştirilmesi gibi tüm olaylar WebSocket üzerinden ilgili kullanıcılara anında iletilir. Bu sayede kullanıcılar sürekli sayfa yenilemek zorunda kalmadan güncel proje akışını takip edebilir. Sistem, aynı anda hem veri depolayan hem de canlı iletişim sağlayan çift yönlü bir akış yapısına sahiptir.

Son olarak, çalışma süreci boyunca kullanıcılar çeşitli raporlama taleplerinde bulunabilir. Bu durumda raporlama modülü gerekli verileri görev modülünden çekerek analiz eder, grafiksel veri setlerini oluşturur ve istenirse PDF/Excel çıktısı üretir. Bu süreç tamamen arka planda çalışır ve kullanıcının diğer işlemlerini etkilemez.



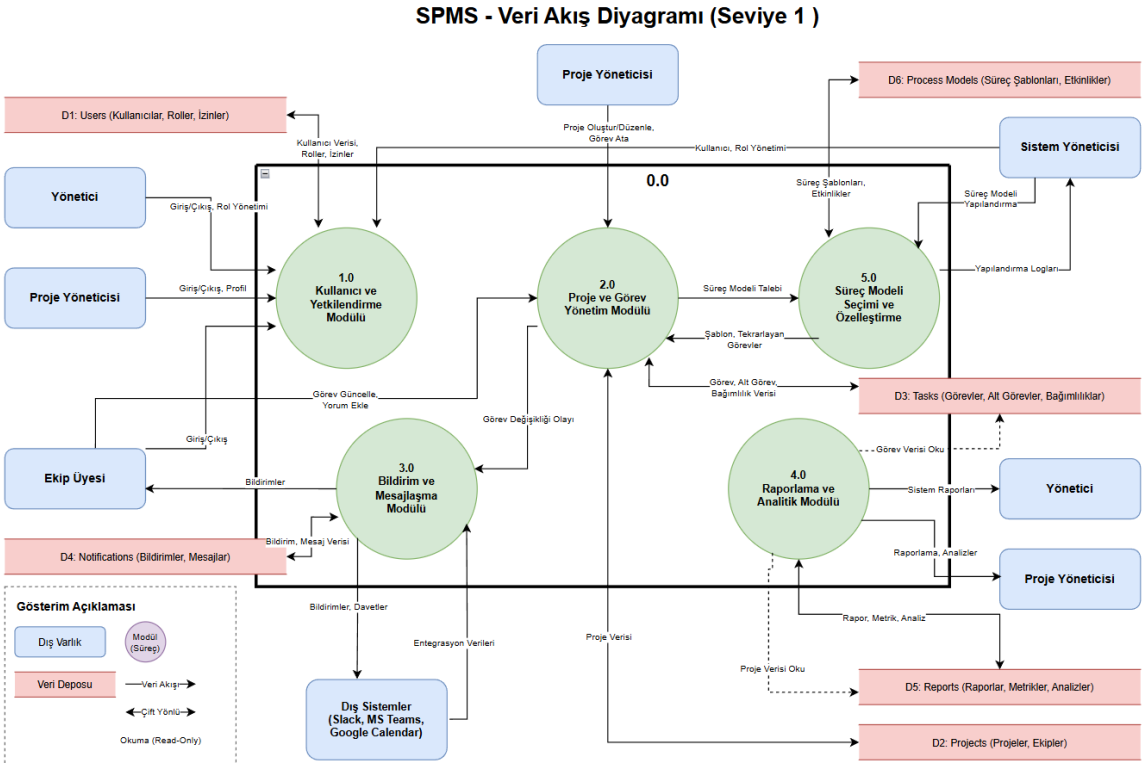
Şekil 4.4. Yönetici Ardışıklık Diyagramı

Genel olarak SPMS'nin çalışma mantığı; kullanıcı rolünün belirlediği sınırlar içinde, süreç modelinin yönlendirdiği kurallarla, modüller arası koordineli etkileşim üzerine kuruludur. Her modül kendi görevinden sorumludur ancak bir modülde gerçekleşen her değişiklik, sistemin diğer bileşenlerine tetikleyici etkiler göndererek SPMS'nin bütünleşik yönetim yapısını oluşturur.

4.3 Arayüz Tasarımı

Bu bölüm, SPMS modüllerinin birbiriyle nasıl veri alışverişi yaptığı, hangi arayüzler üzerinden haberleştiği ve istemci–sunucu yapısında hangi iletişim protokollerinin kullanıldığını açıklar.

SPMS'de arayüzler, modüllerin bağımsız çalışmasını desteklemek için açık, anlaşılır ve genişletilebilir olacak şekilde tasarlanmıştır.



Şekil 4.5. Seviye 1 Veri Akış Diyagramı (VAD)

4.3.1 Kimlik Doğrulama Arayüzü (SPMS-INT-AUTH)

Bu arayüz, kullanıcıların doğrulanması ve oturum yönetimi için kullanılır. Sunucu tarafında FastAPI tabanlı REST endpoint'leri, istemci tarafında React tabanlı form yapısı ile iletişim sağlar.

Başlıca işlevleri:

- Giriş yapma (POST /auth/login)
- JWT üretimi ve doğrulanması

- Kullanıcı bilgisi sorgulama (GET /auth/me)
- Parola sıfırlama taleplerinin işlenmesi

Veri formatı: JSON.

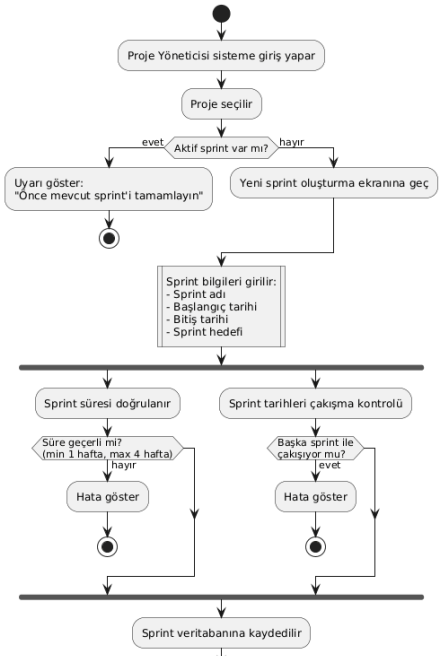
4.3.2 Proje ve Görev Arayüzü (SPMS-INT-TASK)

Görev ve proje yönetimine dair tüm işlemler bu arayüz üzerinden gerçekleştirilir.

Bu arayüz şunları sağlar:

- Proje detaylarını alma
- Yeni görev oluşturma
- Görevi güncelleme / silme
- Görevi sprint veya Kanban sütunları arasında taşıma
- Göreve yorum ekleme
- Görev bağımlılıklarını düzenleme

Proje süreçleri hızlı veri akışı gerektirdiği için tüm bu çağrılar optimize edilmiş REST endpoint'leri ile sağlanır.



Şekil 4.6. Aktivite Diyagramı - Döngü planlama süreci

4.3.3 Bildirim ve Mesajlaşma Arayüzü (SPMS-INT-NOTIF)

Bu arayüz SPMS'nin gerçek zamanlı iletişim altyapısını oluşturur.

- WebSocket bağlantısı üzerinden bildirim akışı
- Kullanıcıya özel mesaj kanalları
- Bildirim veri yapısı (event_type, payload, timestamp)
- Okundu/okunmadı durum güncellemesi

Arayüz, istemci tarafı abonelik sistemiyle çalışır; kullanıcı sadece kendine ait kanalda mesaj alır.

4.3.4 Raporlama Arayüzü (SPMS-INT-REPORT)

Bu arayüz raporlama modülü ile istemci arasında veri alışverişini sağlar.

Sunduğu işlevler:

- Rapor veri setlerini alma
- Grafik veri çıktılarının talep edilmesi
- PDF veya Excel dosyalarının indirilmesi
- Rapor filtrelerinin sunucuya iletilmesi

Veri yapısı hem JSON hem de dosya (binary stream) formatlarını destekler.

4.3.5 Süreç Modeli Arayüzü (SPMS-INT-PROCESS)

Bu arayüz proje süreç modelinin belirlenmesi ve değiştirilmesini yönetir.

İşlevler:

- Kullanılabilir modellerin listesi
- Seçilen modelin sunucuya iletilmesi
- Model değiştirildiğinde proje yapısının yeniden düzenlenmesi
- Model özelleştirme (WIP limitleri, sprint süresi vb.)

Bu arayüz SPMS'nin farklı proje yönetim yöntemlerine uyumlu çalışabilmesini sağlar.

5. YKE DETAYLI PLANI

Bu bölümde SPMS yazılımını oluşturan her modül için, taslakta belirtilen gerekliliklere uygun şekilde; tasarım kararları, modüle özgü kısıtlar, kullanılan diller, işlevsel komutlar, veri giriş-çıkış yapıları ve modülün çalışma sırasındaki mantıksal akışı açıklanmaktadır. Her birim SPMS mimarisindeki bağımsız bir alt yapı taşı olarak değerlendirilmiştir.

5.1. Kullanıcı ve Yetkilendirme Modülü (SPMS-MOD-AUTH)

Bu modül, sistemin giriş kapısıdır ve ER diyagramında yer alan **KULLANICI** ve **ROL** varlıkları (entities) üzerine inşa edilmiştir.

5.1.1. Veri Tabanı Tasarımı ve Tablo Yapıları

Diyagrama uygun olarak aşağıdaki tablolar PostgreSQL veri tabanında oluşturulacaktır:

- **Tablo: ROLES (Roller)**
    - id (PK): Benzersiz rol kimliği (Integer).
    - name: Rol adı (Varchar - Örn: "Admin", "Proje Yöneticisi", "Ekip Üyesi").
    - description: Rolün yetki kapsamını açıklayan metin.
- **Tablo: USERS (Kullanıcılar)**
    - id (PK): Benzersiz kullanıcı kimliği (UUID veya Integer).
    - full_name: Kullanıcının adı ve soyadı.
    - email: Sisteme giriş için kullanılan benzersiz e-posta adresi (Unique Index).
    - password: Güvenli hash algoritması (bcrypt/argon2) ile saklanmış parola.
    - avatar: Profil fotoğrafı dosya yolu veya URL'si.
    - is_active: Kullanıcı hesabının aktif/pasif durumu (Boolean).
    - role_id (FK): Kullanıcının sistemdeki rolünü belirten dış anahtar.

5.1.2. Sınıf ve Nesne Tasarımı

Modül, veri transferi için aşağıdaki DTO (Data Transfer Object) yapılarını kullanacaktır:

- **UserRegisterDTO:** İstemciden gelen kayıt verilerini taşır (email, password, full_name).
- **UserLoginDTO:** Giriş isteğini taşır (email, password).
- **TokenResponseDTO:** Başarılı giriş sonrası dönen veridir (access_token, refresh_token, token_type).
- **UserResponseDTO:** Hassas verilerden (parola) arındırılmış kullanıcı profili.

5.1.3. Algoritmalar ve İş Mantığı

1. **Kayıt (Registration) Algoritması:**
    - Kullanıcıdan gelen email'in sistemde olup olmadığı kontrol edilir.
    - Varsa hata fırlatılır. Yoksa, password alanı "salt" eklenerek hashlenir.
    - Yeni kullanıcı is_active=True ve varsayılan rol (Örn: Ekip Üyesi) ile veri tabanına kaydedilir.
2. **Kimlik Doğrulama (Login) Algoritması:**
    - Kullanıcı email ile veri tabanında aranır.
    - Bulunursa, girilen parola ile veri tabanındaki hash karşılaştırılır.
    - Eşleşme sağlanırsa, kullanıcının id ve role bilgilerini içeren süreli bir **JWT (JSON Web Token)** üretilir ve istemciye gönderilir.
3. **Yetki Kontrolü (Authorization Middleware):**
    - Her API isteğinde, gelen JWT'nin imzası ve süresi kontrol edilir.
    - Token geçerliyse, içindeki role_id okunur ve talep edilen kaynağa erişim yetkisi olup olmadığına bakılır.

5.2. Proje ve Görev Yönetimi Modülü (SPMS-MOD-TASK)

Bu modül, yazılımın çekirdek işlevselliğini oluşturur. Kullanıcıların proje oluşturmasını, görev atamasını, sprint planlamasını ve iş akışlarını takip etmesini sağlar. Veri tabanı tasarımı, SRS belgesinde belirtilen farklı süreç modellerini (Scrum, Kanban) destekleyecek esneklikte yapılandırılmıştır.

5.2.1. Veri Tabanı Tasarımı ve Tablo Yapıları

ER diyagramına uygun olarak ilişkisel yapı şu şekildedir:

- **Tablo: PROJECTS (Projeler)**
    - id (PK): Benzersiz proje kimliği.
    - key: Proje için kısa kod (Örn: "SPMS"). Görev numaralandırmasında kullanılır (SPMS-1, SPMS-2).
    - name: Proje adı.
    - methodology: Seçilen süreç modeli (Enum: 'SCRUM', 'KANBAN', 'WATERFALL').
    - start_date / end_date: Proje zaman aralığı.
- **Tablo: BOARD_COLUMNS (Pano Kolonları)**
    - id (PK): Kolon kimliği.
    - project_id (FK): Hangi projeye ait olduğu.
    - name: Kolon adı (Örn: "Yapılacaklar", "Test", "Tamamlandı").
    - order: Kolonun panodaki sıralaması (Integer).
- **Tablo: SPRINTS (Döngüler)**
    - id (PK): Sprint kimliği.
    - project_id (FK): Bağlı olduğu proje.
    - name: Sprint adı (Örn: "Sprint 1").

- goal: Sprint hedefi.
- start_date / end_date: Sprint süresi.
- status: Durum (Planlanıyor, Aktif, Tamamlandı).

- **Tablo: TASKS (Görevler)**
  - id (PK): Görev kimliği.
  - project_id (FK): Ait olduğu proje.
  - sprint_id (FK, Nullable): Atandığı sprint (Sadece Scrum ise dolu).
  - column_id (FK): Panoda bulunduğu kolon.
  - assignee_id (FK, Nullable): Görevi yapacak kullanıcı (KULLANICI tablosuna referans).
  - title / description: Başlık ve detay.
  - priority: Öncelik (Düşük, Orta, Yüksek, Kritik).
  - points: Story point veya efor değeri.
  - is_recurring: Görevin tekrarlı olup olmadığını belirtir (Boolean).
  - parent_task_id (FK, Nullable): Alt görevler veya bağımlılıklar için kendi kendine referans.

- **Tablo: LOGS (Tarihçe)**
  - id (PK), project_id (FK), action (Yapılan işlem), changes (Eski/Yeni değer JSON), timestamp. Görev ve projelerdeki her değişikliği denetim (audit) amacıyla kaydeder.

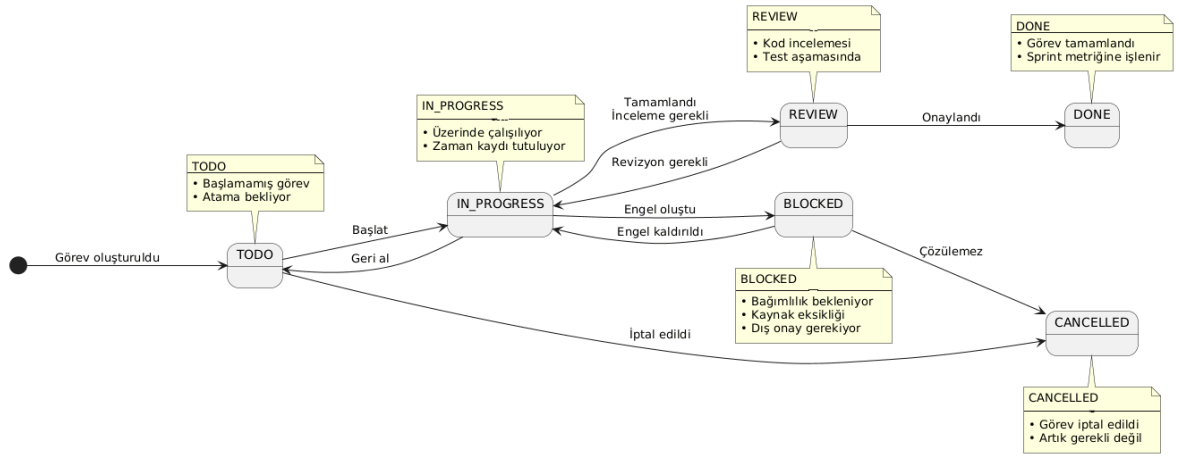
5.2.2. Sınıf ve Nesne Tasarımı

- **ProjectCreateDTO:** name, key, methodology, description.
- **TaskCreateDTO:** title, priority, assignee_id, sprint_id, is_recurring.

- **TaskUpdateDTO:** Görev sürükle-bırak yapıldığında sadece column_id güncelleyen veya detayları değiştiren yapı.
- **KanbanBoardDTO:** Bir projeye ait tüm kolonları ve içindeki görevleri hiyerarşik olarak (Kolon -> Görev Listesi) döndüren yapı.

5.2.3. Algoritmalar ve İş Mantığı

1. **Süreç Modeli Mantığı (Factory Pattern):**
    - Kullanıcı proje oluştururken methodology seçer (Örn: Scrum).
    - Sistem, seçilen metodolojiye göre otomatik olarak varsayılan **Pano Kolonlarını** oluşturur.
        - *Scrum ise:* "Backlog", "In Progress", "Done".
        - *Kanban ise:* "To Do", "In Progress", "Testing", "Done".
    - Scrum seçildiyse Sprint yönetimi özellikleri aktif edilir.



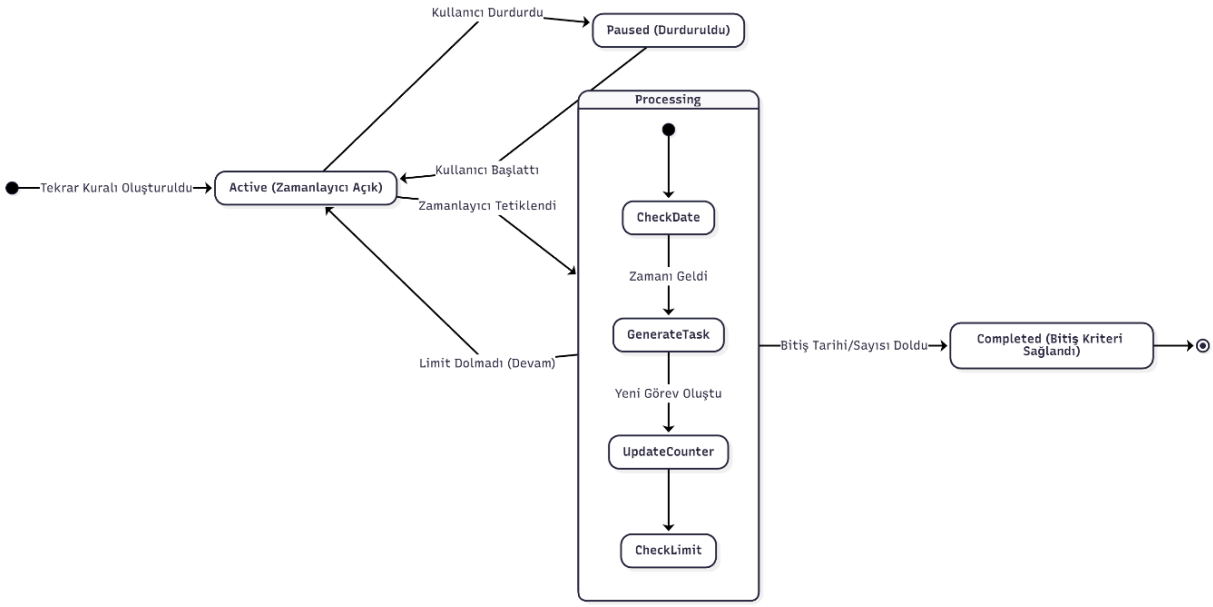
Şekil 5.1. Görev Yaşam Döngüsü Durum Diyagramı

2. **Görev Bağımlılık ve Alt Görev Kontrolü:**
    - Bir görev "Tamamlandı" statüsüne (son kolona) taşınmak istendiğinde, sistem TASKS tablosundaki parent_task_id veya bağımlılık tablosunu kontrol eder.

- Eğer görevin bağımlı olduğu başka bir görev henüz tamamlanmamışsa (SPMS-02.5), sistem işlemi reddeder ve "Önce bağımlı görevi tamamlayın" hatası döner.

3. **Tekrarlayan Görev (Recurring Task) Algoritması:**

   - Arka planda çalışan bir zamanlayıcı (Scheduler), her gün is_recurring=True olan görevleri tarar.
   - due_date veya belirlenen periyot geldiyse, mevcut görevi kopyalayarak yeni bir görev kaydı (INSERT) oluşturur ve bir sonraki tarihi günceller.



Şekil 5.2. Durum Diyagramı: Tekrarlayan görevler

4. **Loglama Mekanizması:**

   - Bir görev üzerinde Update veya Delete işlemi yapıldığında, veri tabanı tetikleyicisi (Trigger) veya uygulama katmanındaki "Interceptor", eski ve yeni veriyi karşılaştırır.
   - Değişiklikleri LOGS tablosuna changes alanına JSON formatında yazar (Örn: {"priority": {"old": "Low", "new": "High"}}).

5.3. Bildirim ve Mesajlaşma Modülü (SPMS-MOD-NOTIF)

Bu modül, SPMS içindeki kullanıcı etkileşimini ve olayların ilgili kişilere iletilmesini sağlar. Sistemdeki asenkron iletişimi yönetir; bir görev güncellendiğinde veya yorum yapıldığında ilgili paydaşları (Proje Yöneticisi, Atanan Kişi vb.) anlık olarak uyarır.

5.3.1. Veri Tabanı Tasarımı ve Tablo Yapıları

ER diyagramına ve iletişim gereksinimlerine uygun tablo yapıları şöyledir:

- **Tablo: COMMENTS (Yorumlar)**
    - id (PK): Yorum kimliği.
    - task_id (FK): Yorumun yapıldığı görev (GÖREV tablosuna referans).
    - user_id (FK): Yorumu yapan kullanıcı (KULLANICI tablosuna referans).
    - content: Yorum metni veya mesaj içeriği.
    - created_at: Mesajın gönderildiği zaman damgasını tutar.
- **Tablo: NOTIFICATIONS (Bildirimler)**
    - id (PK): Bildirim kimliği.
    - user_id (FK): Bildirimin alıcısı olan kullanıcı.
    - message: Bildirim içeriği (Örn: "Ahmet size yeni bir görev atadı.").
    - is_read: Okundu bilgisi (Boolean, Varsayılan: False).
    - type: Bildirim tipi (Enum: 'TASK_ASSIGNED', 'COMMENT_ADDED', 'DEADLINE_APPROACHING').
    - related_entity_id: Bildirimin ilişkili olduğu kaydın ID'si (Örn: Görev ID).

5.3.2. Sınıf ve Nesne Tasarımı

Veri transferi için kullanılan yapılar:

- **CommentCreateDTO:** task_id, content. (Kullanıcı ID token'dan alınır).

- **NotificationResponseDTO:** İstemciye gönderilecek bildirim objesi (id, message, type, is_read, timestamp).
- **WebSocketMessage:** Soket üzerinden iletilen standart mesaj formatı (event, payload).

5.3.3. Algoritmalar ve İş Mantığı

1. **Gerçek Zamanlı İletişim (WebSocket) Mantığı:**
    - Kullanıcı sisteme giriş yaptığında (/auth/login), istemci (React) sunucu ile kalıcı bir **WebSocket** bağlantısı kurar.
    - Her kullanıcı, kendi user_id değeri ile bir soket odasına (room) abone olur.
    - Sunucu, belirli bir kullanıcıya mesaj göndermek istediğinde bu odaya veri paketi (push notification) gönderir.
2. **Olay Güdümlü Bildirim (Event-Driven Notification) Algoritması:**
    - **Adım 1 (Tetikleme):** Görev Modülü'nde bir görev güncellendiğinde (Örn: Atanan kişi değişti), sistem bir TaskAssignedEvent yayınlar.
    - **Adım 2 (Yakalama):** Bildirim Servisi bu olayı dinler (Listener).
    - **Adım 3 (Kayıt):** Servis, yeni atanan kullanıcı için NOTIFICATIONS tablosuna "Okunmadı" durumunda bir kayıt atar.
    - **Adım 4 (İletim):** Eğer kullanıcı o an çevrimiçiyse (WebSocket bağlıysa), bildirim anında ekranına düşer. Çevrimdışı ise bir sonraki girişinde veritabanından okunarak gösterilir.
3. **Yorum ve Mesajlaşma Akışı:**
    - Kullanıcı bir göreve yorum yazdığında, yorum COMMENTS tablosuna kaydedilir.
    - Sistem, o görevi takip eden (watcher) veya göreve atanmış diğer kullanıcılara otomatik bildirim oluşturur.

5.4. Raporlama ve Analitik Modülü (SPMS-MOD-REPORT)

Bu modül, SPMS içindeki operasyonel verileri (görev durumları, tamamlanma süreleri, sprint ilerlemeleri) analiz ederek yöneticilere ve takım üyelerine karar destek mekanizması sunar. Veriler görsel grafikler (Chart.js entegrasyonu) ve indirilebilir dosyalar (PDF/Excel) olarak sunulur.

5.4.1. Veri Kaynakları ve Veri Tabanı Yaklaşımı

Bu modül, ER diyagramında yer alan operasyonel tablolar üzerinden "Okuma" (Read-Only) işlemleri gerçekleştirir. Analiz için kullanılan temel veri kaynakları şunlardır:

- **TASKS (Görevler) & SPRINT:** Anlık durum raporları (Örn: Bekleyen İşler, Sprint İlerlemesi) için kullanılır.
- **LOGS (Tarihçe):** Zaman serisi analizleri için kullanılır. Bir görevin ne zaman "Tamamlandı" durumuna geçtiği veya kimin ne kadar işlem yaptığı bu tablodaki timestamp ve changes alanlarından analiz edilir.
- **PROJECTS (Projeler):** Rapor kapsamını belirlemek için kullanılır.

Not: Performans optimizasyonu amacıyla, çok büyük veri setlerinde raporlar canlı sorgu yerine gece çalışan zamanlanmış görevlerle (Batch Job) önceden hesaplanıp önbelleğe (Cache) alınabilir.

5.4.2. Sınıf ve Nesne Tasarımı

Raporlama verilerini istemciye (Frontend) taşımak için kullanılan yapılar:

- **DashboardStatsDTO:** Genel özet verileri (total_tasks, completed_tasks, overdue_tasks, active_sprints_count).
- **BurndownChartDTO:** Sprint ilerleme grafiği için veri noktaları (date, remaining_points, ideal_burn).
- **UserPerformanceDTO:** Kullanıcı bazlı metrikler (user_id, tasks_completed, average_completion_time).
- **ExportRequestDTO:** İndirilecek raporun filtrelerini içerir (project_id, date_range, format [PDF/Excel]).

5.4.3. Algoritmalar ve İş Mantığı

1. **Burndown Chart (Döngü İlerleme) Hesaplaması (Scrum):**
    - **Girdi:** İlgili sprint_id ve start_date / end_date.
    - **İşlem:**
        - Sprint'in toplam hikaye puanı (story points) hesaplanır.
        - LOGS tablosundan, o sprintteki görevlerin "Done" statüsüne çekildiği tarihler sorgulanır.
        - Sprint başlangıcından bitişine kadar her gün için; (Kalan Puan = Toplam Puan - O güne kadar tamamlanan puan) formülü uygulanır.
    - **Çıktı:** Tarih ve Kalan Puan ikililerinden oluşan bir JSON dizisi.
2. **Kullanıcı Performans Analizi:**
    - Sistem, belirli bir tarih aralığında kullanıcının tamamladığı görevleri TASKS tablosundan çeker.
    - **Zamanında Teslim Oranı:** Görevlerin tamamlanma tarihi (Log tablosundan) ile due_date (Bitiş Tarihi) karşılaştırılır. Gecikme varsa oran düşürülür.
    - Sonuçlar, yönetici ekranında gösterilmek üzere puanlanır.
3. **Rapor Dışa Aktarma (Export) Mantığı:**
    - Kullanıcı "PDF İndir" butonuna bastığında;
    - Backend, ExportRequestDTO içindeki filtrelere göre veriyi veritabanından çeker.
    - Veriler, bir HTML şablonuna (Template) gömülür.
    - HTML şablonu bir PDF dönüştürücü kütüphanesi (Örn: WeasyPrint veya ReportLab) ile binary PDF dosyasına çevrilir.

- Dosya, Content-Type: application/pdf başlığı ile tarayıcıya "Stream" edilir (indirme başlatılır).

5.5. Süreç Modeli Seçimi ve Özelleştirme Modülü (SPMS-MOD-PROCESS)

Bu modül, SPMS'in en ayırt edici özelliği olan "Yöntemden Bağımsız Proje Yönetimi" altyapısını sağlar. Sistem, kullanıcının seçimine göre Scrum, Kanban veya tamamen özel (Custom) bir iş akışı oluşturmasına olanak tanır. Modül, proje oluşturma anında gerekli veri yapılarını (Sprintler, Pano Kolonları) otomatik olarak kurar (Bootstrap).

5.5.1. Veri Tabanı Tasarımı ve Dinamik Yapı

Bu modülün esnekliği, PROJE ve PANO KOLONU tabloları arasındaki ilişkiye dayanır:

- **Tablo: BOARD_COLUMNS (Dinamik İş Akışı):**
  - Geleneksel sistemlerin aksine, görev durumları (Status) sabit bir Enum (To Do, Done vb.) olarak değil, veritabanında dinamik satırlar olarak tutulur.
  - Her BOARD_COLUMN satırı, belirli bir projeye (project_id) aittir.
  - Bu sayede A Projesi "Analiz -> Geliştirme -> Test" akışını kullanırken, B Projesi "Bekleyen -> Tamamlanan" gibi basit bir akışı kullanabilir.
- **Tablo: PROJECTS (Metodoloji Tanımı):**
  - methodology alanı, sistemin arayüz davranışını belirler.
    - **SCRUM:** Arayüzde "Sprint Planlama" butonlarını ve "Backlog" görünümünü aktif eder.
    - **KANBAN:** Sprint özelliklerini gizler, "WIP (Work In Progress) Limitleri"ni aktif eder.
    - **WATERFALL:** Görevler arası "Ardışık Bağımlılık" (Dependency) kurallarını zorunlu kılar.

5.5.2. Sınıf ve Nesne Tasarımı

- **MethodologyTemplateDTO:** Sistemde tanımlı şablonları listeler (Adı, Varsayılan Kolonları, Özellikleri).

- **ColumnDefinitionDTO:** Yeni bir kolon oluşturmak veya sırasını değiştirmek için kullanılır (name, order, wip_limit).
- **WorkflowUpdateDTO:** Projenin akışını toplu güncellemek için kullanılır.

5.5.3. Algoritmalar ve İş Mantığı

1. **Proje Başlatma (Factory Pattern - Şablon Yükleme):**
    - Kullanıcı "Yeni Proje" ekranında bir model seçtiğinde (Örn: Scrum);
    - Sistem **ProjectFactory** sınıfını tetikler.
    - Bu sınıf, PROJECTS tablosuna kaydı ekler ve ardından BOARD_COLUMNS tablosuna varsayılan Scrum setini ("Backlog", "To Do", "In Progress", "Done") ekler (INSERT).
    - Eğer Scrum seçildiyse, otomatik olarak ilk SPRINT kaydını ("Sprint 1") oluşturur.
2. **Özel İş Akışı (Custom SDLC) Algoritması:**
    - Kullanıcı "Pano Ayarları"ndan yeni bir kolon eklediğinde;
    - Sistem, o projenin mevcut kolonlarının order (sıra) değerlerini kontrol eder.
    - Yeni kolon araya eklendiyse, sonraki kolonların sıra numarasını bir artırır (+1 Shift).
    - **Kolon Silme Kontrolü:** Eğer silinmek istenen kolonda aktif görevler (TASKS tablosunda kaydı olan) varsa, sistem kullanıcıya "Bu görevleri hangi kolona taşımak istersiniz?" sorusunu yöneltir ve toplu güncelleme (Bulk Update) yapar.
3. **Metodoloji Değişimi:**
    - Kullanıcı projenin ortasında metodolojiyi (Örn: Scrum'dan Kanban'a) değiştirmek isterse;
    - Sistem, Scrum'a özgü verileri (Sprintler) arşivler ancak silmez.

- Pano kolonlarını korur ve Kanban'a özgü özellikleri (WIP Limit girişleri) aktif hale getirir.

6. GEREKSİNİMLERİN İZLENEBİLİRLİĞİ

Bu bölüm, SPMS Yazılım Tasarım Dokümanında tanımlanan yazılım birimlerinin (modüllerin), Sistem Gereksinim Şartnamesi (SRS) içerisinde tanımlanan YKE gereksinimleriyle olan ilişkisini açıklamaktadır. Gereksinim–tasarım izlenebilirliği; tasarım sürecinin doğruluğunu, modüllerin doğru gereksinimleri karşıladığını ve ileride yapılacak doğrulama/test çalışmalarında her gereksinimin izinin sürülebilmesini sağlamak için hazırlanmıştır. Böylece hem yazılım biriminden gereksinime hem de gereksinimden yazılım birimine doğru çift yönlü izlenebilirlik kurulmuş olur.

6.1 Yazılım Biriminden Gereksinime Doğru İzlenebilirlik

Bu izlenebilirlik türü, Yazılım Tasarım Dokümanında tanımlanan her bir modülün hangi gereksinimleri karşıladığını göstermektedir. SPMS, modüler bir yapıya sahip olduğu için her modül belirli işlevsel gereksinimlere tahsis edilmiştir. Bu doğrultuda:

- **SPMS-MOD-AUTH**, kimlik doğrulama, kayıt/giriş işlemleri, rol bazlı erişim, parola güvenliği ve erişim izinleri gibi SPMS-01.x gereksinimlerinin tamamını karşılar.
- **SPMS-MOD-TASK**, proje oluşturma, görev yönetimi, bağımlılıklar, sprint ve Kanban akışı, tekrarlayan görevler gibi SPMS-02.x gereksinimlerinin karşılanmasından sorumludur.
- **SPMS-MOD-NOTIF**, gerçek zamanlı bildirimler, görev içi yorumlar ve kullanıcı mesajlaşması ile ilgili SPMS-03.x gereksinimlerini karşılar.
- **SPMS-MOD-REPORT**, proje ilerleme raporları, grafiksel sunumlar, PDF/Excel çıktıları ve performans metrikleri ile ilgili SPMS-04.x gereksinimlerini karşılar.
- **SPMS-MOD-PROCESS**, süreç modeli seçimi, özelleştirme, yeni model tanımlama ve proje görünüm yönetimi gibi SPMS-05.x gereksinimlerini karşılar.

Bu eşleştirme sayesinde her yazılım biriminin tasarımının hangi gereksinimlere dayandığı net olarak görülmektedir.

6.2 Gereksinimden Yazılım Birimine Doğru İzlenebilirlik

Bu izlenebilirlik türü, SRS'te tanımlanan her gereksinimin hangi yazılım birimi veya birimleri tarafından karşılandığını belirlemektedir. Böylece hiçbir gereksinimin boşa düşmesi, eksik uygulanması veya yanlış modüle yerleştirilmesi engellenir.

Örneğin:

- **SPMS-01.1 – Kullanıcı kayıt/giriş/çıkış işlemleri** doğrudan **SPMS-MOD-AUTH** modülü tarafından uygulanır.
- **SPMS-02.3 – Görev oluşturma/düzenleme/silme** yalnızca **SPMS-MOD-TASK** modülüne atanmıştır.
- **SPMS-03.1 – Gerçek zamanlı bildirim gönderme** gereksinimi yalnızca **SPMS-MOD-NOTIF** tarafından karşılanır.
- **SPMS-04.3 – PDF/Excel çıktı üretimi** gereksinimi **SPMS-MOD-REPORT** tarafından karşılanır.
- **SPMS-05.1 – Farklı süreç modellerinin desteklenmesi** hem **SPMS-MOD-PROCESS** hem de Task modülü tarafından uygulanır (kural + veri akışı birlikte çalışır).

Bu eşleştirme sayesinde, her gereksinimin uygulanmasından hangi yazılım bileşeninin sorumlu olduğu belirlenmiştir.

6.3 Gereksinim İzlenebilirlik Matrisi

Bu bölümde sunulan **Gereksinim İzlenebilirlik Matrisi**, SPMS sisteminde tanımlanan tüm fonksiyonel gereksinimlerin (SRS: SPMS-01.x – SPMS-05.x) hangi yazılım birimleri tarafından karşılandığını açıkça göstermektedir. Tablo, gereksinimlerin tasarım modülleri ile birebir eşleştirilmesini sağlar ve hem gereksinim → tasarım hem de tasarım → gereksinim yönünde izlenebilirliği garanti eder.

Matriste:

- **ID sütunu**, SRS dokümanında tanımlanan resmi gereksinim kodlarını içerir.
- **YKE Gereksinimi sütunu**, bu gereksinimin özet açıklamasını sunar.
- **İlgili Yazılım Bileşeni(ler)** sütunu, gereksinimin hangi modül veya modüller tarafından karşılandığını belirtir.
- **Açıklama sütunu** ise gereksinimin ilgili modül tarafından nasıl karşılandığını detaylandırır.

Bu yapı sayesinde:

- Her gereksinimin uygulandığı modül açıkça görülebilir,
- Modüller arasında boşa düşmüş, eksik veya fazlalık oluşturan gereksinim kalmadığı doğrulanabilir,
- Test senaryoları gereksinim bazlı olarak çıkarılabilir,
- Modülde yapılacak bir değişikliğin hangi gereksinimleri etkileyeceği kolayca analiz edilebilir,
- Sistem bütünlüğü tasarım aşamasından itibaren korunur.

Gereksinim İzlenebilirlik Matrisi, SPMS tasarımının doğruluğunu garanti eden temel tablodur ve yazılımın geliştirme, test, bakım ve değişiklik yönetimi süreçlerinde referans niteliğindedir.

ID	YKE Gereksinimi	İlgili Yazılım Bileşeni(ler)	Açıklama
SPMS-01.1	Kullanıcıların kayıt, giriş ve çıkış işlemlerinin güvenli yapılabilmesi	SPMS-MOD-AUTH, Backend API	Kayıt ve giriş istekleri Auth modülü tarafından alınır, parola hash'leri doğrulanır ve başarılı durumda JWT üretilir; çıkış işlemi istemci tarafında token'ın temizlenmesi ve oturumun sonlandırılması ile sağlanır.

SPMS-01.2	Her kullanıcıya rol bazlı erişim tanımlanabilmesi	SPMS-MOD-AUTH	Kullanıcı kayıt ve güncelleme akışında rol alanı atanır; tüm backend endpoint'lerinde bu rol Auth modülü tarafından kontrol edilir ve sadece yetkili işlemlere izin verilir.
SPMS-01.3	JWT ile kimlik doğrulama ve parolaların şifreli saklanması	SPMS-MOD-AUTH, Veritabanı Katmanı	Parolalar bcrypt/argon2 ile hash'lenerek Users tablosunda saklanır; girişte doğrulama yapılıp kullanıcı için son kullanma süreli bir JWT üretilir, tüm API erişimleri bu token üzerinden gerçekleştirilir.
SPMS-01.4	Kullanıcı profilinin düzenlenebilmesi, ekip oluşturma/davet	SPMS-MOD-AUTH, SPMS-MOD-TASK	Profil düzenleme ekranı Auth modülünden verileri çeker; ekip oluşturma ve üye daveti ise projeler ve görevler ile ilişkili olduğu için Task modülü üzerinden yürütülür (örneğin proje ekibine üye ekleme).
SPMS-01.5	Projelere erişim izinlerinin proje yöneticisi tarafından yönetilebilmesi	SPMS-MOD-AUTH, SPMS-MOD-TASK	Proje yöneticisi, proje erişim kurallarını Task modülü üzerinden ayarlar; Auth modülü proje erişimi kontrol ederken bu ayarları dikkate alarak "sadece ekibim", "yöneticim" vb. kuralları uygular.
SPMS-01.6	Yetkilendirme yapısının yeni rol tiplerine uygun şekilde ölçeklenebilir olması	SPMS-MOD-AUTH	Rol yapısı veritabanında ayrı bir Roles tablosu ve konfigüre edilebilir yetki haritası ile tanımlanmıştır; yeni rol tipleri eklenebilmesi için Auth modülü rol kontrolünü dinamik yapıdan okur.
SPMS-02.1	Yetkili kullanıcıların yeni proje oluşturması, düzenlemesi, arşivlemesi	SPMS-MOD-TASK, SPMS-MOD-AUTH	Auth modülü kullanıcının rolünü doğrular; Task modülü yeni proje kaydı oluşturur, proje detaylarını günceller ve "arşiv" durumuna alarak proje listesinden filtreler.
SPMS-02.2	Projelere ekip üyelerinin atanabilmesi	SPMS-MOD-TASK	Proje yöneticisi, proje–kullanıcı ilişkisini Task modülü üzerinden yönetir; ilgili kayıtlar ProjectMembers

			tablosunda tutulur ve erişim kontrollerinde kullanılır.
SPMS-02.3	Proje içinde görev oluşturma, düzenleme, silme	SPMS-MOD-TASK	Görev CRUD işlemleri Task modülünde gerçekleştirilir; her görev ilgili proje ile ilişkilendirilir ve gerekli doğrulamalar (yetki, tarih, sorumlu) bu modülde yapılır.
SPMS-02.4	Görevler ve alt görevlerin tanımlanması, öncelik ve durum yönetimi	SPMS-MOD-TASK	Görevler hiyerarşik (üst–alt) yapı destekleyecek şekilde tasarlanmıştır; öncelik (priority) ve durum (todo/in progress/done vb.) alanları görev kartlarının hem backend hem UI tarafında işlenmesini sağlar.
SPMS-02.5	Görevler arası bağımlılıkların tanımlanması	SPMS-MOD-TASK, SPMS-MOD-PROCESS	Task modülü görev bağımlılıklarını kaydeder; Process modülü, seçilen süreç modeline göre bu bağımlılıkların geçerliliğini ve görev akışını (ör. "bitmeden başlayamaz") kontrol eder.
SPMS-02.6	Tekrarlayan görevlerin desteklenmesi	SPMS-MOD-TASK, SPMS-MOD-PROCESS, Veri Katmanı	Tekrarlayan görevler için RepeatingEvents benzeri bir yapı kullanılır; Task modülü tekil görev örneklerini üretir, Process modülü takvim/süreç modeline uyumu kontrol eder.
SPMS-02.7	Tekrarlayan görevlerde "tümü mü yoksa sadece bu örnek mi?" sorusunun sorulması	SPMS-MOD-TASK, Frontend UI	Görev düzenleme akışında, UI kullanıcının seçimini alır; Task modülü gelen karara göre ya yalnızca seçili örneği ya da tüm seriyi günceller.
SPMS-02.8	Tekrarlayan görevler için bitiş kriteri (tarih veya tekrar sayısı)	SPMS-MOD-TASK, Veri Katmanı	Tekrarlayan görev tanımında sonlandırma kriteri alanları tutulur; Task modülü yeni tekrar üretirken bu kriterlere göre görev oluşturmayı durdurur.

SPMS-02.9	Benzer görevler oluştuğunda sistemin uyarı vermesi	SPMS-MOD-TASK	Görev oluşturma sırasında görev başlığı, proje ve tarih aralığına göre benzerlik sorgusu yapılır; benzer kayıt bulunursa kullanıcıya "benzer görev mevcut" uyarısı gösterilir.
SPMS-02.10	Görev geçmişi ve işlem loglarının tutulması	SPMS-MOD-TASK, SPMS-MOD-REPORT	Task modülü görev üzerindeki her değişikliği (durum, sorumlu, tarih) log tablosuna kaydeder; Report modülü bu veriyi daha sonra raporlama için kullanır.
SPMS-02.11	Görev detay sayfasında yorumlaşma ve dosya paylaşımı	SPMS-MOD-TASK, SPMS-MOD-NOTIF	Yorum ve dosya ekleri Task modülünde görev ile ilişkilendirilir; yeni yorum eklendiğinde Notif modülü ilgili kullanıcılara bildirim gönderir.
SPMS-02.12	Görevlerin timeline / Gantt benzeri takvim görünümünde izlenebilmesi	SPMS-MOD-TASK, SPMS-MOD-PROCESS, Frontend UI	Görevlerin başlangıç–bitiş tarihleri Task modülünden alınır; Process modülü süreç modeline göre görünümü şekillendirir ve UI bu veriyi zaman çizelgesi veya Gantt görseline dönüştürür.
SPMS-03.1	Görev ve proje değişiklikleri hakkında gerçek zamanlı bildirim gönderilmesi	SPMS-MOD-NOTIF, SPMS-MOD-TASK	Task modülü değişikliği event olarak üretir; Notif modülü ilgili kullanıcıları belirleyerek WebSocket üzerinden gerçek zamanlı bildirim yayınlar.
SPMS-03.2	Belirli durumlarda ilgili kullanıcılara bildirim gönderilmesi	SPMS-MOD-NOTIF	Bildirim tetikleyicileri (yeni görev, sorumlu değişimi, yorum, durum değişikliği vb.) Notif modülünde kural bazlı olarak tanımlanır ve ilgili kullanıcı listesine bildirilir.
SPMS-03.3	Kullanıcının rolüne göre mesajlaşma yetkisinin sınırlandırılması	SPMS-MOD-NOTIF, SPMS-MOD-AUTH	Mesaj gönderme ekranında Auth modülü kullanıcı rolünü sağlar; Notif modülü yalnızca izin verilen rol kombinasyonlarına (örn. ekip içi, yönetici–üye) mesajlaşma izni verir.

SPMS-03.4	Görev içi mesajlaşma alanı, yorum bırakma	SPMS-MOD-TASK, SPMS-MOD-NOTIF	Görev detayında yazılan yorumlar Task modülünde saklanır; yorum eklendiğinde Notif modülü ilgili proje üyelerine bilgilendirme yapar.
SPMS-03.5	Bildirimlerin hem uygulama içi hem e-posta/push ile iletilmesi	SPMS-MOD-NOTIF, Dış Bildirim Servisleri	Notif modülü, uygulama içi bildirimleri WebSocket ile; e-posta/push bildirimlerini harici servis entegrasyonları üzerinden iletir; kullanıcı tercihleri doğrultusunda kanal seçimi yapılır.
SPMS-03.6	Bildirim tercihlerinin kullanıcı tarafından özelleştirilebilmesi	SPMS-MOD-NOTIF, SPMS-MOD-AUTH	Kullanıcı profil ayarlarında bildirim tercihleri tanımlanır; Notif modülü bildirim üretirken bu tercihleri okuyarak hangi olaylarda, hangi kanaldan bildirim gideceğini belirler.
SPMS-03.7	Mesaj geçmişinin güvenli saklanması ve log tutulması	SPMS-MOD-NOTIF, Veritabanı Katmanı	Gönderilen tüm mesajlar ve sistem içi bildirimler log yapısında saklanır; proje silinse bile erişim izi ve mesaj geçmişi belirlenen süre boyunca korunur.
SPMS-04.1	Proje ilerleme durumlarının grafiksel olarak sunulması	SPMS-MOD-REPORT, SPMS-MOD-TASK, Frontend UI	Report modülü görev ve proje verilerini analiz eder; grafik veri setleri üretir; UI bu setleri grafik bileşenleriyle (ör. Chart.js) kullanıcıya gösterir.
SPMS-04.2	Raporların kullanıcı, görev ve proje bazlı filtrelenebilmesi	SPMS-MOD-REPORT	Rapor API'leri filtre parametreleri (kullanıcı, proje, tarih aralığı vb.) alır; veri sorguları bu parametrelere göre çalıştırılır ve sonuçlar filtreli olarak döndürülür.
SPMS-04.3	Rapor çıktılarının PDF veya Excel olarak dışa aktarılması	SPMS-MOD-REPORT	Raporlama modülü ürettiği istatistikleri PDF/Excel dönüştürme katmanına iletir; çıktı dosyası oluşturularak kullanıcıya indirme linki sunulur.

SPMS-04.4	Kullanıcı performans metriklerinin hesaplanması	SPMS-MOD-REPORT, SPMS-MOD-TASK	Tamamlanan görev sayısı, zamanında teslim oranı gibi metrikler Task modülünden alınan log verisine göre Report modülü tarafından hesaplanır.
SPMS-04.5	Sistem performans verilerinin dashboard üzerinden gösterilmesi	SPMS-MOD-REPORT, Frontend UI	Report modülü metrikleri toplar ve dashboard API'si üzerinden UI'a iletir; UI bu verileri grafik kartlar ve özet kutucuklar halinde gösterir.
SPMS-05.1	Farklı süreç modellerinin (Scrum, Waterfall, Kanban, Iterative) desteklenmesi	SPMS-MOD-PROCESS, SPMS-MOD-TASK	Process modülü süreç modelini seçmek ve kuralları yüklemekle; Task modülü seçilen modele göre görev akışını uygulamakla sorumludur.
SPMS-05.2	Proje oluştururken süreç modelinin seçilmesi ve şablonların otomatik oluşması	SPMS-MOD-PROCESS	Yeni proje oluşturma ekranında kullanıcı model seçer; Process modülü, seçilen modele göre sprint yapısı, Kanban sütunları veya Waterfall aşamaları gibi başlangıç şablonlarını oluşturur.
SPMS-05.3	Süreç şablonlarının özelleştirilebilmesi	SPMS-MOD-PROCESS	Kullanıcı süreçteki adımları, sprint uzunluğunu, toplantı periyotlarını ve diğer parametreleri özelleştirir; Process modülü bu ayarları model konfigürasyonu olarak kaydeder.
SPMS-05.4	İleride yeni süreç modellerinin tanımlanabilir olması	SPMS-MOD-PROCESS	Süreç modeli tanımları konfigürasyon tabanlıdır; yeni model eklemek için yeni bir kural seti ve şablon eklenmesi yeterlidir, modül dinamik olarak bunları yükleyebilir.
SPMS-05.5	Seçilen süreç modeline göre takvim ve tekrarlayan etkinliklerin otomatik planlanması	SPMS-MOD-PROCESS, SPMS-MOD-TASK	Process modülü süreç modeline göre tekrarlayan etkinliklerin (sprint toplantıları vb.) zamanlamasını hesaplar; Task modülü ilgili görevleri veya event kayıtlarını oluşturur.
SPMS-05.6	Projeye modüler görünümler	SPMS-MOD-PROCESS, Frontend UI,	Process modülü projeye hangi görünümlerin (Kanban, Gantt, Liste, Takvim) aktif olacağını tanımlar; UI

	eklenebilmesi (farklı panolar)	SPMS-MOD-TASK	bu görünümleri sunar, Task modülü veri sağlar.
SPMS-05.7	Görevleri Kanban panosu üzerinden görselleştirebilme	SPMS-MOD-TASK, SPMS-MOD-PROCESS, Frontend UI	Task modülü görevlerin durum bilgilerini sağlar; Process modülü Kanban sütun kurallarını belirler; UI görevleri sütunlara yerleştirerek sürükle–bırak işlemlerine izin verir.
SPMS-05.8	Görevleri Gantt şemasıyla görselleştirebilme	SPMS-MOD-TASK, Frontend UI	Görevlerin başlangıç/bitiş tarihleri Task modülünden alınır; UI bu veriyi Gantt şemasına dönüştürerek bağımlılık ilişkileriyle birlikte gösterir.
SPMS-05.9	Görevlerin basit liste veya takvim görünümünde gösterilmesi	SPMS-MOD-TASK, Frontend UI	Task modülü tüm görev verisini sağlar; UI aynı veri setini liste veya takvim görünümünde kullanıcıya sunar; kullanıcı görünüm tipini arayüzden seçebilir.

Tablo 6.3: Gereksinim İzlenebilirlik Matrisi

7. NOTLAR

Bu bölüm, dokümanın daha kolay anlaşılmasını sağlamak için çeşitli açıklamalar, terim tanımları, kullanılan kısaltmalar ve geçmiş bilgiler içermektedir. Aynı zamanda, dokümanda geçen kavramlara dair teknik netlik sunar.

7.1 Genel Açıklamalar

SPMS (Software Project Management System), yazılım geliştirme ekiplerinin proje süreçlerini, görev akışlarını, raporlama mekanizmalarını ve süreç modellerini yönetmek için geliştirilen web tabanlı modüler bir yazılımdır. Bu doküman SPMS'nin yazılım tasarımını teknik bir bakış açısıyla sunmakta olup, sistemin mimarisini, arayüzlerini, işlem akışlarını ve tasarım kararlarını detaylandırır.

Bu bölümde yer alan bilgiler, dokümanın sonraki bölümlerinde kullanılan terimlerin daha net anlaşılmasını amaçlamaktadır.

7.2 Kısaltmalar ve Açıklamaları

Kısaltma	Açılımı	Dokümanda Kullanım Amacı
SDD	Software Design Document	Bu belgenin kendisi (Yazılım Tasarım Dokümanı)
SRS	Software Requirements Specification	SPMS gereksinimlerinin tanımlandığı doküman
SPMS	Software Project Management System	Proje yönetim yazılımının adı
API	Application Programming Interface	Backend servisleri için kullanılan arayüz
UI	User Interface	Kullanıcı arayüzü (React tabanlı)
JWT	JSON Web Token	Kimlik doğrulama için kullanılan token sistemi
ORM	Object Relational Mapping	Veritabanı–nesne eşleme yöntemi (SQLAlchemy)
DB	Database	Sistem veritabanı katmanı
RBAC	Role-Based Access Control	Rol tabanlı erişim kontrolü
CRUD	Create, Read, Update, Delete	Görev/proje işlemlerinin temel dört fonksiyonu
WIP	Work In Progress	Kanban panosunda ilerleyen iş limitleri
REST	Representational State Transfer	API iletişim standardı
TLS/SSL	Transport Layer Security / Secure Sockets Layer	HTTPS şifreleme protokolleri

CI/CD	Continuous Integration / Continuous Deployment	Test ve dağıtım süreçleri
DAG	Directed Acyclic Graph	Görev bağımlılığı doğrulama yapısı
JSON	JavaScript Object Notation	Veri alışveriş formatı

7.3 Terimler ve Tanımlar

Terim	Tanım
Yazılım Birimi (Modül)	SPMS sisteminin işlevsel bir parçasını oluşturan bağımsız geliştirilmiş yazılım bileşeni (AUTH, TASK, NOTIF, REPORT, PROCESS).
Kullanıcı Rolü	Sistemde erişim ve işlem yetkisi belirleyen kategori (Yönetici, Proje Yöneticisi, Ekip Üyesi, Gözlemci).
Görev (Task)	Bir projenin içinde yapılması gereken iş öğesi; açıklama, sorumlu, durum ve öncelik alanlarına sahiptir.
Sprint	Scrum modelinde belirli süreli çalışma dönemi; görevler bu periyotlara dağıtılır.
Kanban Panosu	Görevlerin sütunlar arasında (To Do, In Progress, Done) sürükle–bırak ile ilerlediği yönetim ekranı.
Süreç Modeli	Projenin çalışma yöntemini belirleyen metodoloji (Scrum, Kanban, Waterfall, Iterative).
Bildirim	Görev veya proje üzerinde gerçekleşen bir olayın ilgili kullanıcılara gerçek zamanlı iletimi.
Rapor	Görev durumu, sprint ilerleme, kullanıcı performansı gibi bilgilerin analiz edilerek görselleştirildiği çıktı.

Yineleyen Görev (Repeating Task)	Belirli periyotlarla otomatik tekrar eden görev (ör. haftalık toplantı).
Bağımlılık	Bir görevin başlaması veya bitmesi için başka bir görevin tamamlanmasını gerektiren ilişki.
Token	Kullanıcının kimliğini doğrulamak için kullanılan dijital kimlik anahtarı (JWT).
Arayüz (Interface)	Modüller arasında veri alışverişinin nasıl yapılacağını belirleyen yapı (API, iç arayüzler vb.).
Durum Makinesi (State Machine)	Görevlerin belirli kurallarla durumlar arasında ilerlediği sistem.

İNTİHAL BEYANI

Bu çalışmadaki tüm bilgilerin akademik kurallara ve etik davranışa uygun olarak alındığını ve sunulduğunu ve bu belgede alıntı yaptığımı belirttiğim yerler dışında sunduğum çalışmanın kendi çalışmam olduğunu, Yükseköğretim Kurumları Bilimsel Araştırma Ve Yayın Etiği Yönergesinde belirtilen bilimsel araştırma ve yayın etiği ilkelerine uygun olduğunu beyan ederim.

Numara : 21118080055 22118080006

Ad Soyad : Ayşe Öz Yusuf Emre Bayrakcı

Tarih : 28/11/2025

İmza :