| Doküman Bilgileri | |
|---|---|
| **Hazırlayan** | Yazılım Mimari Müdürlüğü |
| **Versiyon** | v1.0 |
| **Tarih** | 01/07/2025 |
| **Jira Taskı** | YMM-70 |

| Doküman Tarihçesi | | |
|---|---|---|
| **Versiyon** | **Tarih** | **Değişiklik** |
| v1.0 | 01/07/2025 | Staj konusu belirlendi. |

# Yazılım Mimari Müdürlüğü 2025 Yaz Dönemi Staj Programı

- **Takım:** Sedanur Ayhan, Yağmur Başoğlu
- **Staj Süresi:** 20 İş Günü
- **Staj Yeri:** Yazılım Mimari Müdürlüğü

## Stajın Amacı

Bu stajın en temel amacı önce insan yaklaşımına göre, müdürlüğümüz bünyesinde çalışacak birbirini hiç tanımayan stajyerlerin takım olgusunu oluşturmak, takım ruhunu benimsetmek, teknoloji pratiklerini uygulamadan önce gerekli analitik düşünceyi oluşturmak, gerekli akademik araştırmayı yapmak, verilen konu kapsamında mümkün olan tüm teknoloji pratiklerini uygulamak, gerekli olan çıktıları üretmek ve sunmaktır.

Stajyerlerin yazılım geliştirme yaşam döngüsünü mikroservis mimarisi üzerinde deneyimlemeleri, domain bazlı modüler servisler geliştirmeleri, veri yönetimi, güvenlik, loglama gibi konularda pratik yapmaları; Java 21'in modern geliştirme pratiklerini (record, sealed classes, virtual threads vb.) uygulamaları ve en az üç akademik makalenin mikroservis mimarisi perspektifinden incelenerek projeye çıktılarla katkı sunulması hedeflenmektedir.

## Staj Konusu

**PetVerse – Sanal Evcil Hayvan Yönetim Sistemi**
(Mikroservis Mimarisi ve Java 21 Pratikleri ile Geliştirme)

## Öğrenim ve Teknik Gelişim Hedefleri

### Mikroservis Mimari Temelleri

- Mikroservis mimarisi ve DDD’ye giriş
- Servis Keşfi (Eureka), API Gateway kullanımı
- Temel asenkron mesajlaşma (RabbitMQ tercihli)

- Temel Event-Driven Architecture uygulamaları

### Veri Yönetimi

- Her servis için ayrı veri tabanı yaklaşımı
- Saga Pattern'e giriş (basit senaryo)
- Eventual consistency temel kavramları
- Frontend React.js kullanılarak veri görselleştirmesinin yapılması,

### İzleme ve Gözlemlenebilirlik (Observability)

- Sağlık kontrolleri (actuator health)
- Temel loglama (logback + file logging / opsiyonel ELK)
- Basit metrik toplama (Prometheus/Grafana kurulumu)

### Java 21 Özellikleri (*Uygulanabilir Olanlar*)

- Record kullanımı
- Pattern Matching örnekleri
- Sealed class ile sınırlı davranış modelleme
- Virtual Thread ile basit eşzamanlılık örneği

### Güvenlik ve Servisler Arası İletişim

- JWT ile kullanıcı doğrulama
- RestTemplate/WebClient ile temel senkron haberleşme
- Rate limiting (API Gateway seviyesinde)
- SonarQube (Community Edition) kod kalitesinin analizi,

### Deployment ve Versiyonlama

- Docker ile container oluşturma
- Docker Compose ile servis orkestrasyonu
- CI/CD kavramlarına giriş (GitHub Actions pipeline örneği)

## Akademik Katkı

Her stajyer, mikroservis mimarisiyle ilgili **en az 3 akademik makale**yi okuyarak, okuduklarını PetVerse projesine nasıl yansıttığını kısa bir raporla sunacaktır.

### Önerilen Konular

- Mikroservis mimarisi desenleri
- Dağıtık sistemlerde veri tutarlılığı
- Event-driven iletişim yapıları

## Proje Modülleri (Mikroservisler)

| Servisler | | |
|---|---|---|
| **Servis** | **Görevi** | **Teknik Kazanım** |
| PetService | Evcil hayvan bilgileri (CRUD) | Record, JPA, REST API |
| UserService | Kullanıcı kaydı, giriş, JWT güvenlik | Authentication, Authorization |
| ActivityService | Beslenme ve oyun aktiviteleri | Basit event publishing, validation |
| NotificationService | Hatırlatma bildirimleri | RabbitMQ ile basit mesaj gönderimi |
| API Gateway | Servis yönlendirme ve güvenlik | Routing, JWT kontrolü, rate limiting |

*Opsiyonel: ReportService veya MoodService vb. (süre kalırsa eklenebilir)*

## 4 Haftalık İş Takvimi

### 1. Hafta: Teorik Hazırlık ve Proje Kurulumu

- Mikroservis mimarisi makalelerini okuma
- Domain analizi ve servislerin belirlenmesi
- Git repo oluşturulması ve temel branch yapısı kurulması
- Docker Compose ile PostgreSQL ve RabbitMQ kurulumu
- Config Server, API Gateway, PetService ve UserService iskeletlerinin yazılması

*Çıktılar*

- Mimari tasarım dokümanı
- Docker Compose altyapısı
- İlk iki mikroservis iskelet kodları

### 2. Hafta: Çekirdek Servisler ve İletişim Yapıları

- PetService ve UserService'in tamamlanması
- JWT authentication geliştirilmesi
- ActivityService ile RabbitMQ üzerinden bildirim tetikleme
- API Gateway routing işlemleri
- OpenAPI/Swagger dokümantasyonu

*Çıktılar*

- 3 çalışır mikroservis
- JWT login altyapısı

- OpenAPI arayüzü

### 3. Hafta: Gözlemlenebilirlik ve Dayanıklılık

- Health check endpoint'leri
- Logback veya opsiyonel ELK kurulumu
- Prometheus ile metrik toplama ve grafana dashboard
- Circuit breaker (Resilience4j ile) uygulama
- Rate limiting testi (API Gateway)
- React.js ile frontend geliştirilmesi, (*Opsiyonel*)

*Çıktılar*

- Gözlemlenebilirlik kurulum kılavuzu
- Dashboard ekran görüntüleri
- Loglama yapılandırma dosyaları

### 4. Hafta: Sunum ve Dokümantasyon

- Dockerfile'lar ve image üretimi
- Basit GitHub Actions ile CI/CD pipeline kurulumu
- Test senaryoları yazımı (JUnit, Testcontainers örneği)
- SonarQube (Community Edition) kod kalitesi analizi
- Final sunumu ve demo hazırlığı

*Çıktılar*

- Sunum ve tanıtım dokümanı
- Docker imajları ve çalışır demo
- SonarQube (Community Edition) kod kalitesi raporu
- CI/CD örnek pipeline

## Teknoloji Yığını

- **Backend:** Java 21, Spring Boot 3.x
- **Frontend:** React.js
- **Veritabanı:** PostgreSQL
- **Mesajlaşma:** RabbitMQ
- **Gateway:** Spring Cloud Gateway
- **Service Discovery:** Eureka
- **Observability:** Prometheus, Grafana, Logback
- **Güvenlik:** Spring Security + JWT
- **Test:** JUnit 5, TestContainers
- **Deployment:** Docker, Docker Compose
- **CI/CD:** GitHub Actions
- **Kod Kalitesi:** SonarQube (Community Edition)

## Değerlendirme Kriterleri

- **Teknik Yetkinlik (%40)**: Kod kalitesi, desen kullanımı, testler
- **Mimari Anlayış (%30)**: Tasarım diyagramları, sistem yapısı
- **Akademik Katkı (%20)**: Makale özeti, uygulanabilirlik
- **Sunum ve Takım Çalışması (%10)**: Git kullanımı, iletişim ve demo

## Başarı Göstergeleri

- En az 3 mikroservisin çalışır hale getirilmesi
- JWT authentication ve API Gateway yapısının entegrasyonu
- RabbitMQ ile en az bir event-driven haberleşme
- Prometheus ile gözlemlenebilirlik kurulması
- Akademik okuma raporlarının sunulması
- CI/CD pipeline ile otomatik test ve dağıtım yapılması