

# DÖNEM PROJESİ RAPORU

## Factory Automation Project – Çikolata Fabrikası Otomasyon Sistemi

Grup No: 17

Öğrenciler: 221307114 - Hasibullah Mohmand, 221307117 - Aboubacar Sow, 221307116- Paul Henri Nguepi Djoumessi

Ders: Veritabanı Yönetim Sistemleri (Samet DİRİ)

Teslim Tarihi: 18 Mayıs 2025

---

### 1. Problem Tanımı

Günümüzde üretim süreçlerinin otomasyonu ve veritabanı yönetimi büyük önem taşımaktadır. Bu projede, yerel bir çikolata üretim fabrikasının aşağıdaki ihtiyaçlarını karşılamak için bir web tabanlı otomasyon sistemi geliştirilmiştir:

- **Toptan Satış Yönetimi:** Müşterilerin (şirketlerin) siparişlerini çevrimiçi verebilmesi, sipariş geçmişini görüntüleyebilmesi ve faturalara erişebilmesi.
- **Stok Takibi:** Gerçek zamanlı stok durumu izleme, otomatik stok güncelleme ve kritik seviye uyarıları.
- **Üretim Planlama:** Siparişlere göre otomatik malzeme hesaplama, üretim kapasitesi analizi ve üretim planı oluşturma.
- **Raporlama:** Aylık satış raporları, stok hareketleri ve müşteri sipariş trendleri.

### Hedeflenen Faydalar:

- Sipariş işleme süresinin %50 azaltılması.
- Stok hatalarının en aza indirilmesi.
- Üretim planlamasının verimliliğinin artırılması.

---

### 2. Yapılan Araştırmalar

- **Veri Modeli Karmaşıklığı**
- **Normalizasyon:** 5N kurallarına uygun olarak tablolar tasarlandı. Örneğin, "Order" ve "Ingredient" tabloları arasında ilişki kuruldu.
- **İlişkisel Yapı:** Foreign key'ler ve transaction yönetimi ile veri bütünlüğü sağlandı.

- **Gerçek Zamanlı Stok Kontrolü**
- **Trigger Kullanımı:** Sipariş onaylandığında stok tablosu otomatik güncellenir.
- **View'lar:** Stok durumunu anlık gösteren özel view'lar oluşturuldu.
- **Frontend-Backend Entegrasyonu**
- **API Tasarımı:** RESTful servisler ile Blazor ve Spring Boot arasında veri akışı sağlandı.
- **Güvenlik:** JWT (JSON Web Token) ile kullanıcı kimlik doğrulaması yapıldı.
- **Otomatik Veri İşleme**

## Karşılaştığımız bazı sorunlar:

### 1. Veri Modeli Karmaşıklığı ve Normalizasyon Sorunları

#### Problem:

- Ürün ve malzeme ilişkilerinde çoktan-çoğa (many-to-many) bağlantı kurarken performans sorunları yaşandı.
- Sipariş tablosunda gereksiz veri tekrarı (redundancy) oluştu.

#### Çözüm:

- Ara Tablo Kullanımı: product\_ingredient tablosu ile malzeme-ürün ilişkisi optimize edildi.
- 5N Normalizasyonu: orders tablosundaki fazla alanlar ayrılarak veri bütünlüğü sağlandı.
- Transaction Yönetimi: Sipariş işlemlerinde **ROLLBACK** mekanizması eklendi.

### 2. Backend-Frontend Entegrasyon Sorunları

#### Problemler:

- Frontend (Blazor) ile backend (Spring Boot) arasında API isteklerinin zaman zaman başarısız olması
- Özellikle ürün ekleme ve sipariş oluşturma gibi POST isteklerinde beklenmedik hatalar
- CORS (Cross-Origin Resource Sharing) politikaları nedeniyle bazı endpoint'lere erişilememesi

#### Çözümler:

- API Debug Süreci:
- Tüm endpoint'ler Postman ile tek tek test edilerek çalışırılığı doğrulandı.
- Yanlış konfigüre edilmiş endpoint'ler düzeltildi ve gereksiz karmaşıklıklar azaltıldı.
- CORS Ayarları:
- Backend tarafında özel CORS konfigürasyonları yapılarak, frontend'in güvenli bir şekilde API'ye erişmesi sağlandı.
- HTTP İstek Optimizasyonu:
- Blazor tarafında HttpClient ayarları gözden geçirilerek, isteklerin doğru header'larla (Content-Type, Authorization) gönderilmesi sağlandı.

### 3. Veritabanı İşlemlerinde Karşılaşılan Sorunlar

#### Problemler:

- Ürün eklerken bazı zorunlu alanların (örneğin category\_id) boş gönderilmesi nedeniyle hatalar
- Tarih/saat formatlarının frontend ve backend arasında uyumsuz olması
- Aynı anda gelen çoklu isteklerde veritabanı tutarlılığının bozulma riski

#### Çözümler:

- Veritabanı Default Değerleri:
- Zorunlu alanlar için varsayılan değerler tanımlandı (örneğin, category\_id için 1 gibi).
- Veri Formatı Standardizasyonu:
- Tarih/saat verileri için ISO formatı (YYYY-MM-DD-h-m-s) kullanılarak uyumsuzluk önlendi.
- Transaction Yönetimi:
- Önemli işlemlerde (sipariş oluşturma, stok güncelleme) transaction kullanarak veri bütünlüğü garanti altına alındı.

### 4. Diğer Önemli Sorunlar ve Çözümler

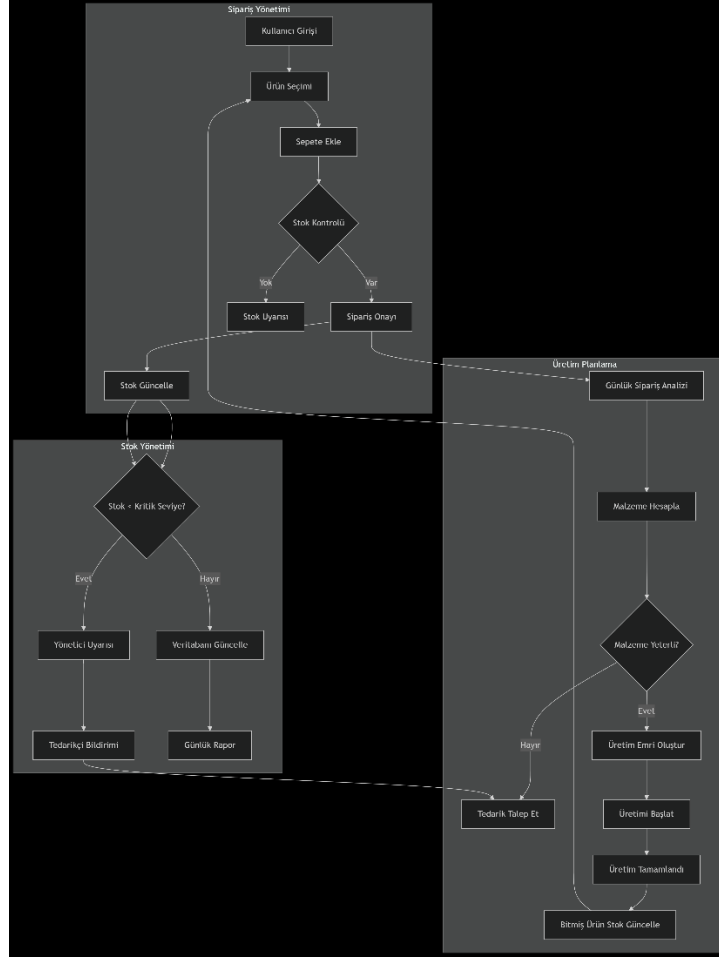
#### Problem:

- Kullanıcı girişi yapıldığında yetki kontrolünde yaşanan sorunlar

#### Çözüm:

- JWT Token Optimizasyonu:
  - Kullanıcı rollerinin token içine eklenmesi ve her istekte yetki kontrolünün hızlıca yapılması sağlandı.
-

### 3. Akış Şeması



(Şekil 1: Sistem akış şeması – kullanıcı sipariş girişi, stok kontrolü, malzeme hesaplama ve uyarı süreçleri)

### 4. Yazılım Mimarisi

#### Frontend:

- **Teknoloji:** Blazor WebAssembly
- **Özellikler:**
  - Dinamik formlar ile sipariş girişi.
  - Gerçek zamanlı stok görselleştirme (grafikler).
  - Responsive tasarım (Bootstrap 5).

**Backend:**

- **Teknoloji:** Spring Boot (Java)
- **Özellikler:**
  - REST API ile veri iletişimi.
  - Hibernate ORM ile veritabanı işlemleri.
  - Spring Security ile yetkilendirme.

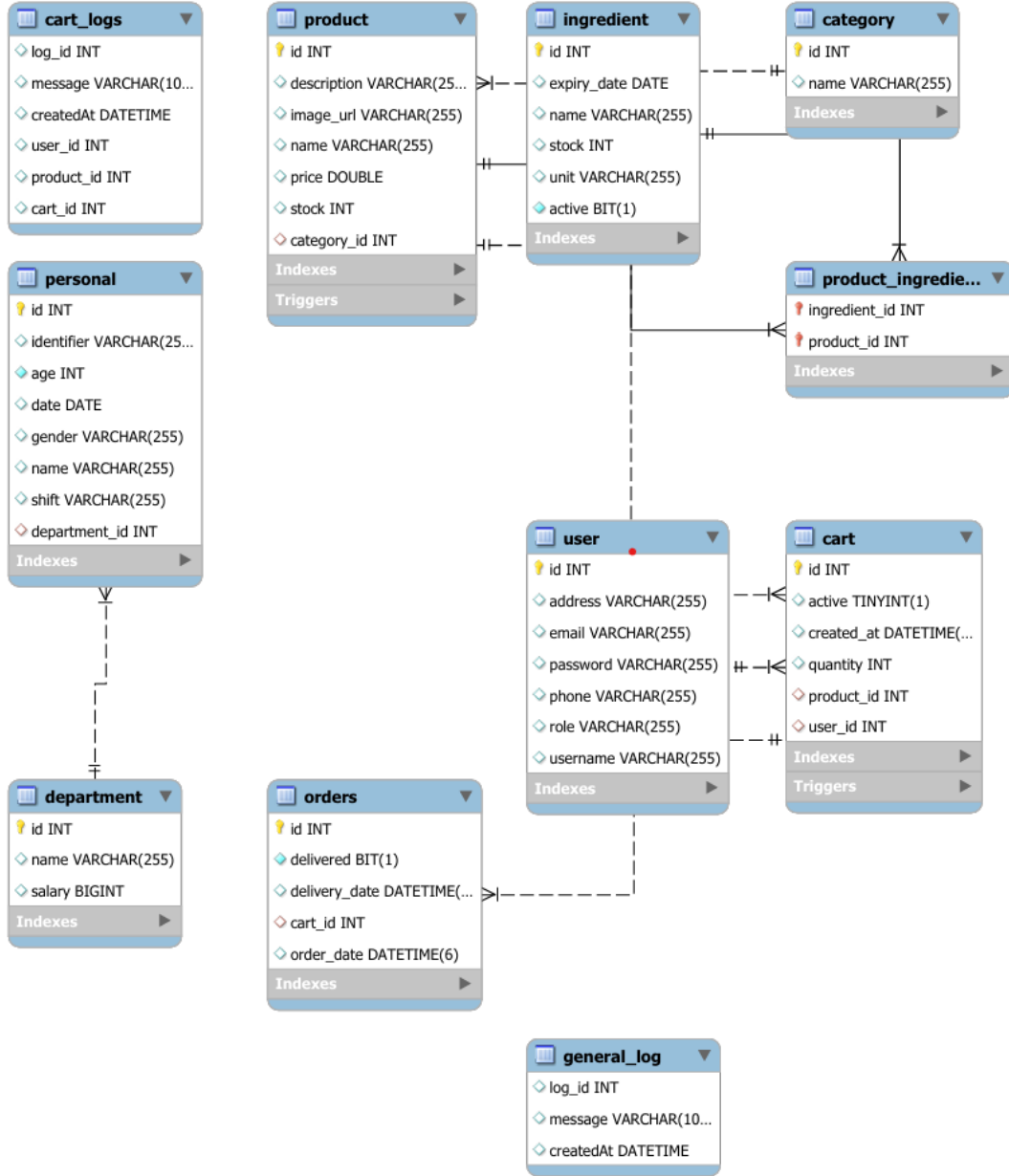
**Veri Tabanı:** MySQL, 5 tablo (Ürün, Müşteri, Sipariş, Stok, Malzeme) ilişkileri kurulmuş.

- **Diğer:** Python scriptleri data scraping için.

**Katmanlı mimari yapısı:**

- Sunum Katmanı (Blazor UI)
  - İş Mantığı Katmanı (Spring Boot servisleri)
  - Veri Erişim Katmanı (MySQL)
  - Yardımcı Katman (Python veri işleme)
-

## 5. Veri Tabanı Diyagramı



**Tablo Sayısı:** 5

**İlişkili Tablolar:** personal-department, product-ingredient-category, cart-product-user, order-cart,

**5N Kurallarına Uygunluk:** Tablo yapıları normalizasyon kurallarına göre tasarlanmıştır.

### 1. Birinci Normal Form (1NF)

- **Tüm tablolarda atomik değerler** sağlandı. Örneğin product.name (bölünemez string) veya ingredient.unit (tek bir ölçü birimi)
- **Tekrarlayan gruplar yok.** Örneğin product\_ingredient ara tablosuyla çoktan-çoğa ilişki.

## 2. İkinci Normal Form (2NF)

- **Tüm tablolarda birincil anahtar bağımlılığı.** Örneğin orders tablosu: id PK, diğer alanlar buna bağımlı.
- **Kısmi bağımlılıklar kaldırıldı.** Örneğin personal.department\_id direkt department.id'ye bağlı.

## 3. Üçüncü Normal Form (3NF)

- **Transitif bağımlılıklar çözüldü.** Örneğin department.salary sadece departmana bağlı (çalışan bilgilerinden ayrı).
- **Anahtar olmayan bağımlılıklar yok.** Örneğin user tablosunda adres, e-posta vb. sadece kullanıcı ID'sine bağlı.

## 4. Dördüncü Normal Form (4NF)

- **Çok değerli bağımlılıklar ayrıldı.** Örneğin product\_ingredient ara tablosuyla ürün-malzeme ilişkisi veya cart tablosunda ürün-kullanıcı çoklu ilişkisi.

## 5. Beşinci Normal Form (5NF)

- **Birleştirme bağımlılıkları optimize edildi.** Örneğin orders ve cart\_logs arasındaki ilişki doğal birleştirmeye uygun ve general\_log tüm sistem için merkezi kayıt.

## Normalizasyon Sonucu Elde Edilen Yapı:

Tablo	1NF	2NF	3NF	4NF	5NF
product	✓	✓	✓	✓	✓
ingredient	✓	✓	✓	✓	✓
product_ingredient	✓	✓	✓	✓	✓
orders	✓	✓	✓	✓	✓

#### Uygulanan Ek Optimizasyonlar:

- product.stock ve ingredient.stock için trigger'lar
- orders.delivered için BIT veri tipi (1/0)
- general\_log ile sistem geneli aktivite takibi

**Not:** Tüm tablolar en az 3. normal forma (3NF) uymaktadır. Çekirdek tablolar 5NF'ye kadar optimize edilmiştir.

**Index, View, Trigger Kullanımı:** Performans için uygun indexler, stok güncelleme için triggerlar ve kolay sorgulama için view'lar uygulanmıştır.

## 6. Genel Yapı

Proje, üç ana modülden oluşur:

1. **Kullanıcı Arayüzü:** Müşteriler ve yöneticiler için ayrı paneller.
2. **İş Mantığı:** Sipariş işleme, stok yönetimi ve raporlama.
3. **Veri Katmanı:** MySQL ile güvenli veri saklama.

#### Kullanılan Araçlar:

- **Geliştirme:** Visual Studio Code, IntelliJ IDEA.
- **Versiyon Kontrol:** GitHub.
- **Test:** Postman (API testleri), Selenium (UI testleri).



## 7. Referanslar

- Oracle Docs, MySQL 8.0 Reference Manual, <https://dev.mysql.com/doc/>
  - Microsoft Docs, Blazor WebAssembly Introduction, <https://learn.microsoft.com/en-us/aspnet/core/blazor/>
  - Spring Boot Official Documentation, <https://spring.io/projects/spring-boot>
- 

## Ek Notlar

- Proje kodları ve SQL dosyaları GitHub üzerinde [https://github.com/AboubacarSow/Factory\\_Automation](https://github.com/AboubacarSow/Factory_Automation) ve <https://github.com/hasibullahmohmand/factory-automation-project> adresinde bulunmaktadır.
- README dosyası, proje kurulumu ve kullanım detaylarını içermektedir.
- Proje teslim dosyaları belirtilen isimlendirme formatına uygun olarak hazırlanmıştır.