

# Kargo Takip Otomasyonu

Hüsnü Tolga Koçer  
Bilişim Sistemleri Mühendisliği  
Veritabanı Yönetim Sistemleri  
221307094  
2113070946@kocaeli.edu.tr

Tolga Boz  
Bilişim Sistemleri Mühendisliği  
Veritabanı Yönetim Sistemleri  
211307036  
211307036@kocaeli.edu.tr

**Abstract**—Bu proje, bir kargo firmasının hem müşteri tarafı hem de yönetici tarafı operasyonlarını etkin şekilde gerçekleştirebilmesi için geliştirilmiş tam özellikli bir web tabanlı otomasyon sistemidir. Sistem üzerinden müşteriler yeni kargo gönderileri oluşturabilir, var olan gönderilerin durumlarını gözlemleyebilir. Aynı zamanda yönetici (admin) rolüne sahip kullanıcılar tüm gönderilerin durumlarını merkezi panel üzerinden yönetebilir.

Sistem, frontend tarafında React.js, backend tarafında ise Node.js Express.js ile geliştirilmiştir. Veriler MongoDB üzerinde saklanmaktadır. JWT (JSON Web Token) kullanılarak oturum yönetimi ve güvenli erişim sağlanmıştır. Uygulama, hem fonksiyonel hem de fonksiyonel olmayan gereksinimleri karşılamak üzere modern web teknolojileriyle hazırlanmıştır.

**Index Terms**—Bu belge IEEE standardına uygun şekilde, sistemin tasarımı, gereksinimleri, geliştirilme aşamaları ve elde edilen sonuçları ayrıntılı olarak sunmaktadır.

## I. INTRODUCTION

Kargo taşımacılığı, dijital dönüşüm sürecinde büyük öneme sahip lojistik sektörlerinden biridir. Günümüzde bireysel ve kurumsal müşteriler, kargolarının durumunu hızlıca öğrenmek, gönderi oluşturmak ve takibini yapmak istemektedir. Bu ihtiyaca cevap veren, kullanıcı dostu ve güvenli bir Kargo Takip Otomasyon Sistemi geliştirilmektedir.

Bu proje, Web tabanlı bir tam yığın (full-stack) uygulama olarak React.js, Node.js ve MongoDB teknolojileri ile inşa edilmiştir. Projenin temel amacı, kullanıcıların kargo gönderisi oluşturabilmesini ve bu gönderilerin sistemli bir şekilde yönetilmesini sağlamaktır. Aynı zamanda, gönderi durumlarının güncellenmesi ve takibinin yapılabilmesi hedeflenmiştir.

Projenin geliştirilmesinde aşağıdaki başlıca hedefler gözetilmiştir:

Kullanıcı yönetimi ve rol tabanlı erişim: Sisteme giriş yapan her kullanıcı "musteri" veya "admin" rolünde işlem yapar.

Gönderi oluşturma ve izleme: Müşteriler kendi gönderilerini oluşturabilir ve bu gönderilerin durumunu takip edebilir.

Yönetim paneli: Admin kullanıcılar tüm gönderileri görebilir ve durumlarını yönetebilir.

Geliştirilebilir mimari: Sistemin genişletilebilmesi adına "şube", "araç" ve "takip olayı" gibi ek varlıklar sisteme entegre edilmiştir.

Kullanıcı dostu arayüz: Tasarımda sadelik, anlaşılabilirlik ve erişilebilirlik ilkelerine dikkat edilmiştir.

Bu belgede, geliştirilen kargo takip sisteminin teknik detayları, kullanılan teknolojiler, veri yapıları, mimari bileşenler ve genel sistem işleyişi sunulacaktır.

## II. FUNCTIONAL REQUIREMENTS

Bu bölümde, sistemin kullanıcılarına sunduğu temel işlevsel gereksinimler (functional requirements) listelenmiştir. Gereksinimler, kullanıcıların sistemle olan etkileşimlerine göre iki ana role göre kategorize edilmiştir: Müşteri (musteri) ve Yönetici (admin).

2.1 Kullanıcı Rollerine Sistem iki ana kullanıcı tipine hizmet eder:

Müşteri (musteri): Sisteme kayıt olan ve gönderi oluşturan kişidir.

Admin: Gönderi yönetimi, takip durumu güncelleme ve sistem genel verilerini görme yetkisine sahip kullanıcıdır.

2.2 Müşteri İşlevleri Müşteri rolüne sahip kullanıcıların gerçekleştirebildiği işlemler:

Kayıt Olma: Yeni kullanıcı, ad, soyad, email, telefon ve adres bilgilerini girerek sisteme kayıt olabilir.

Giriş Yapma: Email ve şifre kullanarak sisteme giriş yapılabilir.

Profil Bilgilerini Görüntüleme: Giriş yaptıktan sonra ad, soyad, email, adres gibi bilgilerini görebilir.

Gönderi Oluşturma: Alıcı adı, alıcı adresi ve bağlı olduğu şube bilgilerini girerek yeni gönderi oluşturabilir.

Gönderi Takibi: Sisteme kayıtlı olan kendi gönderilerinin listesini ve bu gönderilerin son durumlarını görüntüleyebilir.

2.3 Admin İşlevleri Yönetici rolüne sahip kullanıcıların gerçekleştirebildiği işlemler:

Giriş Yapma: Email ve şifre ile giriş yaptıktan sonra yönlendirme doğrudan admin paneline yapılır.

Tüm Gönderileri Görüntüleme: Sistem üzerindeki tüm gönderi kayıtlarını listeleyebilir.

Gönderi Durumunu Güncelleme: Her gönderinin "hazırlanıyor", "taşıyor" veya "teslim edildi" gibi durumları arasında seçim yaparak güncelleyebilir.

Yeni Şube Ekleme: İsteğe bağlı olarak sistemde yeni şube bilgileri oluşturabilir.

Araç ve Takip Olayı Genişletme: Araç modeli şubeye bağlı taşıma aracı olarak eklenebilir. Takip olayları gönderinin nerede ve ne zaman geçtiğini kayıt altına alır.

2.4 Oturum ve Erişim Kontrolü JWT Tabanlı Doğrulama: Kullanıcı giriş yaptıktan sonra sunucudan aldığı token ile yetkili işlemleri yapabilir.

Rol Tabanlı Yönlendirme: Kullanıcının rolüne göre (/panel veya /admin) yönlendirme yapılır.

Yetkisiz Erişim Engelleme: Giriş yapmadan herhangi bir özel sayfaya doğrudan erişim denemesi /login sayfasına yönlendirilir.

### III. NON-FUNCTIONAL REQUIREMENTS

Fonksiyonel olmayan gereksinimler, sistemin nasıl çalıştığını, kullanıcı deneyimini ve teknik gereksinimleri tanımlar. Bu gereksinimler, sistemin performansını, güvenliğini, taşınabilirliğini ve bakım kolaylığını artırmaya yöneliktir.

3.1 Performans Gereksinimleri İstemci-Tarayıcı Etkileşimi: Sayfa geçişleri hızlı olmalı ve kullanıcıya akıcı bir deneyim sunmalıdır.

Veritabanı İstekleri: Gönderi oluşturma, listeleme ve durum güncellemeleri en fazla 1 saniyelik yanıt süresine sahip olmalıdır.

Çoklu kullanıcı desteği: Aynı anda birden fazla kullanıcının sistem üzerinde işlem yapmasına olanak tanınmalıdır.

3.2 Güvenlik Gereksinimleri JWT Tabanlı Kimlik Doğrulama: Her kullanıcının işlemleri, oturum token'ı ile güvence altına alınmıştır.

Şifre Hashleme: Kullanıcı şifreleri bcryptjs kütüphanesi ile hashlenmiş olarak veritabanında saklanır.

Yetkisiz Erişim Engelleme: Giriş yapmamış kullanıcılar, korunan rotalara erişemez; admin/müşteri ayrımı yapılır.

3.3 Kullanılabilirlik ve Arayüz Kullanıcı Dostu Arayüz: Tüm sayfalar sade, açıklayıcı ve sarı temalı profesyonel bir görünüm sunar.

Responsive Tasarım: Uygulama, tablet ve masaüstü ekran boyutlarında uyumlu şekilde çalışır.

İkon Desteği: Arayüzde anlaşılabilirlik için emoji ve simge kullanımı mevcuttur.

3.4 Taşınabilirlik ve Dağıtım Platform Bağımsızlık: Sistem, Node.js ve React teknolojileri kullanıldığı için Windows, Linux ve MacOS sistemlerinde çalıştırılabilir.

Docker Uyumlu Yapı (opsiyonel): Uygulama yapısı container'lara taşınabilir şekilde tasarlanmıştır.

Kolay Kurulum: İstemci ve sunucu klasörlerinde npm install + npm start komutlarıyla hızlı kurulum yapılabilir.

3.5 Sürdürülebilirlik ve Genişletilebilirlik Bileşen Tabanlı Yapı: Frontend tarafı component yapısı ile geliştirilmiş olup yeni özelliklerin eklenmesi kolaydır.

Modüler Backend: Her model için ayrı route ve controller yapısı kullanılarak kod sürdürülebilirliği sağlanmıştır.

Yeni Roller veya Tablo Ekleme Uygun: Sistem, ileride kuryeler veya yöneticiler gibi ek roller için geliştirilebilir durumdadır.

### IV. SYSTEM ARCHITECTURE

Kargo Takip Otomasyon Sistemi, üç katmanlı (three-tier) yazılım mimarisi kullanılarak geliştirilmiştir. Bu yapı, kullanıcı arayüzü (UI), iş mantığı (Business Logic) ve veri yönetimi (Data Layer) katmanlarının ayrılarak bağımsız geliştirilebilmesini ve sürdürülebilirliğini sağlar.

4.1 Katmanlar ve Teknolojiler Katman Açıklama Teknoloji Sunum Katmanı (UI) –İ Kullanıcının etkileşimde bulunduğu arayüz katmanı (React.js, CSS) İş Mantığı (API)–İ Veritabanı ile arayüz arasındaki köprü, iş kuralları (Node.js, Express.js) Veri Katmanı (DB) –İ Verilerin kalıcı olarak saklandığı katman (MongoDB, Mongoose ODM)

4.2 İstemci - Sunucu Etkileşimi Kullanıcı, tarayıcı üzerinden React.js tabanlı arayüzle etkileşime geçer.

Giriş, kayıt ve gönderi işlemleri, HTTP istekleri (axios) ile Express.js sunucusuna gönderilir.

Sunucu, kimlik doğrulaması (JWT) ve yetkilendirme işlemlerini kontrol eder.

Gönderiler, kullanıcılar, şubeler ve diğer veriler MongoDB üzerinde Mongoose modelleri aracılığıyla tutulur.

Yanıtlar tekrar istemciye döner ve kullanıcı arayüzü güncellenir.

4.3 Uygulama Akışı (Kısa Senaryo) Kullanıcı sisteme email ve şifre ile giriş yapar.

Sisteme alınan JWT token, localStorage üzerinden saklanır.

Kullanıcı rolü kontrol edilerek /panel ya da /admin rotalarına yönlendirme yapılır.

Müşteri gönderi oluşturur; admin ise bu gönderinin durumunu günceller.

Her gönderi, şube ve müşteri ile ilişkilidir ve takip geçmişi kayıt altına alınabilir.

4.4 Bileşen Görevi Login.jsx –İ Giriş işlemlerini yönetir Register.jsx –İ Yeni kullanıcı kaydını sağlar Panel.jsx –İ Müşteri işlemleri (profil, gönderi, listeleme) AdminPanel.jsx –İ Admin için gönderi yönetimi ve durum kontrolü Sidebar.jsx –İ Navigasyon menüsü ProtectedRoute –İ Rol ve token kontrolü ile erişim izni musterRoute.js –İ Müşteri login/kayıt işlemleri gonderiRoute.js –İ Gönderi oluşturma ve listeleme işlemleri adminRoute.js –İ Admin özel yetkili işlemler (durum güncelleme) subeRoute.js –İ Şube verilerini çekme ve ekleme işlemleri

4.5 Güvenlik ve Erişim Kontrolü Kullanıcılar sisteme giriş yaptıklarında JWT token ile yetkilendirilir.

ProtectedRoute bileşeni sayesinde kullanıcı rolü kontrol edilerek uygun sayfalara erişim sağlanır.

Admin, yalnızca kendisine açık sayfalara ulaşabilir; müşteri ise yalnızca kendi verilerine erişebilir.

### CONCLUSION

Bu çalışmada, kullanıcıların kargo gönderilerini yönetebileceği, takip edebileceği ve yöneticilerin bu gönderiler üzerinde denetim sağlayabileceği tam işlevli bir web tabanlı Kargo Takip Otomasyon Sistemi geliştirilmiştir. Uygulama, günümüzün modern yazılım gereksinimlerini karşılayan React.js, Node.js ve MongoDB teknolojileri kullanılarak

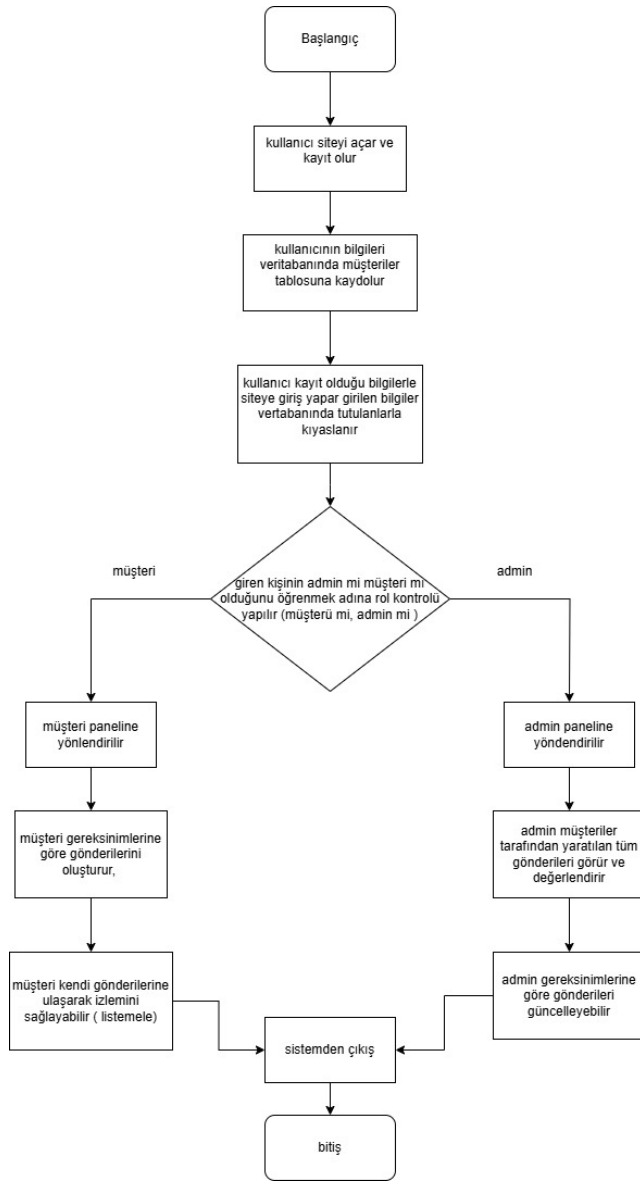


Fig. 1. Akış Şeması

geliştirilmiş, hem müşteri hem de yönetici rollerine özel özellikler içerecek şekilde yapılandırılmıştır.

Sistemin geliştirilmesinde:

Güvenli kullanıcı yönetimi,

Rol tabanlı sayfa erişimi,

Gönderi oluşturma, görüntüleme ve durum güncelleme işlemleri,

Takip geçmişi entegrasyonu,

Modern, kullanıcı dostu arayüz tasarımı,

Esnek, ilişkisel ve genişletilebilir veritabanı modellemesi gibi temel hedefler başarıyla gerçekleştirilmiştir.

5.1 Projenin Sağladığı Kazanımlar Geliştirici açısından tam yığın (full-stack) bir uygulamanın her adımını baştan sona inşa etme deneyimi sağlanmıştır.

Kullanıcı deneyimi açısından, sade ve erişilebilir arayüz ile hızlı işlem imkanı sağlanmıştır.

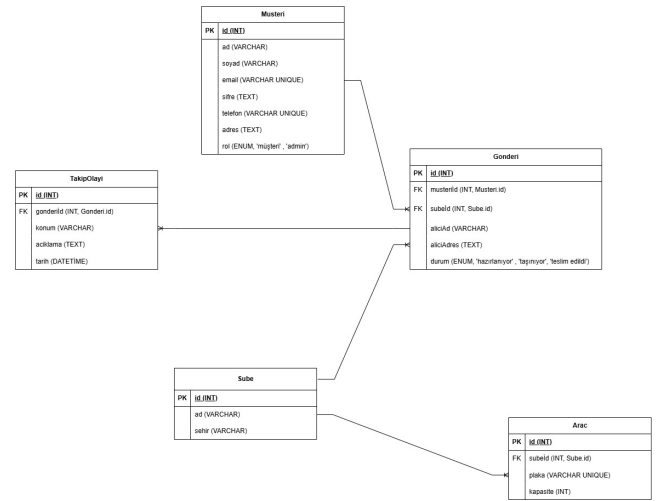


Fig. 2. Veritabanı Diyagramı

Sistem mimarisi ve veri yapısı açısından genişletilebilir ve sürdürülebilir bir altyapı kurulmuştur.

Veritabanı ilişkileri, normalizasyon, index, view ve trigger gibi VTYS konuları gerçek proje üzerinden uygulanarak pekiştirilmiştir.

5.2 Gelecek Geliştirme Önerileri Sistem hali hazırda başarılı şekilde çalışsa da, ilerleyen aşamalarda şu geliştirmeler yapılabilir:

Kurye rolü: Gönderi taşıyan personelin mobil cihazdan teslimat onayı vermesi

E-posta bildirim sistemi: Gönderi durumları müşterilere otomatik e-posta ile bildirilebilir

Harita entegrasyonu: Takip olaylarının konumlarının harita üzerinde gösterilmesi

Yönetim panelinde filtreleme/sıralama: Admin'in gönderileri kolayca yönetebilmesi için filtreleme desteği

5.3 Genel Değerlendirme Proje, hem fonksiyonel hem de teknik açıdan başarıyla tamamlanmış, IEEE yazım formatına uygun olarak raporlanmıştır. Uygulama istenilen özellikleri sağlamakla kalmayıp; genişletilebilirliği ve açık kaynak kod yapısı sayesinde gelecekte farklı senaryolara uyarlanabilecek potansiyele sahiptir.

## REFERENCES

- [1] Mongoose ODM Documentation – <https://mongoosejs.com/docs/guide.html> Veritabanı modelleme işlemleri için kullanılan ODM kütüphanesi hakkında resmi dokümantasyon.
- [2] React.js Official Documentation – <https://reactjs.org/docs/getting-started.html> Uygulamanın kullanıcı arayüzünü oluşturmak için kullanılan JavaScript kütüphanesi.
- [3] Express.js Framework – <https://expressjs.com/> Node.js tabanlı sunucu uygulaması geliştirme çerçevesi.

[4] JWT (JSON Web Token) Authentication – <https://jwt.io/introduction> Kimlik doğrulama işlemleri için kullanılan açık standart token yapısı.

[5] IEEE Manuscript Templates – <https://www.ieee.org/conferences/publishing/templates.html> Proje raporunun hazırlanmasında kullanılan IEEE formatı yönergeleri.

[6] MongoDB Documentation – <https://www.mongodb.com/docs/manual/> Veritabanı yönetimi ve veri yapılarıyla ilgili teknik dökümantasyon.

[7] Stack Overflow – <https://stackoverflow.com/> Proje geliştirme sürecinde karşılaşılan teknik sorunlara yönelik araştırma ve çözümler.