**Yazılım Gereksinimleri Belirtimi (SRS)**

**TrackMate Otomasyon Sistemi**

**Tarih: 18 Mart 2025**

**Hazırlayan: Abdulhadi ELEYYÜB**

**İçindekiler**

1. Giriş
2. Sistem Genel Bakış
3. Fonksiyonel Gereksinimler
4. Fonksiyonel Olmayan Gereksinimler
5. Sistem Mimarisi
6. Veritabanı Tasarımı
7. Kullanıcı Arayüzü
8. Entegrasyon Gereksinimleri
9. Güvenlik Gereksinimleri
10. Uygulama Planı
11. Ekler

**1. Giriş**

**1.1 Amaç**

Bu Yazılım Gereksinimleri Belirtimi (SRS) dokümanı, TrackMate Otomasyon Sistemi'nin detaylı bir tanımını içermektedir. Bu doküman, çok şirketli yönetim sisteminin geliştirilmesi için fonksiyonel ve fonksiyonel olmayan gereksinimleri, sistem mimarisini ve uygulama planını ana hatlarıyla belirtmektedir.

**1.2 Proje Kapsamı**

TrackMate, birden fazla şirketin müşterilerini, ürünlerini, siparişlerini ve faturalarını entegre bir platform üzerinden yönetmelerini sağlamak üzere tasarlanmış kapsamlı bir otomasyon sistemidir. Sistem, merkezi yönetim, kullanıcı yönetimi ve raporlama özellikleri ile iş süreçlerini optimize etmeyi amaçlamaktadır.

**1.3 Tanımlar, Kısaltmalar ve Terimler**

* **SRS**: Yazılım Gereksinimleri Belirtimi
* **API**: Uygulama Programlama Arayüzü
* **UI**: Kullanıcı Arayüzü
* **JWT**: JSON Web Token
* **CRUD**: Oluşturma, Okuma, Güncelleme, Silme
* **ER**: Varlık İlişki

**1.4 Referanslar**

* Proje Ön Raporu
* Veritabanı Şeması

**2. Sistem Genel Bakış**

**2.1 Proje Tanımı**

TrackMate Otomasyon Sistemi, birden fazla şirketin kullanabileceği kapsamlı bir iş yönetim platformudur. Bu sistem ile şirketler müşteri ve ürün yönetimi, sipariş takibi ve fatura işlemlerini entegre bir şekilde gerçekleştirebilirler.

**2.2 Sistem Hedefleri**

* İş süreçlerini dijitalleştirerek verimliliği artırmak
* Birden fazla şirketin tek sistem üzerinden yönetimini sağlamak
* Kullanıcı ve şirket bazlı yetkilendirme ile güvenli bir sistem sağlamak
* Fatura ve belge yönetimini kolaylaştırmak
* Merkezi admin paneli ile sistem yönetimini basitleştirmek

**2.3 Sistem Paydaşları**

* Sistem Yöneticileri: Tüm sistem üzerinde tam kontrol yetkisine sahip kullanıcılar
* Şirket Yöneticileri: Kendi şirketleri üzerinde yönetim yetkisine sahip kullanıcılar
* Şirket Çalışanları: Belirli modüllere erişim yetkisi olan kullanıcılar
* Müşteriler: Sistem üzerinde kayıtlı olan son kullanıcılar

**3. Fonksiyonel Gereksinimler**

**3.1 Admin Paneli Gereksinimleri**

**3.1.1 Şirket Yönetimi**

* Sistem, yeni şirket ekleme, düzenleme ve silme işlevlerini desteklemelidir.
* Her şirket için benzersiz bir tanımlayıcı atanmalıdır.
* Şirketlere ait temel bilgiler (isim, vergi numarası, adres, telefon) saklanmalıdır.
* Sistem yöneticileri tüm şirketlerin listesini görüntüleyebilmelidir.

**3.1.2 Kullanıcı Yönetimi**

* Sistem, kullanıcı ekleme, düzenleme ve silme işlevlerini desteklemelidir.
* Her kullanıcı belirli bir şirket ve rol ile ilişkilendirilmelidir.
* Kullanıcıların durumu (aktif/pasif) yönetilebilmelidir.
* Kullanıcı bilgileri (ad, soyad, e-posta, telefon) saklanmalıdır.

**3.1.3 Sistem Konfigürasyonu**

* Sistem genelinde yapılandırma ayarları yönetilebilmelidir.
* Veritabanı yönetimi ve bakımı yapılabilmelidir.
* Sistem güncelleme ve bakım kontrolleri gerçekleştirilebilmelidir.

**3.2 Şirket Yönetimi Gereksinimleri**

**3.2.1 Personel Yönetimi**

* Şirket yöneticileri kendi personelini ekleyebilmeli, düzenleyebilmeli ve silebilmelidir.
* Personele rol ve yetki ataması yapılabilmelidir.
* Personel aktivasyon/deaktivasyon işlemleri gerçekleştirilebilmelidir.

**3.2.2 Banka Bilgileri Yönetimi**

* Şirketler birden fazla banka hesap bilgisi ekleyebilmelidir.
* Banka bilgileri (banka adı, hesap adı, IBAN, SWIFT, para birimi) saklanmalıdır.
* Fatura oluşturma sırasında kullanılacak banka bilgisi seçilebilmelidir.

**3.3 Ürün Yönetimi**

**3.3.1 Ürün İşlemleri**

* Yeni ürün ekleme, düzenleme ve silme işlevleri desteklenmelidir.
* Ürün bilgileri (ad, açıklama, fiyat, ağırlık, miktar) saklanmalıdır.
* Ürünler belirli bir şirket ile ilişkilendirilmelidir.
* Ürün stok miktarları takip edilebilmelidir.

**3.3.2 Ürün Stok Takibi**

* Sipariş ve fatura oluşturma işlemleri sonrasında stok miktarları güncellenmelidir.
* Kritik stok seviyesine ulaşan ürünler için bildirim üretilebilmelidir.

**3.4 Müşteri Yönetimi**

**3.4.1 Müşteri İşlemleri**

* Yeni müşteri ekleme, düzenleme ve silme işlevleri desteklenmelidir.
* Müşteri bilgileri (ad, telefon, adres) saklanmalıdır.
* Müşteriler belirli bir şirket ile ilişkilendirilmelidir.

**3.4.2 Müşteri İletişimi**

* Müşterilere otomatik bildirim gönderilebilmelidir.
* Müşteri bazlı işlem geçmişi görüntülenebilmelidir.

**3.5 Sipariş Yönetimi**

**3.5.1 Sipariş İşlemleri**

* Yeni sipariş oluşturma, düzenleme ve silme işlevleri desteklenmelidir.
* Siparişler belirli bir müşteri ve şirket ile ilişkilendirilmelidir.
* Sipariş detayları (ürünler, miktarlar, birim fiyatlar, toplam tutar) saklanmalıdır.
* Sipariş durumu (yeni, işlemde, tamamlandı, iptal edildi) takip edilebilmelidir.

**3.5.2 Sipariş Onaylama**

* Sipariş onaylama süreci yönetilebilmelidir.
* Onaylanan siparişler için otomatik fatura oluşturma seçeneği sunulmalıdır.

**3.6 Fatura Yönetimi**

**3.6.1 Fatura İşlemleri**

* Yeni fatura oluşturma, düzenleme ve silme işlevleri desteklenmelidir.
* Faturalar belirli bir müşteri, sipariş ve şirket ile ilişkilendirilmelidir.
* Fatura detayları (ürünler, miktarlar, birim fiyatlar, ara toplam, kargo ücreti, toplam tutar) saklanmalıdır.
* Her fatura için benzersiz bir fatura numarası üretilmelidir.

**3.6.2 Fatura Çıktıları**

* Faturalar PDF formatında oluşturulabilmelidir.
* Fatura şablonları özelleştirilebilir olmalıdır.
* Faturalar müşteriye otomatik olarak e-posta ile gönderilebilmelidir.

**3.7 Raporlama**

**3.7.1 Satış Raporları**

* Dönemsel satış raporları oluşturulabilmelidir.
* Ürün bazlı satış analizi yapılabilmelidir.
* Müşteri bazlı satış analizi yapılabilmelidir.

**3.7.2 Stok Raporları**

* Güncel stok durumu raporlanabilmelidir.
* Düşük stok seviyesindeki ürünler listelenebilmelidir.

**3.7.3 Performans Raporları**

* Şirket bazlı performans ölçümleri görüntülenebilmelidir.
* Kullanıcı bazlı performans analizi yapılabilmelidir.

**3.8 E-posta Yönetimi**

**3.8.1 E-posta Gönderimi**

* SMTP sunucusu üzerinden otomatik ve manuel e-posta gönderimi yapılabilmelidir.
* Şablonlar kullanılarak hızlı ve tutarlı e-posta içerikleri oluşturulabilmelidir.
* Toplu e-posta gönderimi desteklenmelidir.

**3.8.2 E-posta Kayıtları**

* Gönderilen tüm e-postalar veritabanında kaydedilmelidir.
* E-posta durumu (gönderildi, başarısız, beklemede) takip edilebilmelidir.
* Gönderim başarısız olduğunda, hata mesajları kaydedilmelidir.
* Müşteriler ve şirketler bazında e-posta geçmişi görüntülenebilmelidir.

**4. Fonksiyonel Olmayan Gereksinimler**

**4.1 Performans Gereksinimleri**

* Sistem aynı anda en az 1000 kullanıcıyı desteklemelidir.
* Sayfa yüklenme süreleri 3 saniyeden az olmalıdır.
* Veritabanı sorguları 2 saniye içinde tamamlanmalıdır.
* Sistem 7/24 erişilebilir olmalıdır.

**4.2 Güvenlik Gereksinimleri**

* Kullanıcı kimlik doğrulama JWT ile sağlanmalıdır.
* Şirket ve rol bazlı yetkilendirme sistemi uygulanmalıdır.
* Şirket verileri birbirinden izole edilmelidir.
* Tüm şifreler güvenli bir şekilde hash'lenerek saklanmalıdır.
* İletişim SSL/TLS ile şifrelenmelidir.

**4.3 Kullanılabilirlik Gereksinimleri**

* Sistem arayüzü sezgisel ve kullanımı kolay olmalıdır.
* Mobil ve web uygulamaları tutarlı bir kullanıcı deneyimi sunmalıdır.
* Sistem çoklu dil desteği sağlamalıdır.
* Kullanıcı arayüzü, farklı ekran boyutlarına uyum sağlamalıdır.

**4.4 Ölçeklenebilirlik Gereksinimleri**

* Sistem, artan kullanıcı sayısına ve veri hacmine uyum sağlayabilmelidir.
* Yeni şirket ve modül eklemeleri sistem performansını etkilememelidir.

**4.5 Bakım Gereksinimleri**

* Sistem modüler bir yapıda olmalı, bileşenlerin bağımsız olarak güncellenebilmesine olanak tanımalıdır.
* Sistem güncellemeleri, kullanıcı deneyimini minimum düzeyde etkilemelidir.
* Veritabanı yedekleme ve geri yükleme işlemleri otomatikleştirilebilir olmalıdır.

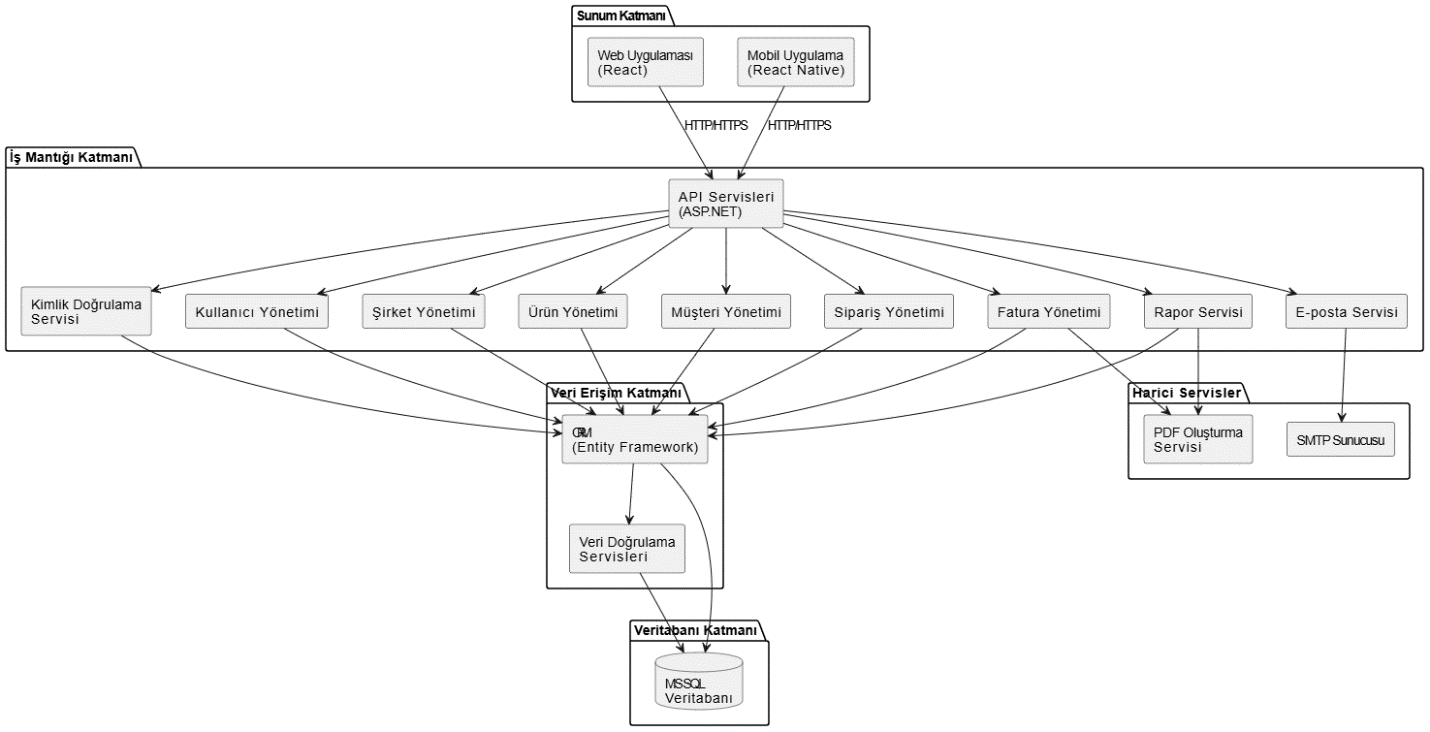
**5. Sistem Mimarisi**

**5.1 Mimari Yapı**

TrackMate Otomasyon Sistemi, çok katmanlı bir mimari yapı kullanacaktır:

1. **Sunum Katmanı**:
   * Web Uygulaması (React)
   * Mobil Uygulama (React Native)
2. **İş Mantığı Katmanı**:
   * API Servisleri (ASP.NET)
   * İş Mantığı Modülleri
3. **Veri Erişim Katmanı**:
   * Veritabanı Erişim Katmanı (ORM)
   * Veri Doğrulama Servisleri
4. **Veritabanı Katmanı**:
   * MS SQL Veritabanı

**5.2 Bileşen Diyagramı**



**5.3 Dağıtım Mimarisi**

* Web ve mobil uygulamalar, RESTful API üzerinden backend servisleriyle iletişim kuracaktır.
* Backend servisleri, veritabanına güvenli bir şekilde erişim sağlayacaktır.
* Tüm sistem bileşenleri, bulut tabanlı bir altyapıda dağıtılacaktır.

**6. Veritabanı Tasarımı**

**6.1 Veri Modeli**

TrackMate sistemi, çok şirketli bir mimariyi destekleyen ilişkisel bir veritabanı yapısı kullanacaktır. Her veri tablosu şirket kimliği ile ilişkilidir, böylece veriler şirketler arasında ayrı tutulur ve erişim kontrol edilir.

**6.2 Veritabanı Şeması**

-- Şirketler Tablosu

CREATE TABLE Companies (

CompanyID INT PRIMARY KEY IDENTITY(1,1),

CompanyName VARCHAR(100) NOT NULL,

TaxID VARCHAR(50) UNIQUE,

Address TEXT,

PhoneNo VARCHAR(20),

CreatedDate DATETIME NOT NULL DEFAULT GETDATE()

);

-- E-posta Kayıtları Tablosu

CREATE TABLE EmailLogs (

ID INT PRIMARY KEY IDENTITY(1,1),

CompanyID INT NOT NULL,

CustomerID INT NULL, -- Müşteriye gönderilmeyen e-postalar için NULL olabilir

Subject VARCHAR(200) NOT NULL,

EmailContent TEXT NOT NULL,

RecipientEmail VARCHAR(100) NOT NULL,

EmailType VARCHAR(50) NOT NULL, -- 'InvoiceNotification', 'OrderConfirmation' vb.

RelatedEntityID INT NULL, -- Fatura ID, Sipariş ID gibi ilgili varlık ID'si

RelatedEntityType VARCHAR(50) NULL, -- 'Invoice', 'Order' gibi

SentDate DATETIME NOT NULL DEFAULT GETDATE(),

Status VARCHAR(20) NOT NULL, -- 'Sent', 'Failed', 'Pending' vb.

ErrorMessage TEXT NULL,

SentBy INT NULL, -- Gönderen kullanıcı ID'si

FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID),

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),

FOREIGN KEY (SentBy) REFERENCES Users(ID)

);

-- Kullanıcılar Tablosu

CREATE TABLE Users (

ID INT PRIMARY KEY IDENTITY(1,1),

CompanyID INT NOT NULL,

UserID VARCHAR(50) NOT NULL UNIQUE,

Fullname VARCHAR(100) NOT NULL,

Email VARCHAR(100) NOT NULL UNIQUE,

Password VARCHAR(100) NOT NULL,

Phone VARCHAR(20),

Role VARCHAR(20) NOT NULL,

Status VARCHAR(10) NOT NULL,

FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID)

);

-- Banka Bilgileri Tablosu

CREATE TABLE CompanyBankDetails (

ID INT PRIMARY KEY IDENTITY(1,1),

CompanyID INT NOT NULL,

BankName VARCHAR(100) NOT NULL,

AccountName VARCHAR(100) NOT NULL,

IBAN VARCHAR(50) NOT NULL UNIQUE,

SWIFT VARCHAR(20),

Currency VARCHAR(3) NOT NULL DEFAULT 'USD',

CreatedDate DATETIME NOT NULL DEFAULT GETDATE(),

FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID)

);

-- Ürünler Tablosu

CREATE TABLE Products (

ProductID INT PRIMARY KEY IDENTITY(1,1),

CompanyID INT NOT NULL,

ProductName VARCHAR(100) NOT NULL,

Description TEXT,

Price DECIMAL(10,2) NOT NULL,

Weight DECIMAL(10,2),

AddDate DATETIME NOT NULL DEFAULT GETDATE(),

Quantity INT NOT NULL,

FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID)

);

-- Müşteriler Tablosu

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY IDENTITY(1,1),

CompanyID INT NOT NULL,

Name VARCHAR(100) NOT NULL,

PhoneNo VARCHAR(20),

Address TEXT,

FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID)

);

-- Siparişler Tablosu

CREATE TABLE Orders (

OrderID INT PRIMARY KEY IDENTITY(1,1),

CompanyID INT NOT NULL,

CustomerID INT,

OrderDate DATETIME NOT NULL DEFAULT GETDATE(),

Status VARCHAR(20) NOT NULL,

Total DECIMAL(10,2) NOT NULL,

CreatedBy INT,

FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID),

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),

FOREIGN KEY (CreatedBy) REFERENCES Users(ID)

);

-- Sipariş Ürünleri Tablosu

CREATE TABLE OrderItems (

ID INT PRIMARY KEY IDENTITY(1,1),

OrderID INT NOT NULL,

ProductID INT NOT NULL,

Quantity INT NOT NULL,

UnitPrice DECIMAL(10,2) NOT NULL,

Total DECIMAL(10,2) NOT NULL,

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),

FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

);

-- Faturalar Tablosu

CREATE TABLE Invoices (

ID INT PRIMARY KEY IDENTITY(1,1),

CompanyID INT NOT NULL,

InvoiceNo VARCHAR(50) NOT NULL UNIQUE,

InvoiceDate DATETIME NOT NULL DEFAULT GETDATE(),

CustomerID INT,

OrderID INT NULL, -- Her sipariş için fatura zorunlu değil

BankID INT NOT NULL, -- Şirketin banka bilgisini tutacak

Subtotal DECIMAL(10,2) NOT NULL,

ShippingCost DECIMAL(10,2),

Total DECIMAL(10,2) NOT NULL,

Currency VARCHAR(3) NOT NULL DEFAULT 'USD',

CreatedBy INT,

CreatedDate DATETIME NOT NULL DEFAULT GETDATE(),

FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID),

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),

FOREIGN KEY (BankID) REFERENCES CompanyBankDetails(ID),

FOREIGN KEY (CreatedBy) REFERENCES Users(ID)

);

-- Fatura Ürünleri Tablosu

CREATE TABLE InvoiceItems (

ID INT PRIMARY KEY IDENTITY(1,1),

InvoiceID INT NOT NULL,

ProductID INT NOT NULL,

Description TEXT,

Quantity INT NOT NULL,

UnitPrice DECIMAL(10,2) NOT NULL,

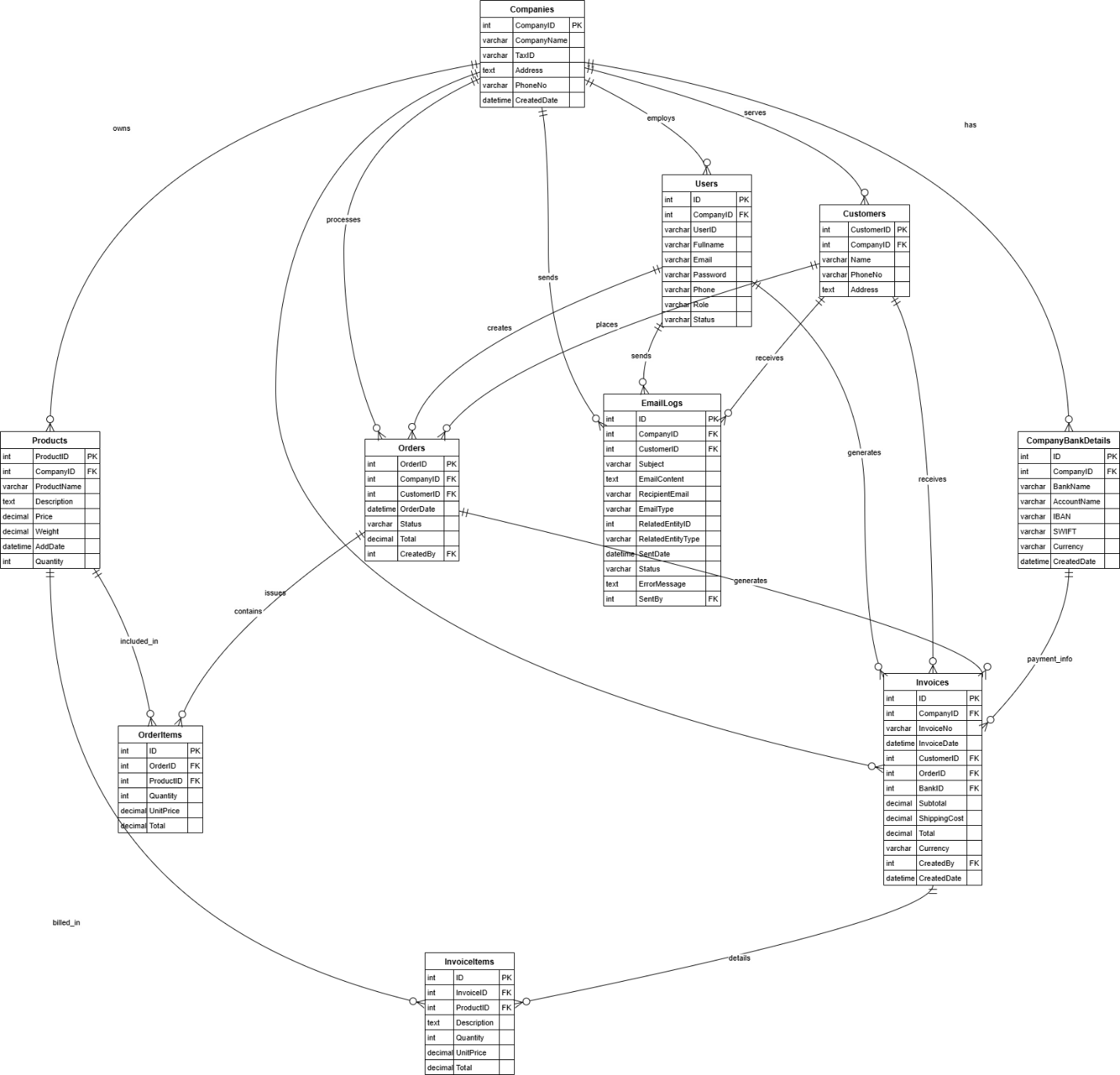
Total DECIMAL(10,2) NOT NULL,

FOREIGN KEY (InvoiceID) REFERENCES Invoices(ID),

FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

);

**6.3 ER Diyagramı**



**7. Kullanıcı Arayüzü**

**7.1 Web Arayüzü**

Web arayüzü React kullanılarak geliştirilecek ve aşağıdaki özellikleri içerecektir:

* Duyarlı tasarım (responsive design)
* Sezgisel kullanıcı deneyimi
* Tablo, grafik ve form bileşenleri
* Özelleştirilebilir tema seçenekleri

**7.2 Mobil Arayüzü**

Mobil arayüzü React Native kullanılarak geliştirilecek ve aşağıdaki özellikleri içerecektir:

* Platform uyumlu tasarım (iOS ve Android için)
* Hızlı erişim için optimize edilmiş arayüz
* Basitleştirilmiş form ve tablo bileşenleri
* Bildirim entegrasyonu

**7.3 Admin Paneli**

Admin paneli, TrackMate sisteminin merkezi yönetim noktası olarak tasarlanacak ve aşağıdaki özellikleri içerecektir:

* Tüm şirketlerin yönetimi
* Kullanıcı ve rol yönetimi
* Sistem yapılandırma ayarları
* Raporlama ve analiz araçları
* Veritabanı yönetim araçları
* Sistem bakım ve güncelleme kontrolleri

**8. Entegrasyon Gereksinimleri**

**8.1 E-posta Entegrasyonu**

* Sistem, otomatik bildirimler için e-posta gönderim hizmeti ile entegre olmalıdır.
* Fatura, sipariş onayı gibi durumlar için şablon tabanlı e-postalar gönderebilmelidir.

**8.2 Doküman İşleme**

* Excel ve PDF formatlarında rapor ve fatura çıktıları oluşturulabilmelidir.
* Dokümanlar, sistem üzerinden indirilebilir olmalıdır.

**8.3 API Entegrasyonu**

* Sistem, üçüncü taraf uygulamalar için RESTful API'ler sunmalıdır.
* API'ler, güvenli erişim için kimlik doğrulama ve yetkilendirme mekanizmaları içermelidir.

**9. Güvenlik Gereksinimleri**

**9.1 Kimlik Doğrulama ve Yetkilendirme**

* Kullanıcı kimlik doğrulama JWT (JSON Web Token) kullanılarak gerçekleştirilmelidir.
* Rol tabanlı erişim kontrolü (RBAC) uygulanmalıdır.
* Şirket bazlı veri izolasyonu sağlanmalıdır.

**9.2 Veri Güvenliği**

* Tüm hassas veriler şifrelenmiş olarak saklanmalıdır.
* Veritabanı erişimi güvenli şekilde yönetilmelidir.
* Kullanıcı şifreleri güçlü hash algoritmaları kullanılarak saklanmalıdır.

**9.3 İletişim Güvenliği**

* Tüm API çağrıları ve veri transferleri SSL/TLS üzerinden gerçekleştirilmelidir.
* API anahtarları ve erişim belirteçleri güvenli bir şekilde yönetilmelidir.

**10. Uygulama Planı**

**10.1 Geliştirme Takvimi**

|  |  |  |  |
| --- | --- | --- | --- |
| Haftalar | Backend (ASP.NET & MS SQL) | Web Geliştirme (React) | Mobil Geliştirme (React Native) |
| 1. Hafta | Gereksinim analizi & planlama, ER diagramı & veritabanı tasarımı, Proje yapısının oluşturulması | - | - |
| 2. Hafta | Kimlik doğrulama (JWT, ASP.NET Identity), Şirket, kullanıcı ve rol yönetimi, Veritabanı bağlantıları | - | - |
| 3. Hafta | Ürün, müşteri ve sipariş yönetimi API’leri, Fatura yönetimi API’leri, Dosya/doküman işleme (Excel, PDF), E-posta bildirim sistemi | - | - |
| 4. Hafta | - | Admin panel tasarımı & temel bileşenler, Backend API entegrasyonu | Giriş & kayıt ekranları, kimlik doğrulama, Ana sayfa ve menü tasarımı |
| 5. Hafta | - | Kimlik doğrulama (JWT, Role-based access), Ürün, müşteri ve sipariş yönetimi ekranları | Backend API bağlantısı kurulumu, Ürün, sipariş, müşteri yönetimi ekranları |
| 6. Hafta | - | Sipariş & fatura oluşturma, Raporlama & doküman yönetimi | Fatura görüntüleme & oluşturma, Bildirim & iletişim sistemleri |
| 7. Hafta | - | Bu haftada eksik oldugum bir ozellik varsa eklemek icin | Bu haftada eksik oldugum bir ozellik varsa eklemek icin |
| 8. Hafta | Test ve hata düzeltme (Backend) | Test ve hata düzeltme (Web) | Test ve hata düzeltme (Mobil) |

**10.2 Test Stratejisi**

* Birim testleri (Unit testing)
* Entegrasyon testleri (Integration testing)
* Kullanıcı arayüzü testleri (UI testing)
* Yük testleri (Load testing)
* Güvenlik testleri (Security testing)

**10.3 Sürüm Yönetimi**

* Git üzerinde versiyon kontrolü yapılacaktır.
* Her ana özellik için ayrı bir branch oluşturulacaktır.
* Düzenli aralıklarla sürüm yayınlanacaktır.

**11. Ekler**

**11.1 Terimler Sözlüğü**

* **Admin Paneli**: Sistem yöneticilerinin kullandığı merkezi yönetim arayüzü.
* **Şirket**: TrackMate sistemi üzerinde kayıtlı olan ve sistemi kullanan tüzel kişilik.
* **Rol**: Kullanıcıların sistem üzerindeki yetkilerini belirleyen gruplandırma.
* **JWT**: JSON Web Token, güvenli kimlik doğrulama için kullanılan bir standart.