

TradeBot Projesi - Yazılım Gereksinim Spesifikasyonu (SRS)

Versiyon: 1.0

Tarih: 16.08.2025

1. Giriş

1.1. Proje Özeti

Bu belge, "TradeBot" adlı çok katmanlı bir al-sat sinyal botu projesinin fonksiyonel ve fonksiyonel olmayan gereksinimlerini tanımlar. Sistem, hisse senedi verilerini PostgreSQL veritabanından okuyacak, bu verileri Teknik Analiz (Katman 1), İstatistiksel Analiz (Katman 2) ve Yapay Zeka (Katman 3) katmanlarında işleyecek ve bir Karar Matrisi (Katman 4) ile nihai bir yatırım tavsiyesi üretecektir. Projenin ana backend'i Java Spring Boot ile, Yapay Zeka servisi ise Python ile geliştirilecektir.

1.2. Proje Kapsamı

Bu SRS, 2 haftalık bir geliştirme süreci sonunda ortaya çıkacak olan **Minimum Uygulanabilir Ürün (Minimum Viable Product - MVP)**'i kapsamaktadır. Bu süreçte hedef, tüm katmanları çalışan ve bir RESTful API üzerinden sonuç üreten bir prototip oluşturmaktır. GraalVM, Kafka entegrasyonu, tam mikroservis mimarisi ve gelişmiş ML modelleri bu 2 haftalık kapsamın dışındadır.

2. Fonksiyonel Gereksinimler

Aşağıdaki gereksinimler, geliştirme önceliğine göre sıralanmıştır.

ID	Gereksinim Adı	Açıklama	Katman	Öncelik	Hafta
FR-01	Veri Katmanı Oluşturma	PostgreSQL'deki stock_data tablosu için bir Spring Data JPA Entity ve Repository sınıfı oluşturulmalıdır.	Veri	Yüksek	1
FR-02	Veri Servisi	Belirli bir hisse senedi ve tarih aralığı için OHLCV verisini veritabanından çeken bir Service katmanı yazılmalıdır.	Veri	Yüksek	1

FR-03	Teknik Analiz Motoru	ta4j kütüphanesi kullanılarak, veritabanından gelen canlı veriyle RSI, MACD vb. indikatörler hesaplanmalı ve bir Sinyal/Skor üretilmelidir.	Katman 1	Yüksek	1
FR-04	İstatistiksel Analiz Motoru	Veritabanı verileri kullanılarak Mean Reversion ve Z-score deviation hesaplamaları yapılmalıdır.	Katman 2	Yüksek	1
FR-05	İstatistiksel Skor Üretimi	İstatistiksel analiz sonuçlarına göre -1 ile +1 arasında bir "Al/Sat Puanı" üreten bir mantık geliştirilmelidir.	Katman 2	Yüksek	1
FR-06	Karar Matrisi Servisi	Gelen tüm analiz sonuçlarını birleştirecek DecisionMatrixServ ice adında bir servis oluşturulmalıdır.	Katman 4	Yüksek	1
FR-07	Karar Matrisi (Mock Entegrasyon)	Katman 1 ve 2'den gelen skorları birleştirmeli, Katman 3'ten geliyormuş gibi sahte (mock) bir veriyi (signal: HOLD, confidence: 0.5) kullanarak ilk "Net Skoru" hesaplamalıdır.	Katman 4	Yüksek	1

FR-08	Çekirdek Motor Testi	Tüm Java akışının (FR-01'den FR-07'ye kadar) birim ve entegrasyon testleri yazılarak doğrulanmalıdır.	Test	Yüksek	1
FR-09	Yapay Zeka Servisi (API)	Python'da FastAPI veya Flask kullanılarak, dışarıdan JSON formatında veri kabul eden /predict endpoint'ine sahip basit bir web sunucusu oluşturulmalıdır.	Katman 3	Yüksek	2
FR-10	Yapay Zeka Servisi (Placeholder)	/predict endpoint'i, henüz gerçek bir ML modeli olmadan, gelen veriye göre sahte bir Sinyal/Güven Skoru ({"signal": "BUY", "confidence": 0.7}) döndürmelidir.	Katman 3	Yüksek	2
FR-11	Servisler Arası İletişim	Java/Spring Boot uygulamasında RestTemplate veya WebClient kullanılarak Python ML servisine ağ üzerinden istek atılmalıdır.	Entegrasyon	Yüksek	2
FR-12	Karar Matrisi (Gerçek Entegrasyon)	DecisionMatrixService içindeki sahte (mock) Katman 3 verisi, Python servisinden gelen	Katman 4	Yüksek	2

		gerçek cevapla değiştirilmelidir.			
FR-13	RESTful API Endpoint	Spring Boot'ta @RestController kullanılarak dışarıdan istekleri karşılayacak POST /analyze/{symbol} endpoint'i tanımlanmalıdır.	API	Yüksek	2
FR-14	Uçtan Uca Akış	/analyze/{symbol} endpoint'i çağrıldığında, tüm analiz akışını (FR-01'den FR-12'ye kadar) tetiklemeli ve nihai kararı JSON formatında döndürmelidir.	API	Yüksek	2
FR-15	Proje Dokümantasyonu	Projenin kurulumunu, çalıştırılmasını ve API kullanımını açıklayan bir README.md dosyası oluşturulmalıdır.	Dokümantasyon	Orta	2

3. Fonksiyonel Olmayan Gereksinimler

ID	Gereksinim Adı	Açıklama
NFR-01	Teknoloji Yığını	Ana backend Java 17+ ve Spring Boot 3+ ile geliştirilecektir. Yapay Zekâ servisi Python 3.9+ kullanacaktır. Veritabanı PostgreSQL olacaktır.
NFR-02	Performans	MVP aşamasında, /analyze endpoint'ine yapılan bir isteğin toplam yanıt süresi 5 saniyenin altında olmalıdır.

NFR-03	Güvenilirlik	Servisler, beklenen veri formatı dışında bir veri geldiğinde hata fırlatmalı ve sistemi kilitlememelidir.
NFR-04	Bakım ve Okunabilirlik	Kod, standart Java ve Python kodlama pratiklerine uygun, anlaşılır ve bakımı kolay bir şekilde yazılmalıdır.

4. Proje Fazları ve Çıktıları

- **1. Hafta Sonu Çıktısı (MVP Çekirdek Motor):**
 - Veritabanından veri okuyabilen, Katman 1 ve Katman 2 analizlerini yapabilen ve bu sonuçları sahte bir Katman 3 verisiyle birleştirerek bir karar üretebilen, test edilmiş bir Java uygulaması.
- **2. Hafta Sonu Çıktısı (Tam Entegre Prototip):**
 - Python ile geliştirilmiş basit bir Yapay Zeka servisinin ana uygulamaya entegre edildiği, tüm sistemin tek bir RESTful API endpoint'i üzerinden uçtan uca çalıştığı, dokümente edilmiş ve sunuma hazır bir proje.