

Uygulamanın Tanımı:

Bir e-ticaret platformunda satın alma uzmanı olarak çalışmaktasınız. Bu rol kapsamında sizden, satın aldığınız ürünlere ait faturaların muhasebe sistemine girişi istenmektedir. Bu girişler yapılırken girilen fatura bedeli aşağıdaki kural doğrultusunda onaylanmakta veya reddedilmektedir.

Kural: Muhasebe sistemine girilecek yeni fatura bilgisi, o satın alım uzmanının şu ana kadar girmiş olduğu ONAYLANMIŞ tüm faturaların toplamını belirlenen limitin üzerine çıkıyorsa ONAYLANMAZ, limitin üzerine çıkarmıyorsa ONAYLANIR.

LİMİT, uygulama süresince tüm satın alma uzmanları için aynıdır.

Her fatura girişi için aşağıdaki bilgileri girilmelidir:

- First Name
- Last Name
- E-mail
- Amount
- Product Name
- Bill No

Örnek işlem bilgisi: "John, Doe, john@doe.com, 30, USB DISC, TR000"

Her satın alma uzmanı Ad, Soyad ve E-mail üçlüsü ile tekil bir şekilde tanımlanır. Aynı Ad ve Soyad bilgisine sahip ama farklı email adresine sahip kişilerin işlemleri ayrı ayrı değerlendirilir.

Örnekler:

İşlem Bilgisi : "John,Doe,john@doe.com,200,TR0001"

Kredi Limit : 200

Sonuç : John Doe'nun işlemi kabul edilir

İşlem Bilgisi : "Jane,Doe,jane@doe.com,201,TR0001"

Kredi Limit : 200

Sonuç : Jane Doe'nun işlemi reddedilir

1. İşlem Bilgisi : "Jane,Doe,jane@doe.com,199,TR0001"

2. İşlem Bilgisi : "Jane,Doe,jane@doe.com,2,TR0002"

Kredi Limit : 200

Sonuç : Jane Doe'nun 2. işlemi reddedilir

Senin görevin aşağıdaki isterleri karşılayacak bir web uygulaması için gerekli REST SERVICE'leri yazmak olacak:

- Fatura bilgisinin kaydı

- Eğer fatura kabul edilirse UYGUN HTTP STATUS'ü ile faturanın kabul edildiğini belirten JSON Response dönmeli
- Eğer fatura kabul edilmezse UYGUN HTTP STATUS'ü ile faturanın kabul edilemediğini belirten JSON Response dönmeli
- Kullanıcıya işlemin başarılı veya başarısız olduğuna dair bilgilendirme gösterimi yapılacaktır. Bu yüzden ORTAK BİR RESPONSE yapısı kurgulanmalıdır.
- Kabul edilen veya reddedilen işlemler ayrı ayrı listelenecektir. Bunun için REST Servis yazılacaktır.

Teknik İsterler:

1. Proje, Maven kullanılarak derlenmeli ve paket oluşturulmalı.
2. Spring Boot kullanılmalı.
3. LİMİT değeri konfigürasyon dosyasından (application.yml) alınmalı.
4. Tercih edilen herhangi bir veritabanı olabilir NOSQL veya RDBMS
5. Object Oriented Programming yaklaşımın uygulanmasına dikkat edilmeli
6. Uygulamanın nasıl çalıştırılacağına dair dokümantasyon yazılmış olmalı

Yapılması opsiyonel olan, artı puan kazandıracak isterler:

1. Servislerin docker ile containerized yapılması.
2. Project içerisinde Postman Collection bulunması.
3. Curl komutlarının paylaşılması.
4. Unit Testlerin yazılmış olması.