Selamlar Developer Team,

Aşağıda size yeni hazırlamış olduğumuz, fırından yeni çıkmış bir case study paylaşmak istiyorum. Bu case study’nin şirket geleceği ve sizin kariyer hedefleriniz içinde çok önemli olacağını düşünmekteyiz.

Bu case study’nin deadline’ı 9 Nisan 2018 saat 15:00’dır. Bir kere her katılan kişiye gerçekten bir ufak sürprizimiz olacak. (Beklentiler lütfen çok aşırıya kaçmasın) 

Çalışmalarınıza isterseniz tek kişi katılabilir, isterseniz yanınızda bir buddy’niz ile geliştirebilirsiniz. (Birincilik ödülü ona göre paylaştırılır) 

Case Study sonrası 1’inciyi seçmede puanlama aşağıdaki kriterlere göre olacaktır. Bu kriterlerin hepsini sağlamak zorunda değilsiniz. Ama nerelerden puan vereceğimizi bilmenizi istiyoruz.

1. Deadline öncesi projeyi teslim etmiş olmak. (15:01’de bu madde geçerli sayılmaz). 15 puan
2. Planlı çalışmanız. Tüm süreci bir doküman haline getirmeniz kod yazmanızdan öncelikli olarak ekstra puan olarak sayılacaktır. Bu konuda çok ekstra bir doküman istemiyoruz. Bu 2 haftalık süre içerisinde işi nasıl parçalara bölüp, gün içerisinde kaç saat ayırdığınız ve nasıl planlı çalıştığınız önemli. 10 puan
3. Çalışmalarınızın raporunu veya gelişmeleri bu 2 hafta içerisinde en az 2 kere paylaşmanız. 5 puan
4. İstenilen Case Study’e göre eksikler yada aklınıza gelen ekstra özellikler olabilir. Bunu geliştirmeniz bittikten sonra ayrı olarak raporlamanız. 5 puan
5. Kod düzgünlüğünü Resharper ile sağlamanız. Projede yarattığınız her sayfada Clean up (Ctrl + E + C) yapmanız. Resharper kullananlar bilir. Her dosyada sağ üst köşede Yeşil tick ikonunun görünmesi  10 puan
6. Projeyi object oriented olarak geliştirmeniz. 25 puan
7. Veritabanı olarak MongoDB kullanmanız. 20 puan
8. Projeyi sıkıştırılmış olarak bizlere yolladığınız zaman sadece connection değerlerini düzeltip çalıştırdığımız zaman sizin lokalinizde çalıştığı gibi sorunsuz çalışması. 5 puan
9. Case study’e katılmanız.  5 puan

Case study aşağıdaki gibidir.

**LOGUE**

Sadece üniversite öğrencilerinin kullanacağı bir portal projesi için bir müşterimiz bize başvurdu. Projenin farklı platformlarda çalışması hedeflendiği için **Web Api ile haberleşmenin sağlanması isteniyor**. Müşterinin talebi üzerine yapılan **her işlemin istek ve sonucunun loglanması isteniyor**. Yaşanacak hız ve sistemsel yük göz önünde bulundurulduğunda **RabbitMQ ile logların asenkron olarak kayıt altına alınmasına karar verildi**. Bu sayede kullanıcılar loglama esnasında oluşacak gecikmelerden etkilenmeyecek ve işlemlerine devam edebilecekler.

Bu projeye başlamadan önce genel bir sunum amacı ile aşağıda belirtilen teknolojileri kullanarak bir demo uygulama yapılması isteniyor. Demo uygulama da **ASP.NET WebAPI ile Token Based Authentication** ile kimlik doğrulama yapacak şekilde bir yapı oluşturulması gerekiyor.

**Web Api’de 3 tane method yazılması gerekiyor**. Birinci method Web Api’de herhangi bir method’da authenticate olabilmesi için Token alabileceği method.

**İkinci method kullanıcı ekleme işlemi**. Bu kayıt ekleme işleminden sonra sistemin bir log numarası döndürmesi gerekiyor kullanıcı (Guid). **Üçüncü method ise bu log numarası ile sistemde kayıt işleminin ne durumda olduğunu gösteren method.**

**1. method hariç sisteme gelen her request’in mutlaka Token kontrolünden geçmesi gerekiyor**. **Token valid değilse sistemin hata vermesi gereklidir.**

**2. method’a gelen bir request olduğu zaman bu request’e özel bir log numarasının sistemin üretimesi gereklidir (Guid). Gelen request öncelikle veritabanında Request tablosuna kaydedilip, RabbitMQ’da Kullanıcı kuyruğuna eklenmelidir. Request tablosundaki Status alanı eğer herhangi bir hata yoksa QUE olmalı, eğer bir hata varsa ERR olmalıdır ve detaylı hata mesajı Response alanına yazılmalıdır.**

Web Api dışındaki yazılacak olan Listener RabbitMQ’daki Kullanıcı kuyruğu FIFO kuralına göre dinlemesi gereklidir. Kuyruktan gerekli isteği aldıktan sonra, istek kuyruktan silinmeli ve bu istek veritabanında Kullanıcı tablosuna kaydedilmelidir. Request tablosundaki Status alanı eğer herhangi bir hata yoksa CMP olmalı, eğer bir hata varsa ERR olmalıdır ve detaylı hata mesajı Response alanına yazılmalıdır. Status CMP durumunda tüm Kullanıcı tablosu JSON olarak dönüştürülüp Response alanına yazılmalıdır.

3. method’dan bir Response nesnesi dönmelidir. Gelen log numarasına göre veritabanındaki Request tablosu sorgulanmalıdır. Sorgu sonucunda Status ve Response alanlarına göre bir Response nesnesi dönmelidir.

*Proje Özeti;*

* Web Api - Token Based Authentication ile token alınacak ve bu token diğer 2 method’da mutlaka kullanılacak (api/account/token),
* Web Api - Kullanıcı Ekleme (api/user/insert),
* Web Api – Log Sorgulama  (api/log/get),
* Windows Service – RabbitMQ kuyruğunu dinleyip, işleme başladığı kayıdı kuyruktan silip, işlem yapacak.

*Proje Gereksinimleri;*

* Web Api – .Net Framework 4.7 ve üzeri veya ASP.NET Core 2.0 ve üzeri,
* RabbitMQ – Güncel Versiyon ve tüm varsayılan ayarları ile,
* Listener – Windows Service veya Console Application - .Net Framework 4.7 ve üzeri veya ASP.NET Core 2.0 ve üzeri,

*Veritabanı;*

* Microsoft SQL Server 2014 ve üzeri veya MongoDB