

## Gittigidiyor Spring Bootcamp Homework 04

Aşağıdaki iş kuralı senaryosuna göre bir önceki ödevlerde hazırladığınız Spring Boot projesini aşağıdaki sorulara göre tekrar düzenleyiniz.

| ENGLISH  | TÜRKÇE  |
|--|---|
| <b>School Management System</b> <ul style="list-style-type: none"><li>A <b>Student</b> who has a <b>name</b>, a <b>birth date</b>, an <b>address</b>, and <b>gender</b> could take <b>zero or more courses</b>.</li><li>Each <b>course</b> has a <b>course name</b>, a <b>course code</b>, and a <b>credit score</b>.</li><li><b>Instructors</b> who have a <b>name</b>, an <b>address</b> and a <b>phone number</b> could instruct <b>at least one or more courses</b>.</li><li>A <b>course</b> could be instructed by <b>only one instructor</b>.</li><li>There are 2 types of <b>instructors</b>; “<b>visiting researcher</b>” or “<b>permanent instructor</b>”.</li><li>A <b>visiting researcher</b> takes an <b>hourly salary</b>, but a <b>permanent instructor</b> takes a <b>fixed salary</b>.</li></ul> <p>Example:<br/><b>Student</b> → Class candidate<br/><b>name</b> → variable candidate<br/><b>zero or more</b> → cardinality candidate</p> | <b>Okul Yönetim Sistemi</b> <ul style="list-style-type: none"><li><b>Adı</b>, <b>doğum tarihi</b>, <b>adres</b> ve <b>cinsiyeti</b> olan bir <b>öğrenci</b> <b>hiç ders almaz veya bir yada birden fazla ders</b> alabilir.</li><li>Her <b>dersin</b> bir <b>ders adı</b>, <b>ders kodu</b> ve <b>kredi puanı</b> vardır.</li><li><b>Adı</b>, <b>adres</b> ve <b>telefon numarası</b> olan <b>eğitmenler</b> <b>en az bir veya daha fazla ders</b> verebilir.</li><li>Bir <b>kurs yalnızca bir eğitmen</b> tarafından verilebilir.</li><li>2 tür <b>eğitmen</b> vardır; “<b>Misafir araştırmacı</b>” veya “<b>daimi eğitmen</b>”.</li><li><b>Misafir araştırmacı</b> <b>saatlik maaş</b> alır, ancak <b>daimi eğitmen</b> <b>sabit maaş</b> alır.</li></ul> <p>Örnek:<br/><b>Öğrenci</b> → Java sınıfı (model) adayı<br/><b>isim</b> → değişken adayı<br/><b>en az bir veya daha fazla</b> → azlık/çokluk adayı</p> |

**SORU-1 :** Controllor katmanında request ve response nesnesi olarak “Data Transfer Object (DTO)” kullanılacaktır.

**SORU-2 :** DTO ve Entity dönüşümleri (mapping) Mapstruct API kullanılarak gerçekleştirilecektir.

**SORU-3 :** Aşağıdaki yeni iş kurallarına uygun gerekli controller/service/repository düzenlemeleri yapılacaktır;

- Öğrenci yaşı 18’den küçük, 40’tan büyük olamaz. Aksi halde **StudentAgeNotValidException** hatası fırlatılır.
- Sistemde aynı telefon numarasına sahip 2 ayrı eğitmen bulunamaz. Tanımlanması durumunda **InstructorAlreadyExistException** hatası fırlatılır.
- Sistemde aynı ders koduna sahip 2 ayrı ders olamaz. Tanımlanması durumunda **CoursesAlreadyExistException** hatası fırlatılır.
- Bir dersi aynı anda en fazla 20 öğrenci alabilir. Aksi halde **StudentNumberForOneCourseExceededException** hatası fırlatılır.
- Tüm hata durumları, Global Exception Handler sınıfı ile yakalanarak uygun **status code** ve **message** ile oluşturulacak ve response olarak dönecektir.

**SORU-4 :** Soru-3'te belirtilen hatalar, fırlatılma tarihi ve hata mesaj bilgisi ile birlikte database içerisinde loglanacaktır (ayrı bir tablo oluşturunuz). Ayrı bir endpoint hazırlanacak ve loglanmış geçmiş hata bilgileri, hata tipi ve/veya tarihe göre sorgulanabilecektir.

**SORU-5 :** Tüm entitylerin create ve modify zamanlarını database üzerinde tutacak şekilde gerekli düzenleme yapılacaktır.

**SORU-6 :** Tüm sistemin fonksiyoneliyetini ve iş kurallarını açıklayacak şekilde bir **README.md** dosyası projeye eklenecektir (İstenirse ekran resimleri eklenebilir). Kullanılan metodlara **javadoc** standartları çerçevesinde commentler eklenecektir.

**ARAŞTIRMA KONUSU :** JVM nedir ve nasıl çalışır? Mimarisi nasıldır? Java 8 ile birlikte JVM mimarisinde ne gibi değişiklikler olmuştur?

**MEET-UP TAVSİYESİ :** Trendyol ile Spring Native & GraalVM'i tanıyoruz ([Link](#)) – [02 Eylül 2021 Perşembe 18:30 - 19:30]