

IADI Sample Exam

October 26, 2015

To answer the following questions, consider the context of an e-commerce application for selling computers. A computer is characterized by its serial number, brand, model, price, and technical features (e.g. CPU, hard drive). Each feature has a serial number, a name, a description of the feature, and a description of the characteristic.

You should answer using Java code under the Spring Boot framework, or IFML/WebRatio notation, whenever applicable.

1 Define the necessary model(s) to denote a computer. Note you don't need to define the corresponding methods, unless you fill the need to do so.

Possible Answer.

```
@Entity
public class Computer {
    @Id
    private long id;
    private String serialNumber;
    private String brand;
    private String model;
    private float price;

    @ManyToMany(mappedBy="iAmUsedInThisComputers")
    private List<Feature> features;
}
```

```
@Entity
public class Feature {
    @Id
    private long id;
    private String serialNumber;
    // e.g. "CPU"
    private String name;
}
```

```

    // e.g. "This features represents the a CPU"
    private String featureDescription;
    // e.g. "I7 @ 3Gz"
    private String characteristic;
    // e.g. "The fastest CPU ever from Intel"
    private String characteristicDescription;

    // this will help searching for all computers with this feature
    @ManyToMany(mappedBy="features")
    private List<Computer> iAmUsedInThisComputers;
}

```

2 Having in mind the typical URLs of a RESTful application, define the mapping for such URLs and corresponding controllers, by implementing such controllers, to:

- create computers,
- to add features to it, and
- to obtain the list of computers in a json representation.

Use the models created in the previous question. Assume there is a repository for computers called `computerRep`. Name the views as you see fit.

Possible Answer.

```

@RequestMapping(value="/computers", method=RequestMethod.POST)
public String addNewComputer(Model model, Computer c) {
    computerRep.save(c);
    model.addAttribute("computers", computerRep.findAll());
    return "computers";
}

```

```

@RequestMapping(value="/computers/{id}/features", method=RequestMethod.POST)
public String addNewFeature(Model model, @PathVariable("id") int id, Feature f) {
    Computer c = computerRep.findOne(id);
    c.addFeature(f);
    computerRep.save(c);
    model.addAttribute("computer", c);
    return "computers/show";
}

```

```

@RequestMapping(value="/computers", method=RequestMethod.GET, produces={"text/plain"},
public @ResponseBody Iterable<Hotel> hotelsAsJson(Model model) {

```

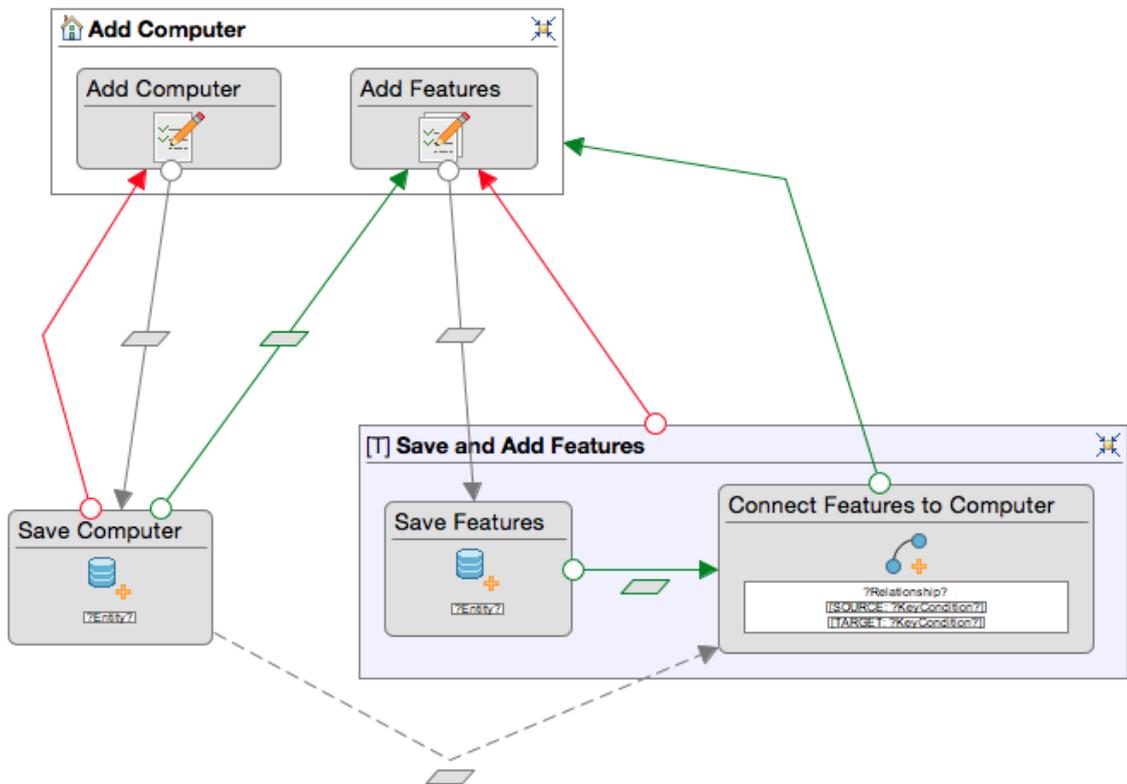
```

    return computerRep.findAll();
}

```

3 Create the IFML/WebRatio specification for the use case where a store administrator inserts a new computer in the system to be sold. Besides the specification, you should describe it as well as you can.

Possible Answer.



Note your need to further detail form fields, binding arguments, etc.

4 Consider the following HTML/ajax/jQuery fragment:

```
$.post("computers/12", {computer: computer})
```

Implement the controller to answer to this request.

Possible Answer.

```

@RequestMapping(value="/computers/{id}", method=RequestMethod.POST)
String addNewFeature(Model model, @PathVariable("id") int id, Computer c) {
    Computer cm = computerRep.findOne(id);
}

```

```
c.setFeatures(cm.getFeatures());
computerRep.save(c);
model.addAttribute("computer", c);
return "computers/show";
}
```