

MTAT.03.229 – Enterprise System Integration

Resit Exam – 17 June 2016

(100 points)

Notes:

- The exam is open-book and open-laptop. Web browsing is allowed
- You are not allowed to communicate with anyone during the exam in any way (except with the lecturer).
- For the solution to Part 1 you must create a Bitbucket repository and push there your code (as often as possible). Invite me as collaborator to such repository. The solution to Part 2 can be submitted on paper or electronically. When electronically, you must include all files (e.g. PDF, DOC or DOCX, MD). If you prefer, you can submit a single ZIP file containing the solutions to part 1 and part 2.
- If you find that there is not enough information in the text below to answer a given question, and you need to make additional assumptions, please write down your assumptions together with your answer.

Part 1. RentIt: Proposing alternatives when a requested plant is not available

In our scenario, a Purchase Order can be rejected by RentIt whenever the requested plant is not longer available for rental (e.g. the plant has been assigned to another customer or the plant has been reported with a failure). Your task will be to modify the classes in the domain model of RentIt to support the following:

- A given plant can be considered as an alternative replacement for other plants (e.g. the plants are of the same nature, with similar work capacities, and similar price; the main difference could be the brand of the plant)
- A purchase order should hold a reference to an eventual replacement. Moreover, when RentIt proposes a replacement to a customer, the purchase order should be considered as `WAITING_CUSTOMER_APPROVAL`.
- For simplicity, we will assume that the purchase order total is computed with the minimum daily rental price of the prices of the originally requested plant and the proposed replacement.

Tasks:

- [15 points] Update RentIt's domain model associated with the bounded context "inventory"
- [15 points] Update RentIt's domain model associated with the bounded context "sales"
- [10 points] Update RentIt's REST adapter and application services to support the notion of plant replacement
- [10 points] Write a sample JSON request/response for
 - Creating a Purchase Order for the case where the RentIt proposes an alternative plant (e.g. the requested plant was not longer available)

- Approving the proposed replacement of a requested plant by the customer

The idea is that the JSON request is the one that you would use with Postman and the response is also what you expect Postman would display after executing the invocation to the REST API

Part 2. RESTful API for CAD simulations

Engineers at automobile design company CaaS routinely run simulations of new automobile designs or automobile component designs in order to study their mechanical properties. Engineers design the automobile or automobile components using a Computer-Aided Design (CAD) tool. The design is exported as a CAD file and uploaded to the main server of the company's computer cluster via an FTP client. Engineers then login to this main server and schedule their simulation on the computer cluster using a command-line tool (the command-line tool is a Python script). Simulations may take between one and 12 hours depending on the complexity of the design (average of 6 hours). However, since the capacity of the computing cluster is limited, it takes several hours between the moment a simulation is scheduled by an engineer and the moment it starts. To monitor the status of a simulation, engineers need to login to the server and use another command-line tool. Engineers can also cancel a simulation using yet another command-line tool. Finally, engineers get an e-mail when their simulation is completed and they need to login to the main server to retrieve the file containing the simulation outputs. They then load this file into their CAD tool in order to analyze the results. Around 20 simulations per day are executed on CaaS's computer cluster. CaaS has 20 engineers who routinely run simulations and another 20 engineers who run simulations occasionally.

Several problems have been identified with the current way of running simulations. First, the process is cumbersome for engineers, and it is difficult to teach it to new engineers who join the company. As CaaS is foreseeing a significant expansion in the next year, and plans to double its number of engineers, it will become crucial to make this process simpler. Also, given this planned expansion of the company, it is clear that the computer cluster is reaching its limits and scaling up the cluster would be expensive. To scale up, CaaS is planning to rent servers from cloud computing providers on a per-hour basis, to run some of the non-critical simulations (simulations of the most critical vehicle models would still be done in the local computer cluster). The fact that external cloud-based servers will be used will make the procedure for starting and monitoring simulations more complex and engineers are against this additional complexity, given that it takes away time from them that could be used for more productive purposes. Finally, a third issue with the current simulation process is that no history of past simulations is kept. Sometimes one engineer runs a simulation, fetches the results into his/her laptop, and when another engineer wants to inspect the results of this simulation, it is difficult for them to retrieve these results. It has become evident that engineers need a searchable and browsable archive of all previous simulations.

In parallel, IT managers intend to start internally billing (also known as "IT chargeback") on a monthly basis for completed simulations, in order to offset growing computing costs. The cost of a completed simulation will be proportional to its duration, and will depend on whether the simulation is "critical" or "non-critical".

To address the above needs, you are asked to design a RESTful Application Programming Interface (API) for a simulation environment to be deployed in the main server of the computer cluster. Engineers will be able to start, monitor and retrieve the output of their simulations directly from their CAD tool by means of a special CAD tool plugin that will be developed for them. This plugin will be developed in Java Standard Edition (Java SE), which is the plugin development technology supported by the CAD tool. The plugin will interact with the server-side simulation environment via HTTP operations, using JSON as the resource representation format.

Given the inherent complexity of a CAD file, its content is not directly as part of the JSON file to be used in the RESTful interactions. Instead, the CAD files are stored in a “document management system” (DMS). The DMS associates a URL to each CAD file, which can then be used for retrieving a given CAD file by using a HTTP GET request on the corresponding URL. Simulation results will also be stored in the DMS – the DMS provides operations for pushing files into it.

You are requested to design an API for the simulation environment. The interface should provide all operations that the CAD tool plugin would need to invoke to create and manage simulations on the server-side simulation environment. This includes the ability to create a simulation; to cancel a scheduled or a running simulation; to retrieve the current state of simulations (e.g. scheduled, running, completed, ...); to view a simulation (including the link to the simulation output if completed); and to query the full history of created simulations.

Task 2 [7.5 points]. Design a domain model for the CAD simulation platform based on the above description.

Task 3 [7.5 points]. Based on the domain model, design a *resource model* using the RESTful service design method presented in the lectures.

Task 4 [15 points]. Specify a state machine capturing the lifecycle of a simulation.

Task 5 [20 points] Specify an apiary blueprint of the RESTful interface that the CAD simulation platform will provide to its users. For each operation on a resource, you should specify an example of a request/response.