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 grid 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 between 2 and 6 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 12 engineers who routinely run simulations and another 10 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 capacity of the computer cluster will quickly reach 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 less 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.

**Task 1 [35 points]** To address the above problems, 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.

Concretely, you should specify an interface for the simulation environment. The interface should provide all operations that the CAD tool plugin would need to invoke to start and manage simulations on the server-side simulation environment. This includes the ability to retrieve the full history of past simulations and the ability to retrieve the history of past simulations filtered by one or both of the following criteria:

- The time window when the simulations were run (captured by a start date and an end date of the time window)
- A keyword, which may be the simulation’s unique identifier, part of the simulation’s name or part of the simulation’s description given by the engineer who started the simulation.

For each operation provided by the simulation environment you need to specify:

- The name of the operation
- A brief explanation of what the operation does
- A list of input and output parameters of the operation. For each parameter, give a name, a brief explanation of the meaning of the parameter, whether the parameter is “required” or “optional”, and a description of how the parameter will be encoded, i.e. whether the parameter will be transferred using URL tunnelling (form-encoded) or in the body of the request/response, and in the latter case, what Internet media type (Content-type) will be used.
- The relative URL where the operation will be available.
- Whether the operation uses the GET, POST, PUT or DELETE method.
- A list of HTTP status codes that the operation may return and for each code, 1 or 2 sentences explaining under what conditions this status code is returned.

**Task 2 [15 points]**

Given the API specification you wrote for Task 1, write the skeletons of a set of Java classes to implement this specification using JAX-RS and JAXB. Skeletons should be given for Java classes corresponding to resources (JAX-RS) as well as for Java classes corresponding to documents (JAXB). The skeleton of a class should include:

- The class name and JAX-RS or JAXB annotations associated with this class
- For each method, its name, input parameters and return type, JAX-RS or JAXB annotation associated with the method and JAX-RS or JAXB annotation associated with each parameter (where applicable).
- For each attribute, its name, type and JAXB annotation where applicable.

You do not need to specify method bodies.

Alternatively, if you prefer, you can give the class skeletons (with equivalent information as above) in C# instead of Java.