DESCRIPTION OF THE PROJECT TASK FROM THE OF INFORMATION SYSTEM INTEROPERABILITY

According to the given topic (REST API URL) for the project task implement a "backend" and "frontend" system that implements the following functionalities:

1. A REST API interface that includes a service (endpoint) that will be called by the POST method and send an XML file. The XML file must contain arbitrary data for the entity that is bound to the domain of the default REST API interface. The default entity must first be validated, check whether all default data are correct using XSD file validation, and only then save it in the system. In case of errors, it is necessary to display validation errors to the user. (LO2 – 5 points, LO3 – 2 points, LO5 – 4 points, LO7 – 2 points)
2. A REST API interface that includes a service that will be called by the POST method and send an XML file. The XML file must contain arbitrary data for the entity that is bound to the domain of the default REST API interface. The default entity must first be validated, check if all default data are correct using RNG file validation, and only then save it in the system. In case of errors, it is necessary to display validation errors to the user. (LO2 – 5 points, LO3 – 2 points, LO5 – 4 points, LO7– 2 points)
3. A SOAP interface that includes a service that receives a term by which to search for an entity. Before that, an XML file must be generated on the "backend" containing data retrieved from one of the REST API methods according to the given topic. The entered term, which is the input data of the SOAP method, must be used to filter only those records that match the given term with the help of XPath and the prepared XML file, and return them as a result of the SOAP method call. (LO3 – 2 points, LO5 – 4 points)
4. Using JAXB, check the prepared file from the previous step to see if it complies with the default XSD file and return validation messages if the data on the XML file is not valid. (LO2 – 4 points, LO7 – 2 points)
5. Create an XML-RPC server application that, using DHMZ (https://vrijeme.hr/hrvatska_n.xml), will enable retrieving the current temperature according to the given city name or part of the city name. If there are multiple entries that match part of the city name, they should all be printed. (LO3 – 2 points)
6. Use the REST API with the given topic and implement the calling of at least one service (endpoint) and print the results of the call for the given input parameters or implement your own REST API interface with JWT tokens (access and refresh) that will enable calling all four REST API methods (GET, POST, PUT and DELETE) over the selected entity. (LO3 – 2 points, LO4 – 12 points, LO5 – 10 points, LO6 – 2 points, LO7 – 6 points)
7. Write a client desktop or web application (Java or C#) that will contain a graphical interface and enable users to call the service from the first six steps. (LO1 – 2 points, LO3 – 2 points)