**Instructions**

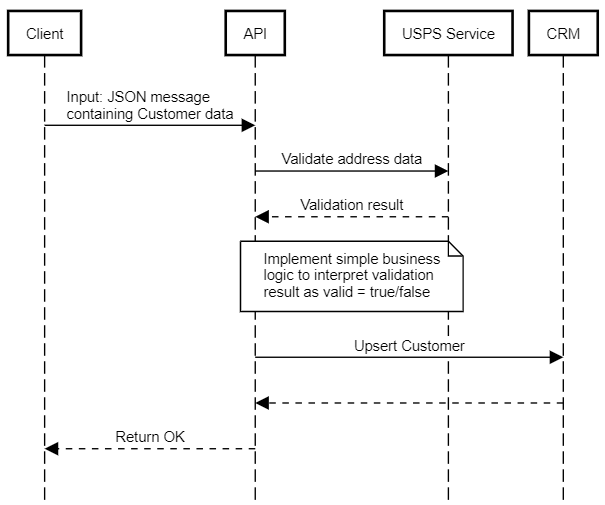
Please feel free to make reasonable assumptions on implementation details that have not been specified and document assumptions made along with rationale for the decision. If you run into any blocking issues or have questions, please email the Minitab hiring manager for clarification if time permits otherwise document the issue(s) and proceed with implementing remaining requirements.

Please upload source code and any accompanying explanatory documentation to GitHub or any other publicly accessible Git Repo/provider and provide the public repository link to Minitab for review when you are finished. Minitab does not plan to review a published instance of your completed API, and as such there is no need to publish the application publicly; a source repository link is the only expected deliverable.

Best of luck!

**Requirements**

Develop a C# .NET Core or .NET Framework RESTful Web API with one endpoint that accepts a JSON-formatted input (see sample payload below). The responsibility of the API endpoint is to integrate customer information passed from an external client into “CRM” (Fake/Stub repository implementation).



Incorporate simple address validation (boolean – is valid/is not valid only) into the endpoint’s business logic using a freely available web API like USPS address validation services (<https://www.usps.com/business/web-tools-apis/welcome.htm#developers>) to ensure only valid addresses are saved to CRM with customer information. If a valid address was not provided, the customer record should be saved without an address.

Include at least 3 automated test cases with your solution.

Hint: Favor a simple solution that showcases your knowledge of development concepts/techniques. Exhaustive error handling and validation is not necessary or expected. This project is expected to take 2-4 hours to complete.

**Sample Input**

{

"CustomerName": "Fred Flintstone",

"CustomerEmail": "fred.flintstone@bedrock-llc.com",

"Address": {

"Line1": "10 South LaSalle St.",

"City": "Chicago",

"State": "IL",

"PostalCode": "60603",

"Country": "US",

}

}

Hint: You do not need to support non-US addresses or additional address lines for the purpose of this exercise.

**CRM Repository Fake/Stub Code**

A sample CRM Fake/Stub interface is provided below. Implementation can be as simple as printing the serialized data to the Console/log stream or as complicated as writing to cloud storage – the choice is yours.

public interface ICrmRepository

{

Task UpsertCustomer(Customer customer);

}