Final Exam

You are a developer for Custom Core Apps (CCA), a software development shop that helps small businesses setup an online presence by building out web applications. One of your clients, Sven Slomsko, has contacted your business analyst about a website your team has been developing for his new pizzeria, Hot Mess Slices. The original deal was that your team would develop the front-end, while he would write the back-end in some COBOL-like language called Natural. As it turns out, this language requires an expensive license to use, and he does not have that kind of money. Worse off, Hot Mess Slices is set to open in a couple days, and he cannot build up hype for the grand opening without a proper back-end. The front-end engineers completed the development of the customer-facing portion of the site the other day, and Sven is begging that you write the back-end for him by whatever means necessary. If you and your team can get it done by tonight, he promises he will throw you a pizza party on the house.

Requirements

The back-end shall be an ASP.NET Core Web API written in C#. It will take requests from your already completed front-end application, and simple Angular app. The Angular app presents a simple menu where the user can order one or more pizzas. This menu includes options to:

* Pick a type of crust
* Pick a type of sauce
* Pick how much cheese will be on a pizza.
* Pick a crust flavor.
* Pick toppings.

When the user has completed filling out what pizza(s) they wish to order, they will be directed to a form where they fill out personal information. This includes input boxes for:

* User first name
* User last name
* Middle initial
* Address 1
* Address 2
* Apt number
* State
* Zip
* Email
* Credit card number to be used on order
* Mobile phone number

When the user submits this info, a JSON payload will be sent to the back-end where the following actions should happen:

* The order and user info needs to be saved to a database.
* A confirmation text with the users order info should be sent to the mobile number provided.

For the sake of simplicity, assume every user that orders from the site is new, even if user information from the form matches earlier entries.

The front-end also will eventually have a page where employees can view a list of all orders and their statuses (Not started, in progress, and completed). This means the back-end will need to have the ability to pull order data from the database and send it to the front-end.

CCA likes to make use of unit tests to ensure code quality, so you will need to write tests that:

* Ensure the order data pull works correctly
* Ensure that saving order/user data works as intended

Remember, unit tests are only supposed to test as few actions as possible per test. Also take note that if you rely on an actual persistent database and/or any infrastructure for your tests, they are technically considered integration tests (which is not what we want).

Sven has an account with Twilo to handle sending texts from the back-end to a cell phone. He will provide you with the token needed to use it, but you can access the docs that discuss configuration here:

<https://www.twilio.com/docs/sms/quickstart/csharp-dotnet-core>

It does not make sense to have 8 programmers working on so few items independently, so you will be all working in tandem using Mob Programming. One person will code while the rest of your team tells them what to write. You will rotate coders every 10 minutes. Time will be kept by your business analyst. If you need clarification on the requirements, ask your analyst (DO NOT MAKE ASSUMPTIONS ON BEHALF OF THE USER)!

A final note: you have no control over the data the front end sends you, so it’s important to ensure you are doing server-side validation. If you find any issues on the front-end that could impact customer experience, bring it to the attention of your business analyst. It will be their job to communicate these issues with the front-end team so they can get the code fixed before deployment.