01 Task

“Ensuring the Acceptance Criteria are met, build a C# Web API that connects to an instance of a database and persists the contents of the Meter Reading CSV file.

We have provided you with a list of test customers along with their respective Account IDs (please refer to Test\_Accounts.csv). Please seed the Test\_Accounts.csv data into your chosen data storage technology and validate the Meter Read data against the accounts.”

Solution

I have created my solution which has got following projects.

* BusinessLayer

All business logic

* DataAccessLayer

To connect with the database. I have used SQL Server database.

Note: Please chabge the connection string in the application Setting.JSON file.

* ClientLanguage

Defined all language here.

* ENSEKCoreAPI

Web API that connects to an instance of a database. Defined all the API methods.

* UnitTestENSEKCore.Tests

It is nUnitTest project.

* ENSEKCoreWebApp

It is front-end web app where you consume WebAPIs.

02 USER STORY

“As an Energy Company Account Manager, I want to be able to load a CSV file of Customer Meter Readings So that we can monitor their energy consumption and charge them accordingly”

Solution

User should be able to load a CSV file.

03 ACCEPTANCE CRITERIA

MUST HAVE

Create the following endpoint:

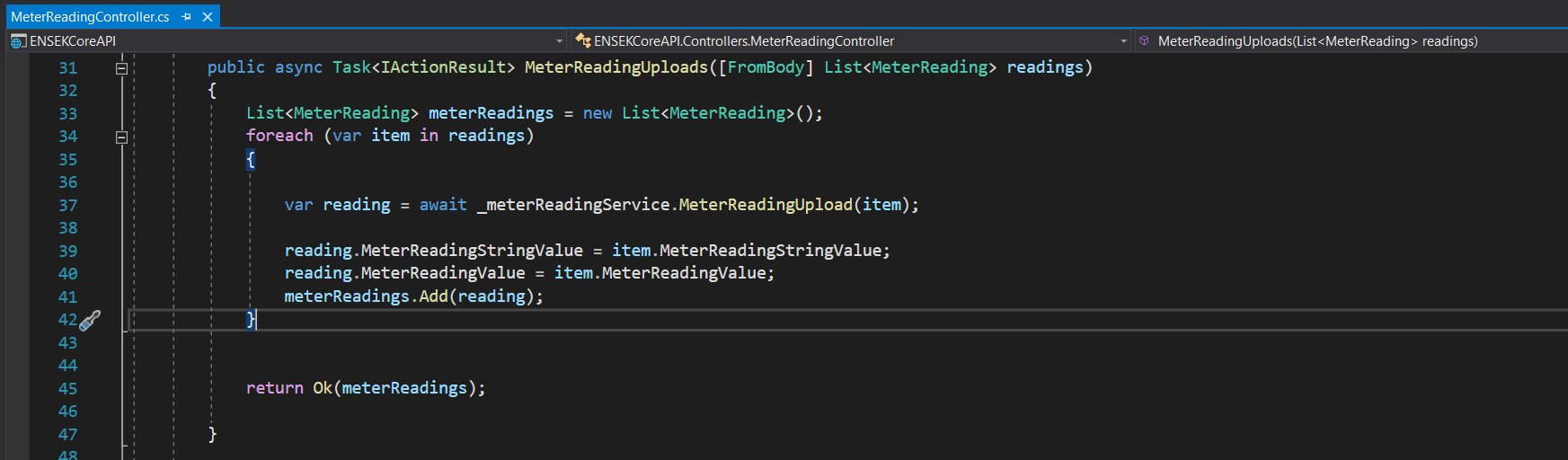
POST => /meter-reading-uploads

Solution

I have create an endpoint => meter-reading-uploads

Project => ENSEKCoreAPI

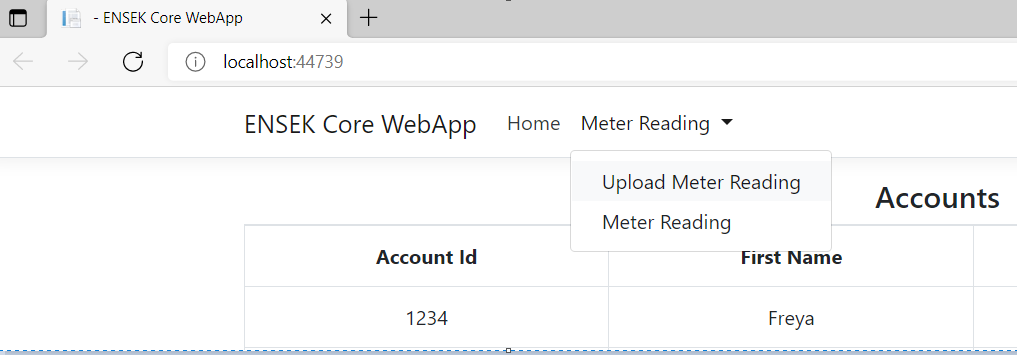
Controllers/MeterReadingController.cs

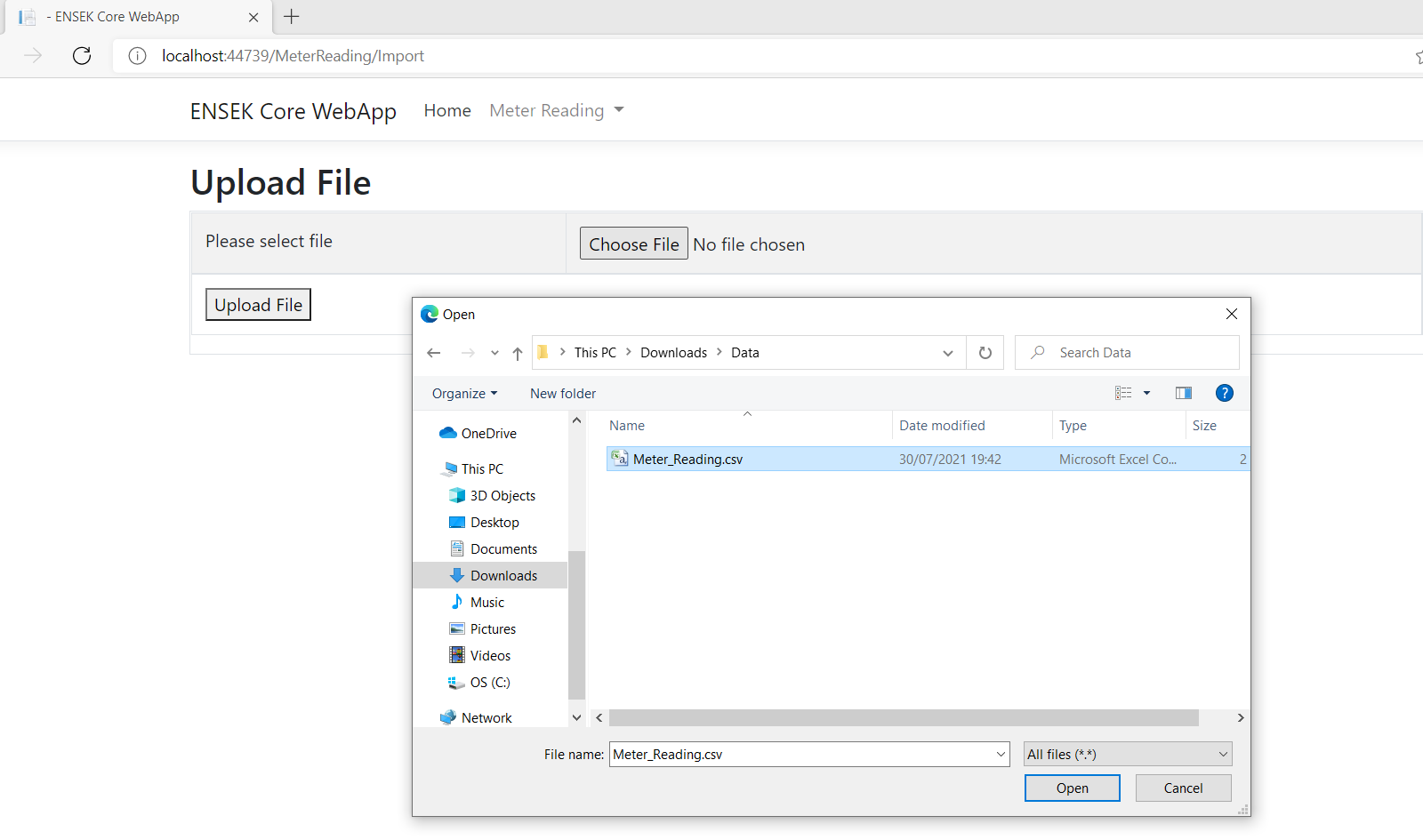


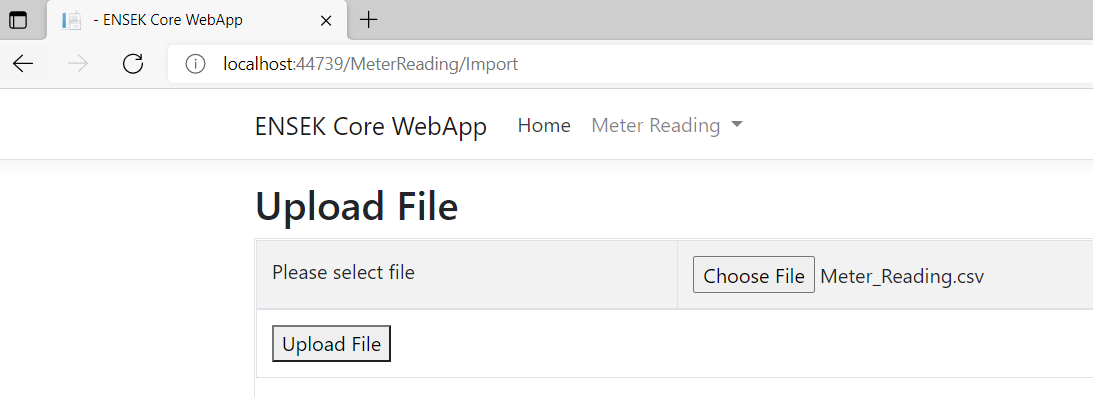
The endpoint should be able to process a CSV of meter readings. An example CSV file has been provided (Meter\_reading.csv)

Solution

* Run ENSEKCoreWebApp
* Go to Meter Reading > Upload Meter Reading



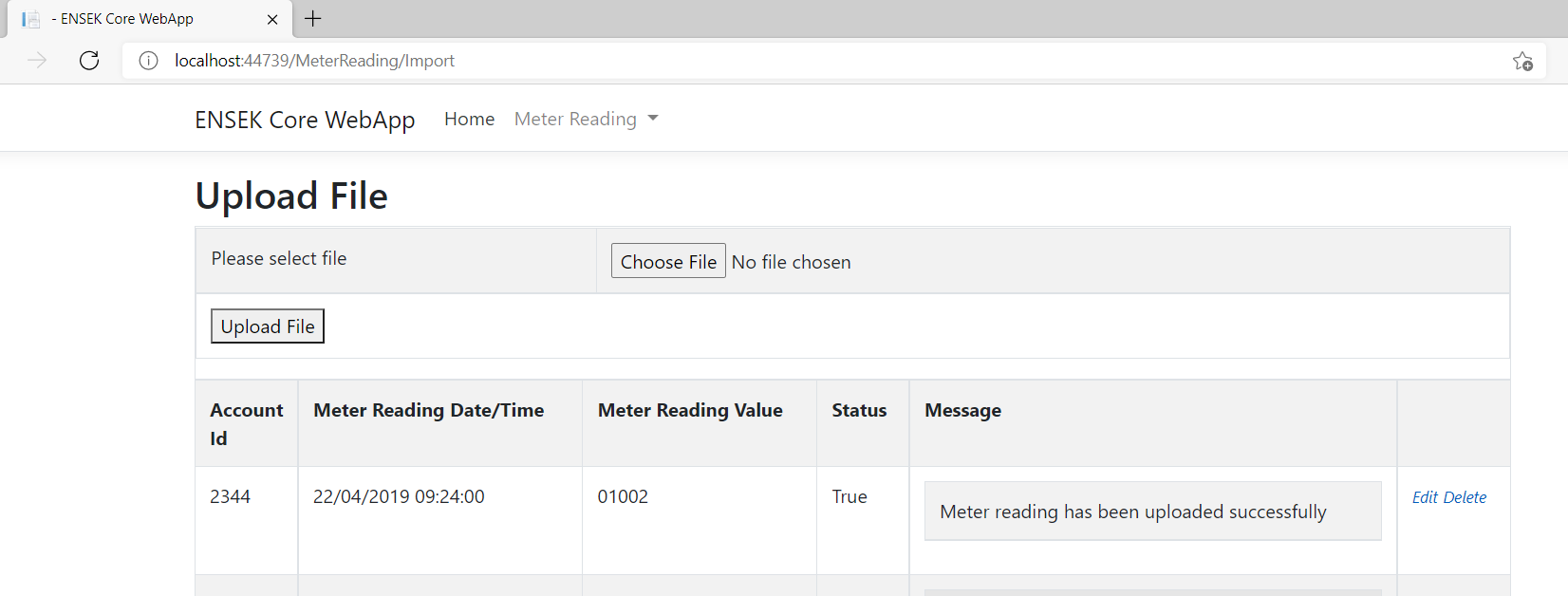




Each entry in the CSV should be validated and if valid, stored in a DB.

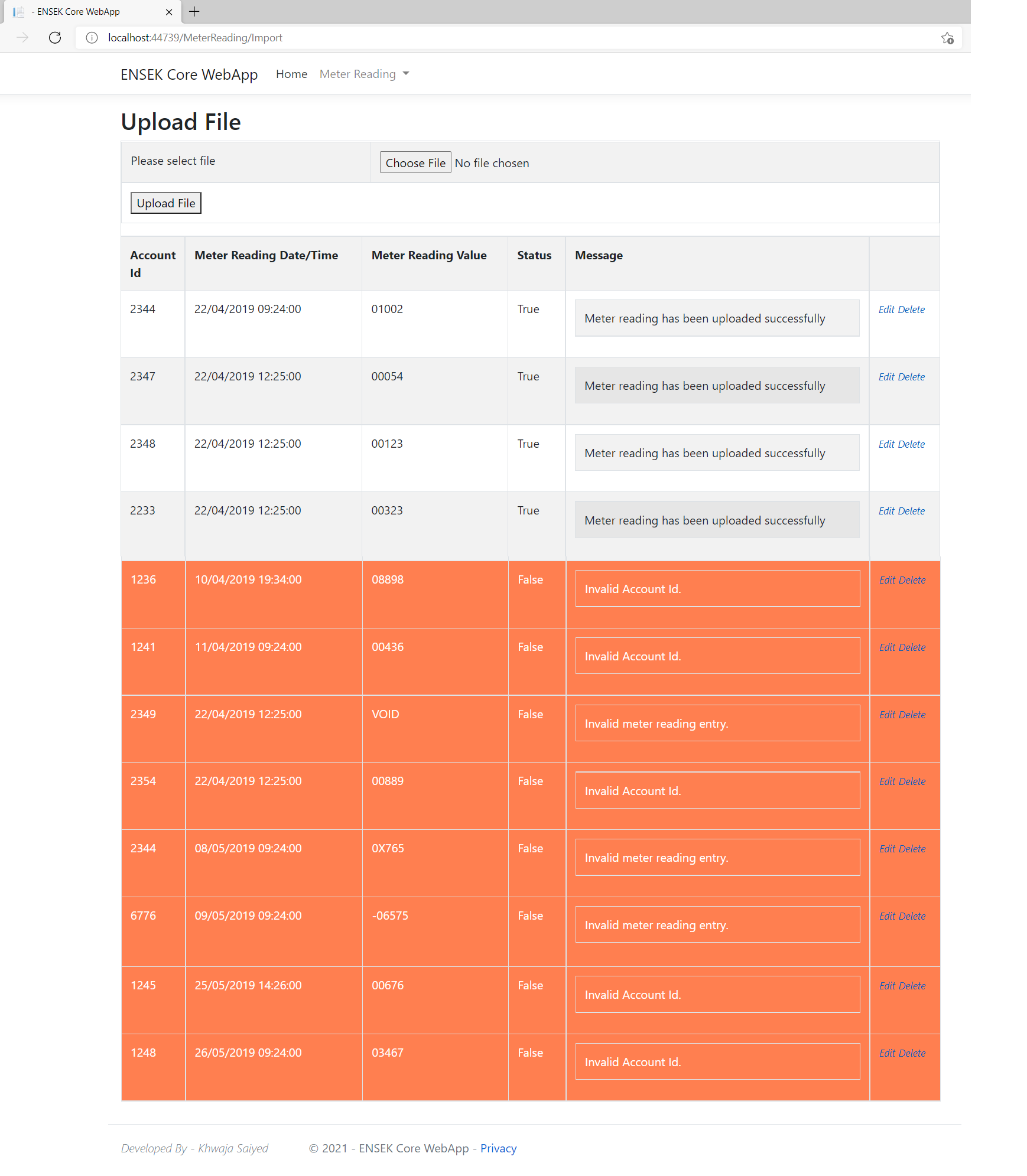
Solution

Upload process should validate each record and stored in a database if there is no issue



After processing, the number of successful/failed readings should be returned.

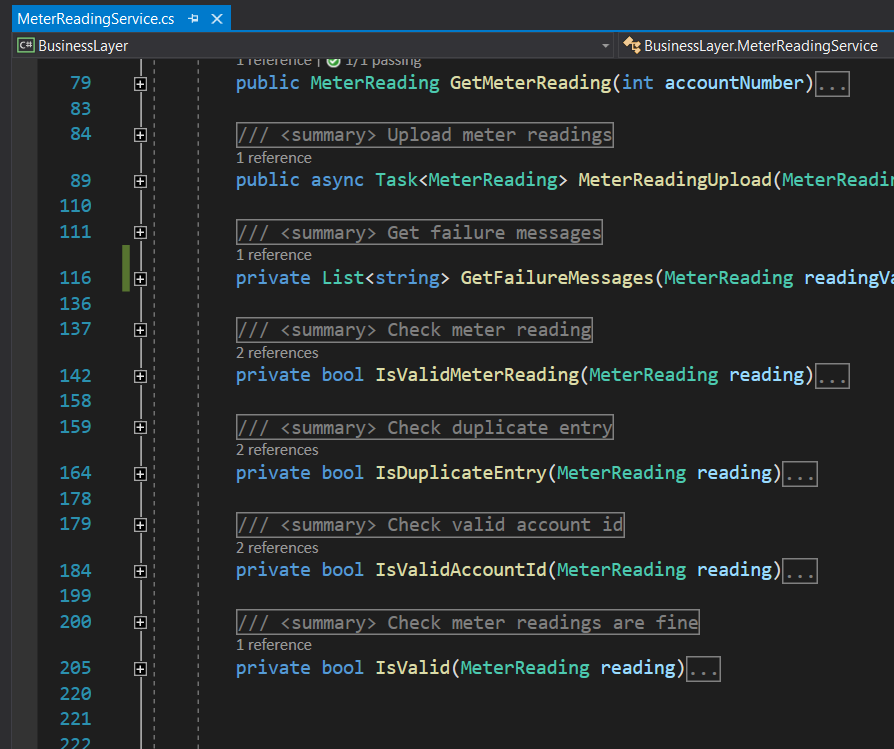
User should be able to see successful/failed messages.



Validation:

Solution

Validations are applied in **BusinessLayer** > **MeterReadingService.cs**

****

You should not be able to load the same entry twice

Upload process will NOT process the same entry twice.

A meter reading must be associated with an Account ID to be deemed valid

Upload process will check valid Account ID

Reading values should be in the format NNNNN

Upload process will make sure reading values are in the format NNNNN

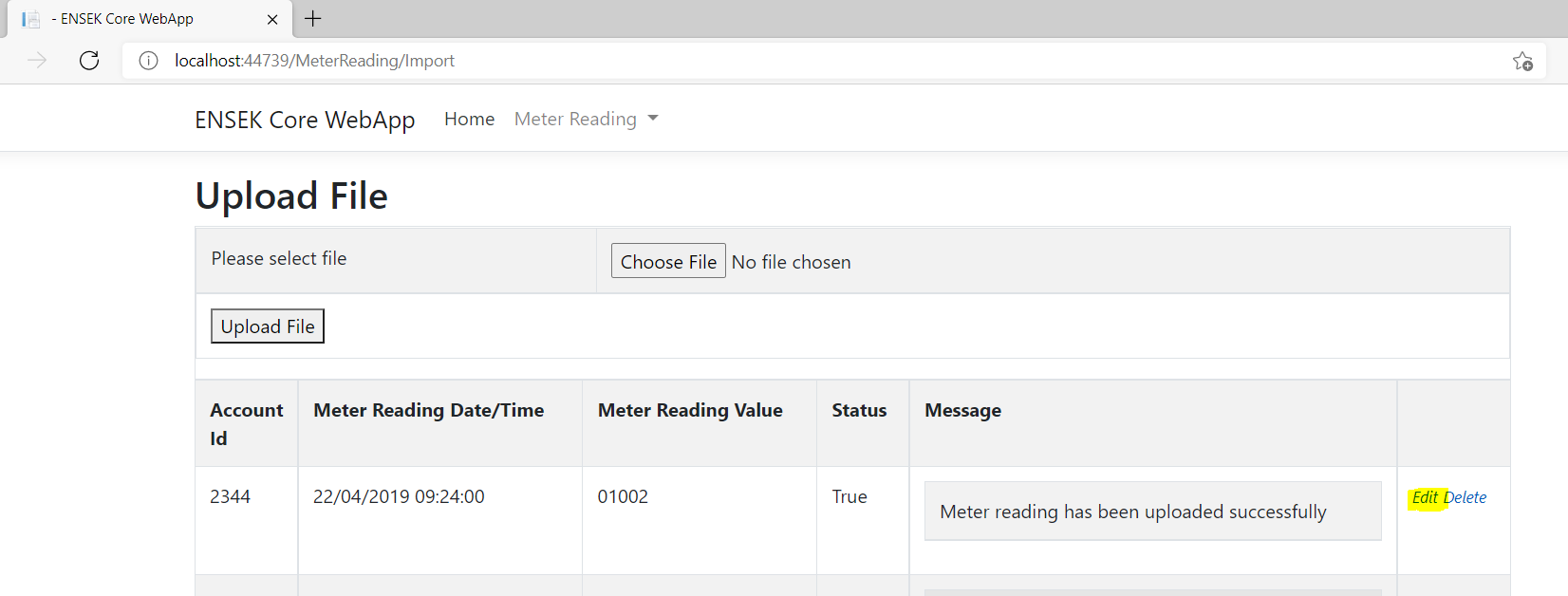
NICE TO HAVE

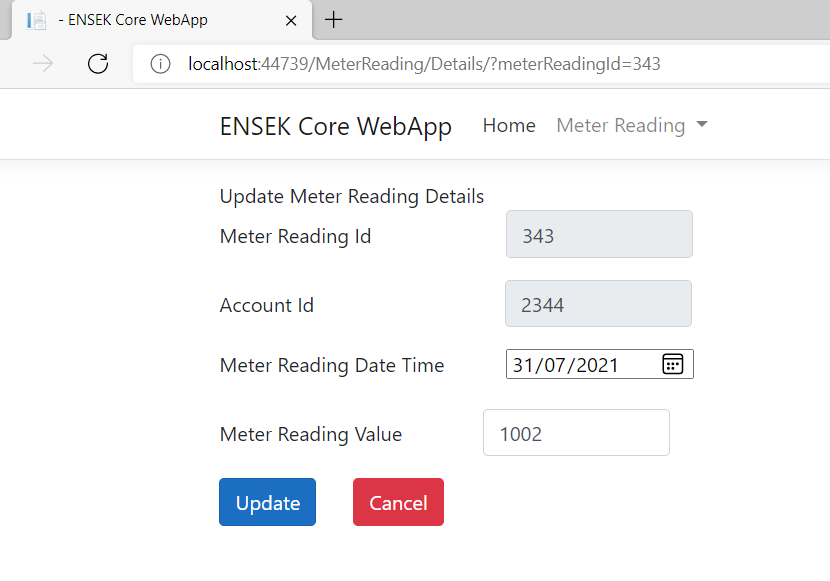
Create a client in the technology of your choosing to consume the API. You can use angular/react/whatever you like

I have used created front-end Web App to used JQuery to consume the API.

Other functionality such as CRUD for accounts/individual meter readings

I have tried to add Edit functionality for accounts & individual meter readings





Meter Reading Value updated from 1002 to 1012.

