Dear Reviewer,

First up, thanks for giving me the opportunity.

**Understanding of the Requirement:**

Need to compute Normal, Min & Max resistance for a Resistor by providing four band colors. So, I have formulated a Resistor Object which has four bands & Resistance Values property (Please refer Core/Models). Also, I have assumed all 4 bands are required fields in order to compute resistance.

**Implementation Details:**

1. **TDD using MSTest & NSubstitute(Mockup Framework – Added from Nuget) –**

* Found in ElectronicCodeCalculator.Tests
* Business/ResistorBusinessTest
* Controllers/ResistorControllerTest

1. **Web Development using ASP.NET MVC & Jquery**
2. **Domain Driven Design using Onion Architecture (**[**Jeffrey Palermo**](https://jeffreypalermo.com/author/jeffreypalermo/)**) –**

Divided into 4 layers as follows:

**ElectronicCodeCalculator.Core**

* Models/Entities, Interfaces, Constants
* Independent & forms the central layer of Architecture
* All other layers would depend on the core.

**ElectronicCodeCalculator.Business**

Implements the Interface defined in Core.

**ElectronicCodeCalculator(Web):**

* UI Controls bound to Core Models.
* Controller Actions calls Business layer to do CalculateOhmValue

**ElectronicCodeCalculator.Infrastructure –**

* Not required in this project as there is no need to interact with Database or Sharepoint or other ORM.

1. **Dependency Injection – Unity Container(Unity.MVC from Nuget)**

* Contoller do not instantiate Business object.
* Instead, unity provides the required instance through **constructor Injection**.
* Required instance would be provided for a requested Interface as per the configuration defined in UnityConfig. RegisterTypes.

1. **Reactive forms using Bootstrap**
2. **TDD Results & sample output found in ElectronicCodeCalculator/** **Results Screenshots**

**Requirement Suggestion:**

**Ideally, my preferred interface definition would be given below as follows:**

public interface IOhmValueCalculator

{

/// <summary>

/// Calculates the Ohm value of a resistor based on the band colors & return a Resistance Object which has Min, Normal & Max Resistance fields.

/// </summary>

/// <param name="ObjResistor">Resistorr object for which Resistance needs to be computed..</param>

Resistance CalculateOhmValue(Resistor objResistor);

}