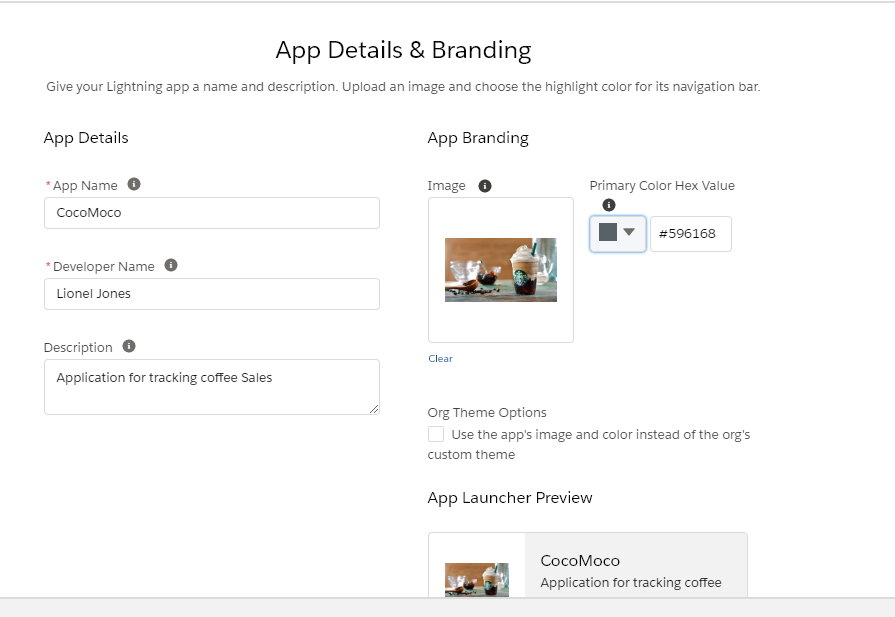
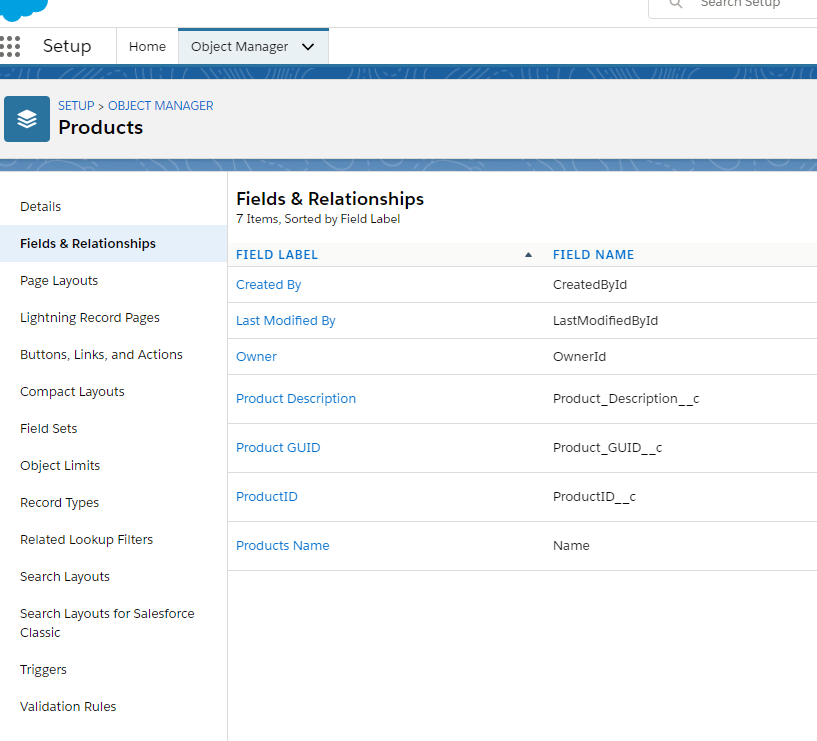
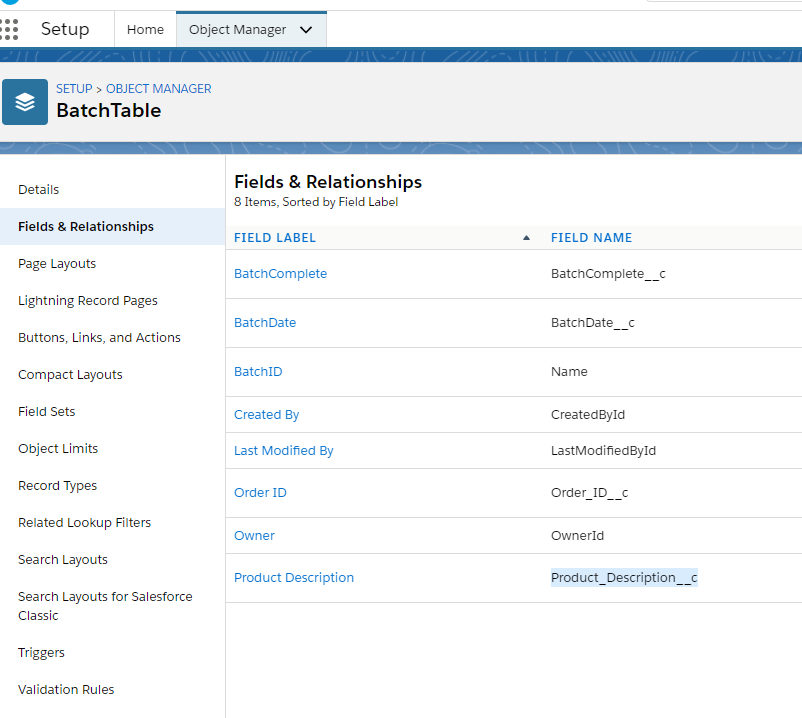
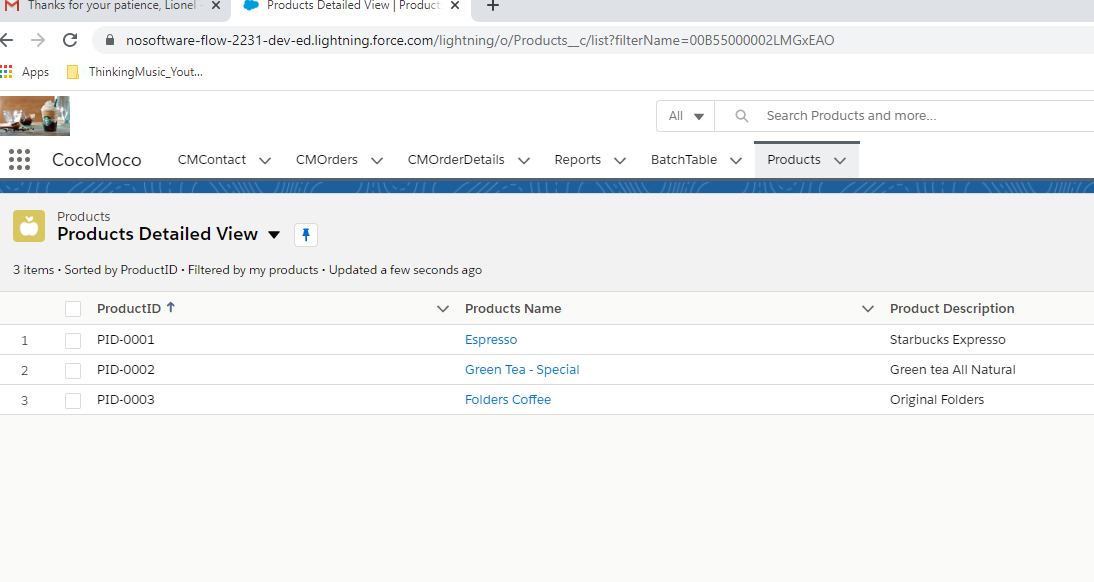
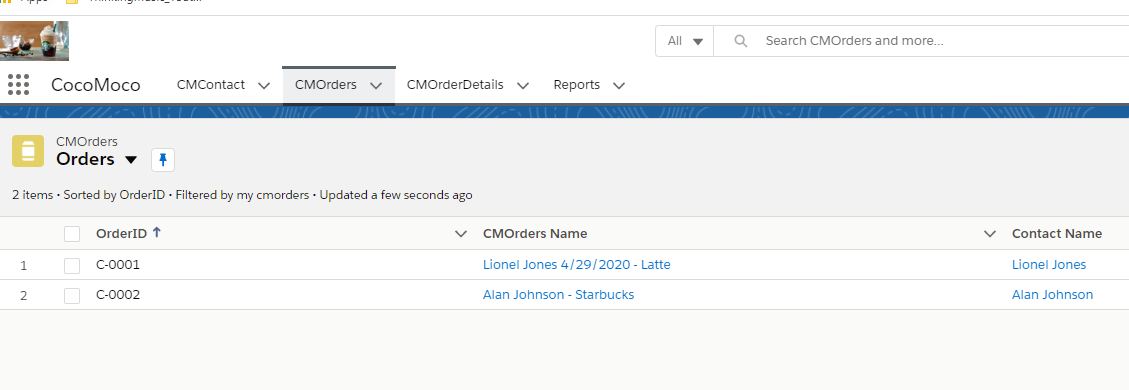
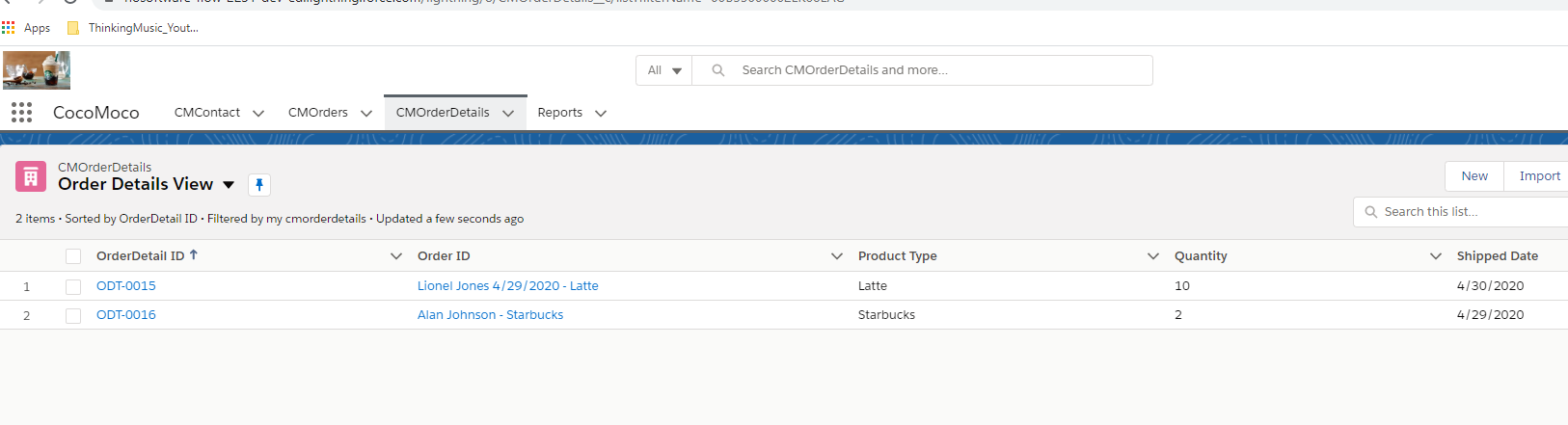
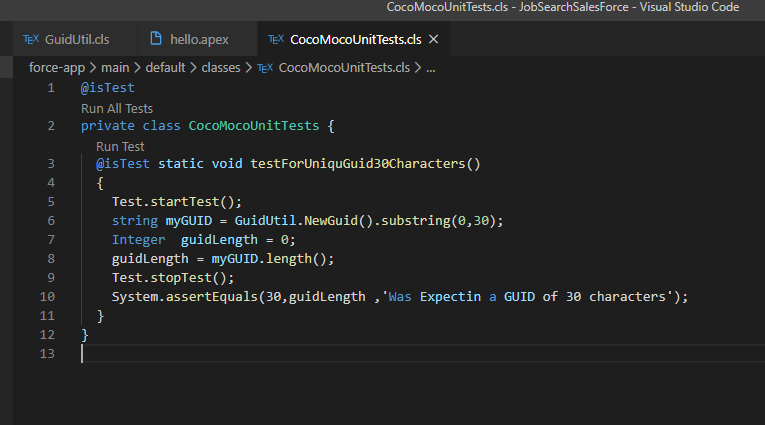
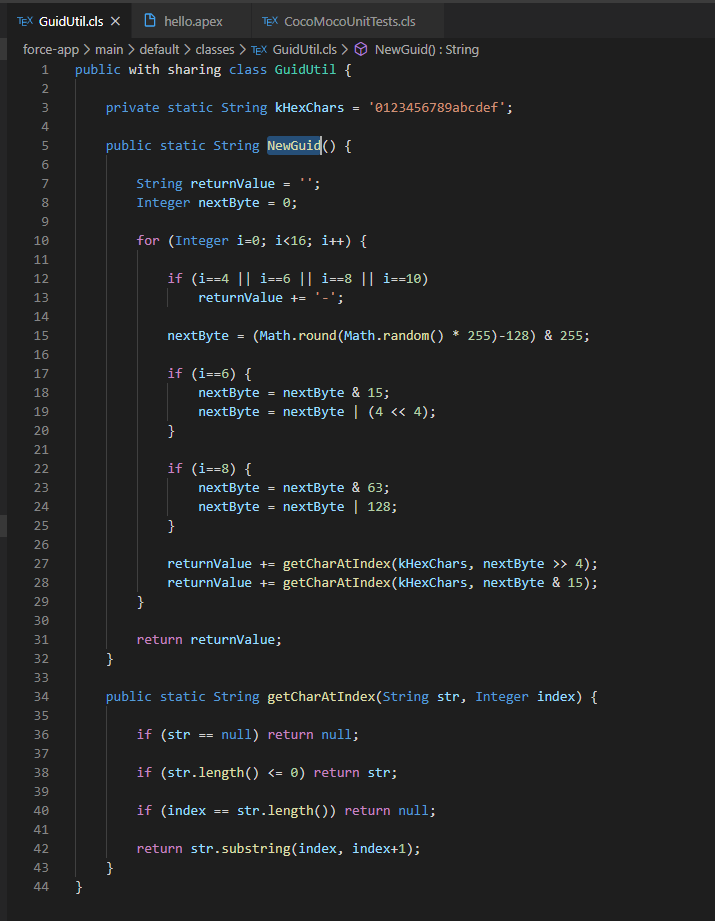
**System Design Specification – CocoMoco Coffee**Date: 4/29/2020  
  
Contributor(s)  
Lionel Jones, Developer  
  
**Business/Functional Requirement:**  
**High Level**CocoMoco is an online coffee roaster, selling coffee direct to consumer. Each bag of coffee has a unique serial number that allows the business to track quality issues and help customers understand the specifics of the roasting and production of their beans.  
  
**Tech Stack:**  
Salesforce – SAAS

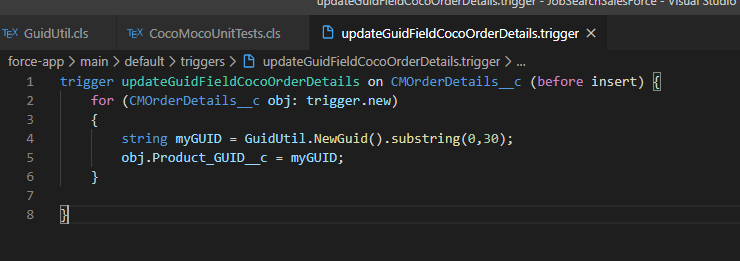
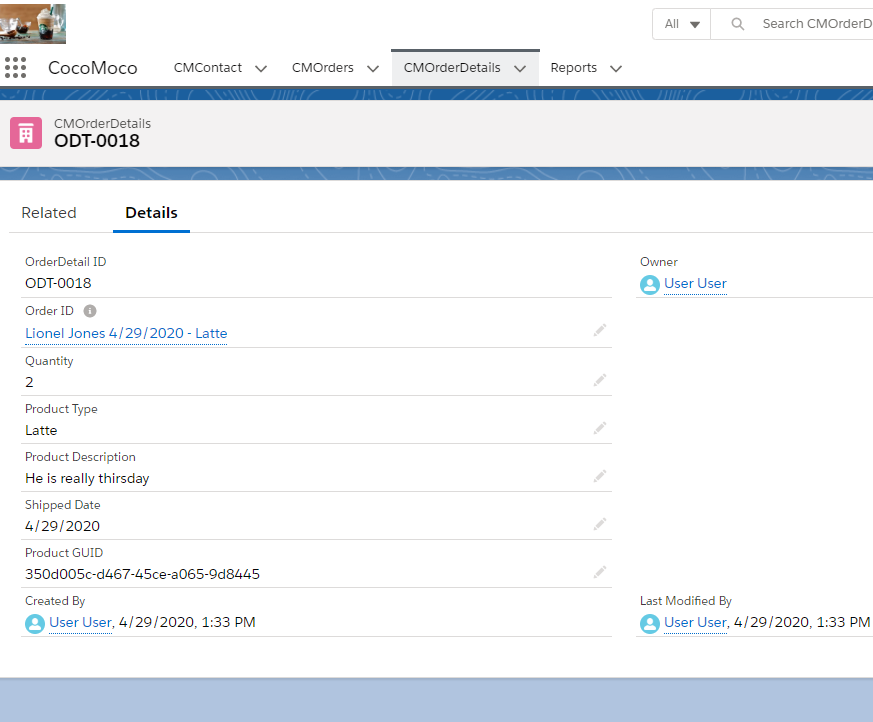
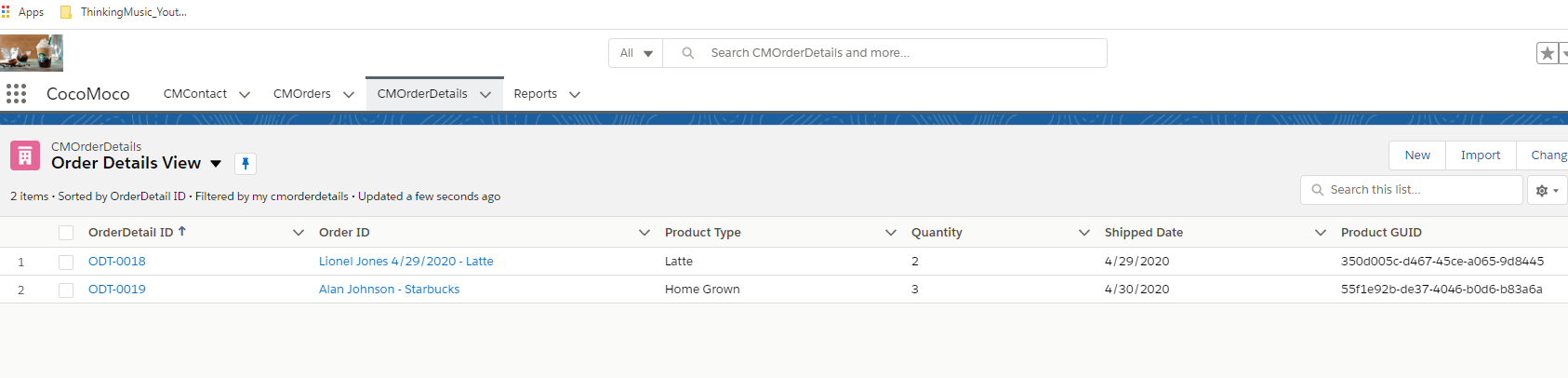
**Tools:**Visual Studio Code (Class Creation – Unit Testing)  
Visual Studio 2019 (Development of REST API)

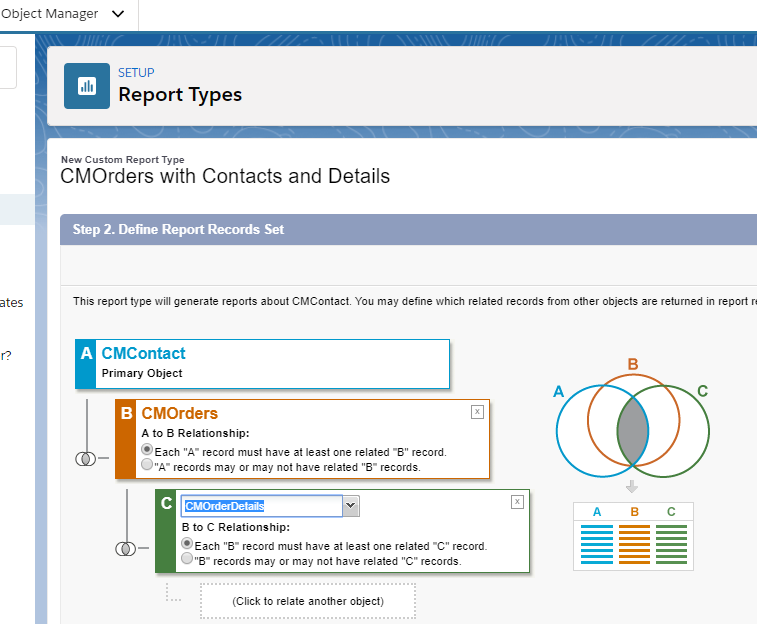
**Custom App:**  
  
**Database Schema (Sales force)**Sales Force Custom Objects:  
**CMContact**  
**Fields**:  
Name  
Address  
Email Address  
Phone  
  
**CMOrders** **Fields**:  
**CMContact** ->Relationship to CMContact (lookup)  
**OrderID (Auto Gen)**  
Order Date  
Notes

**CMOrderDetails ->Relationshop to CMOrders (lookup)**  
**Fields:**GUID  **OrderID**->Relationship to **CMOrders** (lookup)Quantity  
Product Type: (**Pick list**) Latte, Regular, Starbucks, HomeGrown  
Product Description (Could be placed in a separate entity, but for simplicity, I will leave it here)  
Shipped Date

**Products Table (Will discuss on code review) – at the very end, wanted to re-factor**  
  
  
**BatchTable**This will work with my Bactch Class (on the last page)  


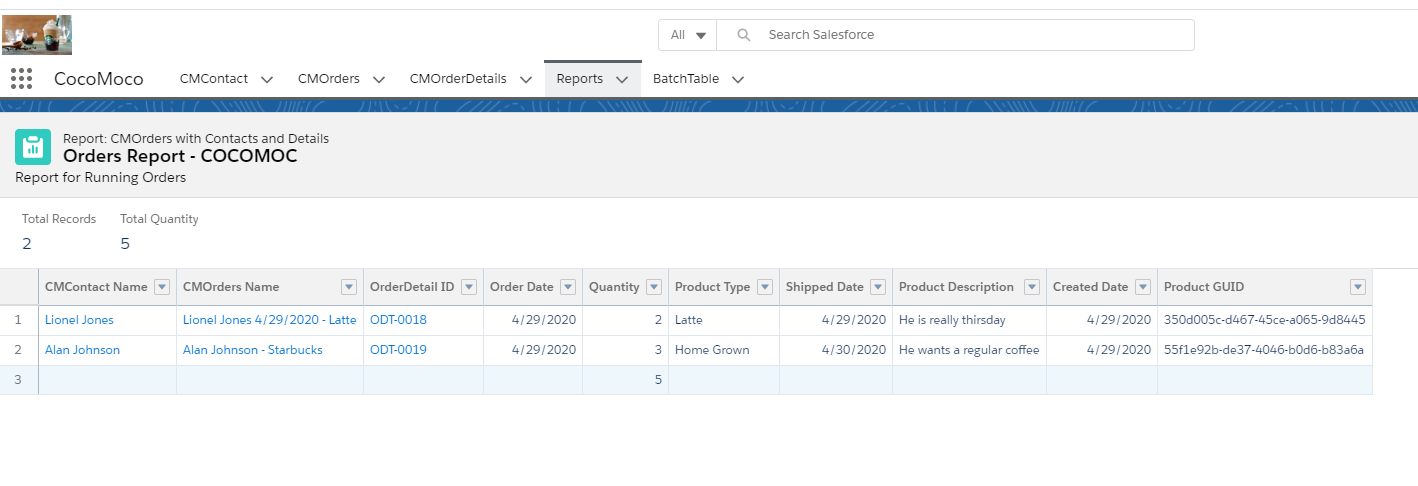
Ending App:  
  
  
  
Unit Test  
I create a Unit Test for creating a 30 character GUID  
Here is he Unit Test:  
  
  
  
I created a trigger on the CMOrderDetails, create a 30 character guid on before insert

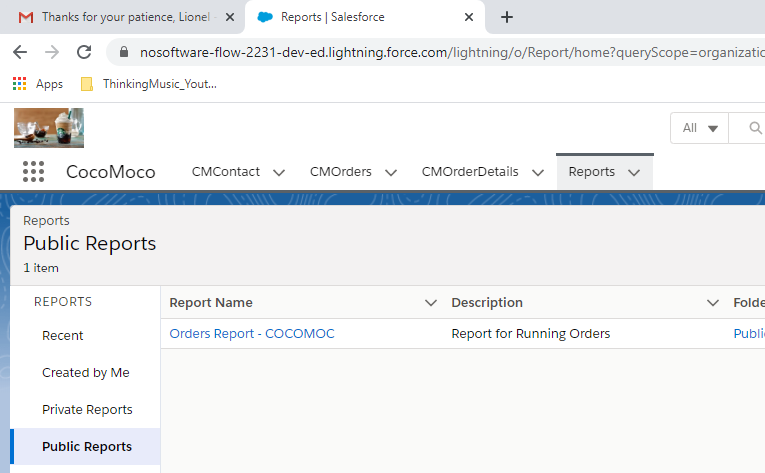
  
  
  
  
**For the Custom Report, I created a Custom Report Type:**



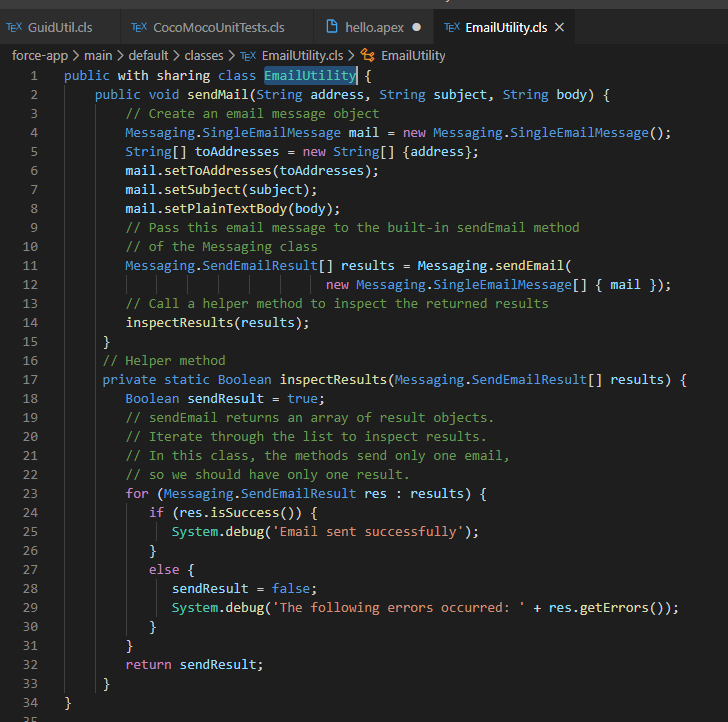
Then I created this report:

**With a date filer to show the last 24 months**

  
  
**Stored under Public Reports**



I created an email Utility class:



These are my Unit Tests

@isTest

private class CocoMocoUnitTests {

  @isTest static void testForUniquGuid30Characters()

  {

    Test.startTest();

    string myGUID = GuidUtil.NewGuid().substring(0,30);

    Integer  guidLength = 0;

    guidLength = myGUID.length();

    Test.stopTest();

    System.assertEquals(30,guidLength ,'Was Expectin a GUID of 30 characters');

  }

  @isTest static void testCanCreateContact()

  {

      //brute force

      List<CMContact\_\_c> mycontacts = new List<CMContact\_\_c>();

      CMContact\_\_c c = new CMContact\_\_c(name='New Contact');

      insert c;

      Test.startTest();

      mycontacts = [SELECT Id FROM CMContact\_\_c];

      Test.stopTest();

      System.assert(mycontacts.size() > 0, 'Was expecting to find at least one contact');

  }

  @isTest static void testAddContactWithOrder()

  {

    List<CMContact\_\_c> mycontacts = new List<CMContact\_\_c>();

    CMContact\_\_c c = new CMContact\_\_c(name='New Contact');

    insert c;

    List<CMOrders\_\_c> myOrders = new List<CMOrders\_\_c>();

    CMOrders\_\_c o = new CMOrders\_\_c(

        name='Starbucks Coffe Order',

        CMContact\_\_c = c.id

    );

    insert o;

    Test.startTest();

    myOrders = [SELECT Id FROM CMOrders\_\_c];

    Test.stopTest();

    System.assert(myOrders.size() > 0, 'Was expecting to find at least one order');

  }

  @isTest static void testDidEmailSucessfullySend()

  {

    Boolean sendResult = false;

    EmailUtility em = new EmailUtility();

       Test.startTest();

       sendResult = em.sendMail('lioneljones5116@gmail.com', 'Order Processed',

       'Your order was process with success on '

       + ' great success');

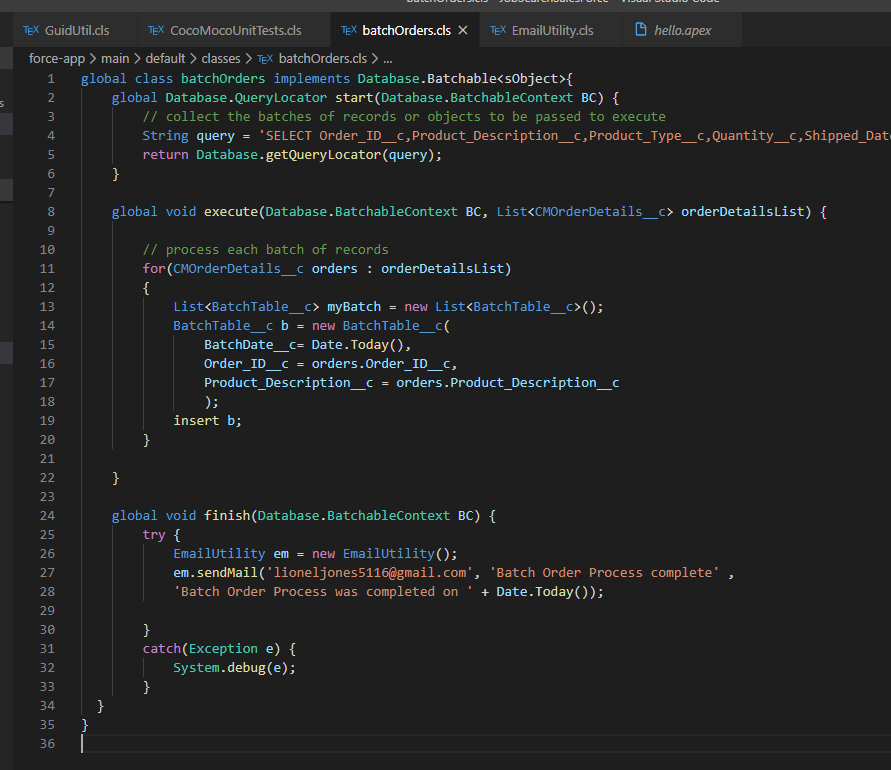
       Test.stopTest();

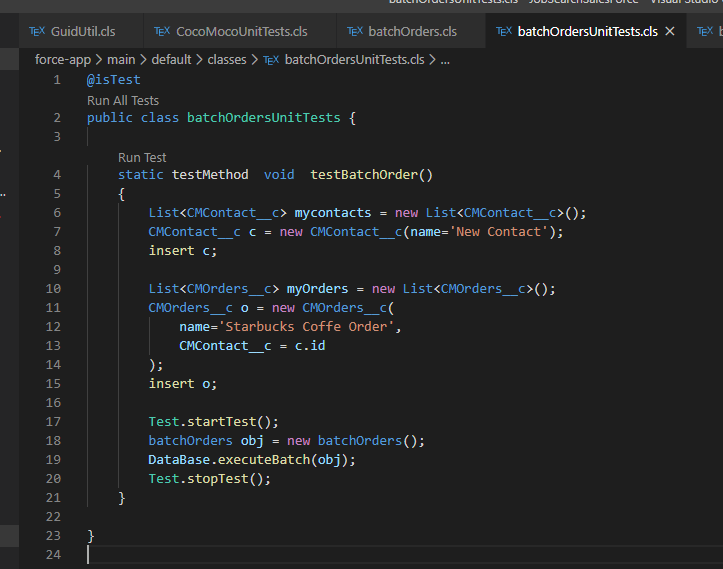
       System.assertEquals(sendResult,true, 'Expected a return of true');

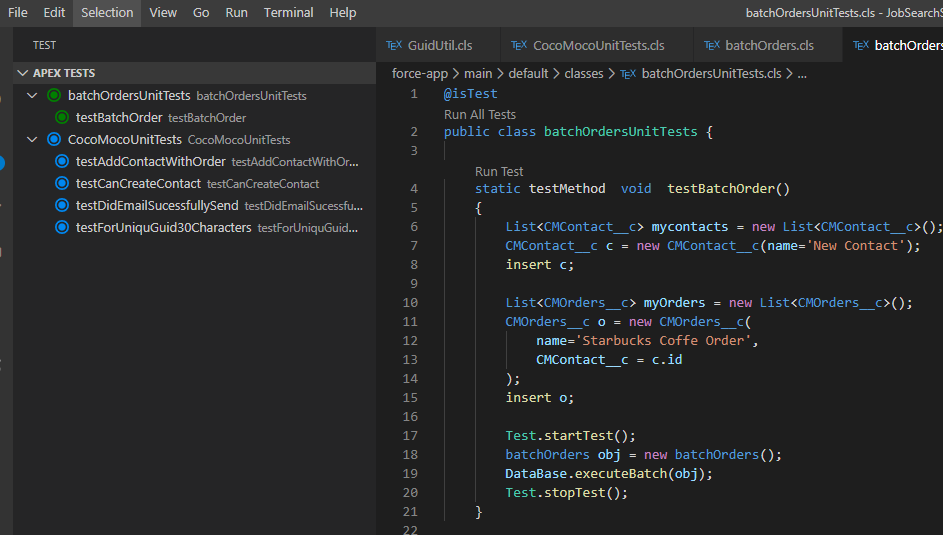
  }

}

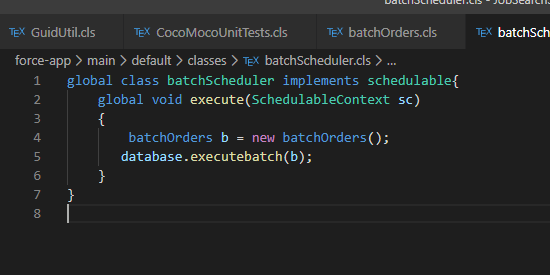
For Batching, I created a BatchClass



I wrote a unit test for the batch class:  
  
All of my unit tests passed:



I created a batch scheduler



I then created a scheduled job:

