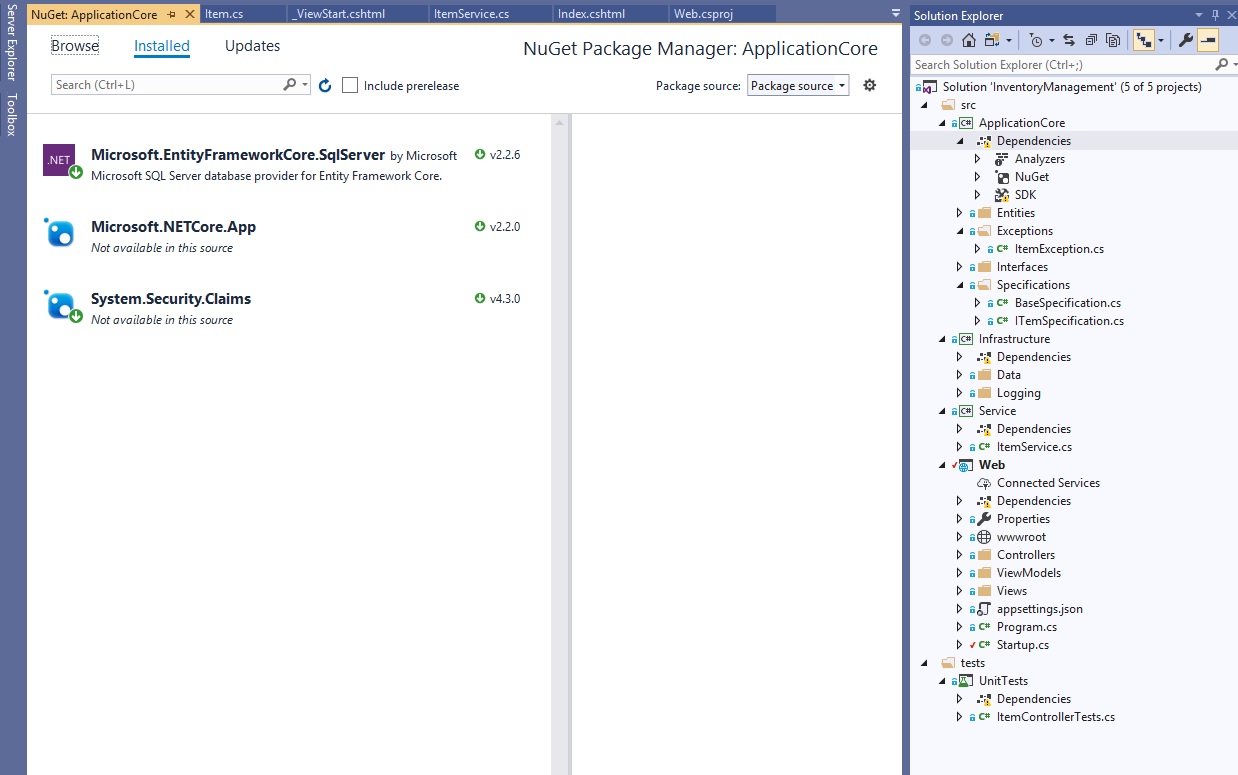
Inventory Project Management Assessment Details

1. Software Installed – (Attached Screenshot)

Asp.NetCore, EntityFrameworkCore



1. To run the application used Swagger Tool (Installed Swashbuckle.AspNetCore) to consume RESTful Web-services.

Here is the sample reference path <https://localhost:portno/swagger/index.html>

1. With reference of Design Problem document (Problem 3) , Inventory management

I followed the below instructions to Generate Report 1 and 2

*a) create itemName costPrice sellingPrice*

Whenever Mr. X wants to add a new item to his store he issues a create command. This command creates a new item in the inventory with the given cost price and selling price. The prices are rounded off to two decimal places.

*b) delete itemName*

If Mr. X decides not to sell an item anymore, then he simply issues a delete command. This command will remove the item from the inventory.

*c) updateBuy itemName quantity*

Whenever Mr. X purchases additional quantity of the mentioned item, then he issues a updateBuy command. This command should increase the quantity of the mentioned item.

*d) updateSell itemName quantity*

Whenever Mr. X sells some item, then he issues a updateSell command. This command should deduct the quantity of the mentioned item.

*e) report*

Whenever Mr. X wants to view his inventory list he issues the report command. This command should print the current inventory details in the specified format sorted by alphabetical order. Apart from printing the inventory it has to report on the profit made by Mr. X since last report generation.

Where profit is calculated by: ∑ (sellingPrice-costPrice) of the sold items multiplied by no. of items sold- costPrice of the deleted items.

1. Here are the sample Input using Swagger Api
2. To create new items used HttpPost and end point is **/api/item/Create**

Create Book01 10.50 13.79

**Input Json: { "name": "Book01", "costPrice": 10.50, "sellPrice": 13.79, "quantity": 0, "id": 0}**

Create Food01 1.47 3.98

**Input Json: { "name": "Food01", "costPrice": 1.47, "sellPrice": 3.98, "quantity": 0, "id": 0}**

Create Med01 30.63 34.29

**Input Json: { "name": "Med01", "costPrice": 30.63, "sellPrice": 34.29, "quantity": 0, "id": 0}**

Create Tab01 57.00 84.98

**Input Json: { "name": "Tab01", "costPrice": 57.00, "sellPrice": 84.98, "quantity": 0, "id": 0}**

1. To updateBuy Items used HttpPut and end point is **/api/item/UpdateBuy/{ItemName}**

updateBuy Tab01 100

**InputName: Tab01 and JsonValue: { "quantity": 100 }**

updateBuy Food01 500

**InputName: Food01 and JsonInput: { "quantity": 500 }**

updateBuy Book01 100

**InputName: Book01 and JsonInput: { "quantity": 100 }**

updateBuy Med01 100

**InputName: Med01 and JsonInput: { "quantity": 100 }**

1. To updateSell Items used HttpPut and end point is **/api/item/UpdateSell/{ItemName}**

updateSell Tab01 2

**InputName: Tab01 and JsonInput: { "quantity":2}**

updateSell Food01 1

**InputName: Food01 and JsonInput: { "quantity":1}**

updateSell Food01 1

**InputName: Food01 and JsonInput: { "quantity":1}**

updateSell Tab01 2

**InputName: Tab01 and JsonInput: { "quantity":2}**

1. Generated Report 1 – (Output: Json format)

{

"totalValue": 10317.06,

"profitSincePreviousReport": 116.94,

"itemReportViewModels": [

{

"itemName": "Book01",

"boughtAt": 10.5,

"soldAt": 13.79,

"availableQty": 100,

"value": 1050

},

{

"itemName": "Food01",

"boughtAt": 1.47,

"soldAt": 3.98,

"availableQty": 498,

"value": 732.06

},

{

"itemName": "Med01",

"boughtAt": 30.63,

"soldAt": 34.29,

"availableQty": 100,

"value": 3063

},

{

"itemName": "Tab01",

"boughtAt": 57,

"soldAt": 84.98,

"availableQty": 96,

"value": 5472

}

]

}

========================================================================

1. Followed Below Steps to generate Report 2

Delete Record

============

delete Book01

Create New Record

=================

create Mobile01 10.51 44.56

**JsonInput: { "name": "Mobile01", "costPrice": 10.51, "sellPrice": 44.56, "quantity": 0, "id": 0}**

UpdateBuy

=========

updateBuy Mobile01 250

**JsonInput: { "quantity": 250 }**

UpdateSell

===========

updateSell Tab01 5

**JsonInput: { "quantity": 5 }**

updateSell Food01 5

**JsonInput: { "quantity": 5 }**

updateSell Mobile01 4

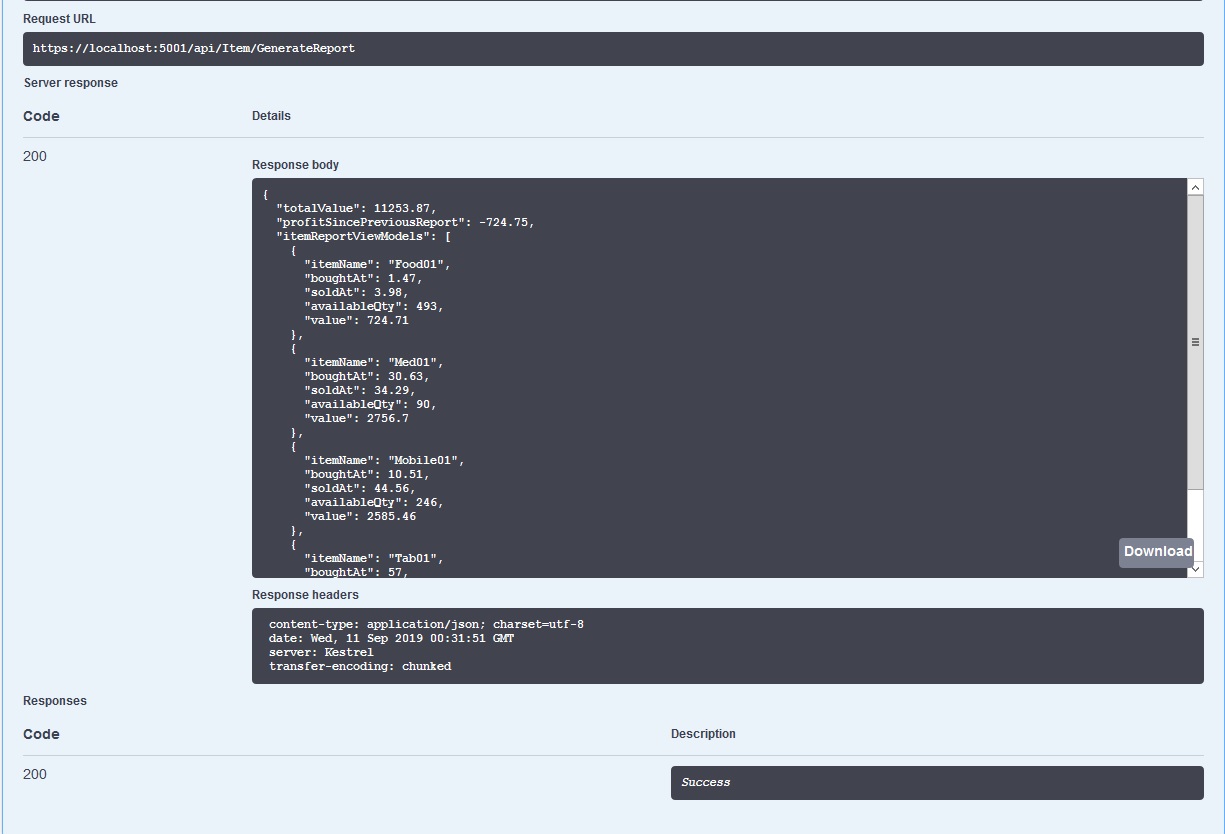
**JsonInput: { "quantity": 4 }**

updateSell Med01 10

**JsonInput: { "quantity": 10 }**

Generate Report 2 (ScreenShot Attached with Json data)

=================



{

"totalValue": 11253.87,

"profitSincePreviousReport": -724.75,

"itemReportViewModels": [

{

"itemName": "Food01",

"boughtAt": 1.47,

"soldAt": 3.98,

"availableQty": 493,

"value": 724.71

},

{

"itemName": "Med01",

"boughtAt": 30.63,

"soldAt": 34.29,

"availableQty": 90,

"value": 2756.7

},

{

"itemName": "Mobile01",

"boughtAt": 10.51,

"soldAt": 44.56,

"availableQty": 246,

"value": 2585.46

},

{

"itemName": "Tab01",

"boughtAt": 57,

"soldAt": 84.98,

"availableQty": 91,

"value": 5187

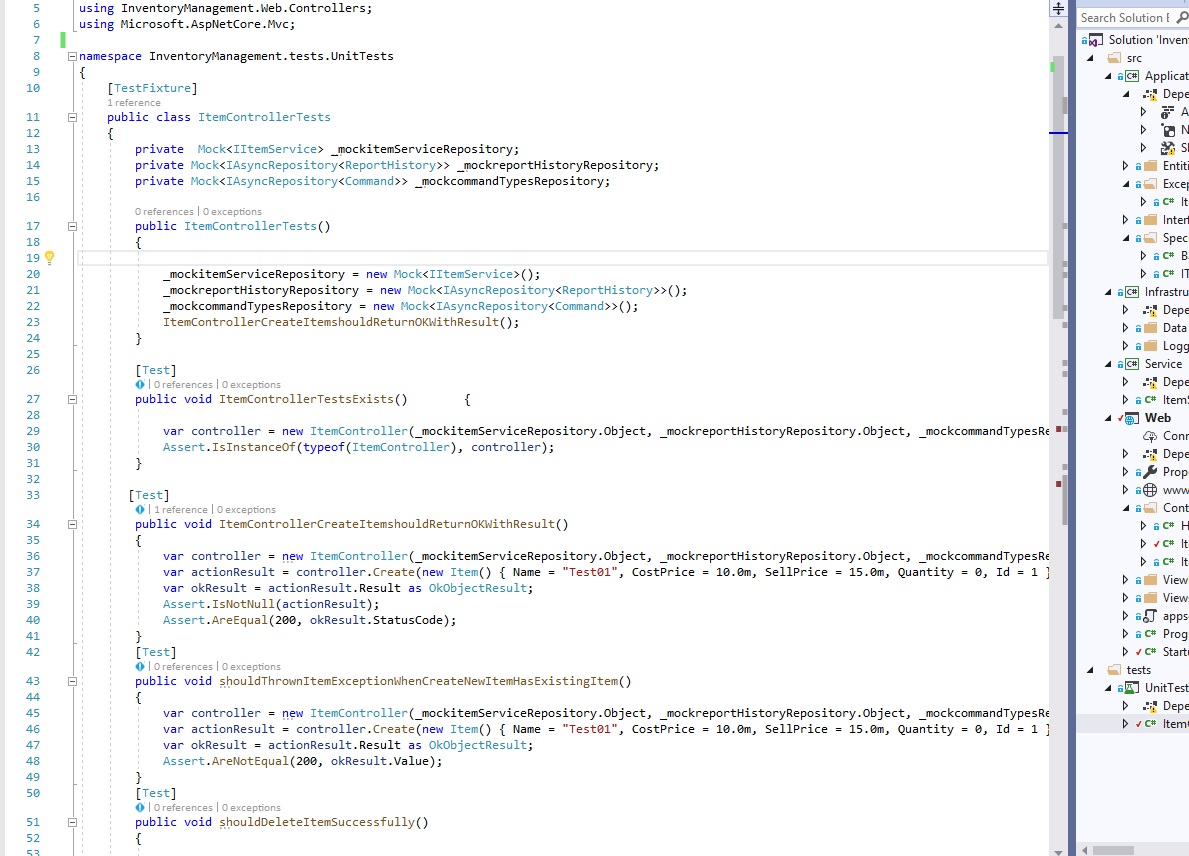
}

]

}

=====================================================================

1. Test Case ScreenShots:-



1. Test Case Passed - (Attached Screenshot)

