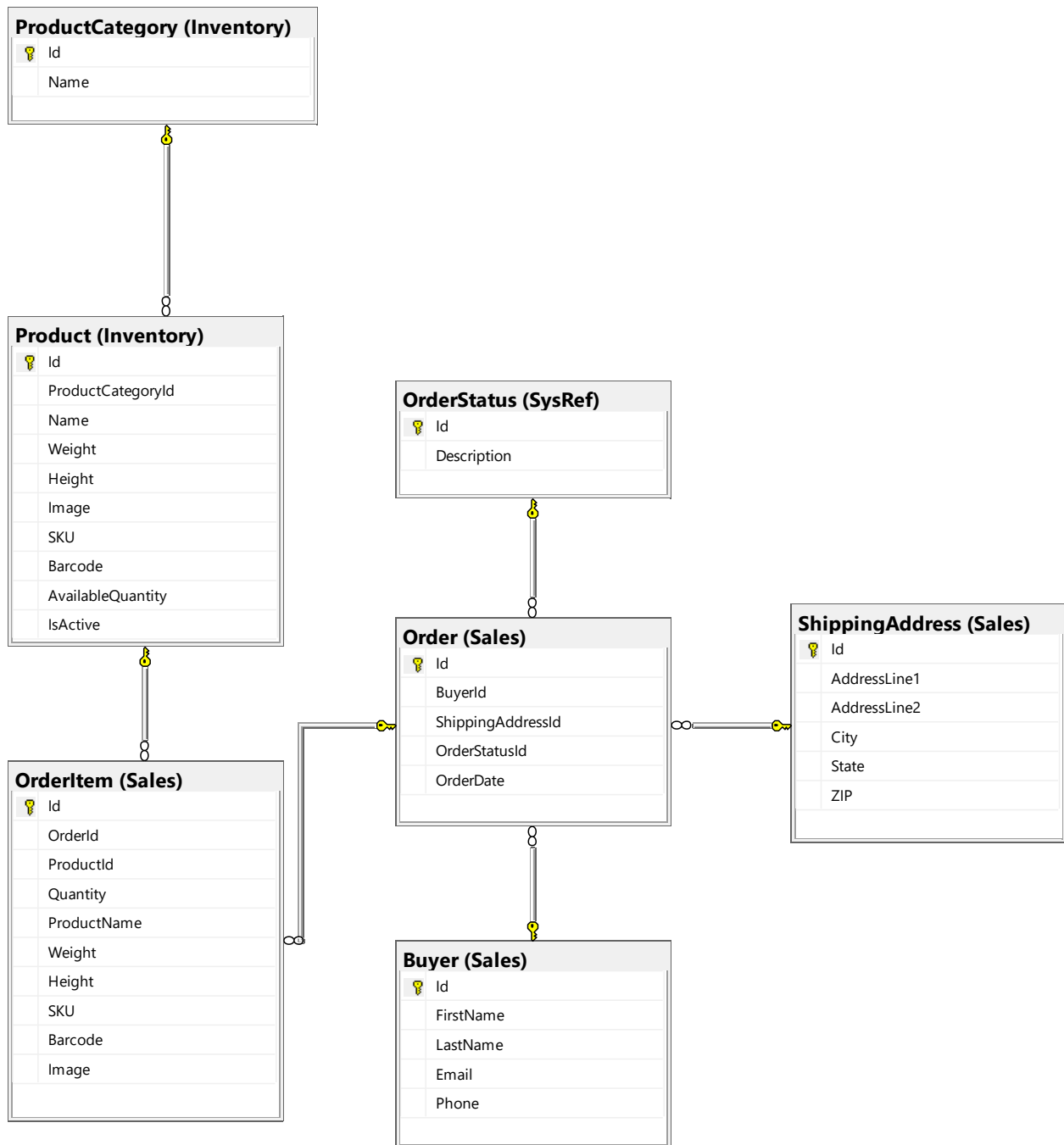


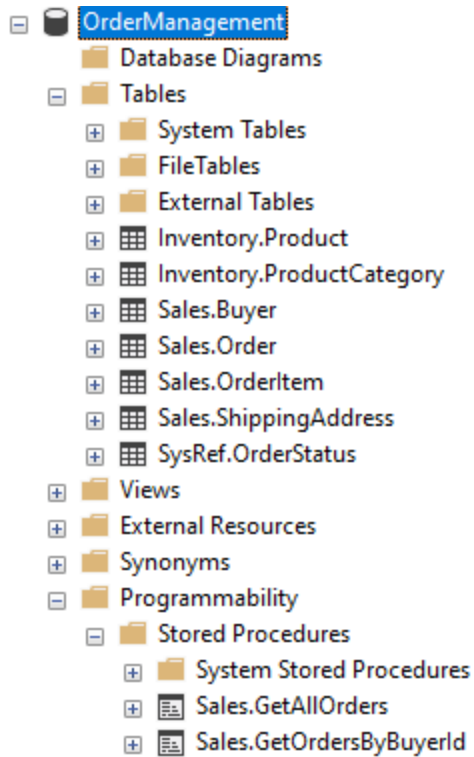
Order Management System Web API (RESTful Services)

Contents

Data Model	2
Application Design.....	3
Business Operations Implementation Approach and Analysis	4
Place Order Operation Analysis.....	4
Update Order Operation Analysis	5
Delete Order Operation Analysis.....	5
View Orders by Buyer (Customer Role) Analysis:.....	5
View Orders All Order (Administrator Role) Analysis	6
My Other Github Open Source Works:.....	7
Multi-service and multi-level dependency injection framework for .NET.....	7
Modular, scalable and pluggable architecture using IocServiceStack.NET.....	7
You can also check all other repositories here	7

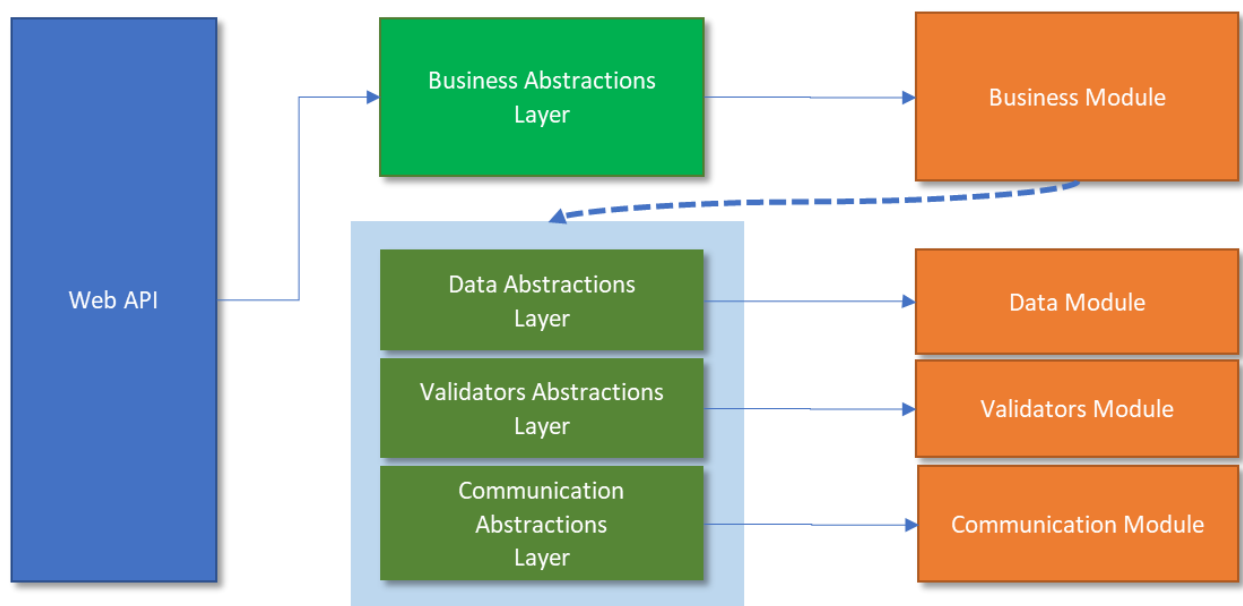
Data Model

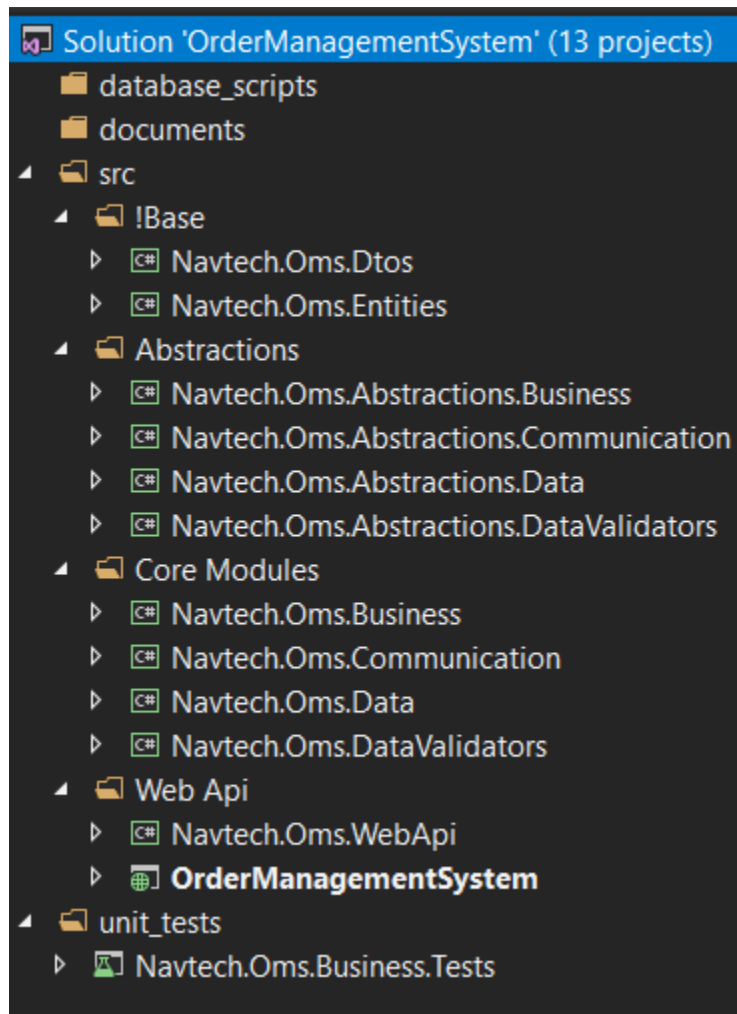




Application Design

This is modular, pluggable and scalable architecture. No module is directly depending on the other (no direct reference), it depends on the abstraction layers. I have Used `IoCServiceStack.NET` is a DI Framework, and I am the author of this framework. `IoCServiceStack.NET` framework that I developed in 2016. And I have developed several applications using this framework.





Business Operations Implementation Approach and Analysis

Place Order Operation Analysis

- Create Order:
 - Validate Buyer/Customer information, Shipping Address
 - Shipping Address needs be validated whether it's possible to deliver the customer specified location
 - Validate whether items that are in basket available in stock
 - Save Buyer Information (If buyer is already existing in the system, we can tag current order with existing buyer id)
 - Save Shipping Address (If shipping is already existing in the system, we can tag current order with existing shipping address id)
 - Save order record in system
 - Save order items

- Update Inventory
- Send notification

Update Order Operation Analysis

- Update Order:
 - Order to be allowed to update only when the status of order is draft or not yet dispatched.
 - Validate Buyer/Customer information, Shipping Address
 - Shipping Address needs be validated whether it's possible to deliver the customer specified location
 - Validate whether items that are in basket available in stock
 - API needs be deigned to receive only items that are modified in basket. So, we receive state of each item that are only modified/added/deleted
 - Save Buyer Information
 - Save Shipping Address
 - Save order record in system with last update time
 - Save order items (based on state of item)
 - Update Inventory

Delete Order Operation Analysis

- Delete Order:
 - Order to be allowed to delete only when the status of order is draft or not yet dispatched.
 - We don't need to delete the buyer and shipping information if we maintain this as master data. Need to design different production data model.
 - Delete order items
 - Delete order
 - Delete shipping address
 - Delete buyer

View Orders by Buyer (Customer Role) Analysis:

- View Your Orders:
 - Based on user login, we can get current User object for current thread, this information will be assigned to current thread while authorize the current request
 - If we want to show customer order by sending open link, then buyer id must be in encrypted format in URL

```

ALTER PROCEDURE [Sales].[GetOrdersByBuyerId]
@BuyerId int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    SELECT O.Id as OrderId, B.FirstName, B.LastName, B.Email, B.Phone,
           S.AddressLine1, S.AddressLine2, S.City, S.State, S.ZIP,
           COUNT(OI.Id) as ItemsCount
    FROM [Sales].[Order] O
    JOIN [Sales].[Buyer] B ON O.BuyerId = B.Id
    JOIN [Sales].[ShippingAddress] S ON O.ShippingAddressId = S.Id
    JOIN [Sales].[OrderItem] OI ON O.Id = OI.OrderId

    WHERE O.BuyerId = @BuyerId

    GROUP BY O.Id, B.FirstName, B.LastName, B.Email, B.Phone,
           S.AddressLine1, S.AddressLine2, S.City, S.State, S.ZIP;

END

```

View Orders All Order (Administrator Role) Analysis

- View Your Orders:
 - Based on user login, we can get current User object for current thread, this information will be assigned to current thread while authorize the current request
 - Need to get all customers orders

```

BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    SELECT O.Id as OrderId, B.FirstName, B.LastName, B.Email, B.Phone,
           S.AddressLine1, S.AddressLine2, S.City, S.State, S.ZIP,
           COUNT(OI.Id) as ItemsCount
    FROM [Sales].[Order] O
    JOIN [Sales].[Buyer] B ON O.BuyerId = B.Id
    JOIN [Sales].[ShippingAddress] S ON O.ShippingAddressId = S.Id
    JOIN [Sales].[OrderItem] OI ON O.Id = OI.OrderId

    GROUP BY O.Id, B.FirstName, B.LastName, B.Email, B.Phone,
             S.AddressLine1, S.AddressLine2, S.City, S.State, S.ZIP;

END

```

My Other Github Open Source Works:

Multi-service and multi-level dependency injection framework for .NET

Source Code: <https://github.com/rjinaga/locServiceStack>

Documentation: <https://rjinaga.github.io/locServiceStack/>

Nuget: <https://www.nuget.org/packages/locServiceStack/>

Modular, scalable and pluggable architecture using locServiceStack.NET

Source Code: <https://github.com/rjinaga/EnterpriseProjectStructure.net>

You can also check all other repositories here

<https://github.com/rjinaga>