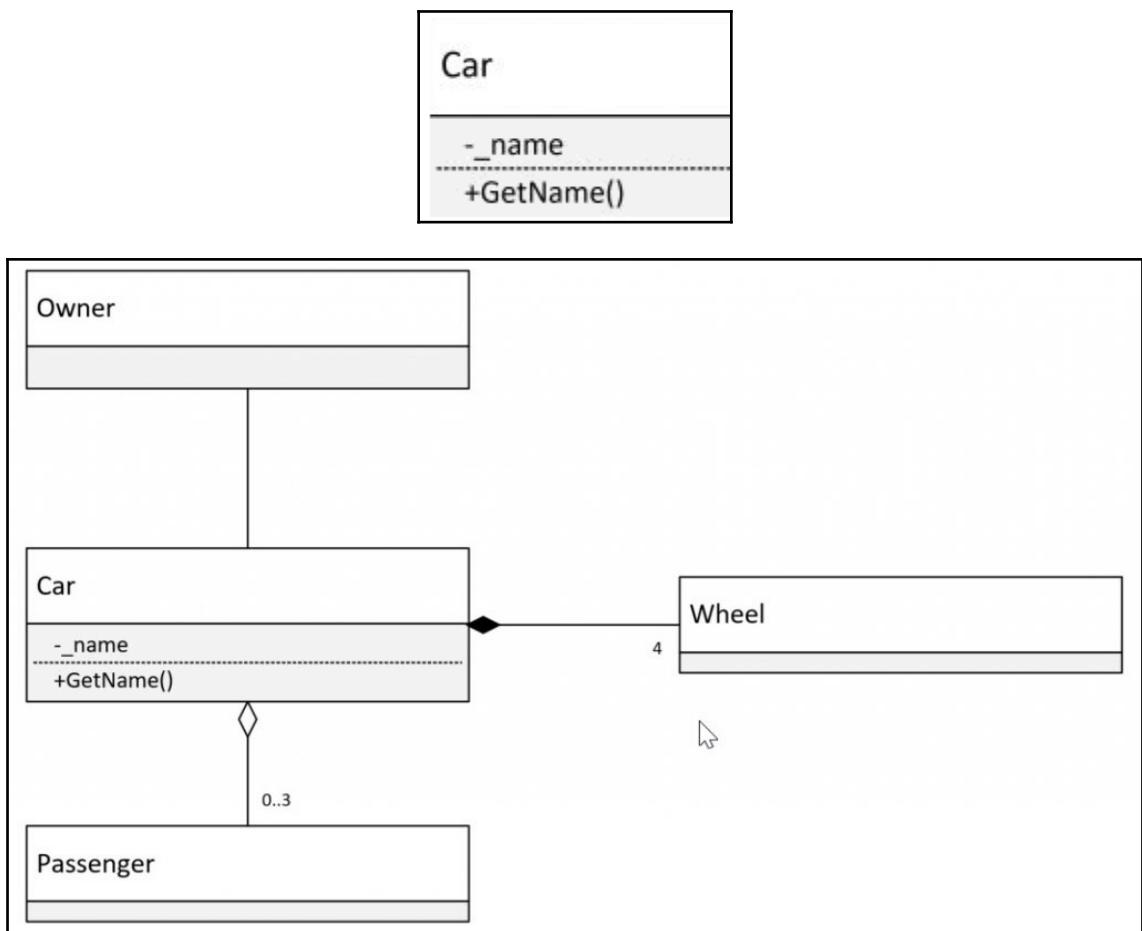
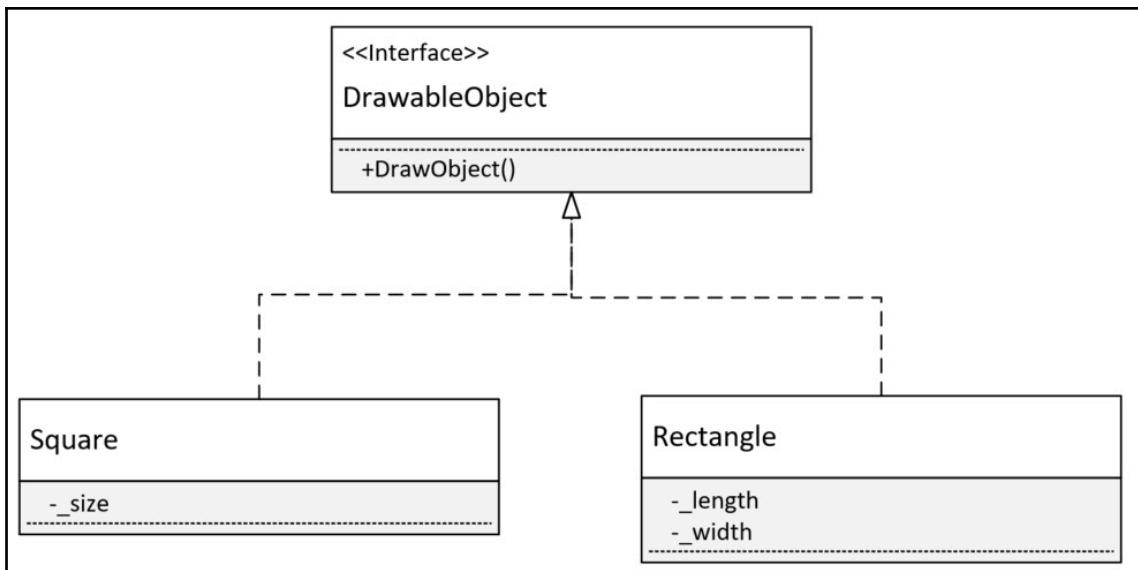
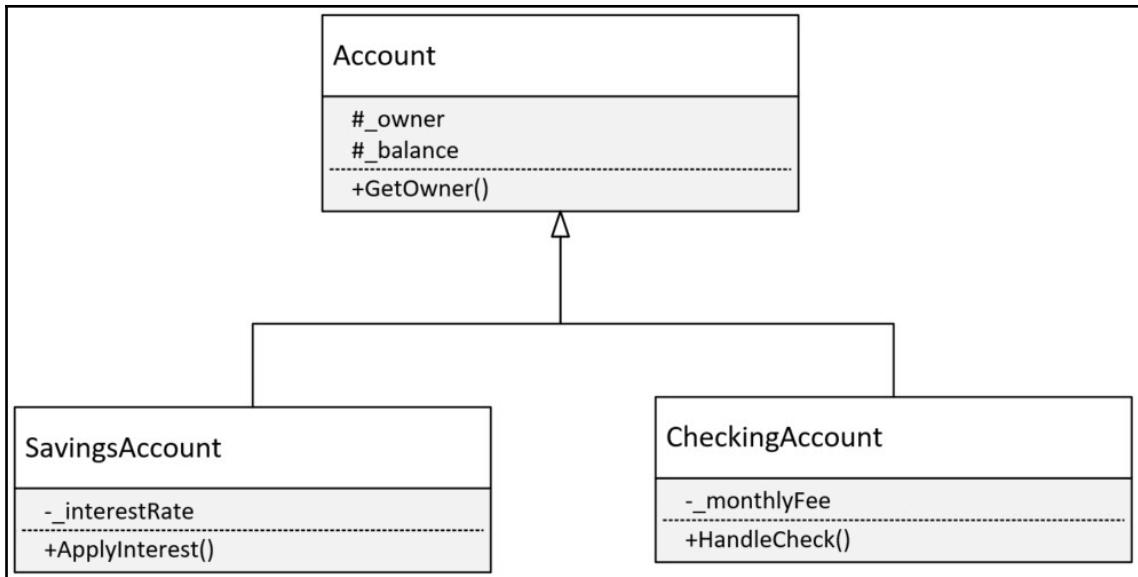
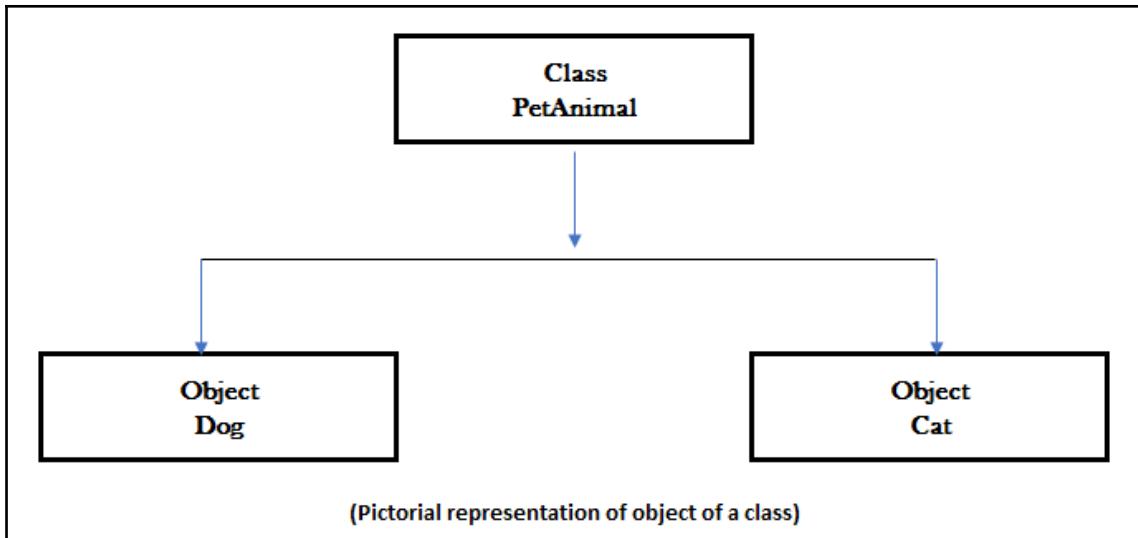


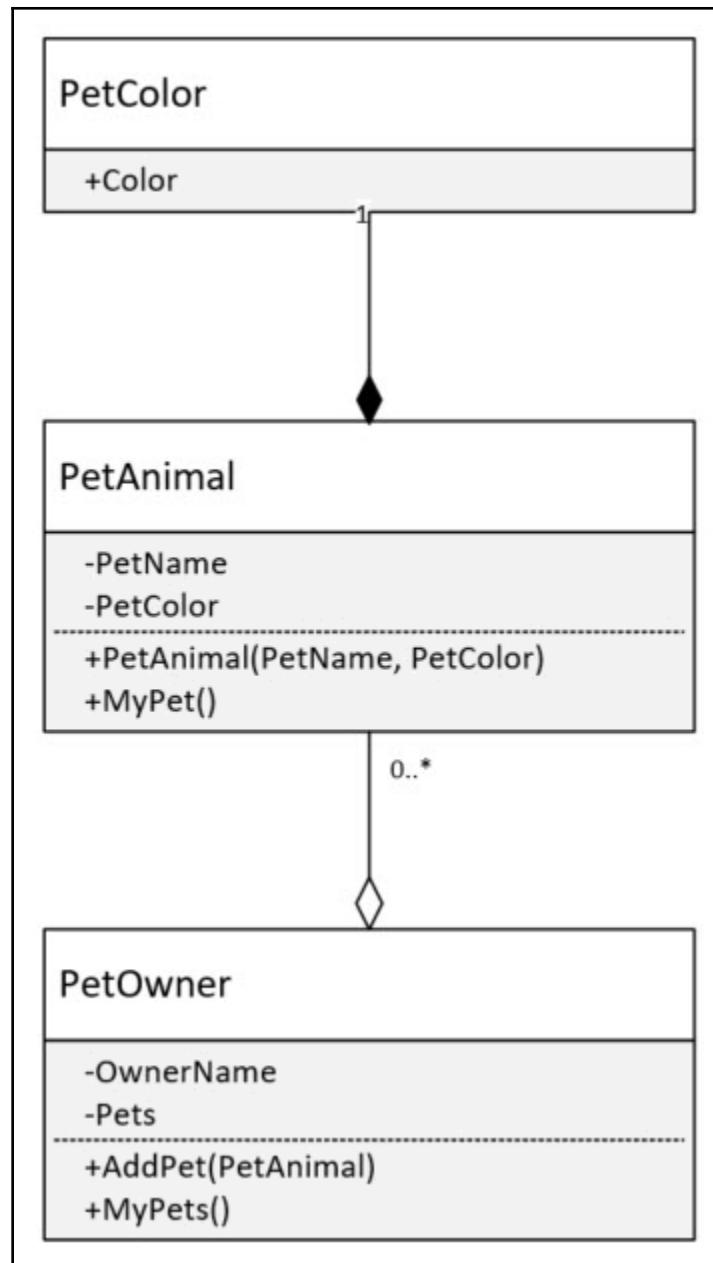
Graphic Bundle

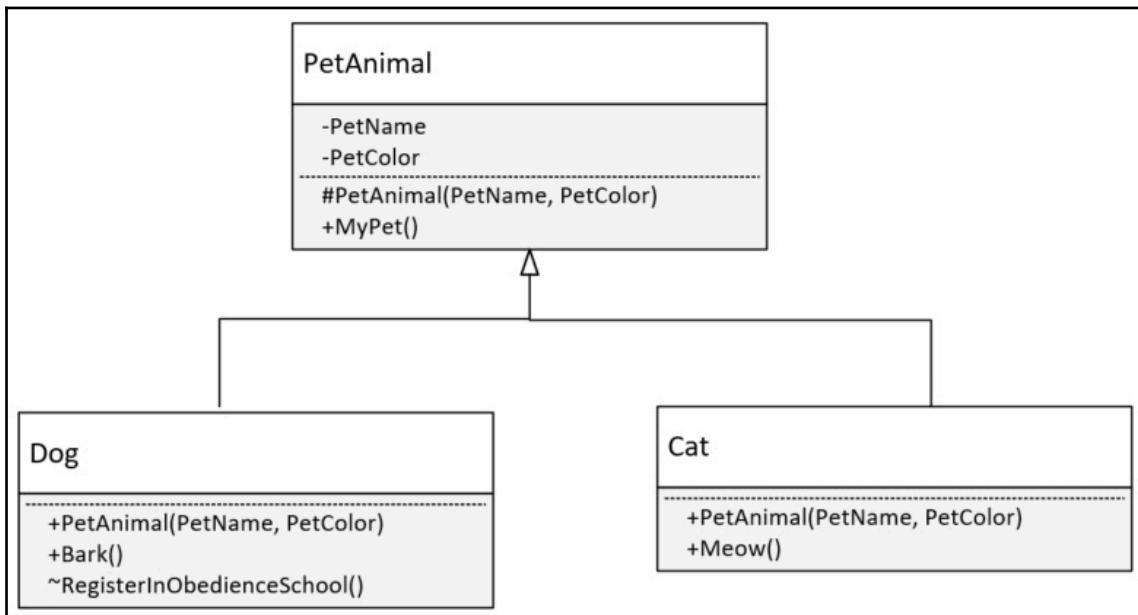
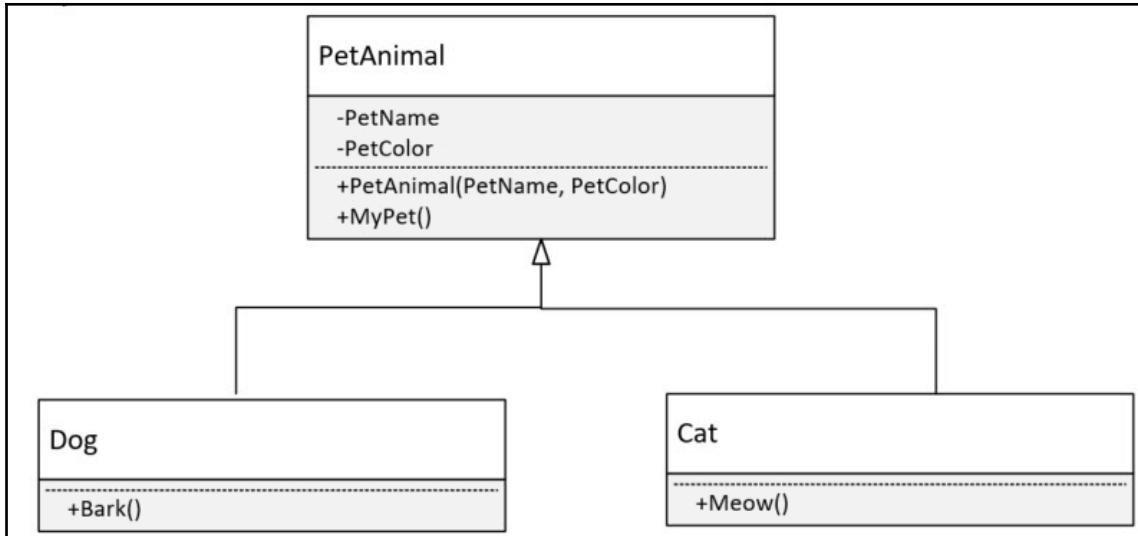
Chapter 1: Overview of OOP in .NET Core and C#









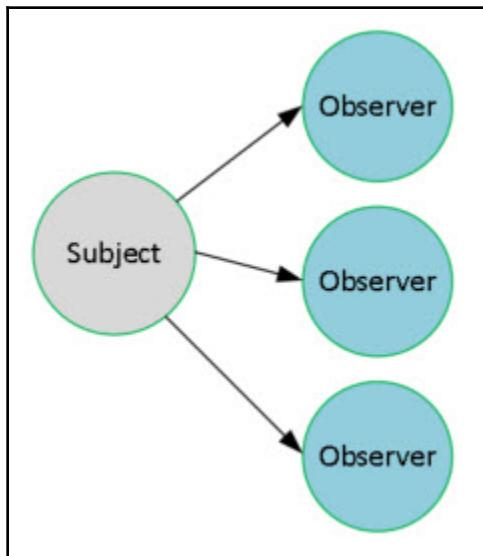


Chapter 2: Modern Software Design Patterns and Principles

```
C:\Program Files\dotnet\dotnet.exe
First Message!
Second Message with a beep!
Third Message with a beep and in red!
Not Decorated...
```

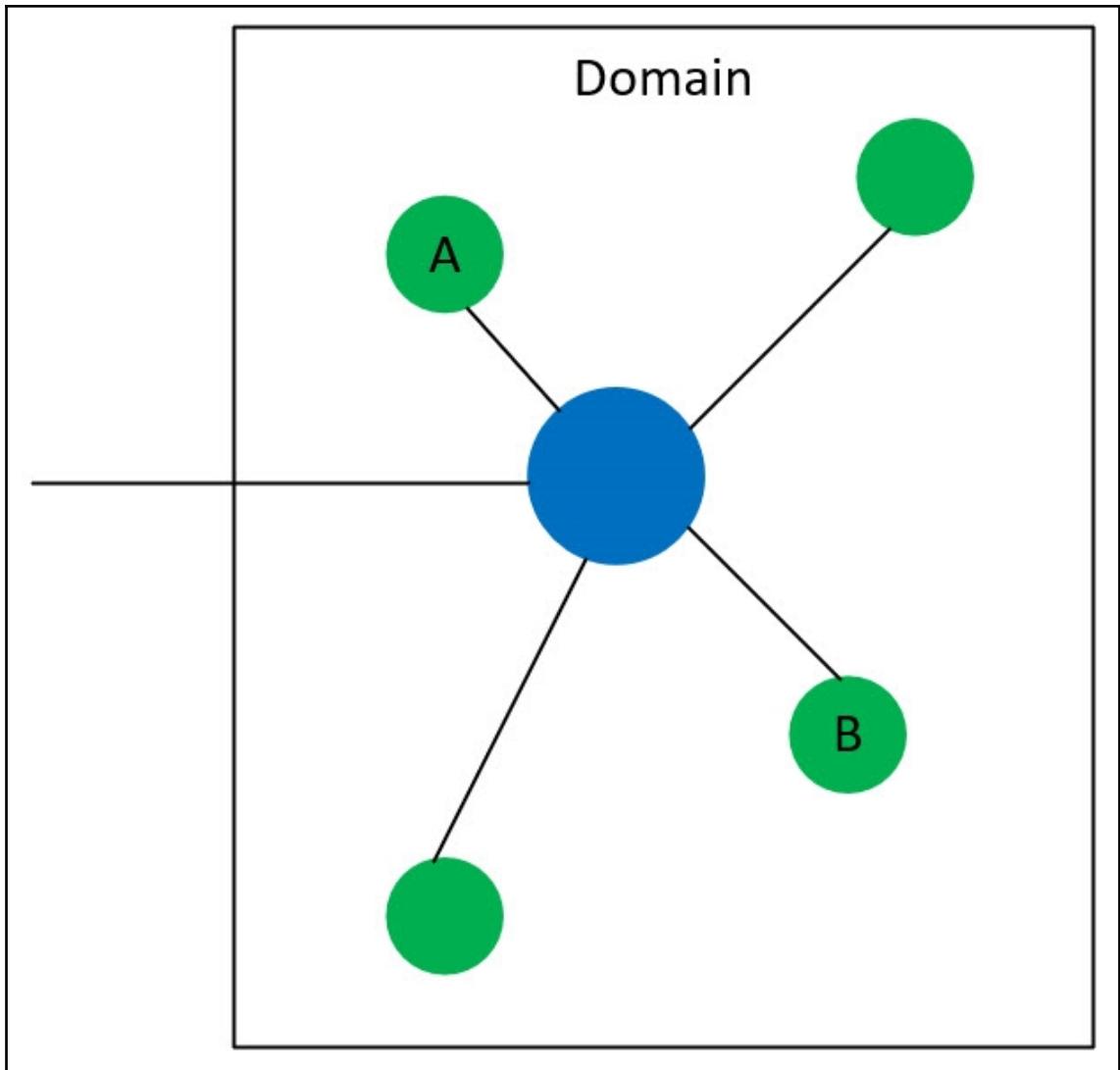
```
C:\Program Files\dotnet\dotnet.exe
Car 1 is dirty
Detailer providing Dirty services.

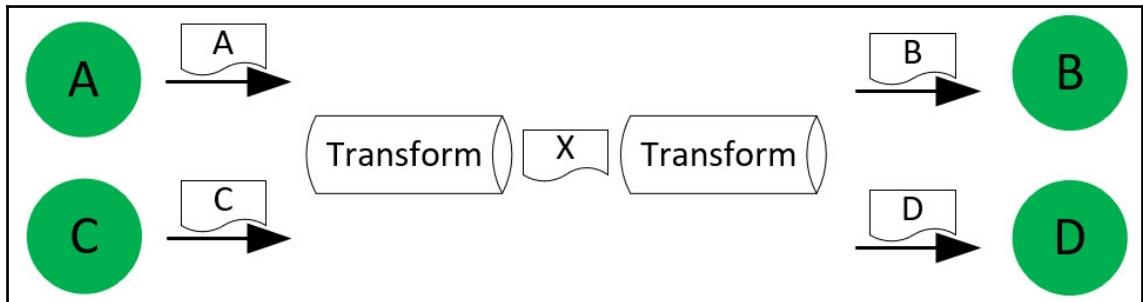
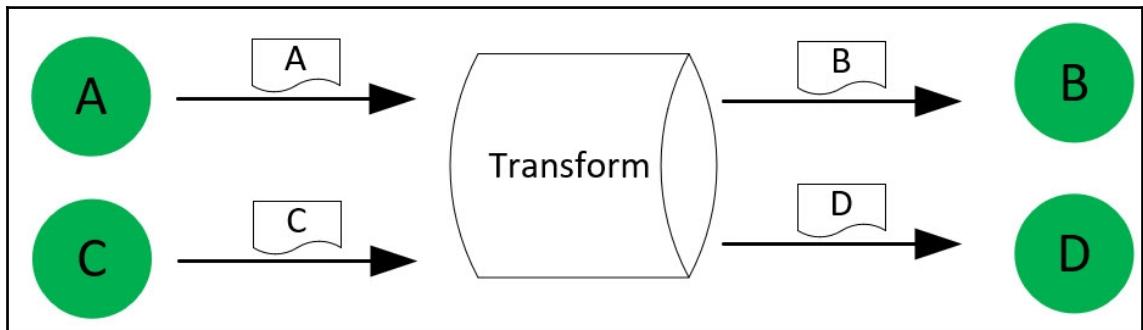
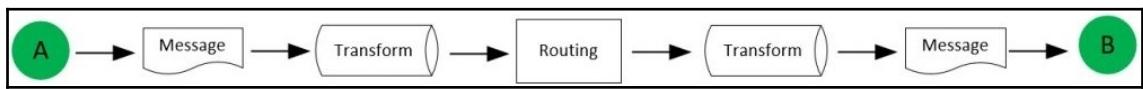
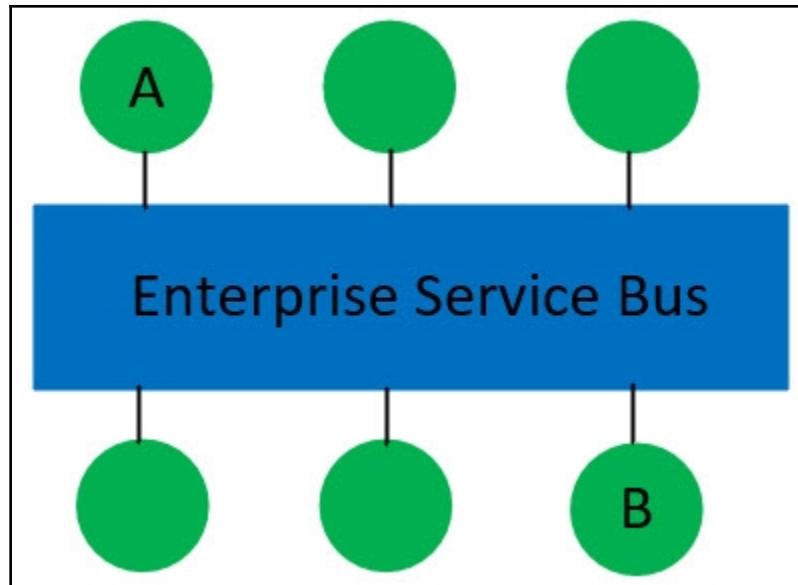
Car 2 requires full service
Mechanic providing EngineTune services.
WheelSpecialist providing WheelAlignment services.
QualityControl providing TestDrive services.
Detailer providing Dirty services.
```

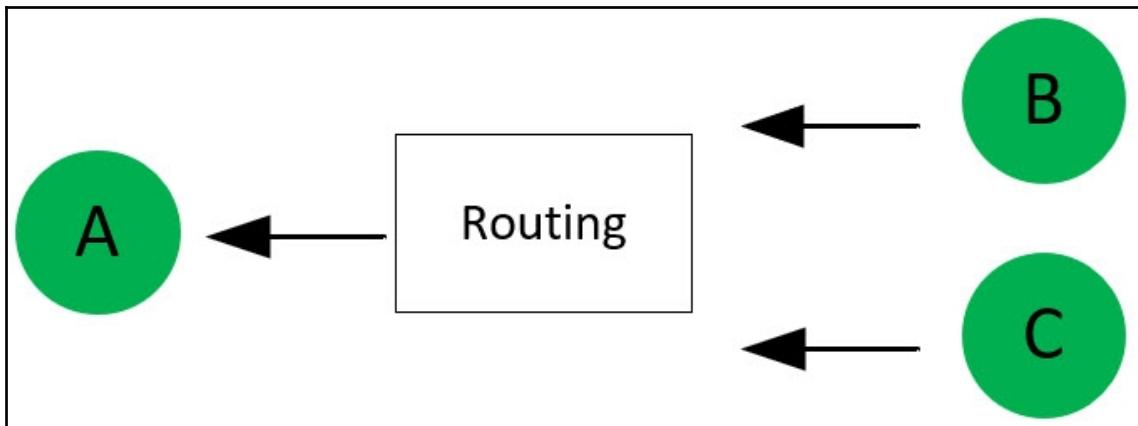
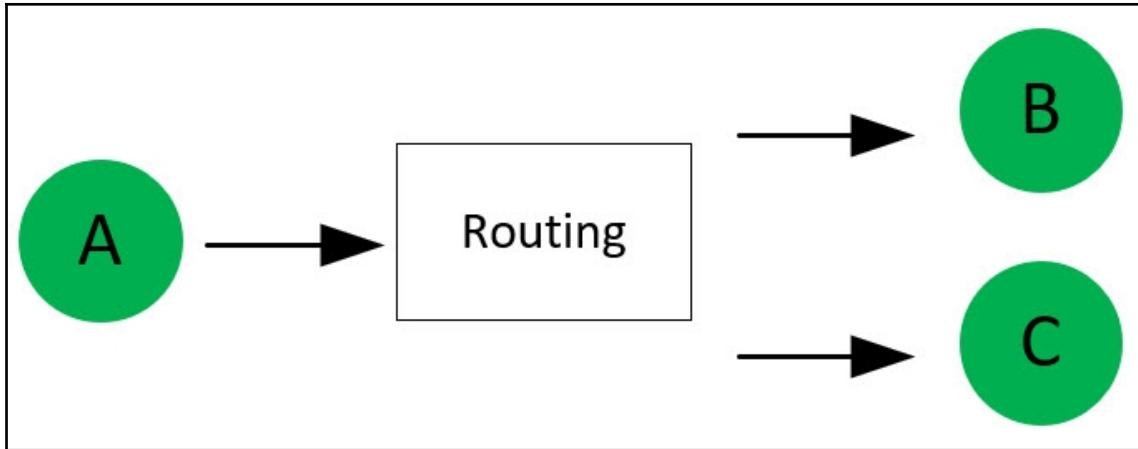


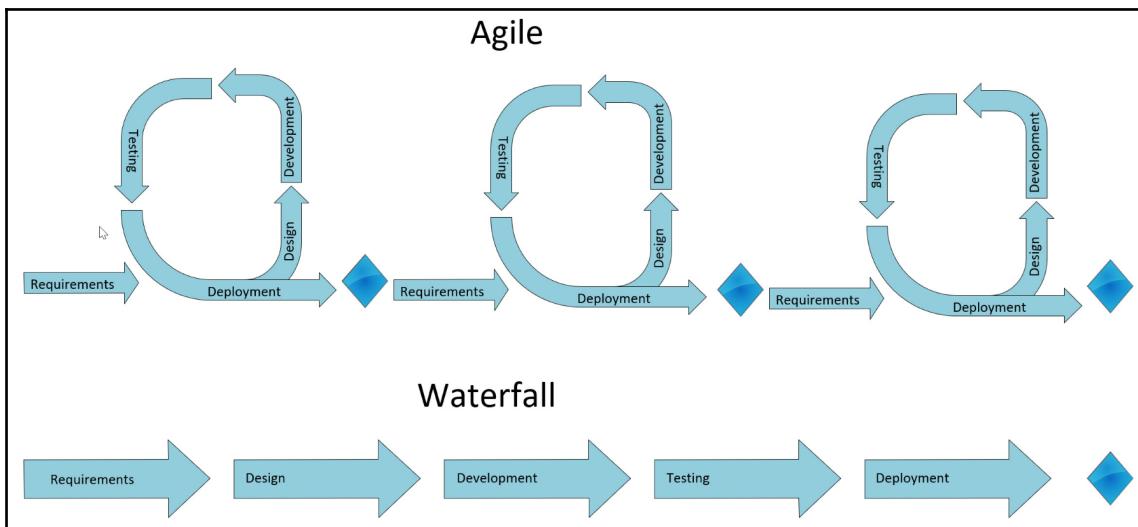
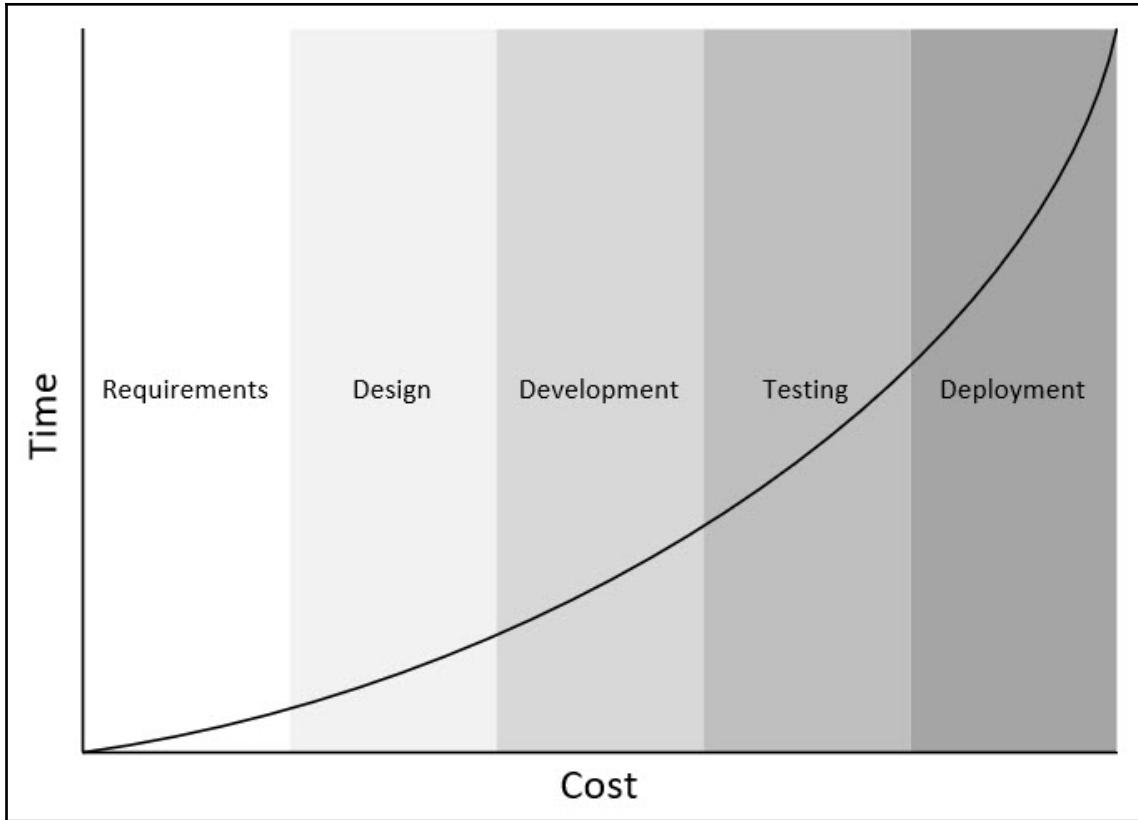
```
cs C:\Program Files\dotnet\dotnet.exe

Hello World!
I observer the new quantity value of 12.
I observer the new quantity value of 12.
I observer the new quantity value of 12.
I observer the new quantity value of 17.
I observer the new quantity value of 17.
I observer the new quantity value of 17.
Enter a key to quit.
```

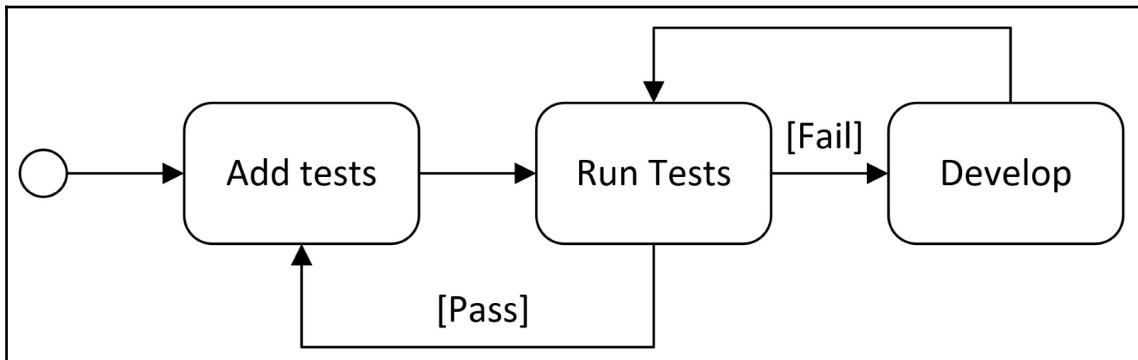
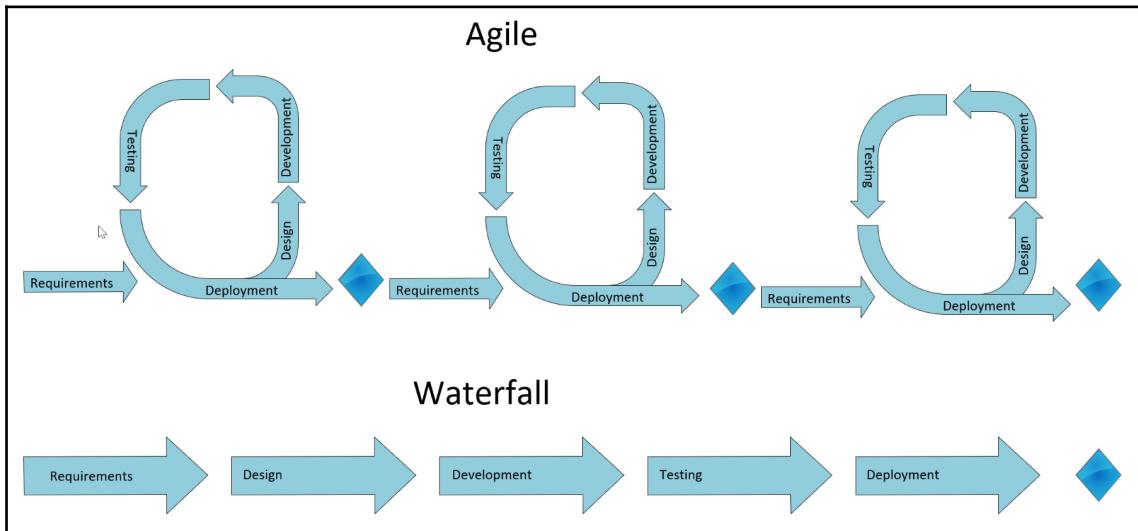


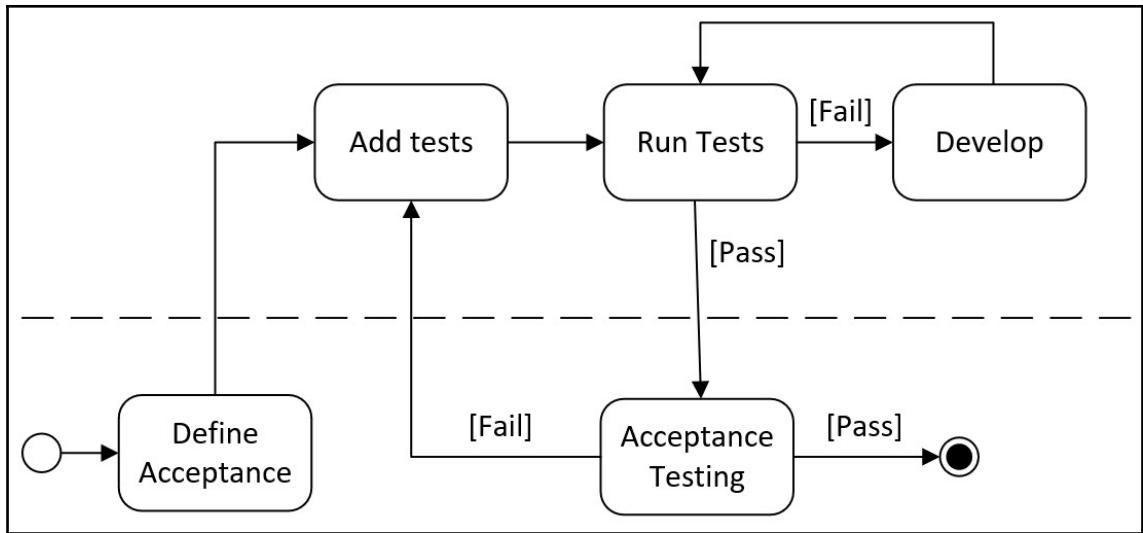






Chapter 3: Implementing Design Patterns - Basics Part 1





```
Administrator: Windows PowerShell
PS J:\git> mkdir FlixOne

Directory: J:\git

Mode          LastWriteTime    Length Name
----          -----          ---- 
d---          24/05/2018 2:51 PM          FlixOne

PS J:\git> cd .\FlixOne\
PS J:\git\FlixOne> dotnet new sln
The template "Solution File" was created successfully.
PS J:\git\FlixOne> dir

Directory: J:\git\FlixOne

Mode          LastWriteTime    Length Name
----          -----          ---- 
-a---         24/05/2018 2:51 PM          540 FlixOne.sln

PS J:\git\FlixOne> ■
```

Graphic Bundle

```
PS J:\git\FlixOne> dotnet new classlib --name FlixOne.InventoryManagement
The template "Class library" was created successfully.

Processing post-creation actions...
Running 'dotnet restore' on FlixOne.InventoryManagement\FlixOne.InventoryManagement.csproj...
Restoring packages for J:\git\FlixOne\FlixOne.InventoryManagement\FlixOne.InventoryManagement.csproj...
Generating MSBuild file J:\git\FlixOne\FlixOne.InventoryManagement\obj\FlixOne.InventoryManagement.csproj.nuget.g.props.
Generating MSBuild file J:\git\FlixOne\FlixOne.InventoryManagement\obj\FlixOne.InventoryManagement.csproj.nuget.g.targets.
Restore completed in 113.91 ms for J:\git\FlixOne\FlixOne.InventoryManagement\FlixOne.InventoryManagement.csproj.

Restore succeeded.
```

```
PS J:\git\FlixOne> dotnet new console --name FlixOne.InventoryManagementClient
The template "Console Application" was created successfully.

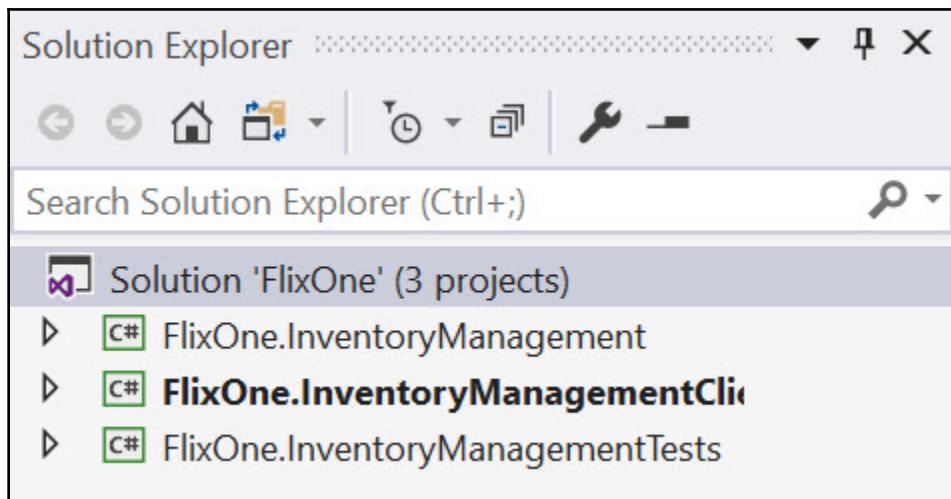
Processing post-creation actions...
Running 'dotnet restore' on FlixOne.InventoryManagementClient\FlixOne.InventoryManagementClient.csproj...
Restoring packages for J:\git\FlixOne\FlixOne.InventoryManagementClient\FlixOne.InventoryManagementClient.csproj...
Generating MSBuild file J:\git\FlixOne\FlixOne.InventoryManagementClient\obj\FlixOne.InventoryManagementClient.csproj.nuget.g.props.
Generating MSBuild file J:\git\FlixOne\FlixOne.InventoryManagementClient\obj\FlixOne.InventoryManagementClient.csproj.nuget.g.targets.
Restore completed in 197.76 ms for J:\git\FlixOne\FlixOne.InventoryManagementClient\FlixOne.InventoryManagementClient.csproj.

Restore succeeded.
```

```
PS J:\git\FlixOne> dotnet new mstest --name FlixOne.InventoryManagementTests
The template "Unit Test Project" was created successfully.

Processing post-creation actions...
Running 'dotnet restore' on FlixOne.InventoryManagementTests\FlixOne.InventoryManagementTests.csproj...
Restoring packages for J:\git\FlixOne\FlixOne.InventoryManagementTests\FlixOne.InventoryManagementTests.csproj...
Installing Microsoft.TestPlatform.ObjectModel 15.7.0.
Installing Microsoft.TestPlatform.TestHost 15.7.0.
Installing MSTest.TestFramework 1.2.1.
Installing MSTest.TestAdapter 1.2.1.
Installing Microsoft.NET.Test.Sdk 15.7.0.
Generating MSBuild file J:\git\FlixOne\FlixOne.InventoryManagementTests\obj\FlixOne.InventoryManagementTests.csproj.nuget.g.props.
Generating MSBuild file J:\git\FlixOne\FlixOne.InventoryManagementTests\obj\FlixOne.InventoryManagementTests.csproj.nuget.g.targets.
Restore completed in 2.71 sec for J:\git\FlixOne\FlixOne.InventoryManagementTests\FlixOne.InventoryManagementTests.csproj.

Restore succeeded.
```



Test Explorer ▾ X

Run All | Run... | Playlist : All Tests ▾

Chapter1 (5 tests)

- ◀ ⚠ FlixOne.InventoryManagementTests (5) 30 ms
 - ⚠ AddInventoryCommand_Successful 24 ms
 - ⚠ GetInventoryCommand_Successful 1 ms
 - ⚠ HelpCommand_Successful < 1 ms
 - ⚠ QuitCommand_Successful < 1 ms
 - ⚠ UpdateQuantity_Successful < 1 ms

UpdateQuantity_Successful [Copy All](#)

Source: [UpdateQuantityCommandTests.cs](#) line 26

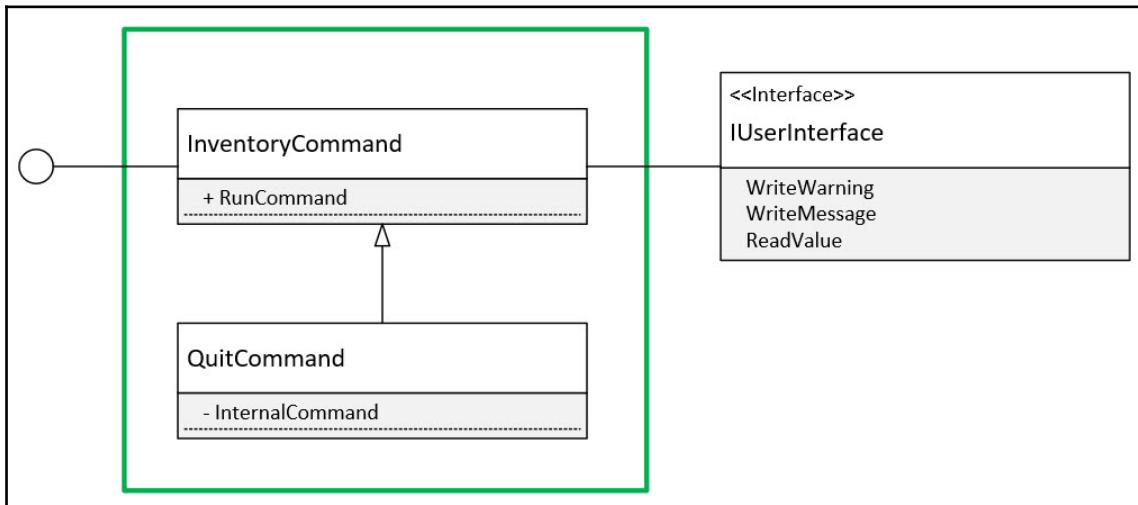
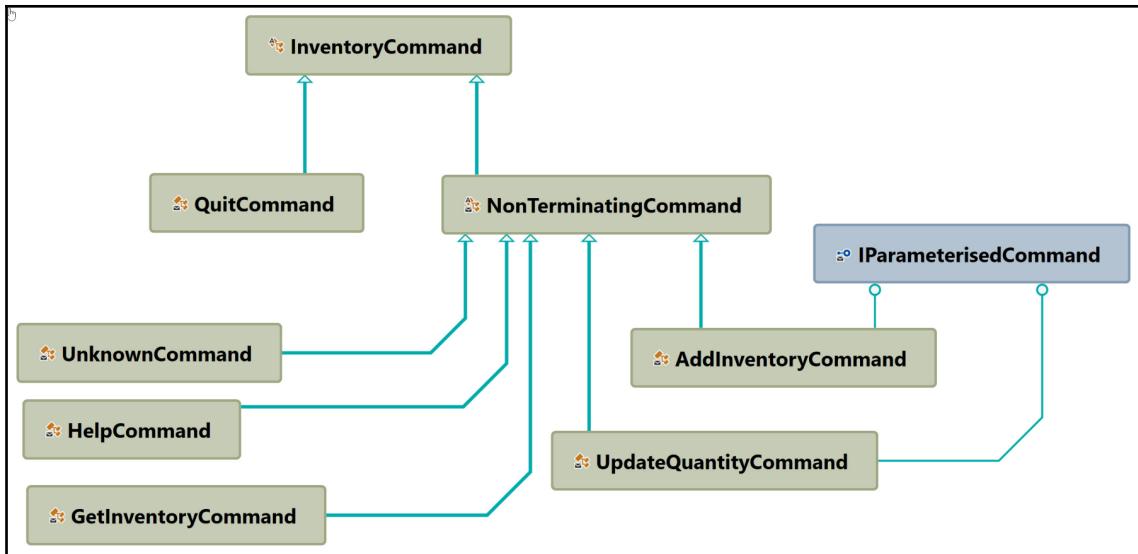
⚠ Test Skipped - UpdateQuantity_Successful

Message: Assert.Inconclusive failed. UpdateQuantity_Successful has not been implemented.

Elapsed time: 0:00:00.0009626

◀ StackTrace:

[UpdateQuantityCommandTests.UpdateQuantity_Successful\(\)](#)



Chapter 4: Implementing Design Patterns - Basics Part 2

The screenshot shows a test runner interface with the following details:

- Project: FlixOne (13 tests)
- Test Suite: InventoryContextTests (1 test)
- Test Result: MaintainBooks_Successful (7 ms, successful)
- Other Tests:
 - AddInventoryCommandTests (1)
 - GetInventoryCommandTests (1)
 - HelpCommandTests (1)
 - InventoryCommandFactoryTests (6)
 - QuitCommandTests (1)
 - UnknownCommandTests (1)
 - UpdateQuantityCommandTests (1)

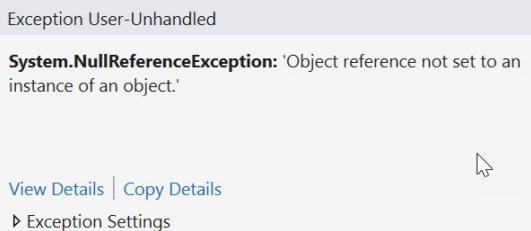
```
public bool UpdateQuantity(string name, int quantity)
{
    _books[name].Quantity += quantity;
    return true;
}
```

A tooltip for the variable `_books` is shown, indicating it has a count of 28.

Graphic Bundle

```
public bool AddBook(string name)
{
    _books.Add(name, new Book { Name = name });
    return true;
}

3 references
public bool UpdateQuantity(string name, int quantity)
{
    _books[name].Quantity += quantity;
    return true;
}
```



```
public bool UpdateQuantity(string name, int quantity)
{
    _books[name].Quantity += quantity;
    return true;
}
```

_books | Count = 27

Test Explorer

Run All | Run... | Playlist : All Tests

FlixOne (13 tests) 1 failed

- ▷ ✓ AddInventoryCommandTests (1) 14 ms
- ▷ ✓ GetInventoryCommandTests (1) 4 ms
- ▷ ✓ HelpCommandTests (1) < 1 ms
- ▷ ✓ InventoryCommandTests (1) 6 ms
- ◀ ✗ InventoryContextTests (1) 51 ms
 - ✗ **MaintainBooks_Successful** 51 ms
 - ▷ ✓ QuitCommandTests (1) < 1 ms
 - ▷ ✓ UnknownCommandTests (1) < 1 ms
 - ◀ ✓ UpdateQuantityCommandTests (1) 1 ms
 - ✓ UpdateQuantity_ExistingBook_Successful 1 ms

MaintainBooks_Successful Copy All

Source: [InventoryContextTests.cs](#) line 15

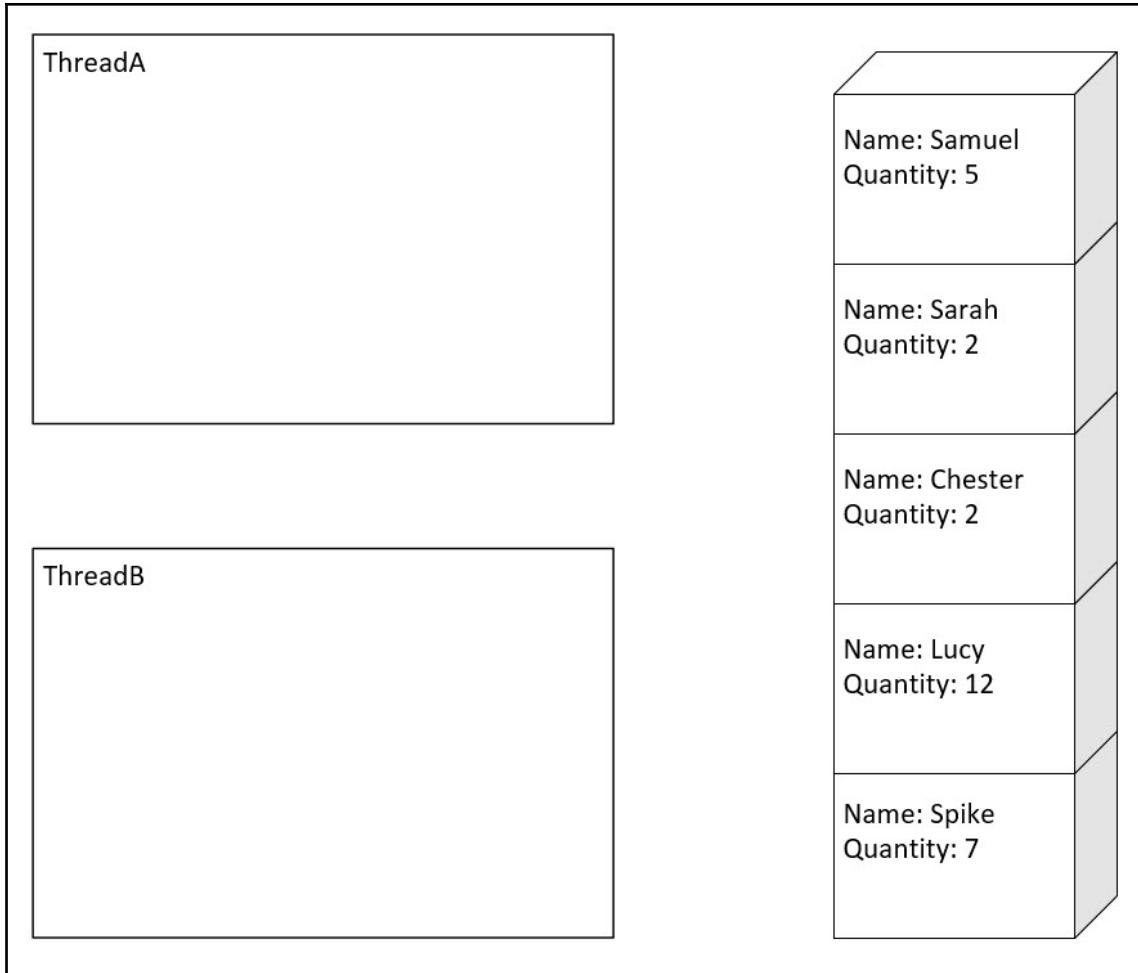
✗ Test Failed - MaintainBooks_Successful

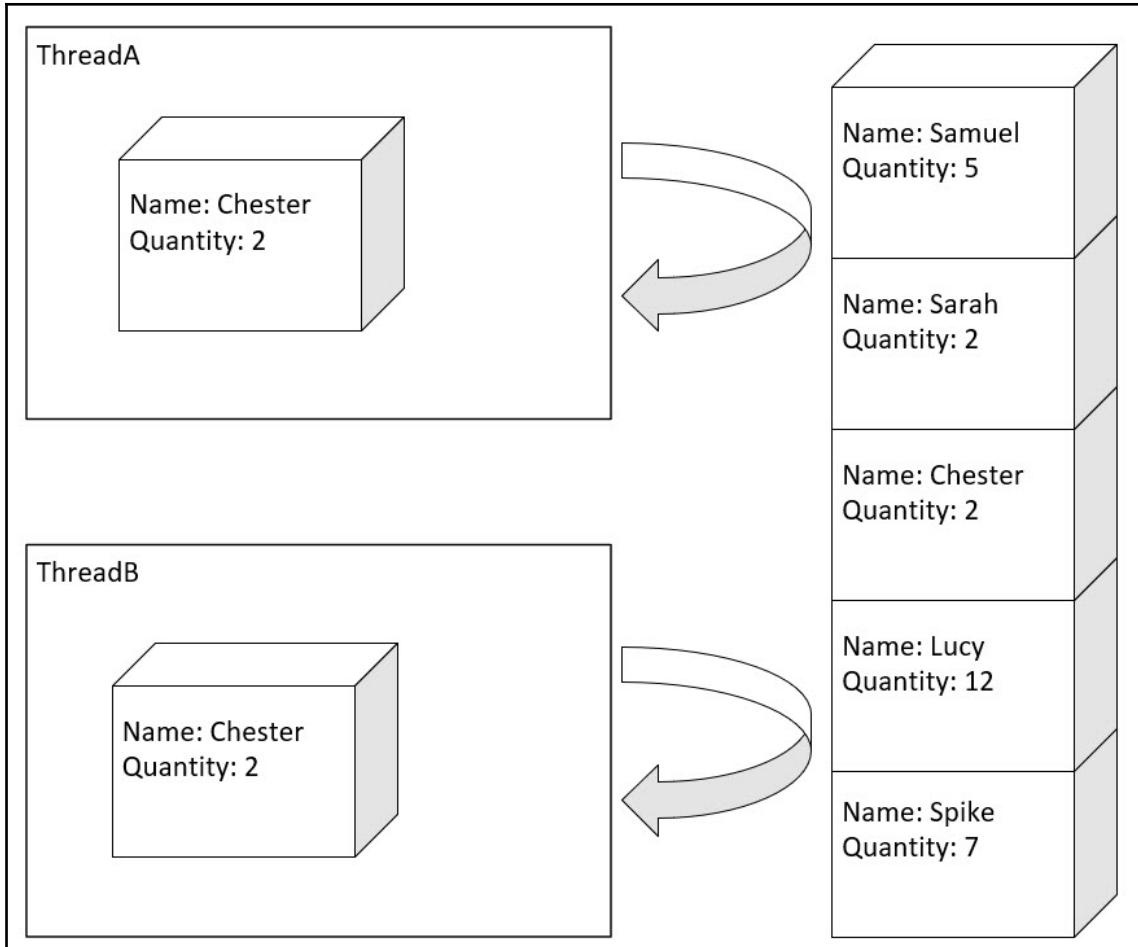
Message: Assert.AreEqual failed. Expected:<0>. Actual:<-4>.

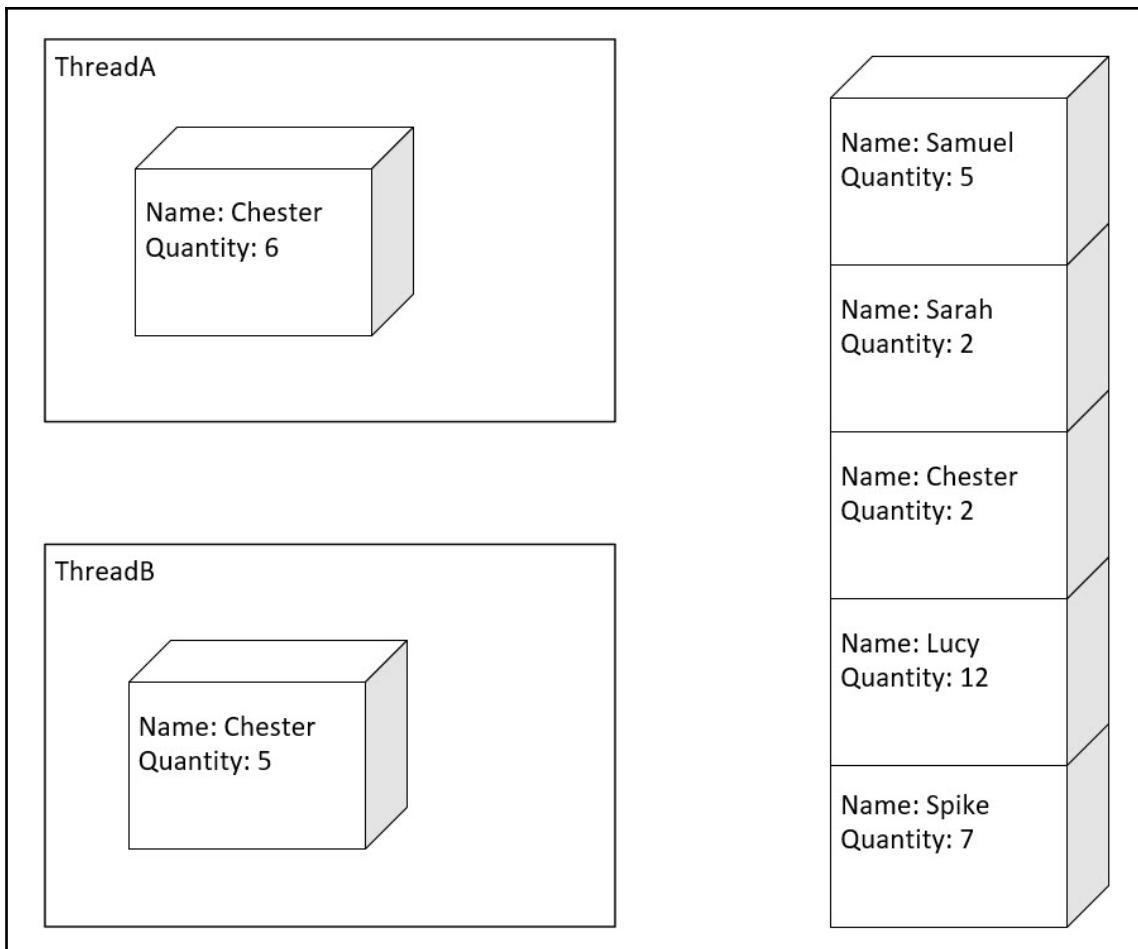
Elapsed time: 0:00:00.0514803

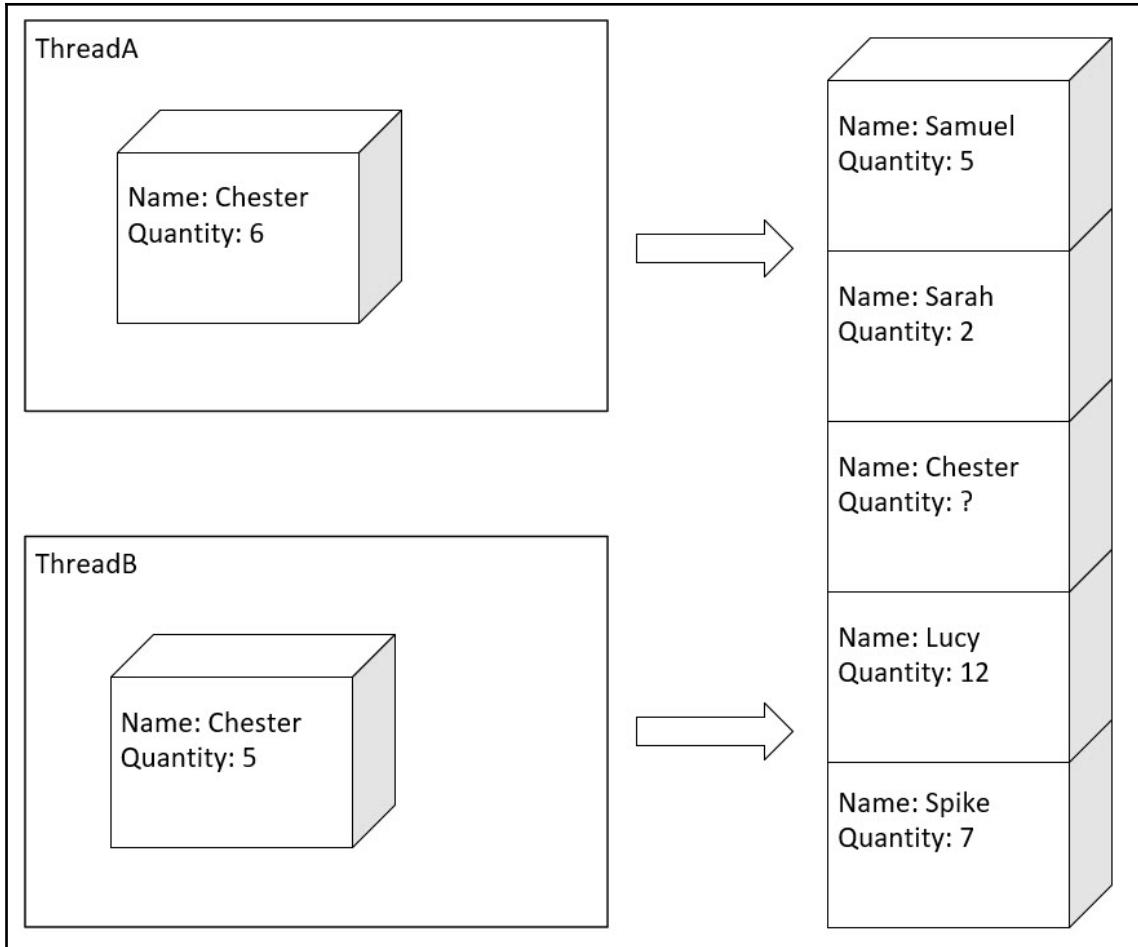
◀ StackTrace:

[InventoryContextTests.MaintainBooks_Successful\(\)](#)









- The application is a console application
 - print a welcome message that includes the version of the assembly
 - loops until a quit command is given
 - if a given command is not successful or not understood, then print a helpful message
- The application is driven by simple case-insensitive text commands
- Each command has a short form of a single character and a long form

- The application is a console application
 - print a welcome message that includes the version of the assembly
 - loops until a quit command is given
 - if a given command is not successful or not understood, then print a helpful message
- The application is driven by simple case-insensitive text commands
- Each command has a short form of a single character and a long form

```
*****
*                               Welcome to FlixOne Inventory Management System
*                               v1.0.0.0
*****
USAGE:
    addinventory (a)
    getinventory (g)
    updatequantity (u)
    quit (q)
    ?
Examples:
    New Inventory
    > addinventory
    Enter name:The Meaning of Life

    Get Inventory
    > getinventory
    The Meaning of Life      Quantity:10
    The Life of a Ninja     Quantity:2

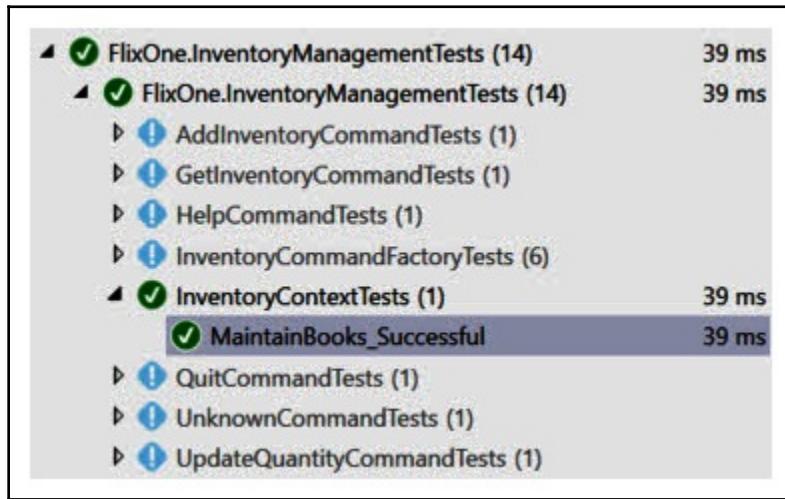
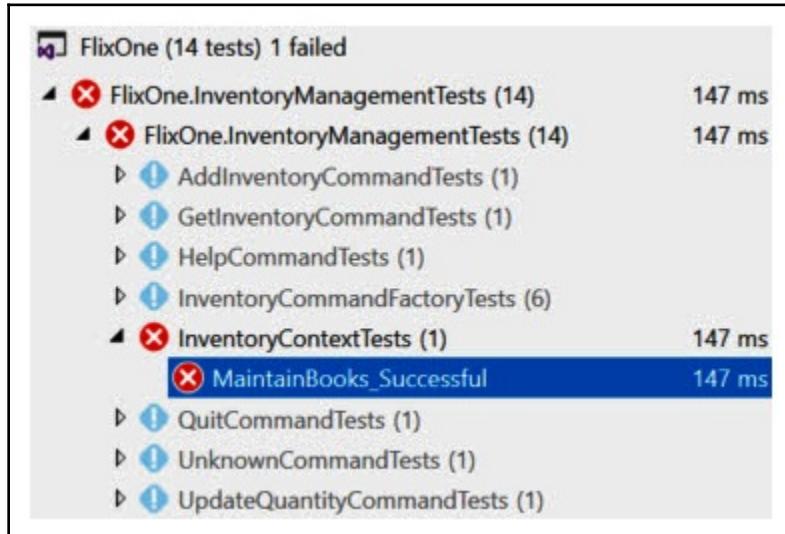
    Update Quantity (Increase)
    > updatequantity
    Enter name:The Meaning of Life
    11
    11 added to quantity

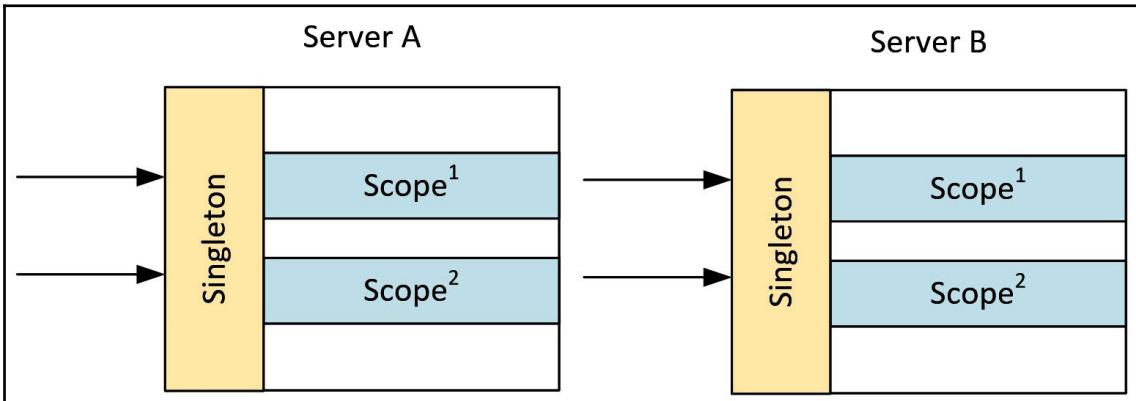
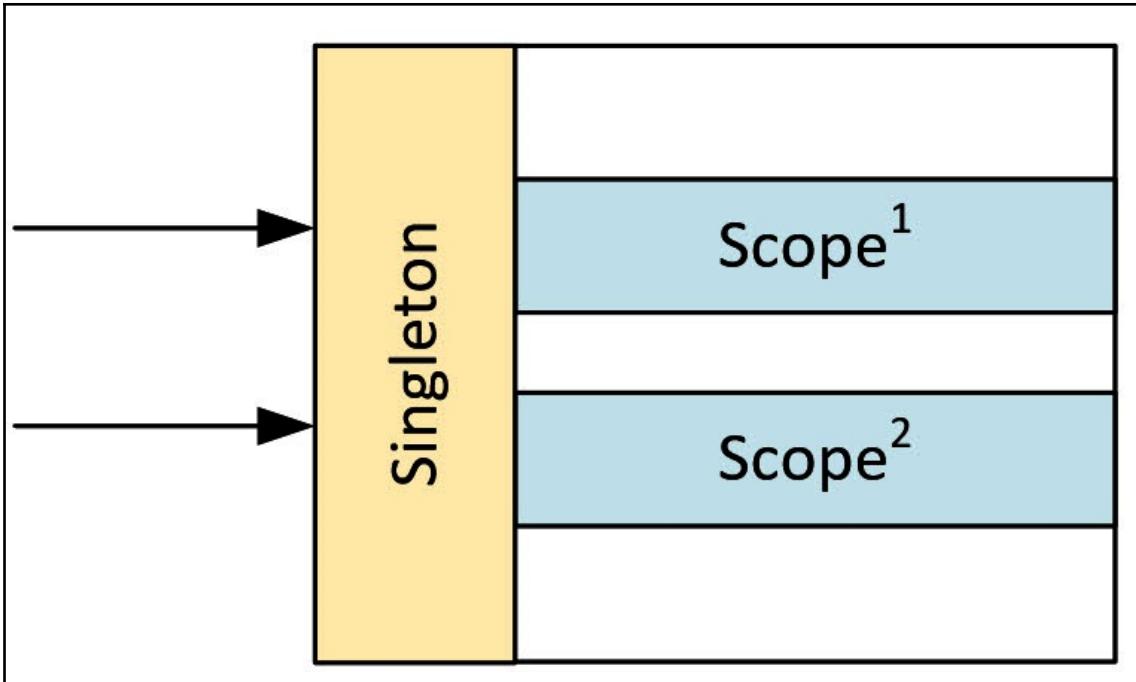
    Update Quantity (Decrease)
    > updatequantity
    Enter name:The Life of a Ninja
    -3
    3 removed from quantity

    > -
```

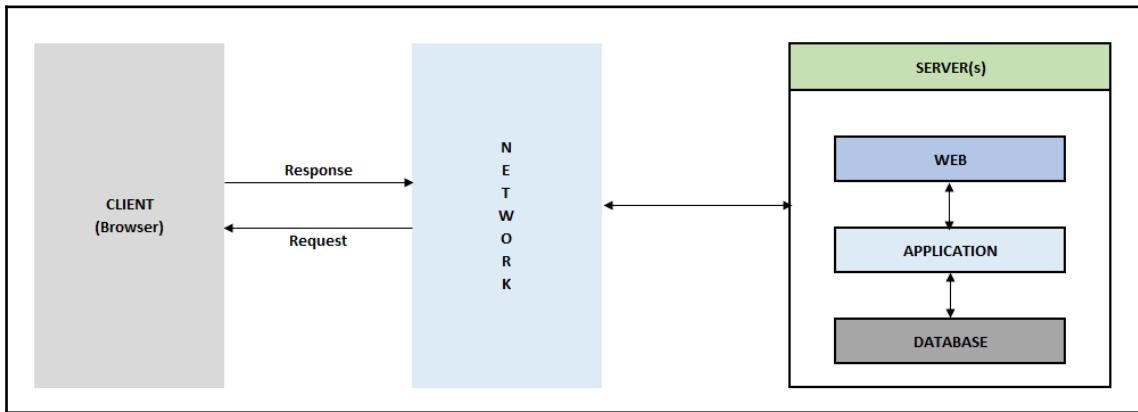
```
> addinventory
Enter name:Godfather
> u
Enter name:Godfather
Enter quantity:12
> g
Godfather           Quantity:12
> notsurecommand
Unable to determine the desired command.
Enter ? to view options.
> q
Thank you for using FlixOne Inventory Management System
CatalogService has completed.
```

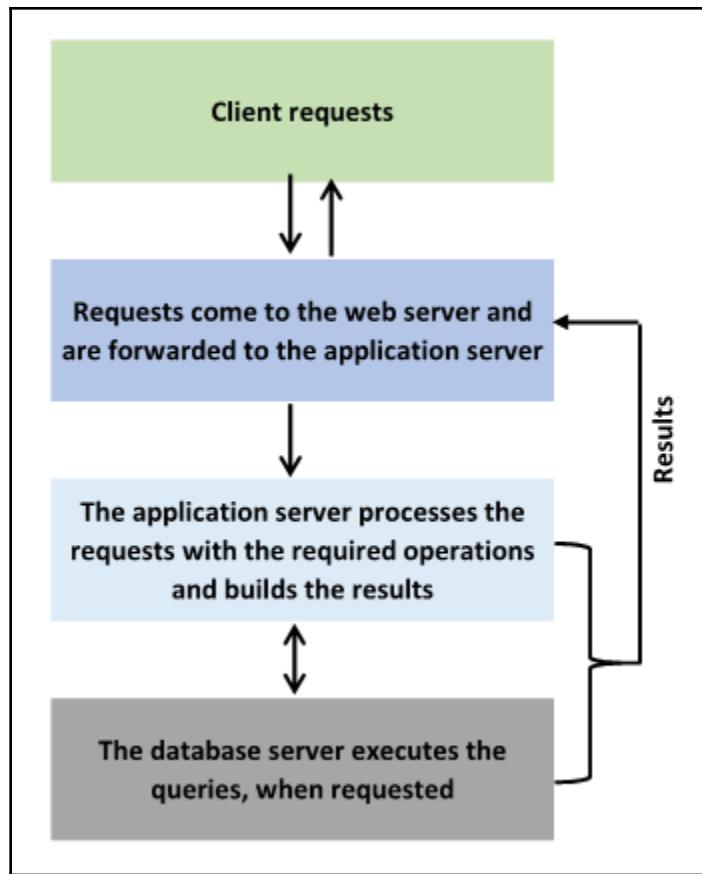
Chapter 5: Implementing Design Patterns - .NET Core

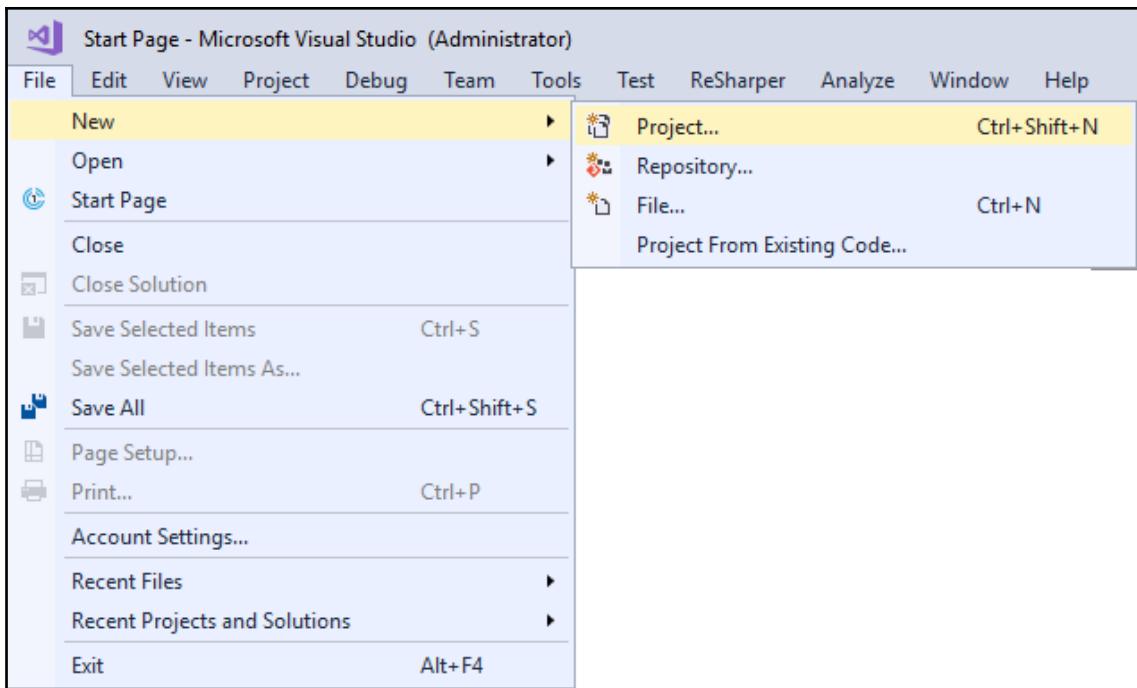


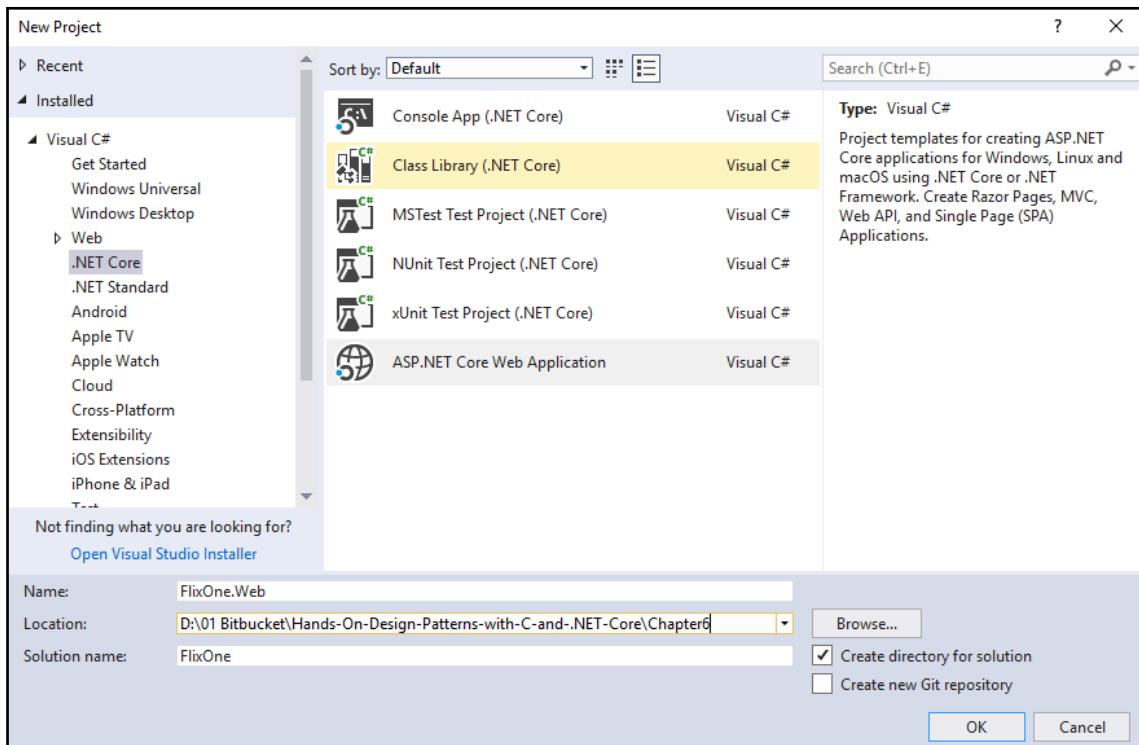


Chapter 6: Implementing Design Patterns for Web Applications - Part 1

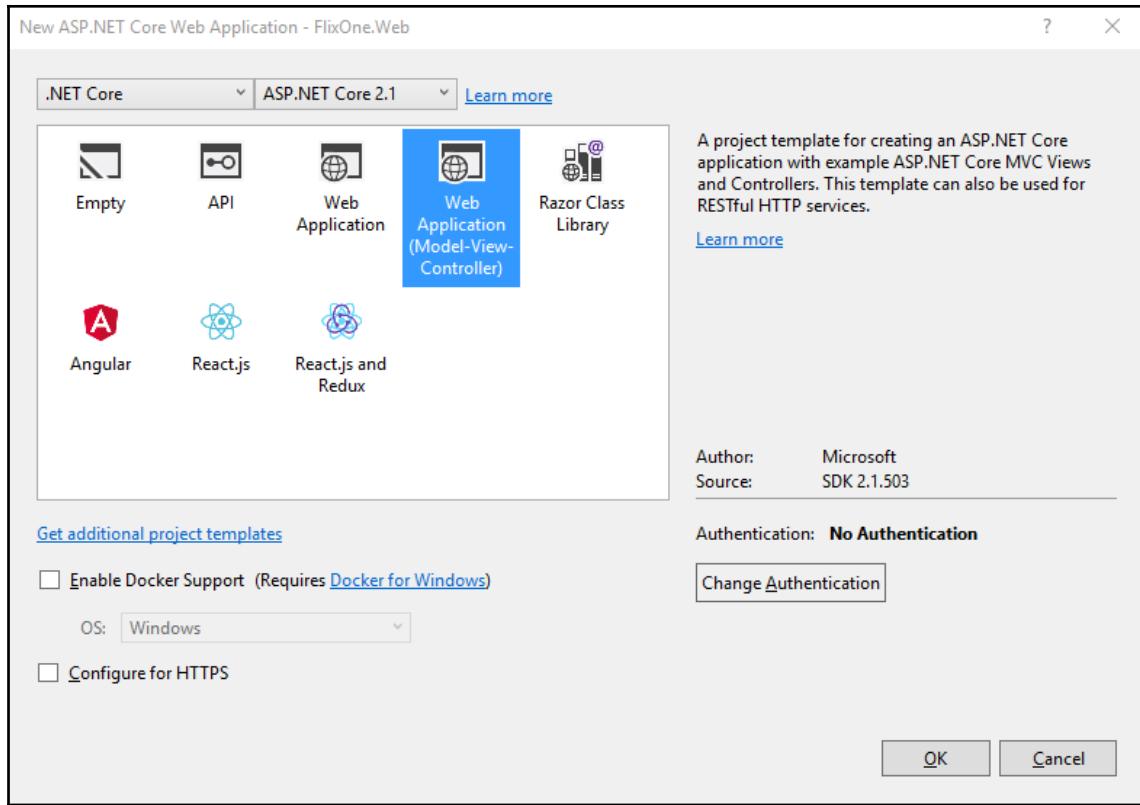


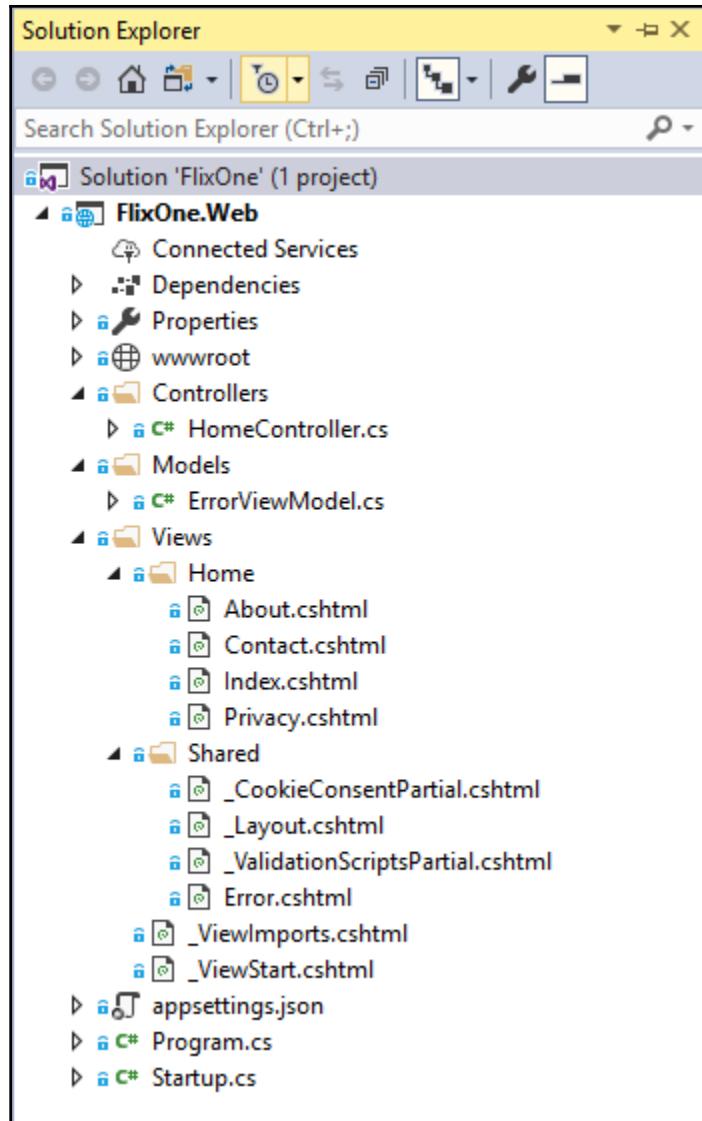






Graphic Bundle





FlixOne.Web Home About Contact

Microsoft Azure

Learn how Microsoft's Azure cloud platform allows you to build, deploy, and scale web apps. [Learn More](#)

Application uses

- Sample pages using ASP.NET Core MVC
- Theming using Bootstrap

How to

- Add a Controller and View
- Manage User Secrets using Secret Manager
- Use logging to log a message
- Add packages using NuGet
- Target development, staging or production environment

Overview

- Conceptual overview of what is ASP.NET Core
- Fundamentals of ASP.NET Core such as Startup and middleware
- Working with Data
- Security
- Client side development
- Develop on different platforms
- Read more on the documentation site

Run & Deploy

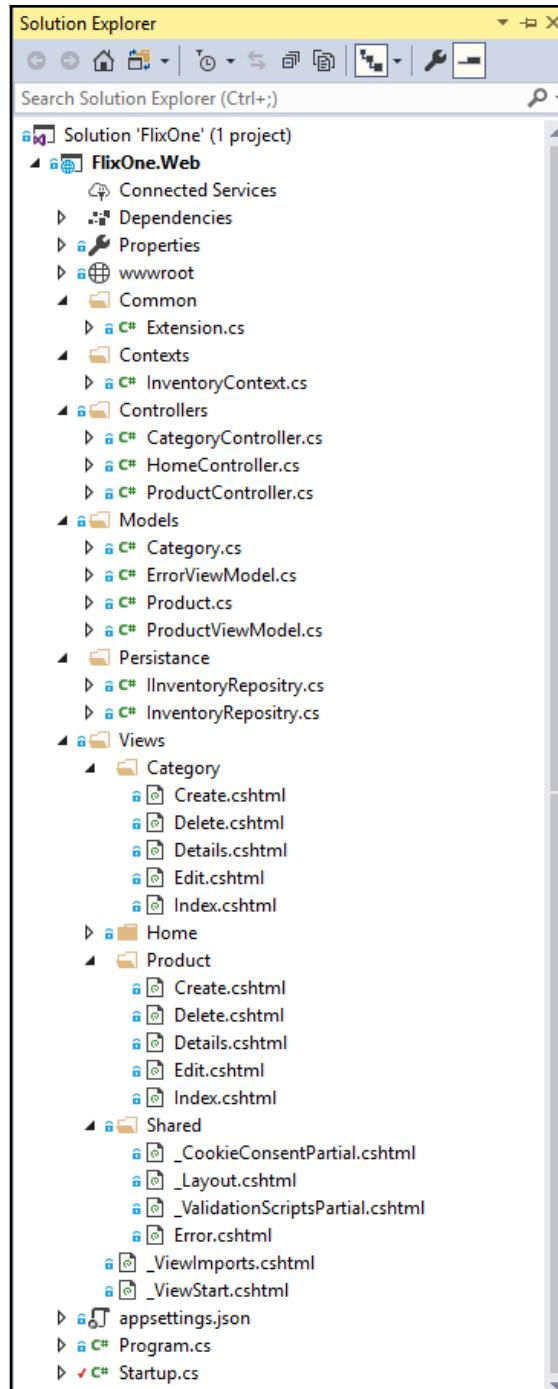
- Run your app
- Run tools such as EF migrations and more
- Publish to Microsoft Azure Web Apps

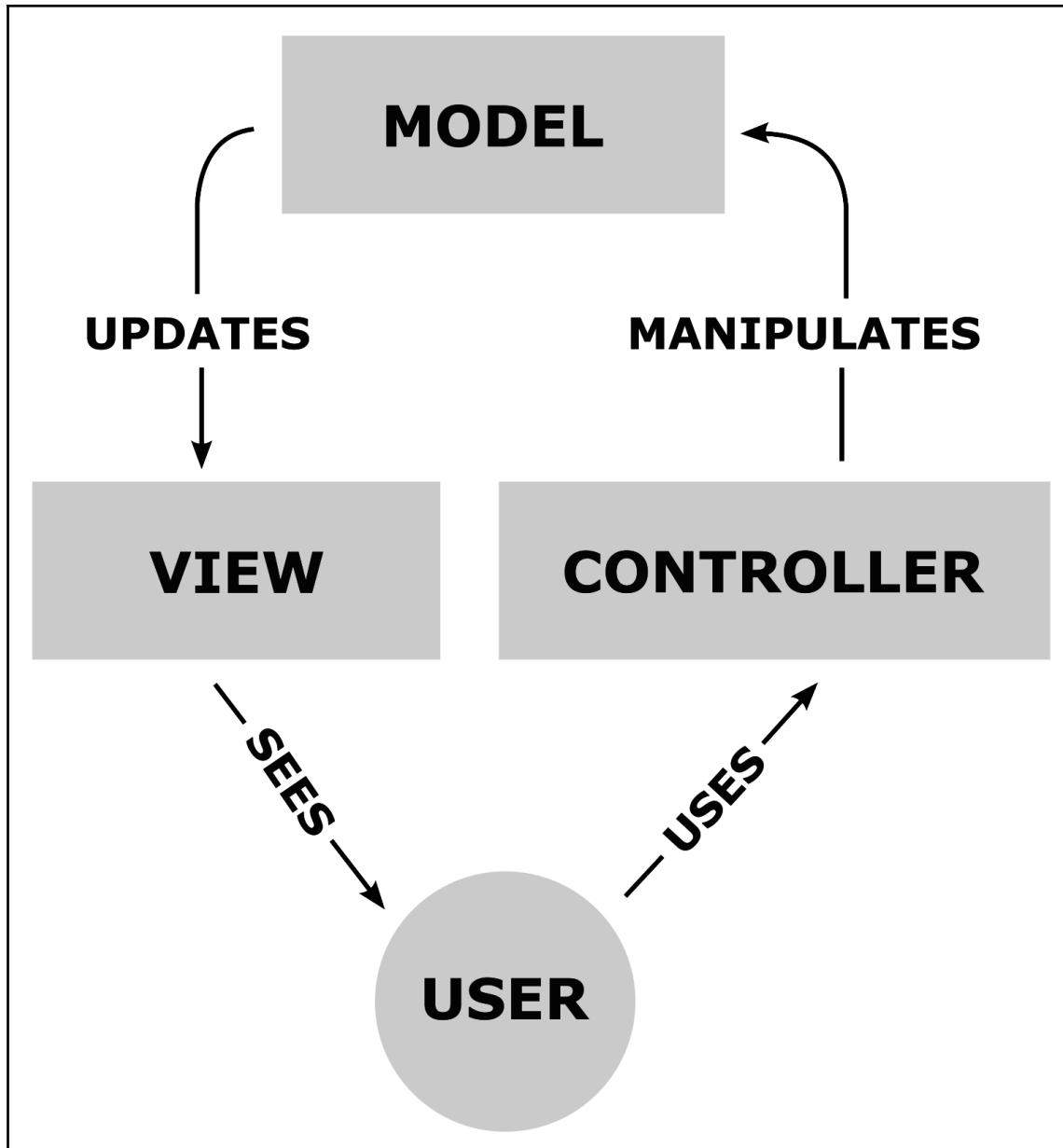
An unhandled exception occurred while processing the request.

InvalidOperationException: Unable to resolve service for type 'FlixOne.Web.Persistance.IInventoryRepository' while attempting to activate 'FlixOne.Web.Controllers.ProductController'.
Microsoft.Extensions.DependencyInjection.ActivatorUtilities.GetService(IServiceProvider sp, Type type, Type requiredBy, bool isDefaultParameterRequired)

```
// This method gets called by the runtime. Use this method to add services to the container.
public void ConfigureServices(IServiceCollection services)
{
    services.AddTransient<IInventoryRepository, InventoryRepository>();
    services.AddDbContext<InventoryContext>(o => o.UseSqlServer(Configuration.GetConnectionString("FlixOneDbConnection")));
    services.Configure<CookiePolicyOptions>(options =>
    {
        // This lambda determines whether user consent for non-essential cookies is needed for a given request.
        options.CheckConsentNeeded = context => true;
        options.MinimumSameSitePolicy = SameSiteMode.None;
    });

    services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_1);
}
```





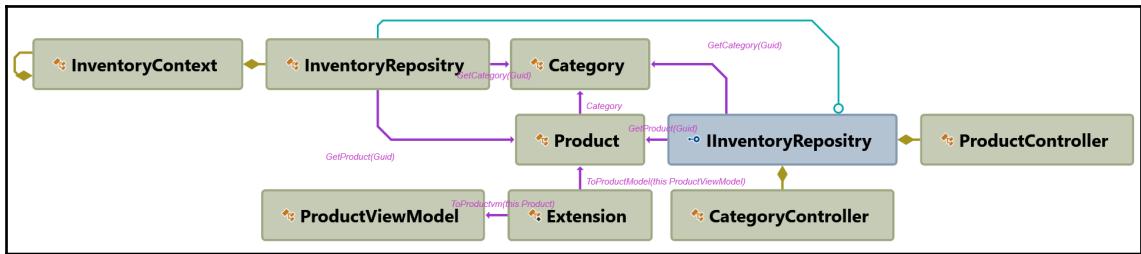
FlixOne.Web Home Products About Contact

Product Listing

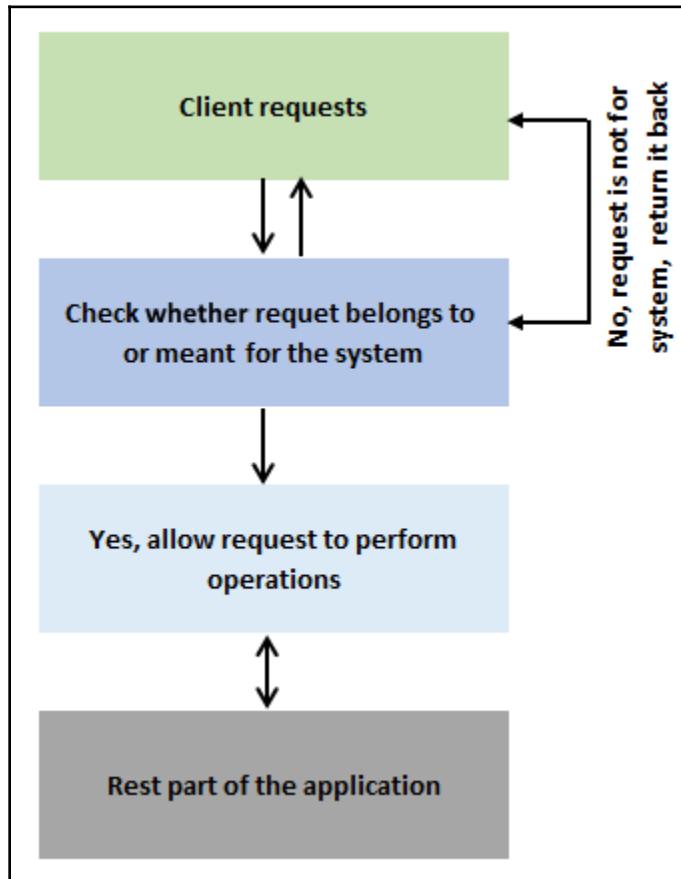
Create New

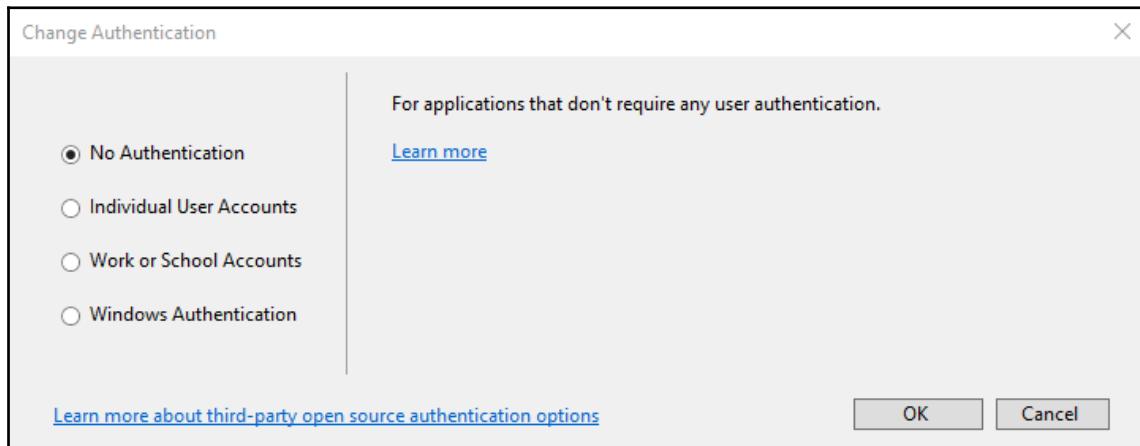
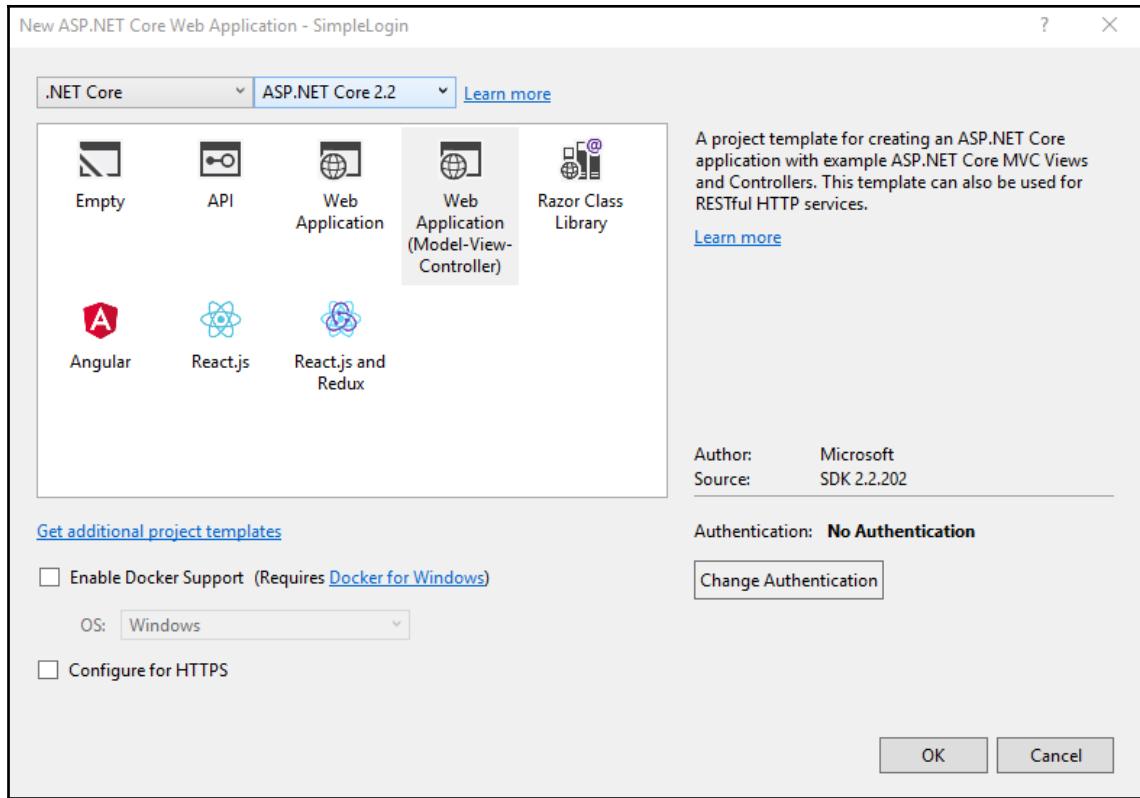
Product Name	Description	Image	Price (INR)	Category Name	Action
Mango	A juicy mango		40.00	Fruit	Edit Details Delete
Apple	Red apple		100.00	Fruit	Edit Details Delete
Orange	Fruity oranges		35.00	Fruit	Edit Details Delete

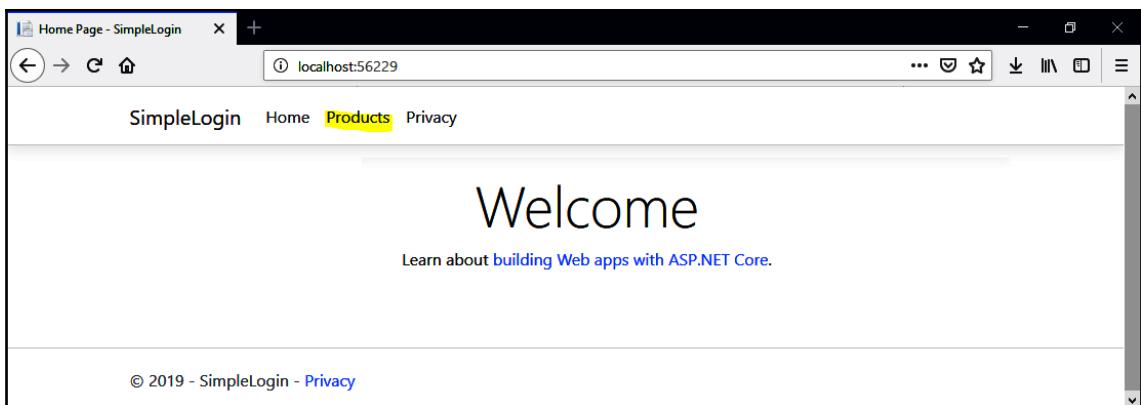
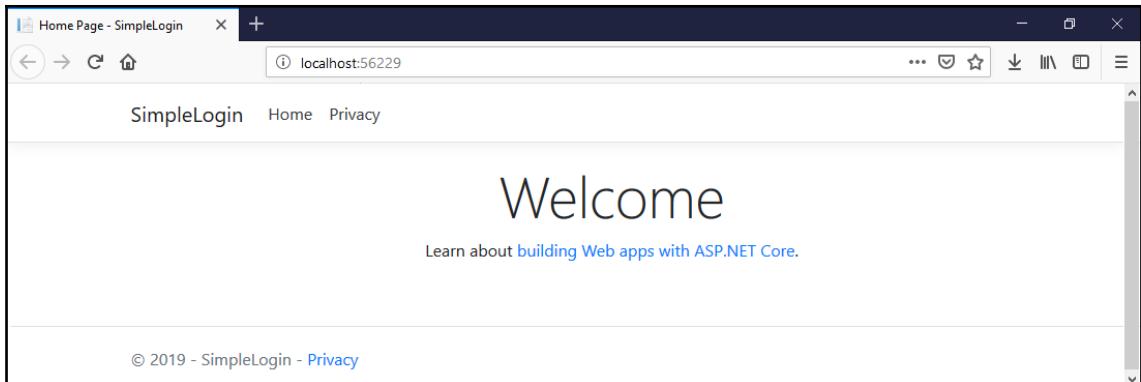
© 2019 - FlixOne.Web



Chapter 7: Implementing Design Patterns for Web Applications - Part 2







Product Name	Description	Image	Price (INR)	Category Name	Action
Mango	A juicy mango		40.00	Fruit	Edit Details Delete
Apple	Red apple		100.00	Fruit	Edit Details Delete
Orange	Fruity oranges		35.00	Fruit	Edit Details Delete

© 2019 - SimpleLogin - [Privacy](#)

```
public class InventoryContext : DbContext
{
    public InventoryContext(DbContextOptions<InventoryContext> options)
        : base(options)
    {
    }

    public InventoryContext()
    {
    }

    public DbSet<Product> Products { get; set; }
    public DbSet<Category> Categories { get; set; }
    public DbSet<User> Users { get; set; }
}
```

Login to the system

Username

Password

Remember Me on current System

[Login](#)

Not a member, [Register](#).

Login to the system

Username

Password

Remember Me on current System

Login

Not a member, [Register](#).

SimpleLogin [Home](#) [Products](#) [Privacy](#) **Welcome Gaurav**, [Logout](#)

Welcome

This sample application demonstrate the simplest way to authenticate users.

© 2019 - SimpleLogin - [Privacy](#)

Login to the system

• Invalid login attempt

Username

Password

Remember Me on current System

Login

Not a member, [Register](#).

SimpleLogin [Home](#) [Products](#) [Privacy](#) [Login](#)

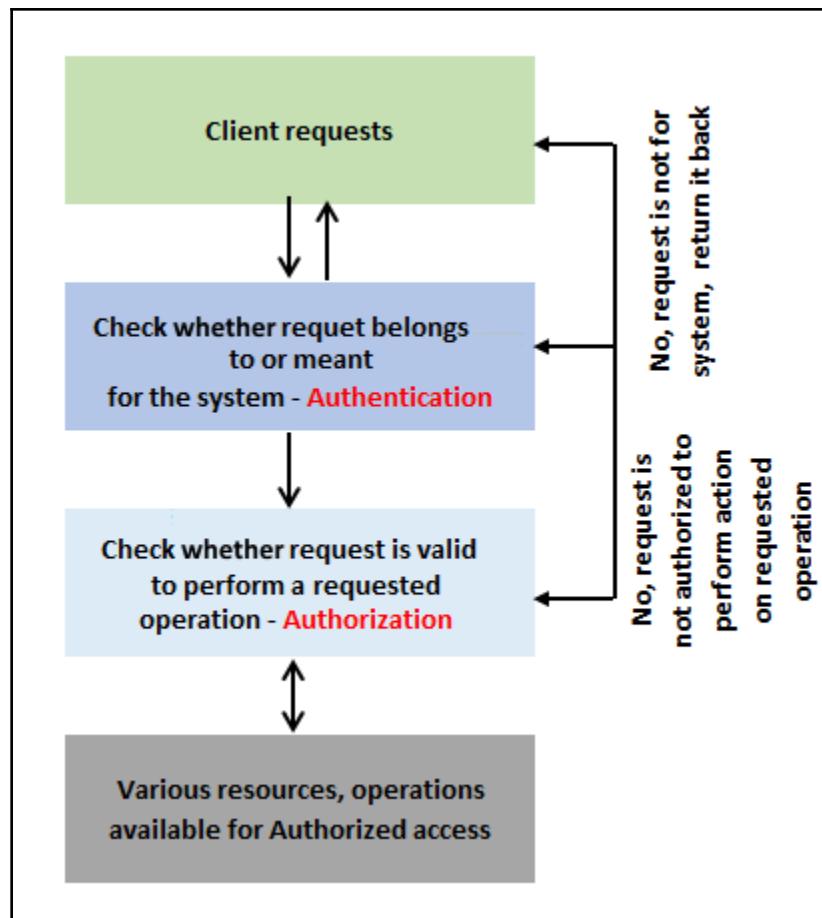
SimpleLogin [Home](#) [Products](#) [Privacy](#) **Welcome Gaurav**, [Logout](#)

SimpleLogin Home Privacy Login

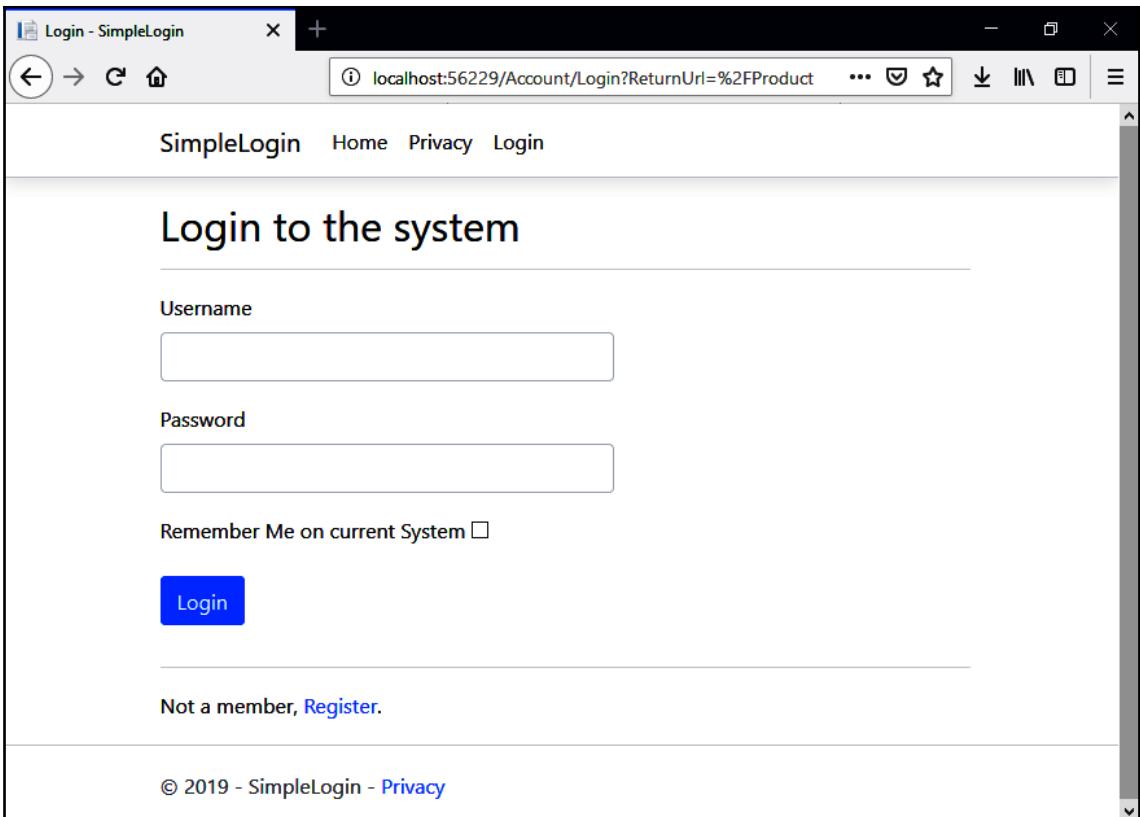
Welcome

This sample application demonstrate the simplest way to authenticate users.

© 2019 - SimpleLogin - [Privacy](#)



```
8  namespace SimpleLogin.Controllers
9  {
10     [Authorize]
11     public class ProductController : Controller
12     {
13         private readonly IInventoryRepository _inventoryRepository;
14
15         public ProductController(IInventoryRepository inventoryRepository) => _inventoryRepository = inventoryRepository;
16
17         public IActionResult Index() => View(_inventoryRepository.GetProducts().ToProductvm());
18
19         public IActionResult Details(Guid id) => View(_inventoryRepository.GetProduct(id).ToProductvm());
20
21         public IActionResult Create() => View();
--
```



Product Listing

[Create New](#)

Product Name	Description	Image	Price (INR)	Category Name	Action
Mango	A juicy mango		40.00	Fruit	Edit Details Delete
Apple	Red apple		100.00	Fruit	Edit Details Delete
Orange	Fruity oranges		35.00	Fruit	Edit Details Delete

```
public class InventoryContext : DbContext
{
    public InventoryContext(DbContextOptions<InventoryContext> options)
        : base(options)
    {
    }

    public InventoryContext()
    {
    }

    public DbSet<Product> Products { get; set; }
    public DbSet<Category> Categories { get; set; }
    public DbSet<User> Users { get; set; }
    public DbSet<Role> Roles { get; set; }
}
```

```
private void SignInCookie(LoginViewModel model, User user)    model = {LoginViewModel}, user = {User}
{
    var claims = new List<Claim>  claims = Count = 5
    {
        new Claim(ClaimTypes.Name, user.FirstName),
        new Claim(ClaimTypes.Email, user.EmailId),
        new Claim(ClaimTypes.NameIdentifier, user.Id.ToString())
    };

    if (user.Roles != null)
    {
        string[] roles = user.Roles.Split(",");
        roles = {string[2]}

        claims.AddRange(roles.Select(role => new Claim(ClaimTypes.Role, role)));
    }
}

var identity = claims[0];
var principal = claims[1];
var props = new AuthenticationProperties { IsPersistent = model.RememberMe };

.HttpContext.SignInAsync(CookieAuthenticationDefaults.AuthenticationScheme, principal, props).Wait();
}
```

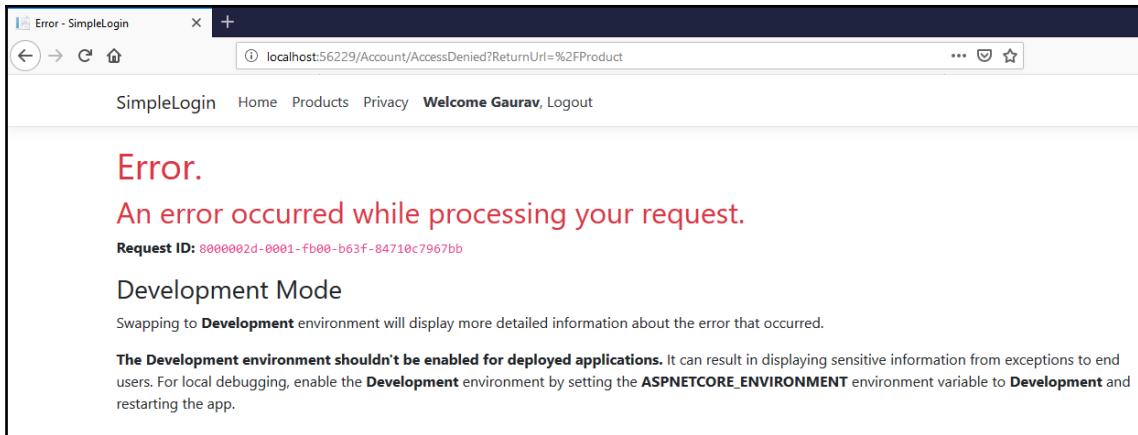
[Authorize(Roles = "Admin,Manager")]

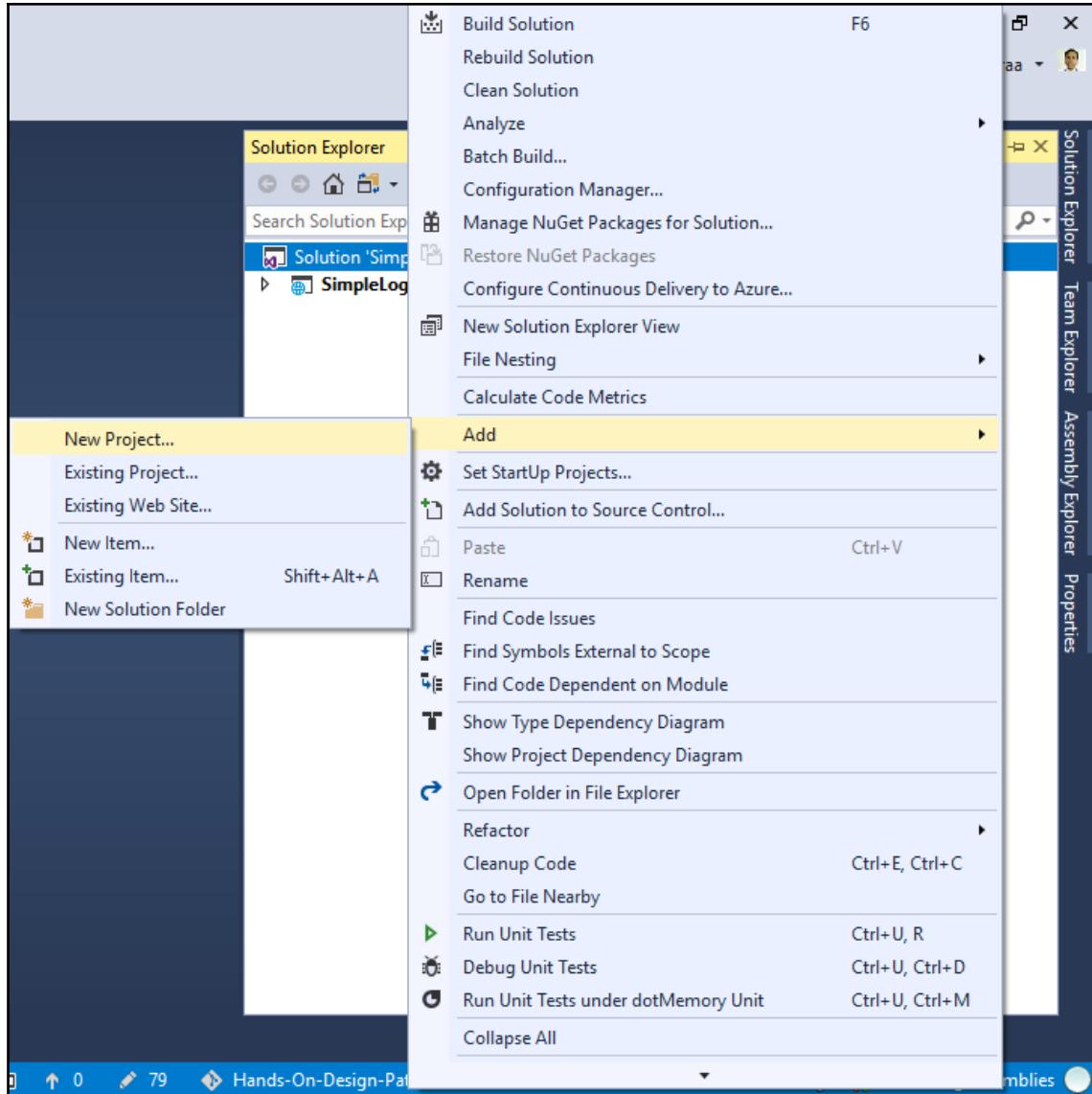
SimpleLogin Home Products Privacy **Welcome Gaurav**, Logout

Product Listing

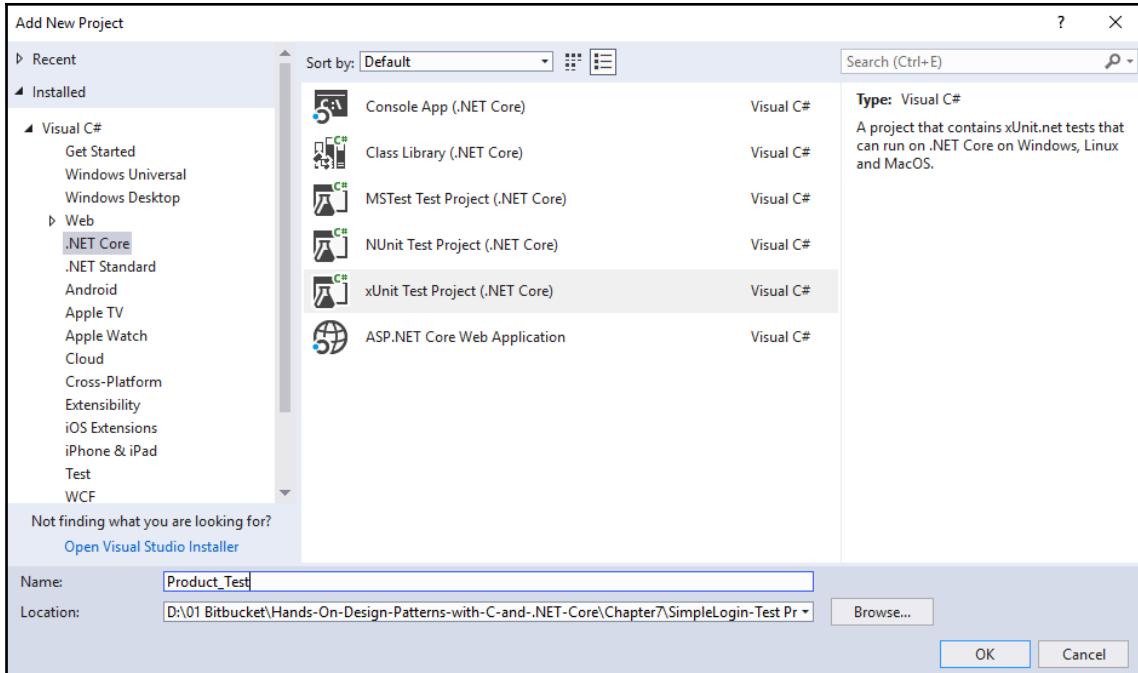
[Create New](#)

Product Name	Description	Image	Price (INR)	Category Name	Action
Mango	A juicy mango		40.00	Fruit	Edit Details Delete
Apple	Red apple		100.00	Fruit	Edit Details Delete
Orange	Fruity oranges		35.00	Fruit	Edit Details Delete





Graphic Bundle



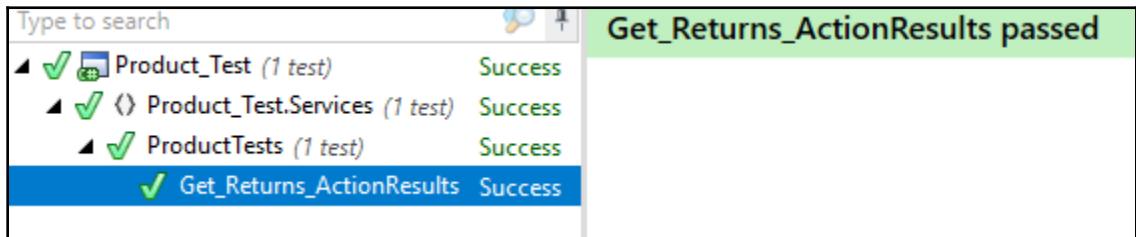
The screenshot shows a code editor with the following C# test code:

```
namespace Product_Test.Services
{
    public class ProductTests
    {
        [Fact]
        public void Get_Returns_ActionResults()
        {
            // Arrange
            var mockRepo = new Mock<IProductRepository>();
            mockRepo.Setup(repo => repo.[< ->]).  
The type or namespace name 'Mock<>' could not be found (are you missing a using directive or an assembly reference?)  
var controller = new ProductController(mockRepo);  
The type or namespace name 'IProductRepository' could not be found (are you missing a using directive or an assembly reference?)  
// Act
            var result = controller.GetList();  
Cannot resolve symbol 'IProductRepository'  
Show potential fixes (Ctrl+.)  
// Assert
            var viewResult = Assert.IsType<OkObjectResult>(result);
            var model = Assert.IsAssignableFrom<IEnumerable<ProductViewModel>>(viewResult.Value);
            Assert.NotNull(model);
            Assert.Equal(2, model.Count());
        }
    }
}
```

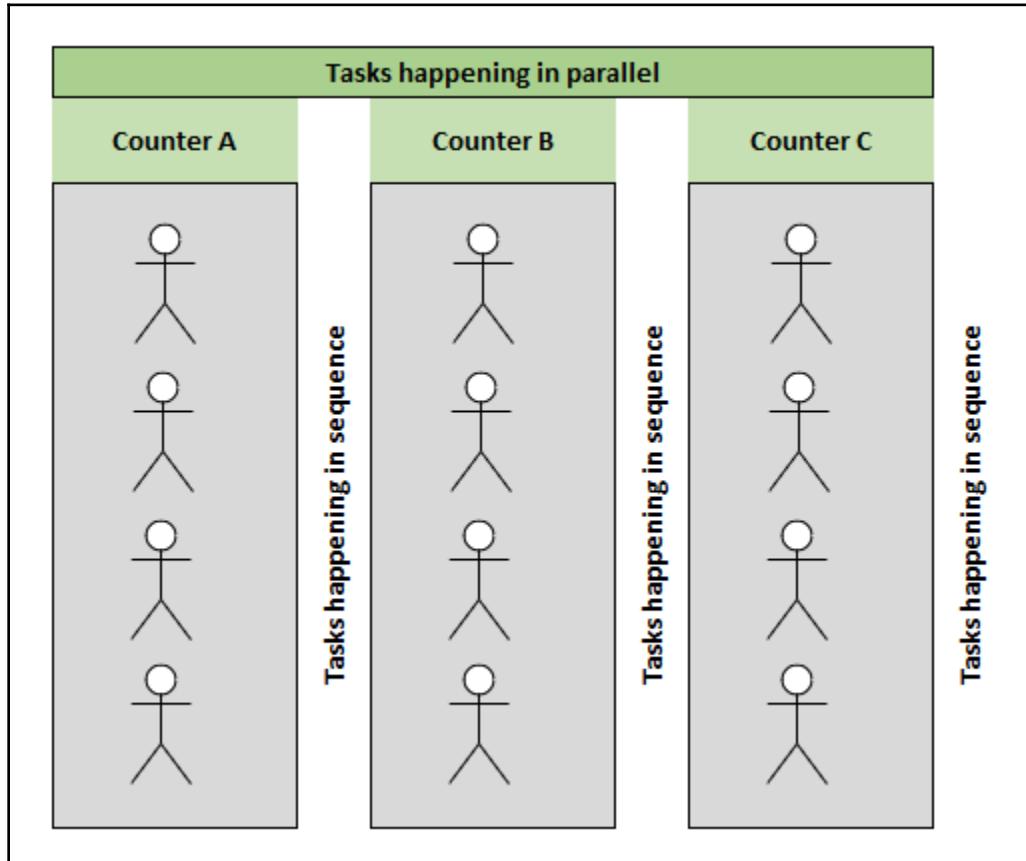
A tooltip is displayed over the 'Mock<IProductRepository>' line, indicating that the type or namespace 'Mock<IProductRepository>' could not be found, suggesting a missing using directive or assembly reference.

Graphic Bundle

```
Package Manager Console
Package source: All | Default project: Product_Test
PM> install-package moq
Restoring packages for D:\01 Bitbucket\Hands-On-Design-Patterns-with-C-and-.NET-Core\Chapter7\SimpleLogin-Test Project\Product_Test\Product_Test.csproj...
GET https://api.nuget.org/v3-flatcontainer/moq/index.json
OK https://api.nuget.org/v3-flatcontainer/moq/index.json 1022ms
GET https://api.nuget.org/v3-flatcontainer/moq/4.10.1/moq.4.10.1.nupkg
OK https://api.nuget.org/v3-flatcontainer/moq/4.10.1/moq.4.10.1.nupkg 12ms
GET https://api.nuget.org/v3-flatcontainer/system.reflection.typeextensions/index.json
OK https://api.nuget.org/v3-flatcontainer/system.reflection.typeextensions/index.json 1391ms
GET https://api.nuget.org/v3-flatcontainer/system.reflection.typeextensions/4.5.1/system.reflection.typeextensions.4.5.1.nupkg
OK https://api.nuget.org/v3-flatcontainer/system.reflection.typeextensions/4.5.1/system.reflection.typeextensions.4.5.1.nupkg 17ms
Installing System.Reflection.TypeExtensions 4.5.1.
Installing Moq 4.10.1.
Installing NuGet package moq 4.10.1.
100 %
```



Chapter 8: Concurrent Programming in .NET Core



3-counters are serving...
Next person from row
Person A is collecting ticket from Counter A
Person C is collecting ticket from Counter C
Person B is collecting ticket from Counter B

```
3-counters are serving...
Person A is collecting ticket from Counter A
Person B is collecting ticket from Counter B
Next person from row
Person C is collecting ticket from Counter C
```

```
Person 1 is collecting ticket from counter A.
Person 2 is collecting ticket from counter A.
Person 3 is collecting ticket from counter A.
Person 4 is collecting ticket from counter A.
Person 5 is collecting ticket from counter A.
Person 1 is collecting ticket from counter B.
Person 2 is collecting ticket from counter B.
Person 3 is collecting ticket from counter B.
Person 4 is collecting ticket from counter B.
Person 5 is collecting ticket from counter B.
Person 1 is collecting ticket from counter C.
Person 2 is collecting ticket from counter C.
Person 3 is collecting ticket from counter C.
Person 4 is collecting ticket from counter C.
Person 5 is collecting ticket from counter C.
```

```
await Task.Delay(1000);
```

 (awaitable) `Task Task.Delay(int millisecondsDelay) (+ 3 overloads)`
Creates a task that completes after a time delay.

Usage:
`await Delay(...);`

Exceptions:
`ArgumentOutOfRangeException`

```
public async void ShowMessage()
```

This async method lacks 'await' operators and will run synchronously. Consider using the 'await' operator to await non-blocking API calls, or 'await Task.Run(...)' to do CPU-bound work on a background thread.

Show potential fixes (Ctrl+.)

Serving messages!

```
Person 1 is collecting ticket from counter A.  
Person 2 is collecting ticket from counter A.  
Person 3 is collecting ticket from counter A.  
Person 4 is collecting ticket from counter A.  
Person 5 is collecting ticket from counter A.  
Person 1 is collecting ticket from counter C.  
Person 2 is collecting ticket from counter C.  
Person 3 is collecting ticket from counter C.  
Person 4 is collecting ticket from counter C.  
Person 5 is collecting ticket from counter C.  
Person 1 is collecting ticket from counter B.  
Person 2 is collecting ticket from counter B.  
Person 3 is collecting ticket from counter B.  
Person 4 is collecting ticket from counter B.  
Person 5 is collecting ticket from counter B.
```

```
Person 1 is collecting ticket from counter C.  
Person 2 is collecting ticket from counter C.  
Person 3 is collecting ticket from counter C.  
Person 4 is collecting ticket from counter C.  
Person 5 is collecting ticket from counter C.  
Person 1 is collecting ticket from counter B.  
Person 2 is collecting ticket from counter B.  
Person 3 is collecting ticket from counter B.  
Person 4 is collecting ticket from counter B.  
Person 5 is collecting ticket from counter B.  
Person 1 is collecting ticket from counter A.  
Person 2 is collecting ticket from counter A.  
Person 3 is collecting ticket from counter A.  
Person 4 is collecting ticket from counter A.  
Person 5 is collecting ticket from counter A.
```

Id	Name	Qty
1	Book#1	10
2	Book#2	20
3	Book#3	30
4	Book#4	40
5	Book#5	50
6	Book#6	60
7	Book#7	70
8	Book#8	80
9	Book#9	90
13	Book#13	130
10	Book#10	100
14	Book#14	140
15	Book#15	150
11	Book#11	110
16	Book#16	160
12	Book#12	120
17	Book#17	170
18	Book#18	180
19	Book#19	190

Total Processes Running on the machine:4

Processing complete. Press any key to exit.

Id	Name	Qty
1	Book#1	10
2	Book#2	20
3	Book#3	30
4	Book#4	40
5	Book#5	50
6	Book#6	60
7	Book#7	70
8	Book#8	80
9	Book#9	90
10	Book#10	100
11	Book#11	110
12	Book#12	120
13	Book#13	130
14	Book#14	140
15	Book#15	150
16	Book#16	160
17	Book#17	170
18	Book#18	180
19	Book#19	190
Total Processes Running on the machine:4		
Processing complete. Press any key to exit.		

48775 books out of 50000 total books,having Qty in stock more than 12250.

Chapter 9: Functional Programming Practices

FlixOne.Web Home Products About Contact Privacy

Product Listing

[Create New Product](#)

Product Name	Description	Image	Price (INR)	Discount Rate	Discount (INR)	Net Price (INR)	Category Name	Action
Mango	A juicy mango		40.00	5.00	2.00	38.00	Fruit	Edit Details Delete
Apple	Red apple		100.00	10.00	10.00	90.00	Fruit	Edit Details Delete
Orange	Fruity oranges		35.00	3.00	1.05	33.95	Fruit	Edit Details Delete

© 2019 - FlixOne.Web - [Privacy](#)

FlixOne.Web Home Products About Contact Privacy

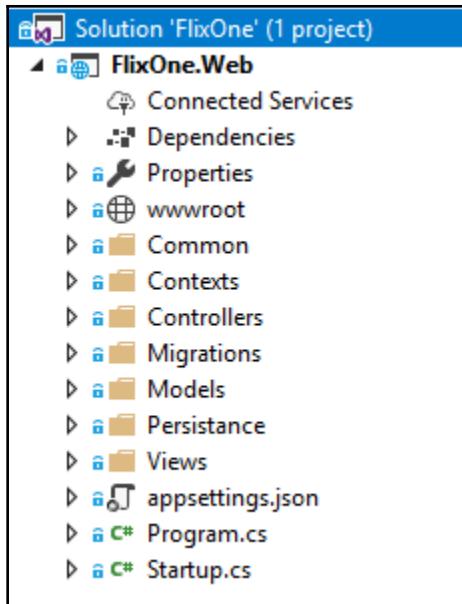
Product Discount Listing

[Back to Product](#)

Product Name	Price (INR)	Discount Rate	Description	Active?	Remarks, if any
Mango	40.00	5%	Seasonal Discount	Yes	-
Mango	40.00	105%	Wrong discount, should be ignored	No	Discount rate is invalid, hence will not consider in price calculations.

[Back to Product](#)

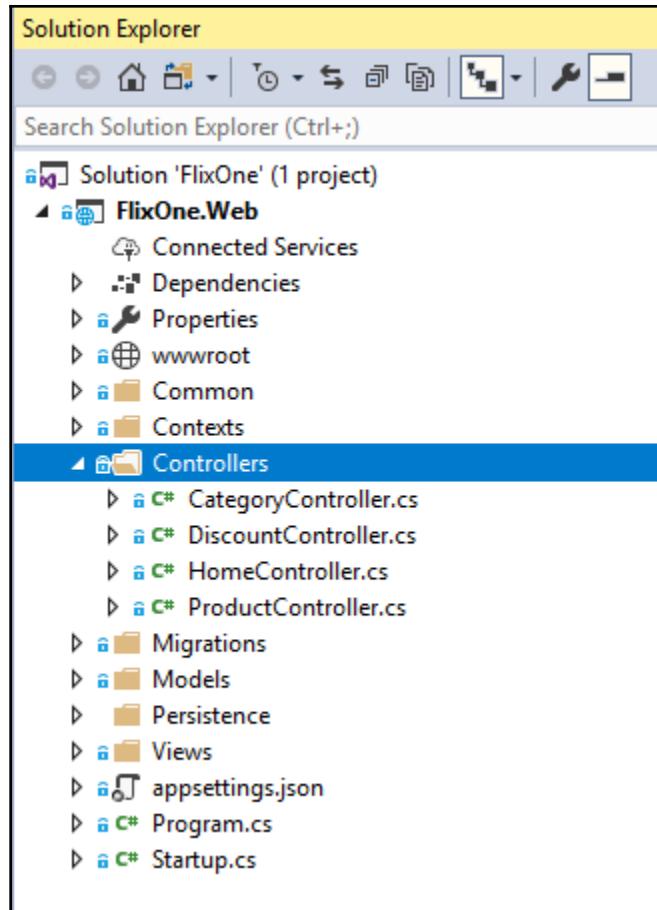
© 2019 - FlixOne.Web - [Privacy](#)



An unhandled exception occurred while processing the request.

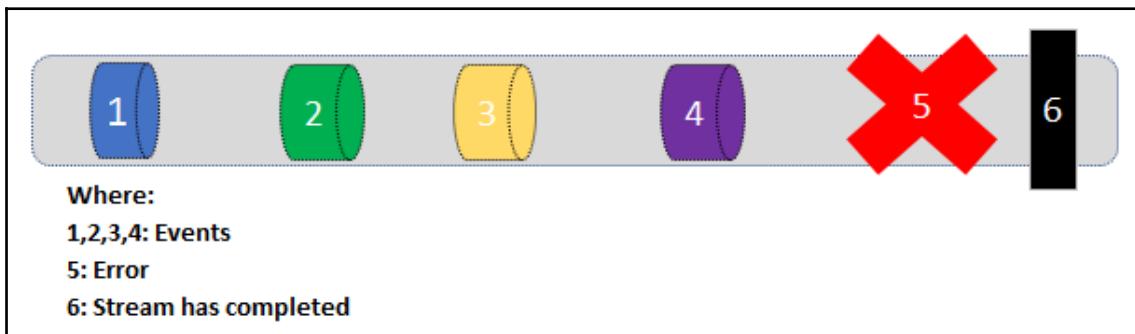
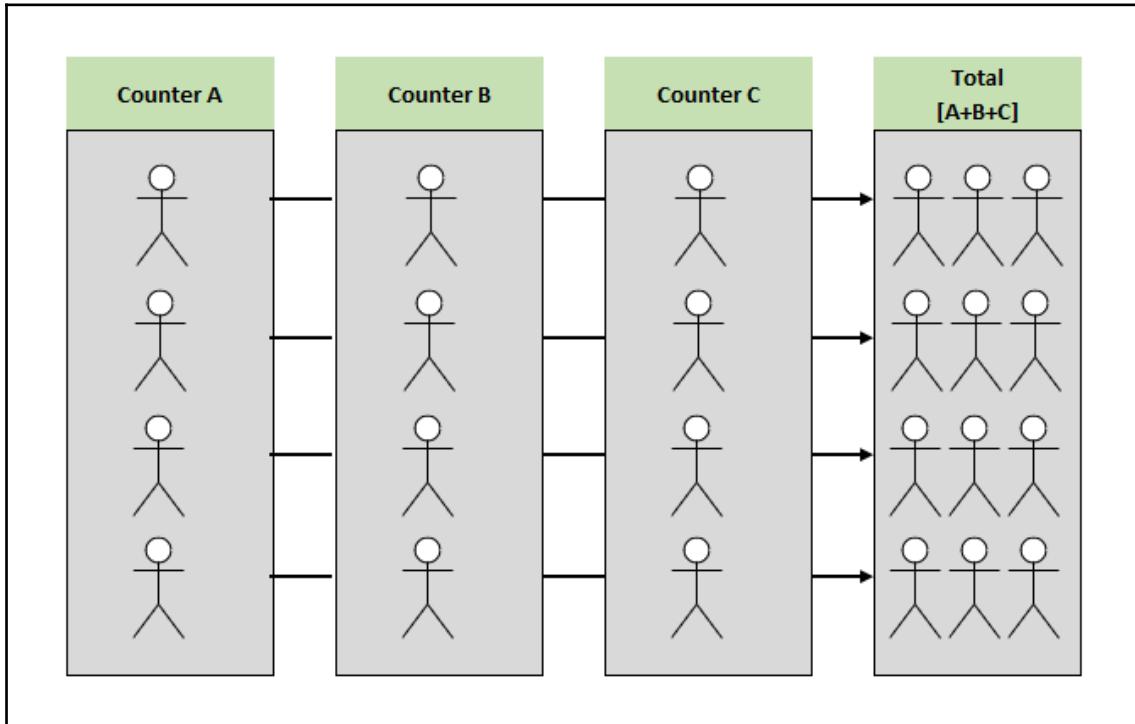
InvalidOperationException: Unable to resolve service for type 'FlixOne.Web.Common.IHelper' while attempting to activate 'FlixOne.Web.Persistence.InventoryRepository'.

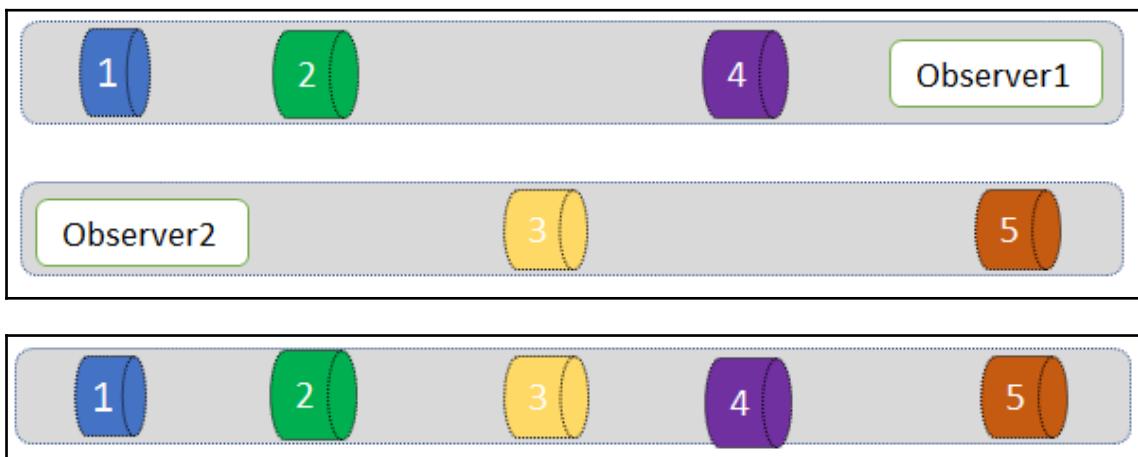
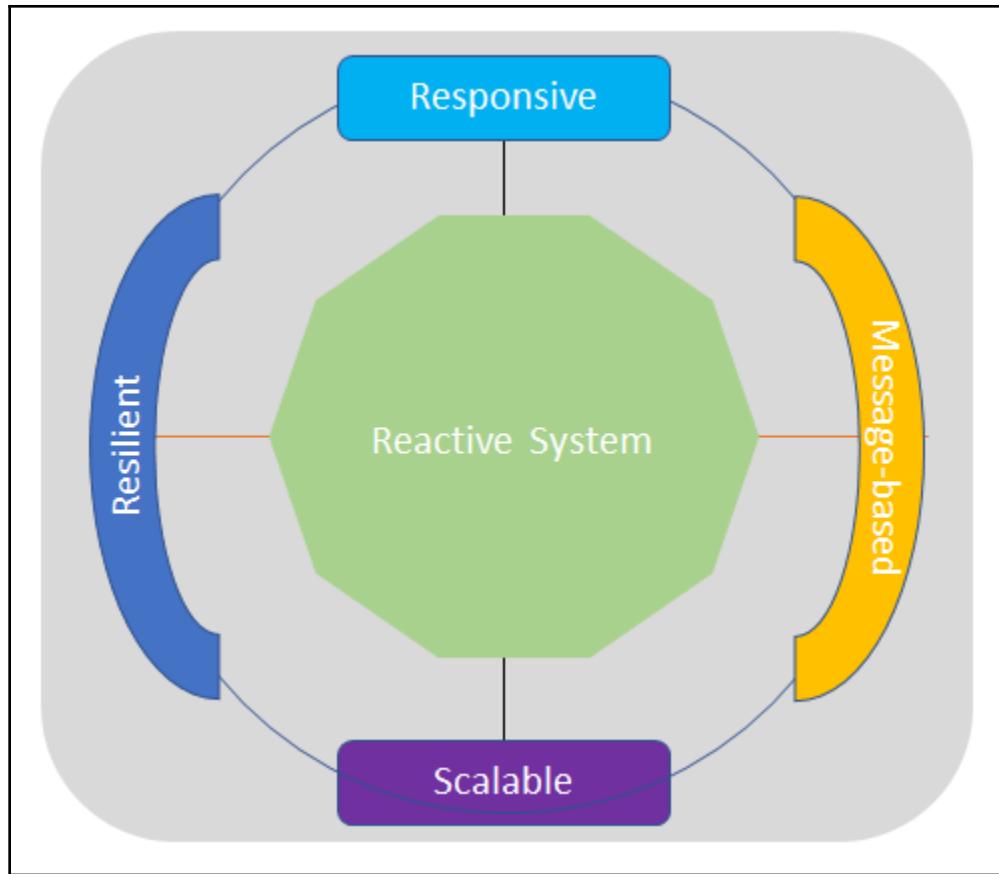
```
Microsoft.Extensions.DependencyInjection.ServiceLookup.CallSiteFactory.CreateArgumentCallSites(Type serviceType, Type implementationType, CallSiteChain callSiteChain, ParameterInfo[] parameters, bool throwIfCallSiteNotFound)
```

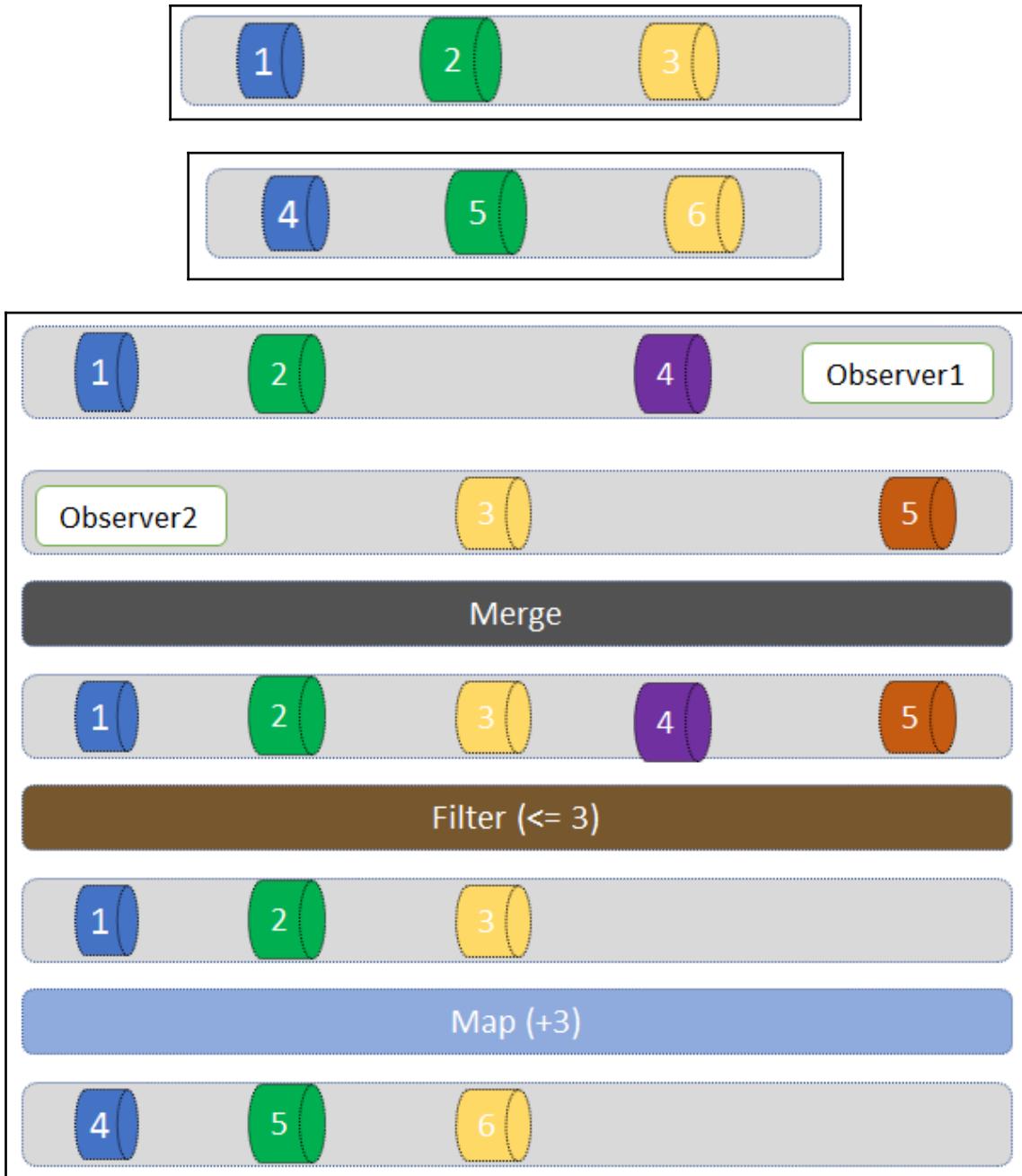


Product Discount Listing					
Back to Product					
Product Name	Price (INR)	Discount Rate	Description	Active?	Remarks, if any
Mango	40.00	5%	Seasonal Discount	Yes	-
Mango	40.00	105%	Wrong discount, should be ignored	No	Discount rate is invalid, hence will not consider in price calculations.
Orange	35.00	3%	Fruit dhamaka	Yes	-
Orange	35.00	32%	Pitch Discount	No	-
Apple	100.00	10%	Special discount	Yes	-
Apple	100.00	2%	Seasonal Discount	No	-
Apple	100.00	125%	Discarded	No	Discount rate is invalid, hence will not consider in price calculations.

Chapter 10: Reactive Programming Patterns and Techniques







```
Enter comma separated number (0-9):
```

```
Enter comma separated number (0-9): 1,2,4  
Enter comma separated number (0-9): 3,5
```

```
Counter1:      1      2      4  
Counter2:      3      5
```

```
Merge:      1      3      2      5      4
```

```
Filter (<= 3):  1      3      2
```

```
Map (+ 3):    4      6      5
```

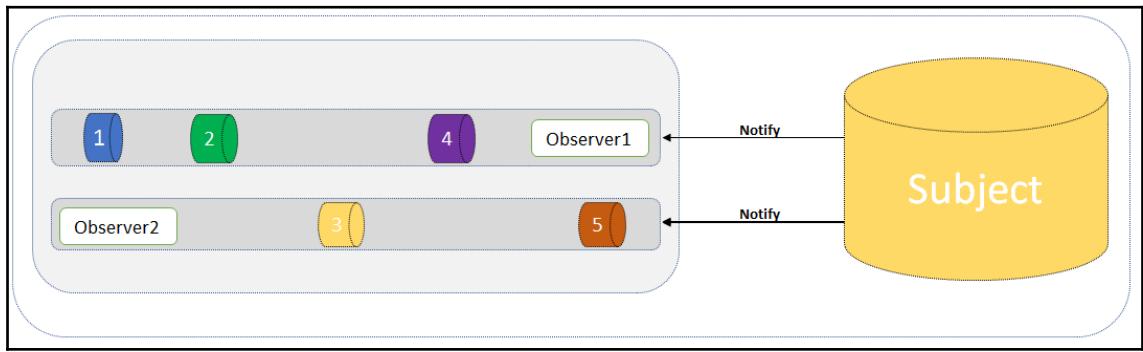
```
Counter1:      1      2      4
Counter2:      3      5

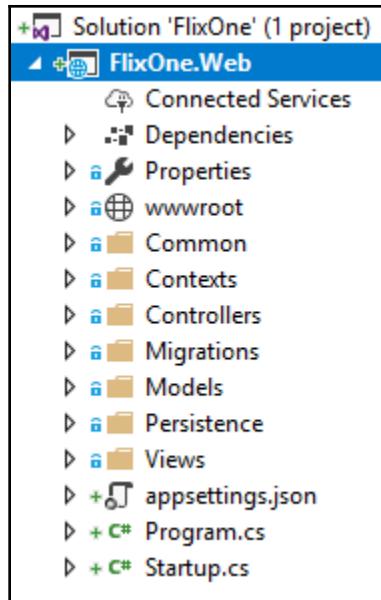
Merge:         1      3      2      5      4

Filter (<= 3): 1      3      2

Map (+ 3):    4      6      5

Press any key...
```





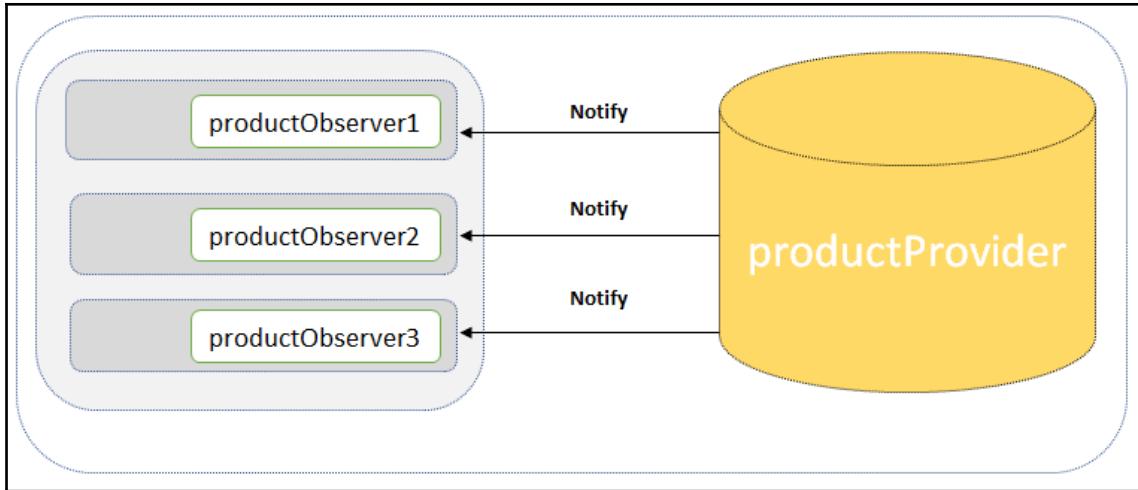
FlixOne.Web Home Products About Contact Privacy

Product Listing

[Create New Product](#) | [Products Discount Listing](#) | [Audit Report](#)

Product Name	Description	Image	Price (INR)	Discount Rate	Discount (INR)	Net Price (INR)	Category Name	Action
Mango	A juicy mango		40.00	5.00	2.00	38.00	Fruit	Edit Details Delete
Apple	Red apple		100.00	10.00	10.00	90.00	Fruit	Edit Details Delete
Orange	Fruity oranges		35.00	3.00	1.05	33.95	Fruit	Edit Details Delete

© 2019 - FlixOne.Web - [Privacy](#)



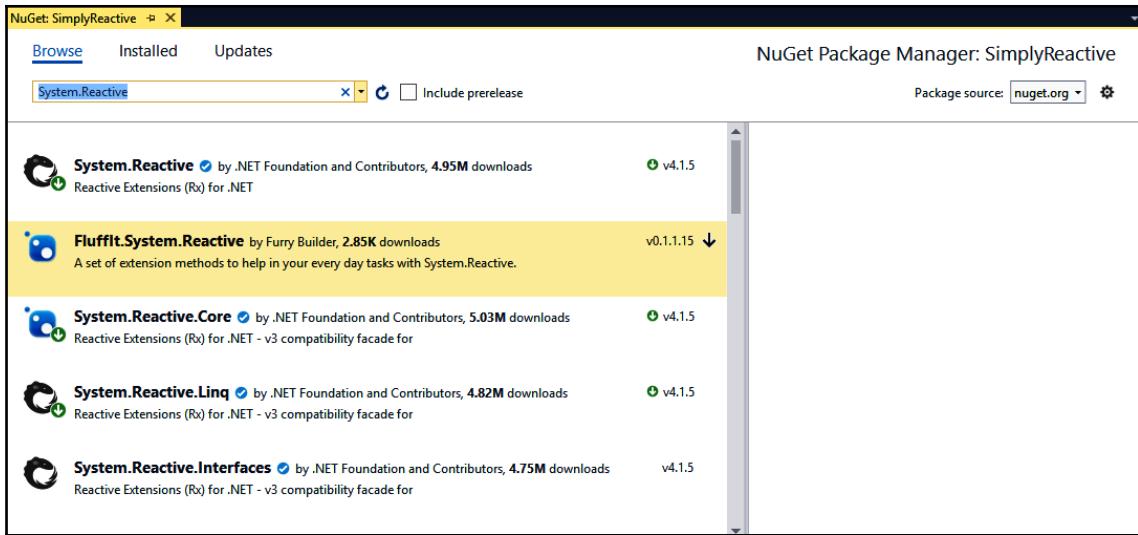
Product Audit Report

Product Listing

IsSuccess	Message
<input checked="" type="checkbox"/>	Reporter:mango. Product - Name: Mango, Price:40.00,Desc: A juicy mango
<input checked="" type="checkbox"/>	Reporter:apple. Product - Name: Mango, Price:40.00,Desc: A juicy mango
<input checked="" type="checkbox"/>	Reporter:apple. Product - Name: Apple, Price:100.00,Desc: Red apple
<input checked="" type="checkbox"/>	Reporter:orange. Product - Name: Mango, Price:40.00,Desc: A juicy mango
<input checked="" type="checkbox"/>	Reporter:orange. Product - Name: Apple, Price:100.00,Desc: Red apple
<input checked="" type="checkbox"/>	Reporter:orange. Product - Name: Orange, Price:35.00,Desc: Fruity oranges

© 2019 - FlixOne.Web - [Privacy](#)

Graphic Bundle



The screenshot shows a web application titled 'FlixOne.Web' with a navigation bar including Home, Products, About, Contact, and Privacy. The main content area is titled 'Product Listing' and contains a table of products:

Cat Name	Name	Description	Image	Price	Discount Rate	Discount	Net Price	
Fruit	Orange	Fruity oranges		35.00	3.00	1.05	33.95	Edit Details Delete
Fruit	Mango	A juicy mango		40.00	5.00	2.00	38.00	Edit Details Delete
Fruit	Apple	Red apple		100.00	10.00	10.00	90.00	Edit Details Delete

Product Listing

[Create New Product](#) | [Products Discount Listing](#) | [Audit Report](#)

Search by: Search

Cat Name	Name	Description	Image	Price	Discount Rate	Discount	Net Price	
Fruit	Mango	A juicy mango		40.00	5.00	2.00	38.00	Edit Details Delete
Fruit	Apple	Red apple		100.00	10.00	10.00	90.00	Edit Details Delete
Fruit	Orange	Fruity oranges		35.00	3.00	1.05	33.95	Edit Details Delete

FlixOne.Web [Home](#) [Products](#) [About](#) [Contact](#) [Privacy](#)

Product Listing

[Create New Product](#) | [Products Discount Listing](#) | [Audit Report](#)

Search by: Search

Cat Name	Name	Description	Image	Price	Discount Rate	Discount	Net Price	
Fruit	Apple	Red apple		100.00	10.00	10.00	90.00	Edit Details Delete
Fruit	Mango	A juicy mango		40.00	5.00	2.00	38.00	Edit Details Delete
Fruit	Orange	Fruity oranges		35.00	3.00	1.05	33.95	Edit Details Delete

FlixOne.Web [Home](#) [Products](#) [About](#) [Contact](#) [Privacy](#)

Product Listing

[Create New Product](#) | [Products Discount Listing](#) | [Audit Report](#)

Search by: Search

Cat Name	Name	Description	Image	Price	Discount Rate	Discount	Net Price	
Fruit	Mango	A juicy mango		40.00	5.00	2.00	38.00	Edit Details Delete

Graphic Bundle

FlixOne.Web Home Products About Contact Privacy

Product Listing

[Create New Product](#) | [Products Discount Listing](#) | [Audit Report](#)

Search by: Search

Cat Name	Name	Description	Image	Price	Discount Rate	Discount	Net Price

Product Listing

[Create New Product](#) | [Products Discount Listing](#) | [Audit Report](#)

Search by: Search

Cat Name	Name	Description	Image	Price	Discount Rate	Discount	Net Price
Fruit	Apple	Red apple		100.00	10.00	10.00	90.00
Fruit	Mango	A juicy mango		40.00	5.00	2.00	38.00
Fruit	Orange	Fruity oranges		35.00	3.00	1.05	33.95

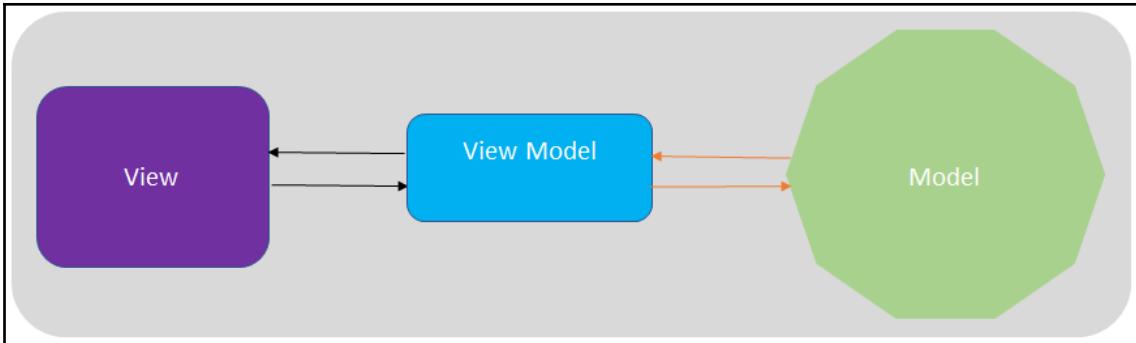
Product Listing

[Create New Product](#) | [Products Discount Listing](#) | [Audit Report](#)

Search by: Search Clear

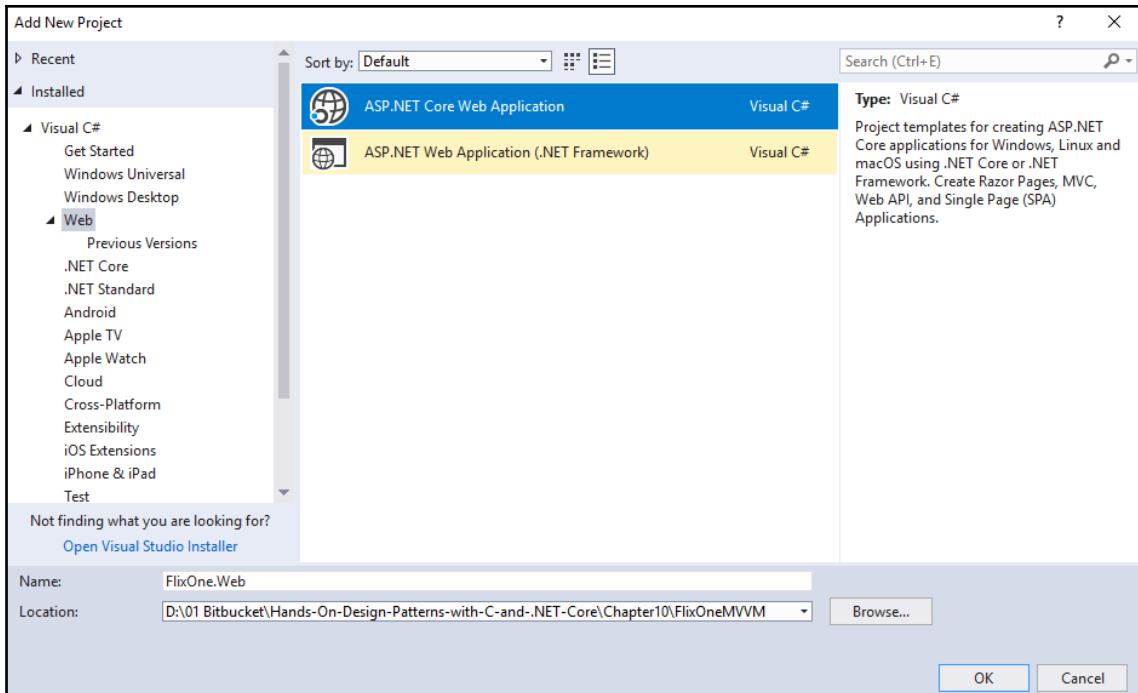
CategoryName	ProductName	ProductDescription	ProductImage	ProductPrice	ProductDiscountRate	ProductDiscount	ProductNetPrice
Fruit	Apple	Red apple		100.00	10.00	10.00	90.00
Fruit	Mango	A juicy mango		40.00	5.00	2.00	38.00
Fruit	Orange	Fruity oranges		35.00	3.00	1.05	33.95

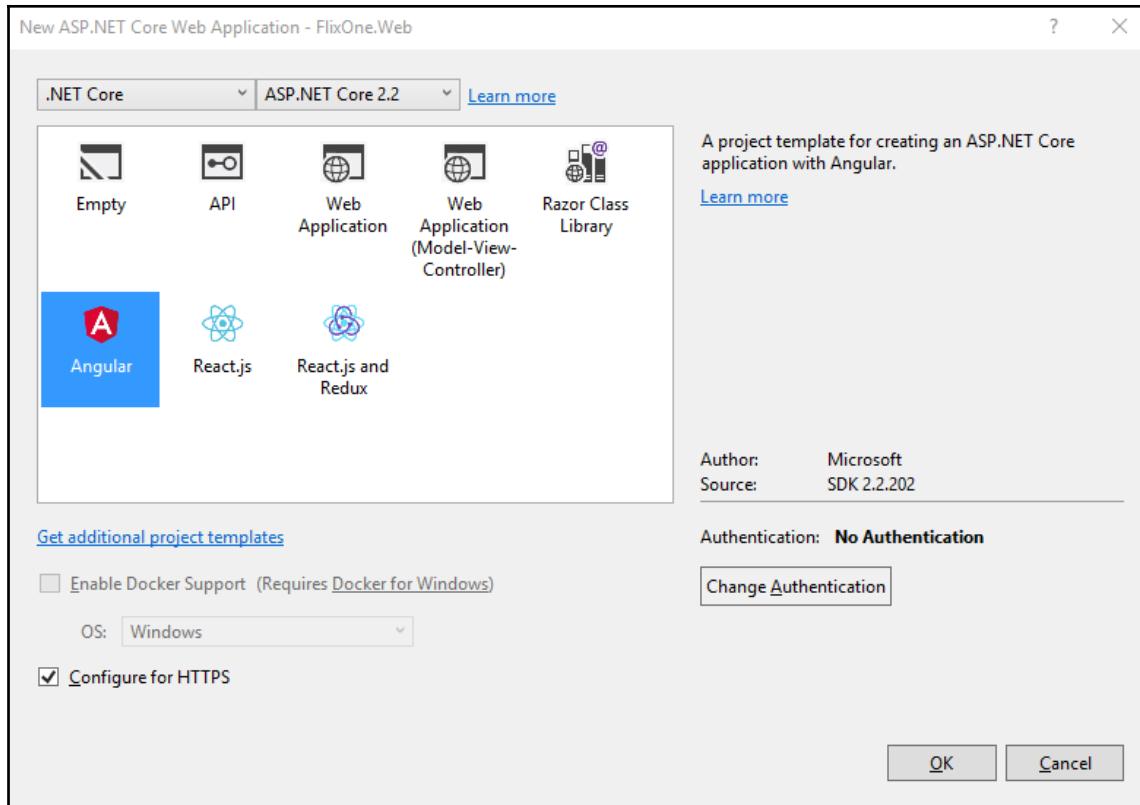
[Previous](#) [Next](#)

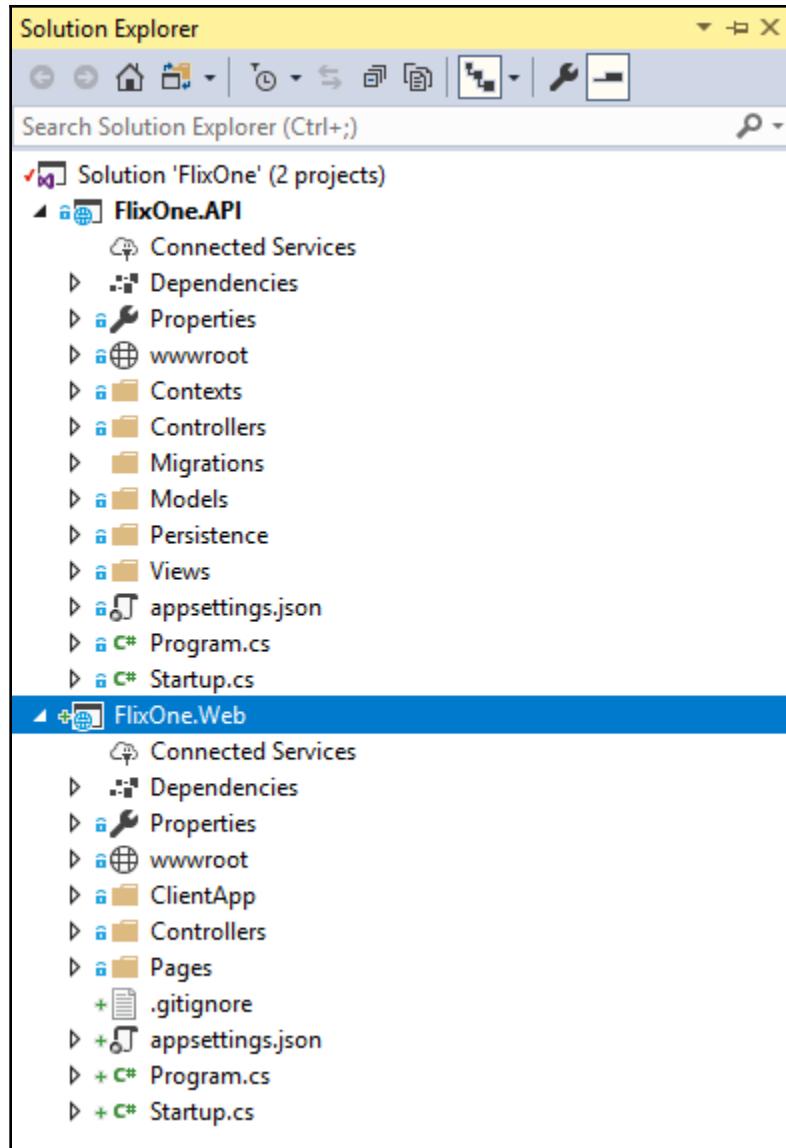


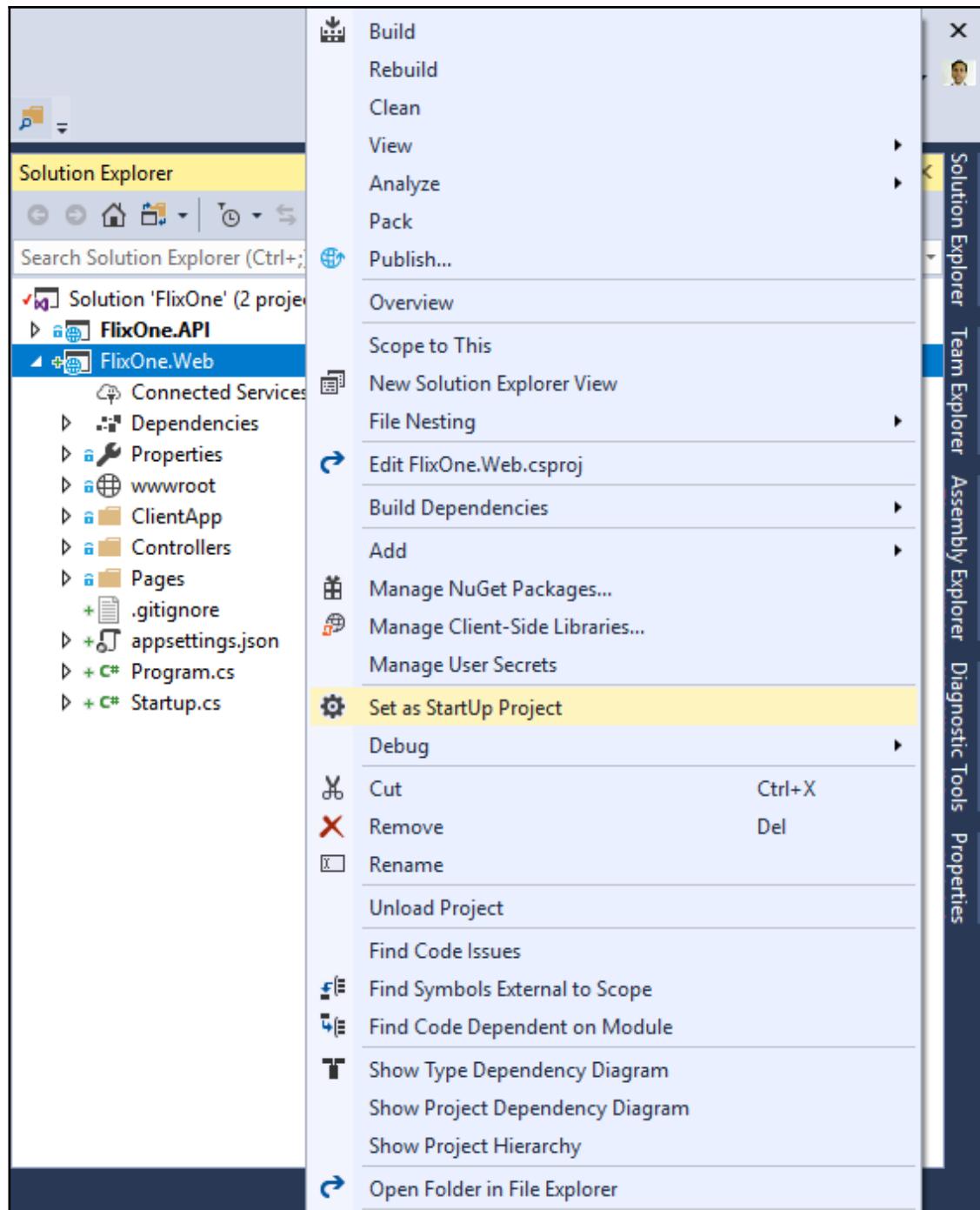
The screenshot shows the Swagger UI interface for "Product APIs". At the top, there is a navigation bar with "swagger" and a dropdown menu "Select a spec" set to "Product API V1". Below the header, the title "Product APIs" is displayed with a small "v1" badge. A link to "/swagger/v1/swagger.json" is shown. The main content area is titled "Product" and contains a list of API endpoints:

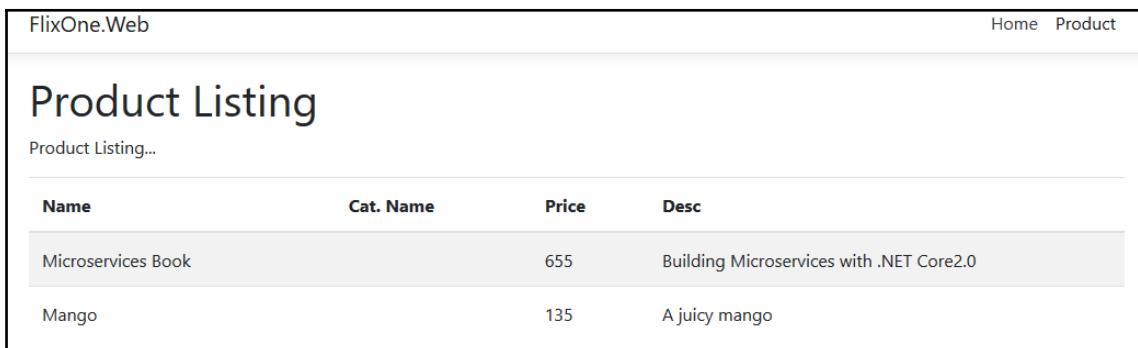
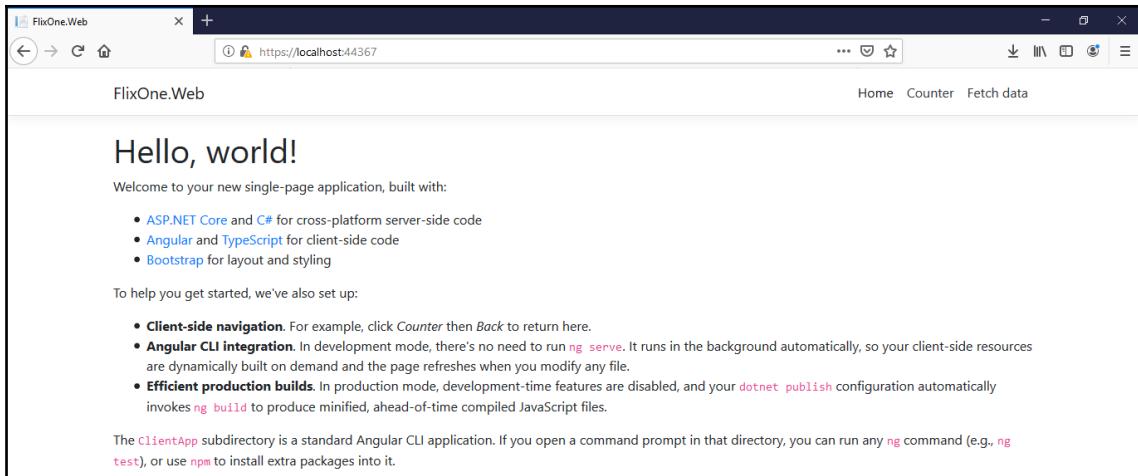
- GET /api/product/productlist** Product Listing
- GET /api/product/product/{productid}** Retrieve product for a specific product using productid
- POST /api/product/addproduct** Add new product
- PUT /api/product/updateproduct/{productid}** Update an existing product
- DELETE /api/product/deleteproduct/{productid}** Remove a specific product





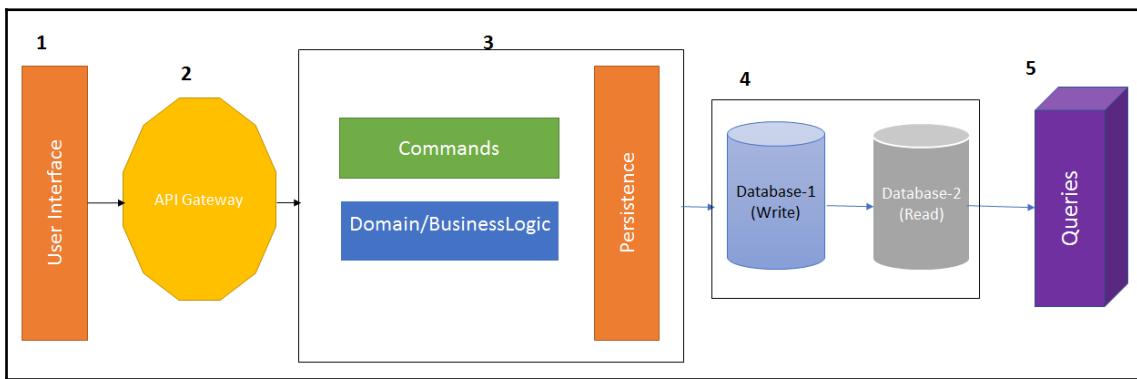


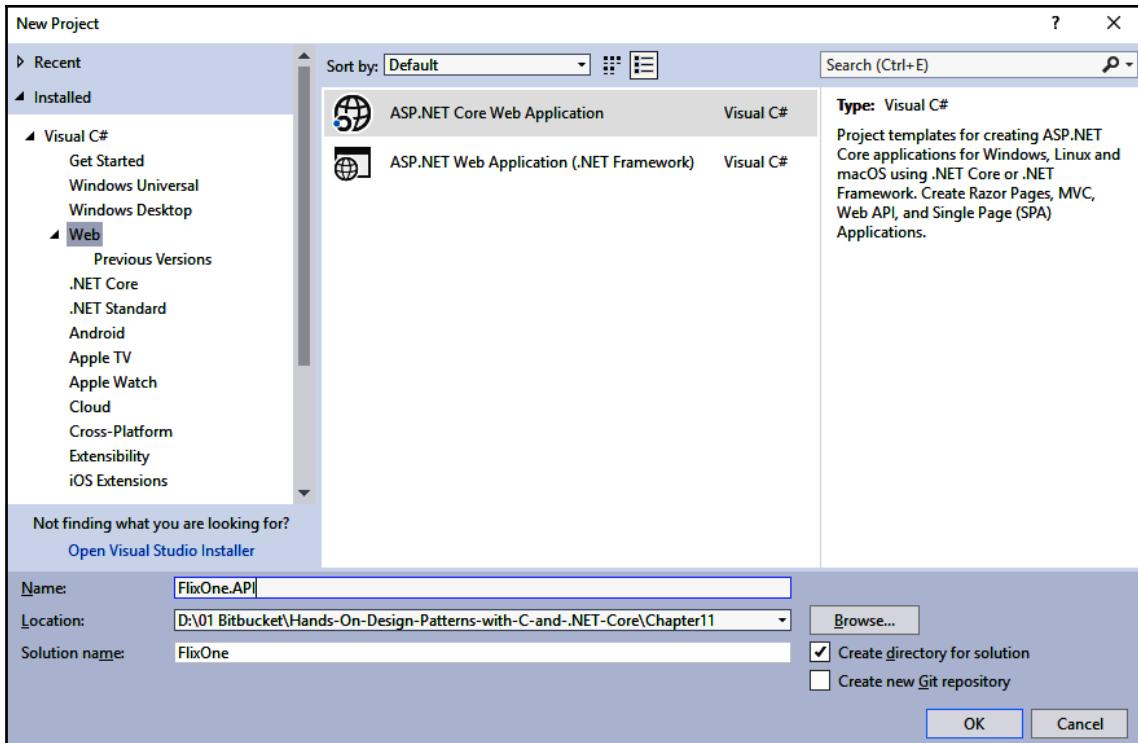




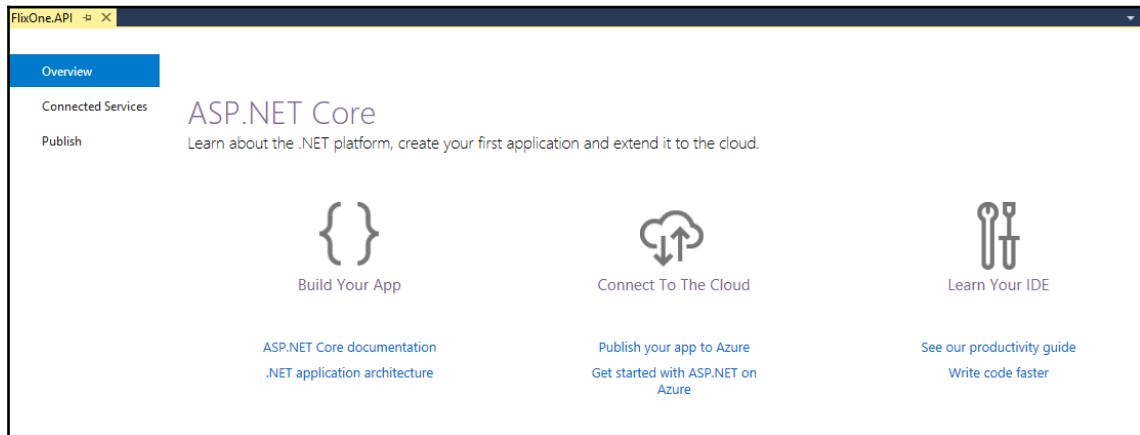
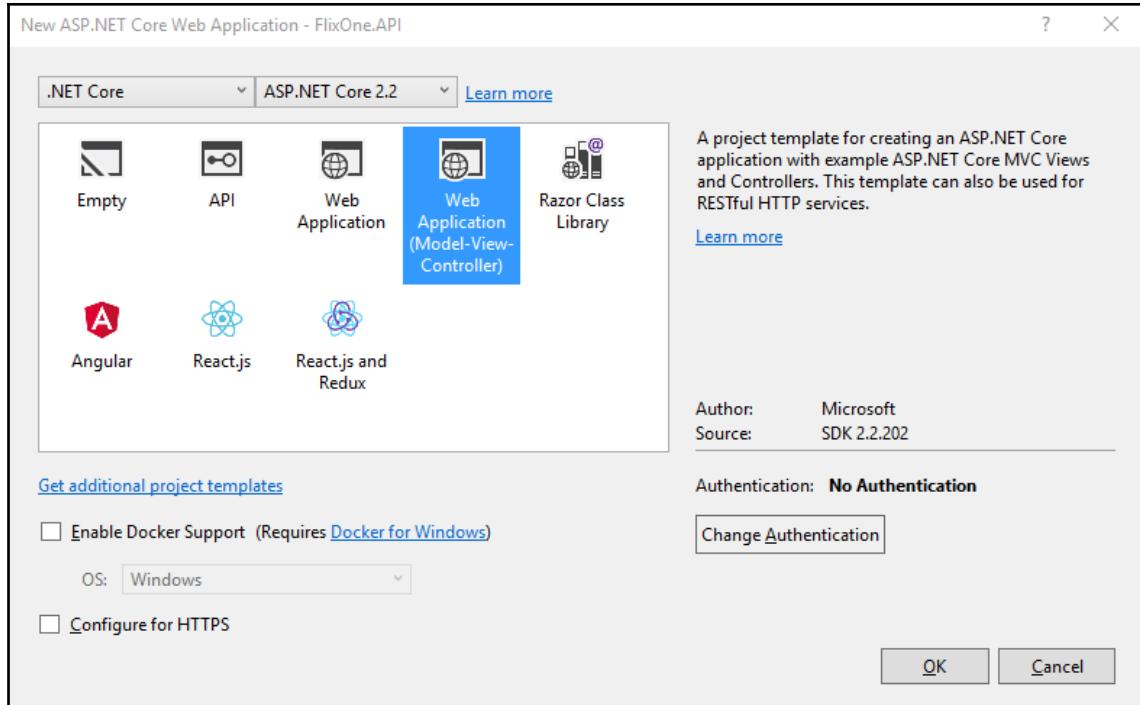
Chapter 11: Advanced Database Design and Application Techniques

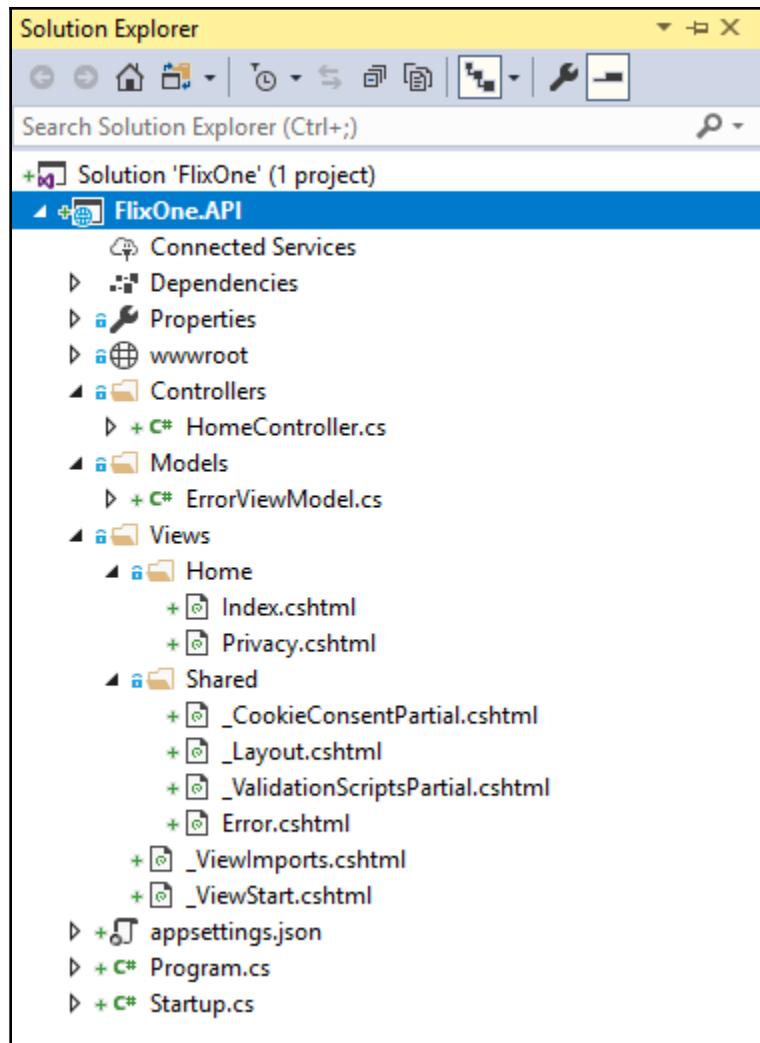
Date	Purchaser	Amount	Price
5/05/2019	West Country Produce	100	\$ 200.00
5/05/2019	Jim's Freshest	30	\$ 60.00
5/07/2019	West Country Produce	-25	-\$ 50.00
5/07/2019	Marlborough Fair	70	\$ 140.00
5/07/2019	West Country Produce	12	\$ 30.00
5/07/2019	Marlborough Fair	15	\$ 30.00
6/07/2019	Coopers Collective	80	\$ 160.00
		282	\$ 570.00

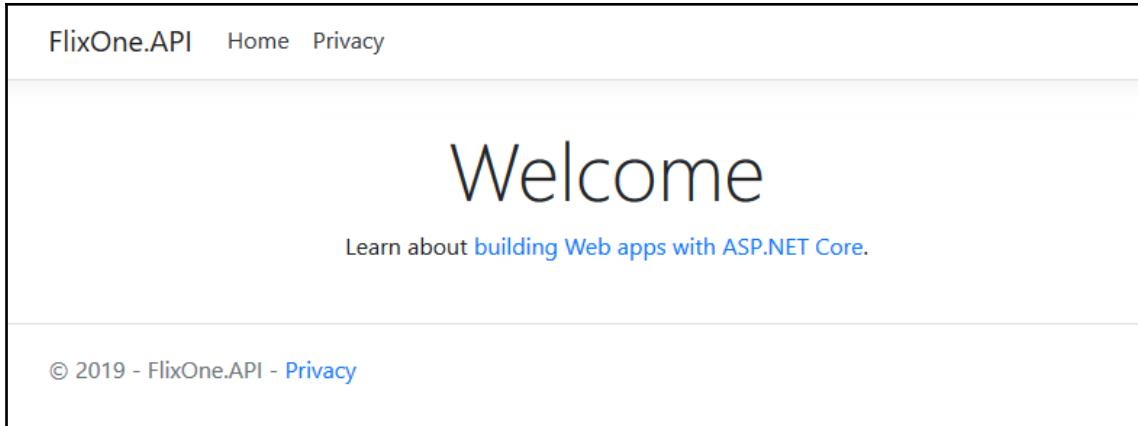


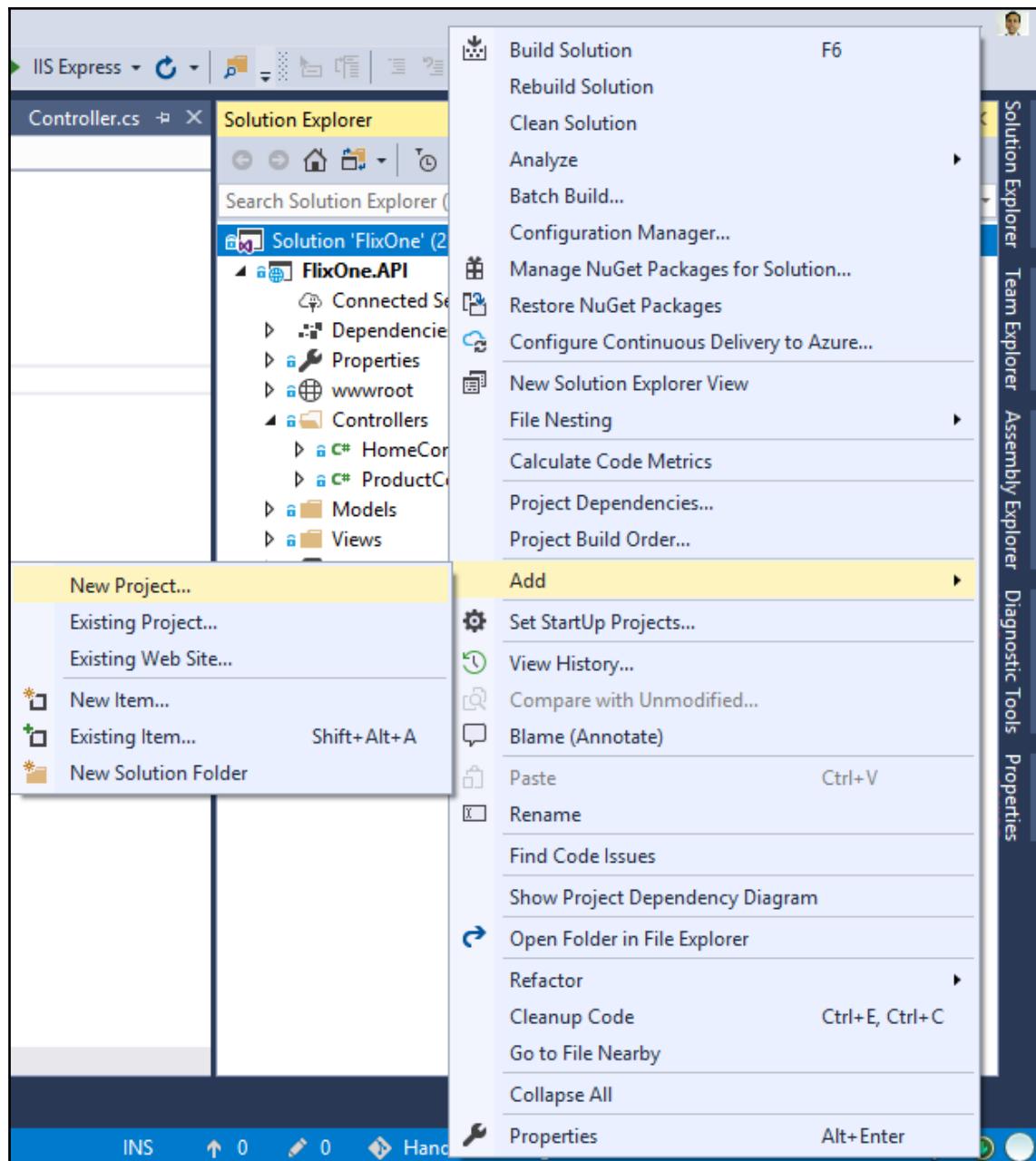


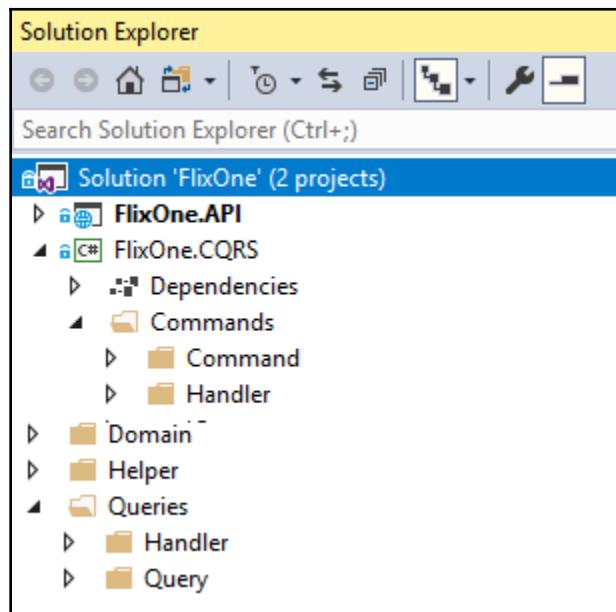
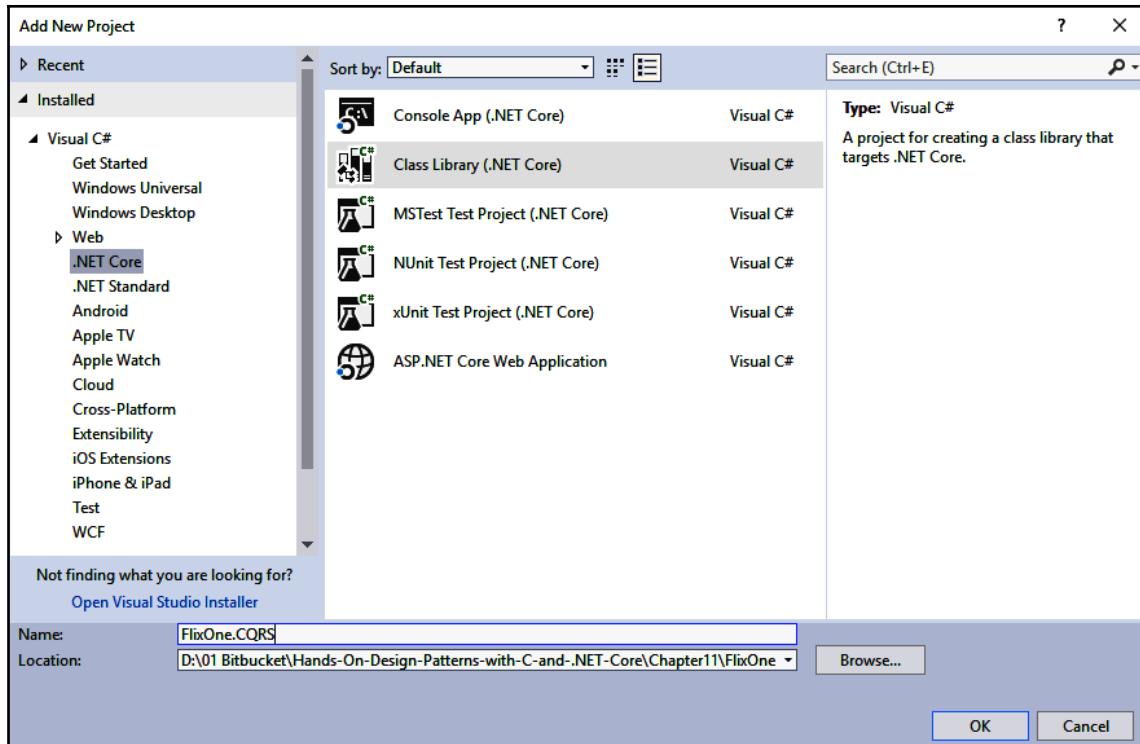
Graphic Bundle

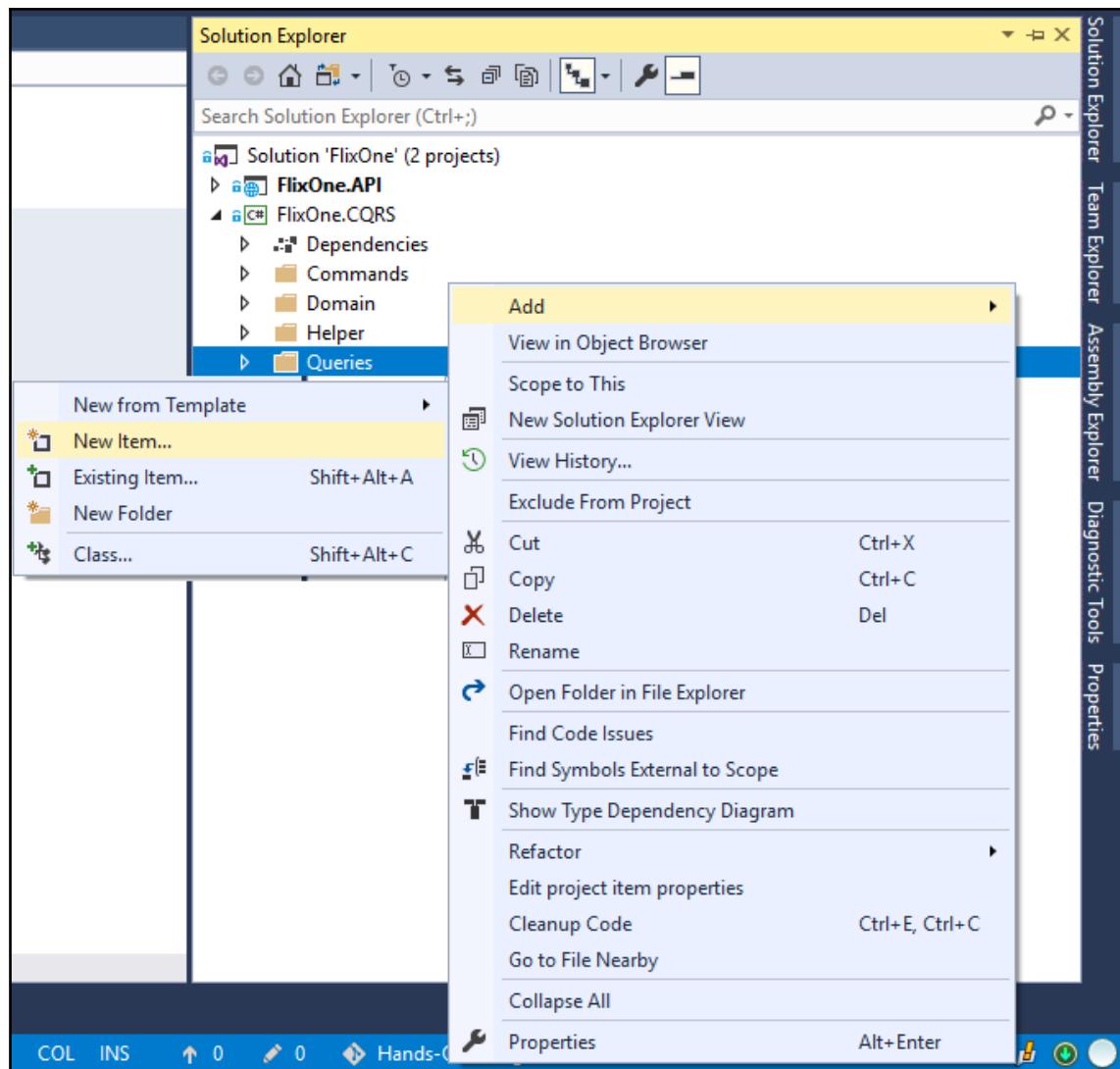


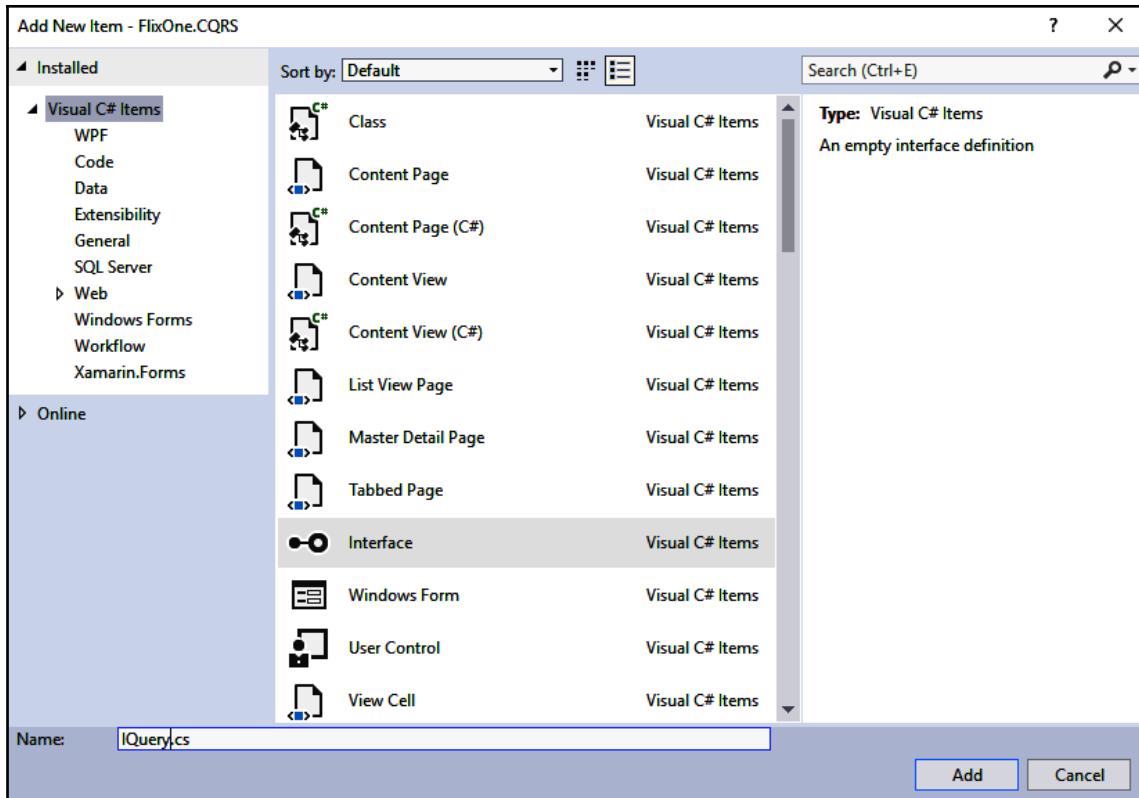


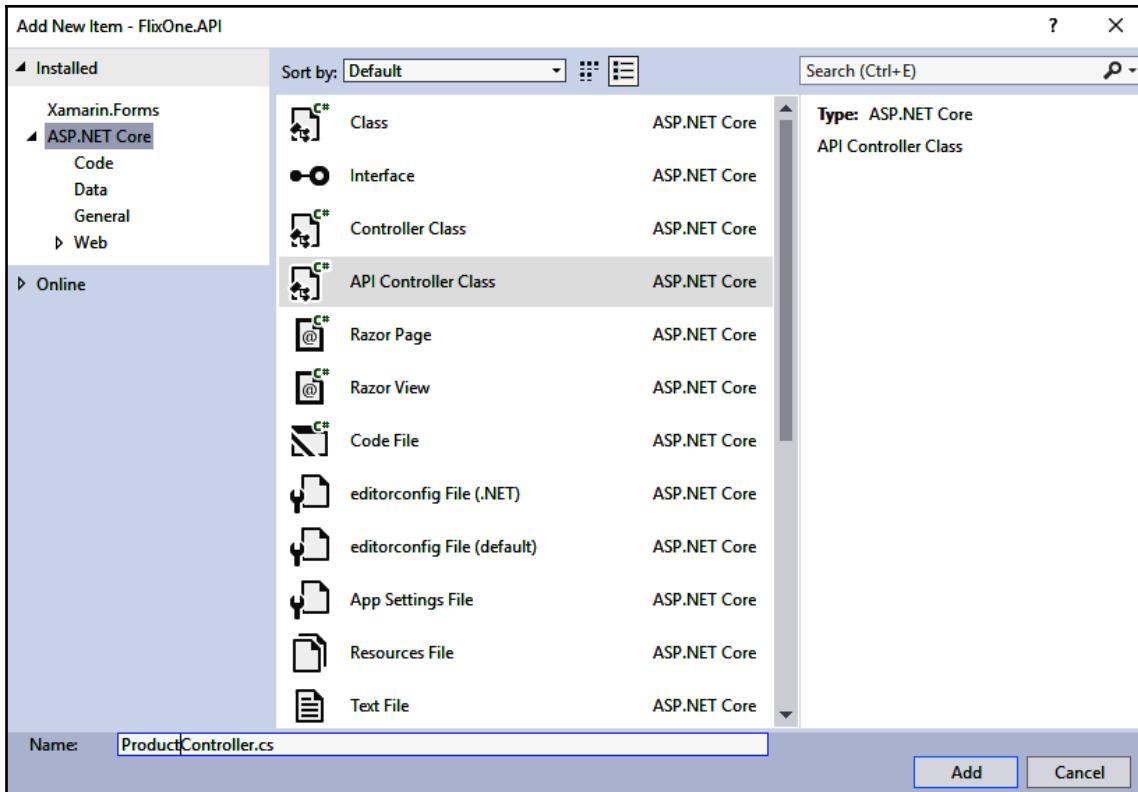












Browse Installed Updates

SwashBuckle.AspNetCore

NuGet Package Manager: FlixOne.API

Package source: nuget.org

SwashBuckle.AspNetCore

Version: Latest stable 4.0.1

Install

Options

Description

Swagger tools for documenting APIs built on ASP.NET Core

Version: 4.0.1

Author(s): Swashbuckle.AspNetCore

License: View License

Date published: Wednesday, November 7, 2018 (11/7/2018)

Project URL: <https://github.com/domaindrivendev/Swashbuckle.AspNetCore>

Report Abuse: <https://www.nuget.org/packages/Swashbuckle.AspNetCore/4.0.1/ReportAbuse>

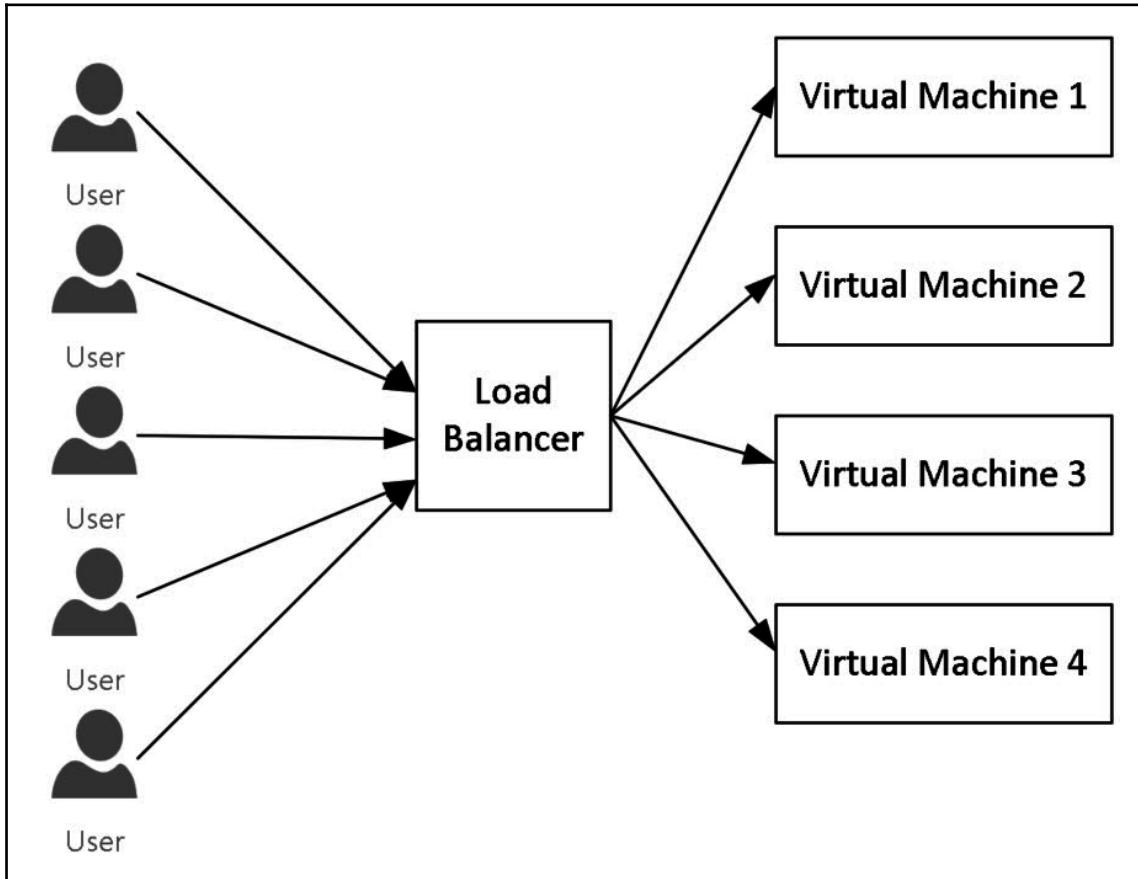
Tags: discovery, documentation, aspnet, help, swagger, webapi, aspnetcore

Dependencies

The screenshot shows the Swagger UI interface for a "Product APIs" section. At the top, there is a green header bar with the "swagger" logo and a dropdown menu labeled "Select a spec" set to "Product API V1". Below the header, the title "Product APIs" is displayed with a small "v1" badge. A link to "/swagger/v1/swagger.json" is shown below the title. The main content area is titled "Product" with a dropdown arrow icon. It lists four API endpoints:

- GET /api/Product** (blue button)
- POST /api/Product** (green button)
- GET /api/Product/{id}** (blue button)
- DELETE /api/Product/{id}** (red button)

Chapter 12: Coding for the Cloud



Weekday Autoscale  

Scale mode Scale based on a metric Scale to a specific instance count

Rules

Scale out

When	(Average) CpuPercentage > 60	Increase count by 2
------	------------------------------	---------------------

Scale in

When	(Average) CpuPercentage < 30	Decrease count by 1
------	------------------------------	---------------------

[+ Add a rule](#)

Instance limits Minimum Maximum Default

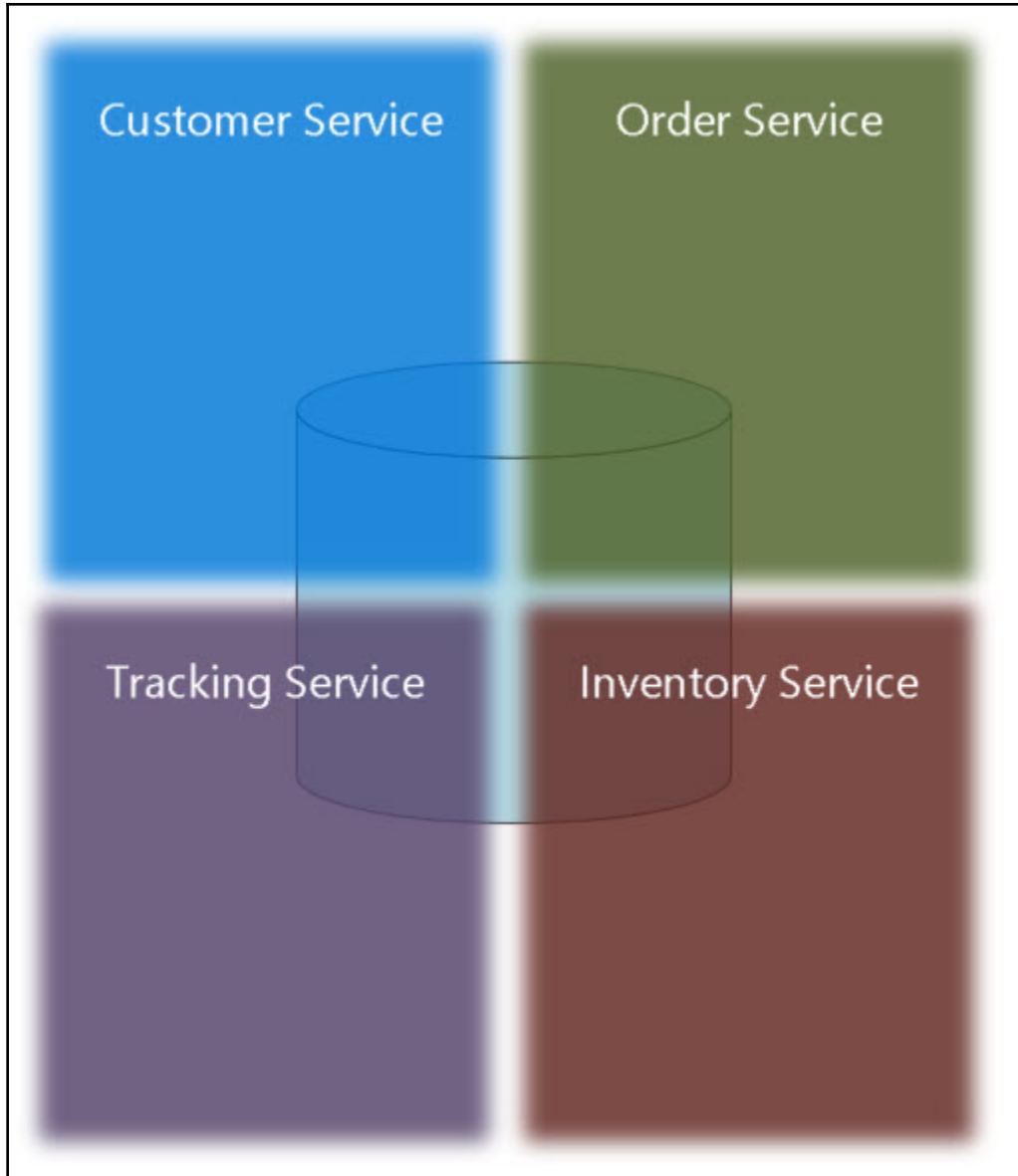
Schedule Specify start/end dates Repeat specific days

Repeat every Monday Tuesday Wednesday Thursday Friday
 Saturday Sunday

Timezone

Start time

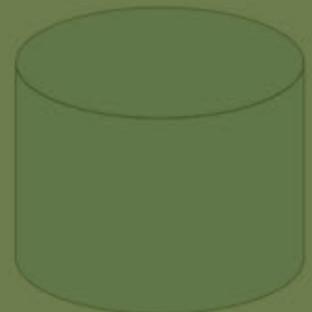
End time



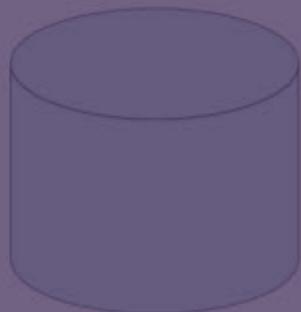
Customer Service



Order Service

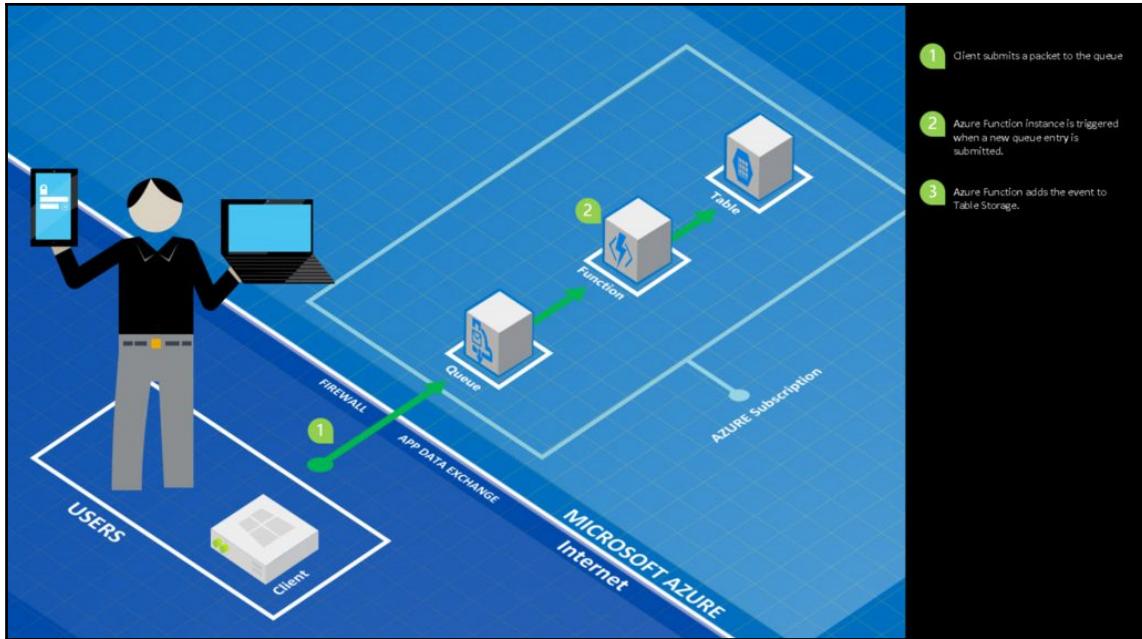


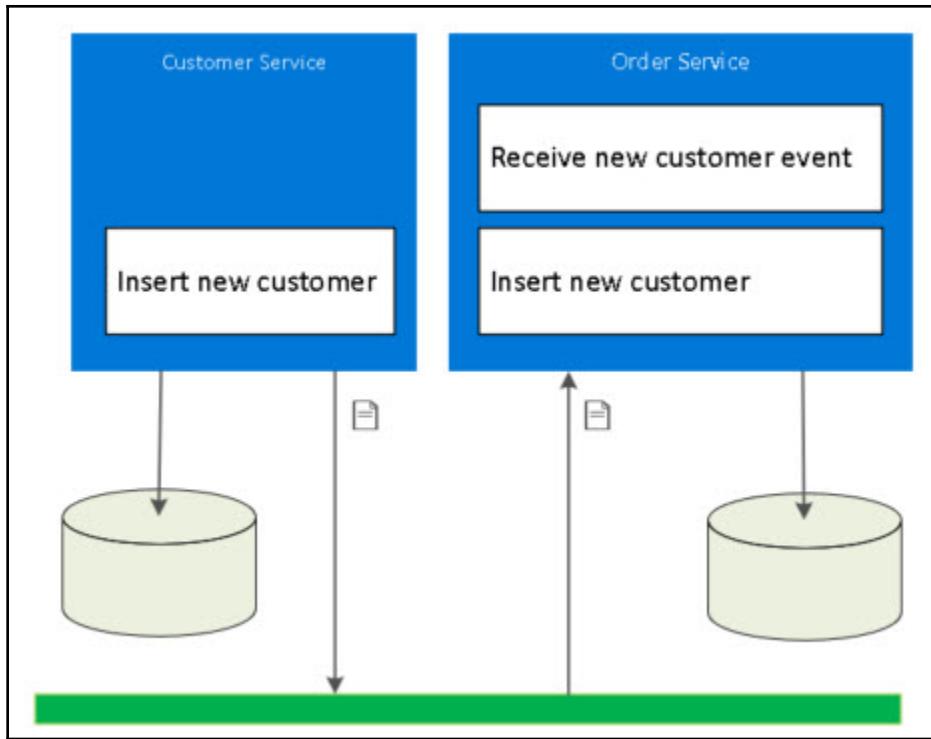
Tracking Service

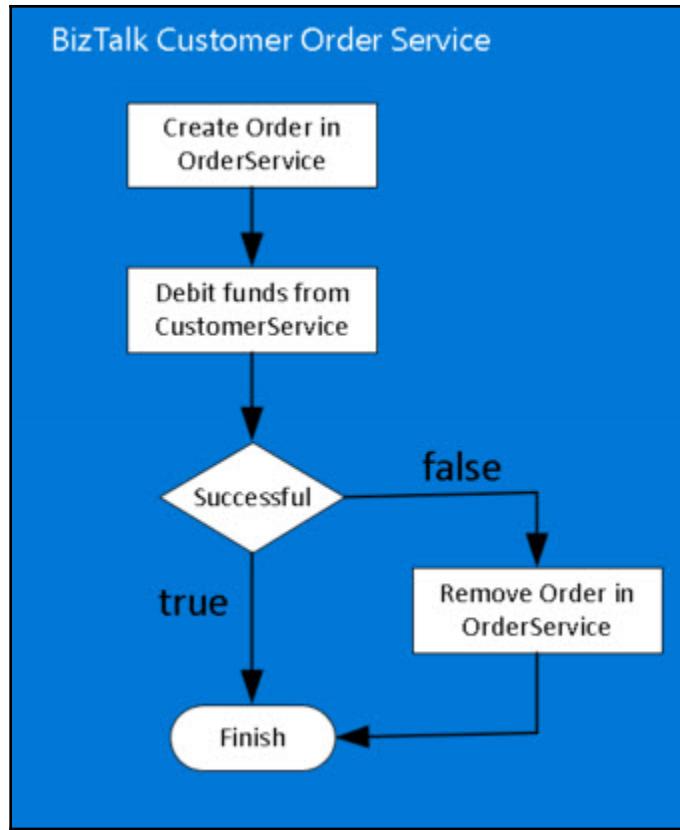


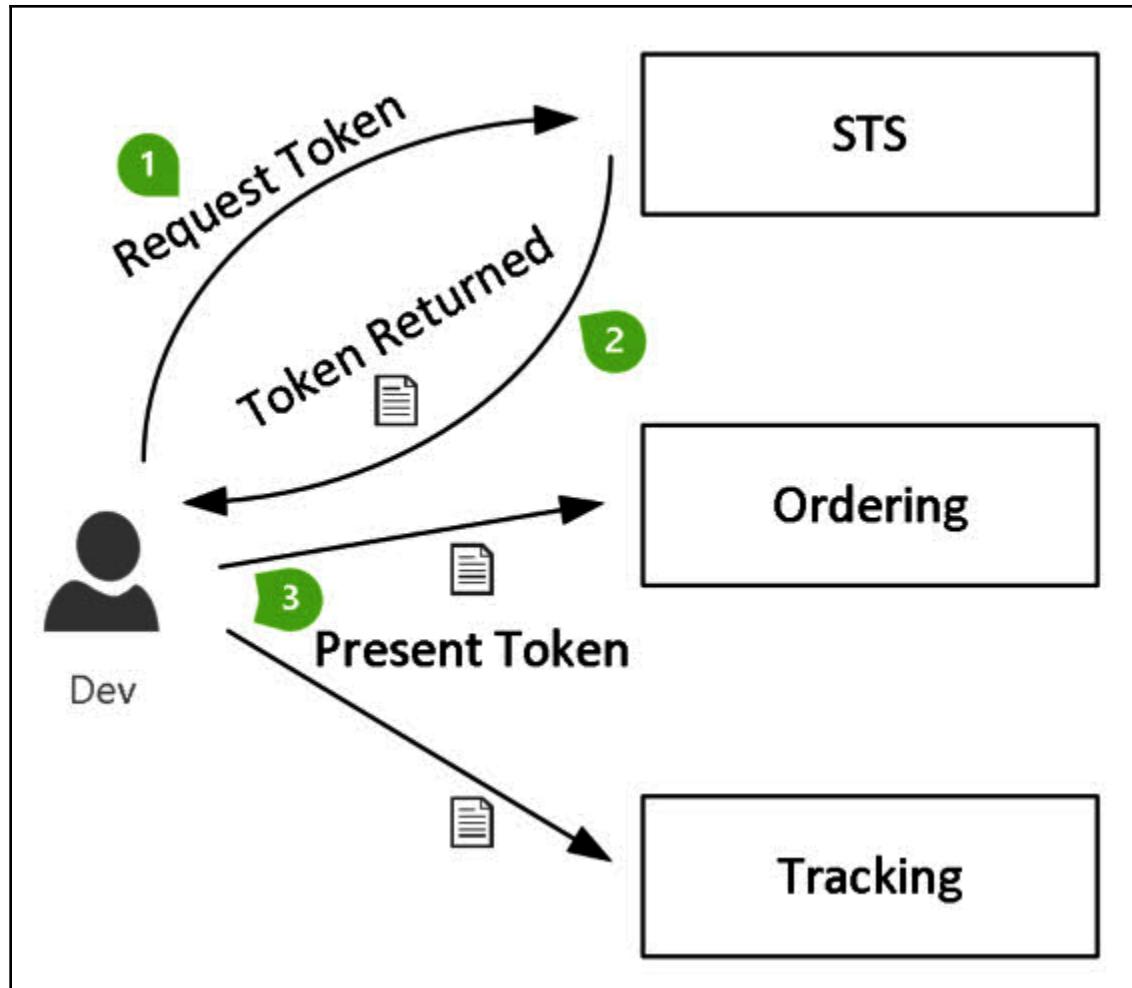
Inventory Service

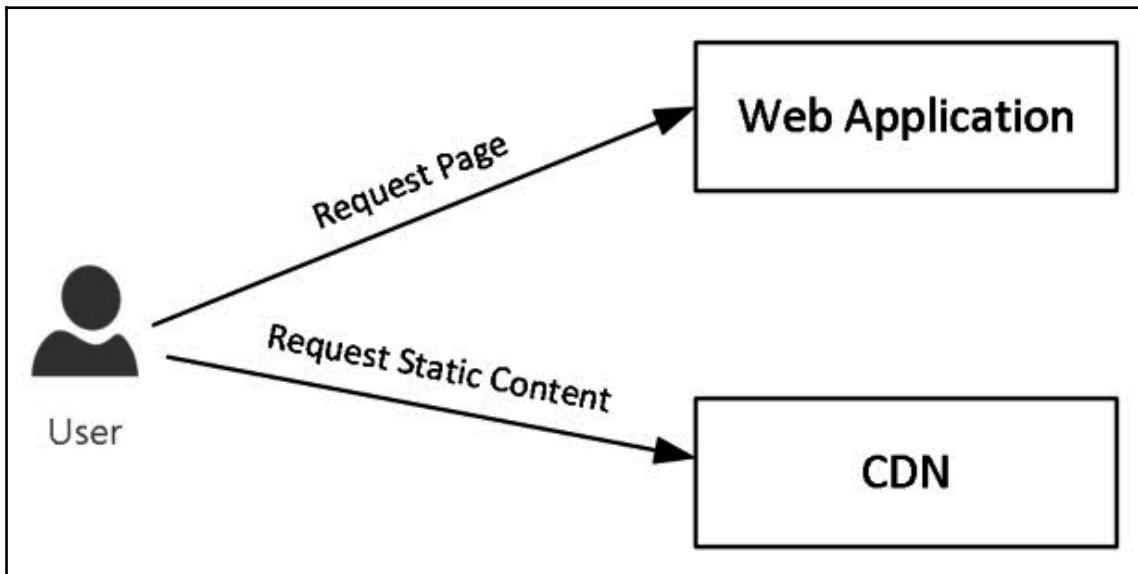
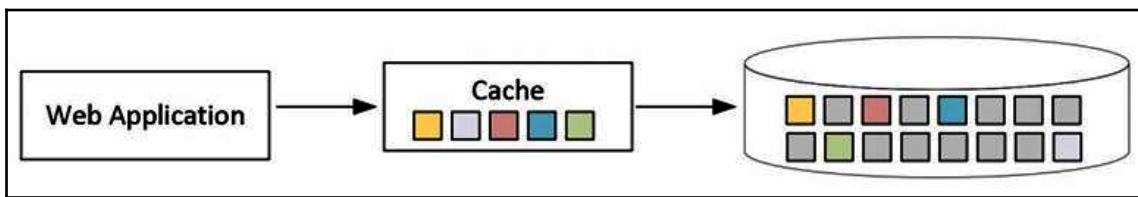
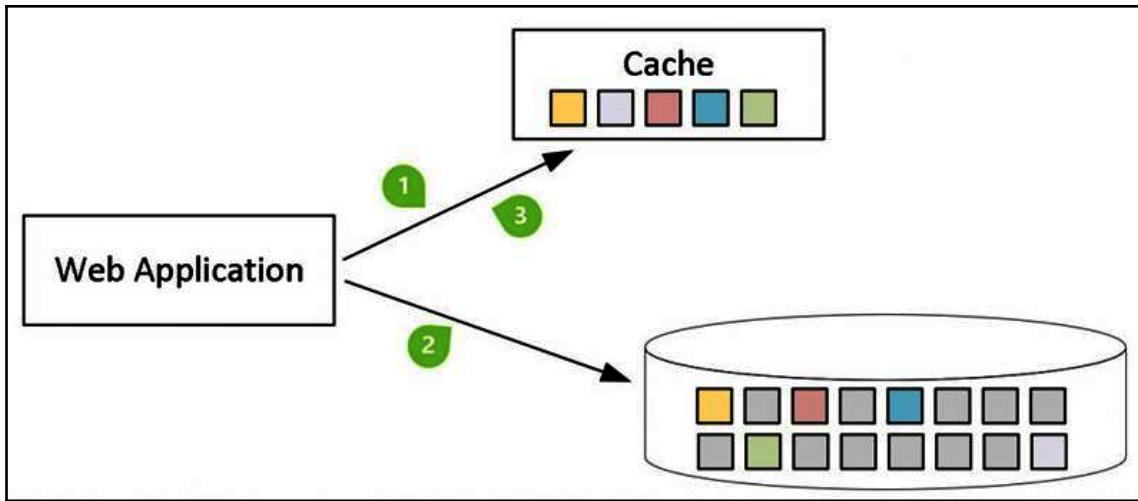


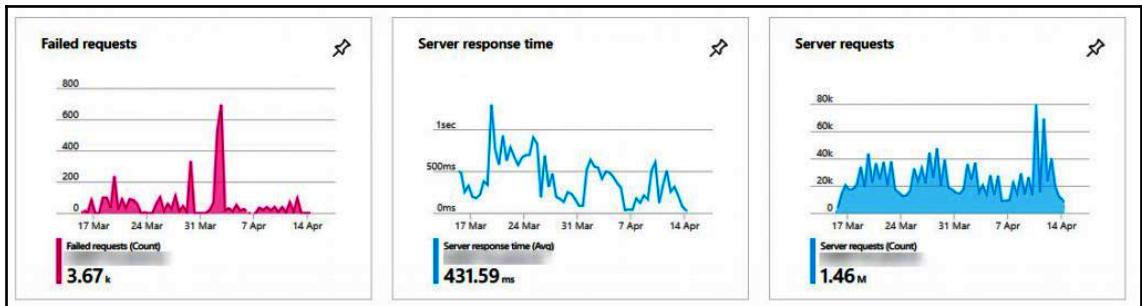
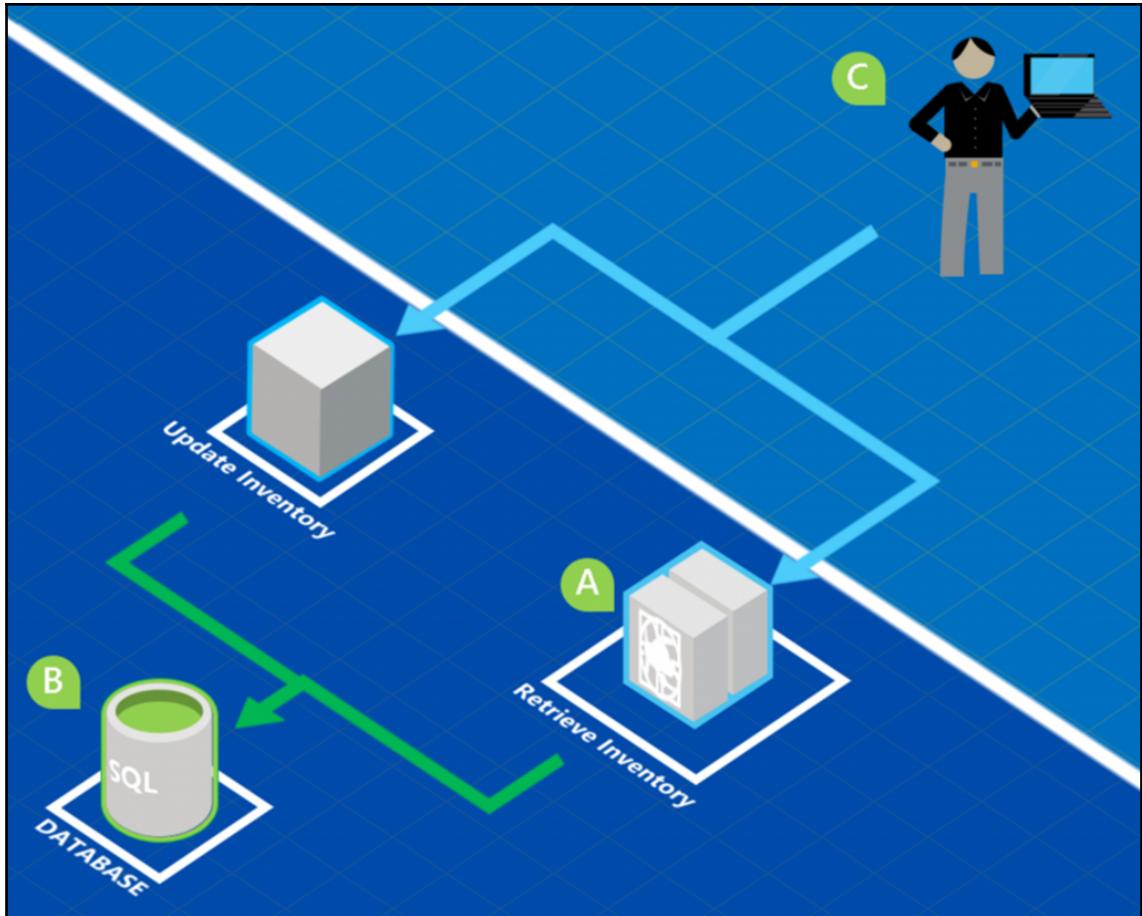


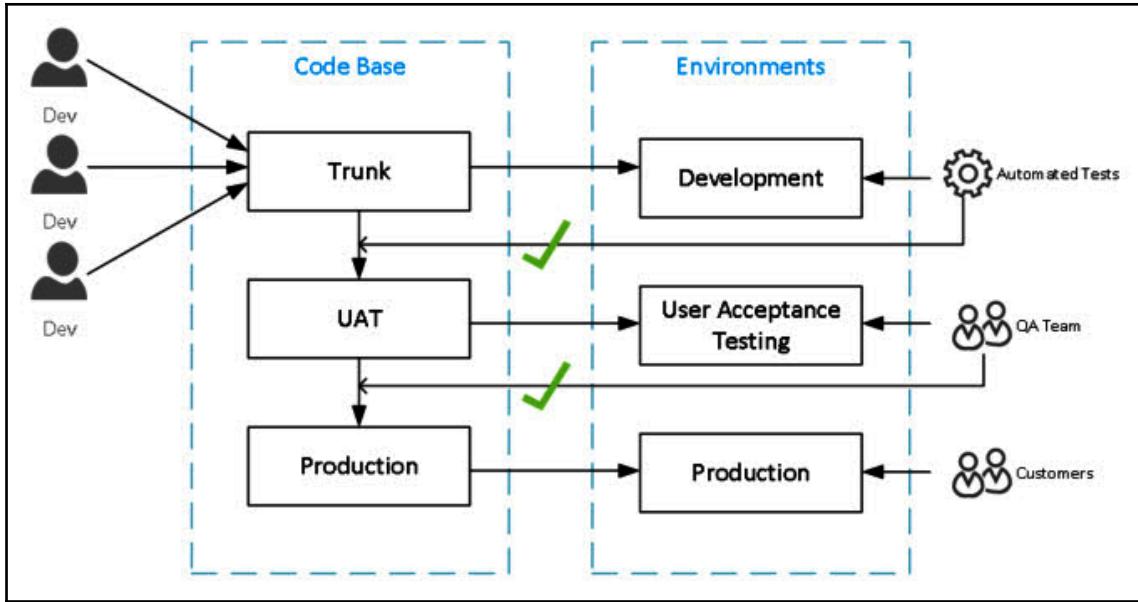




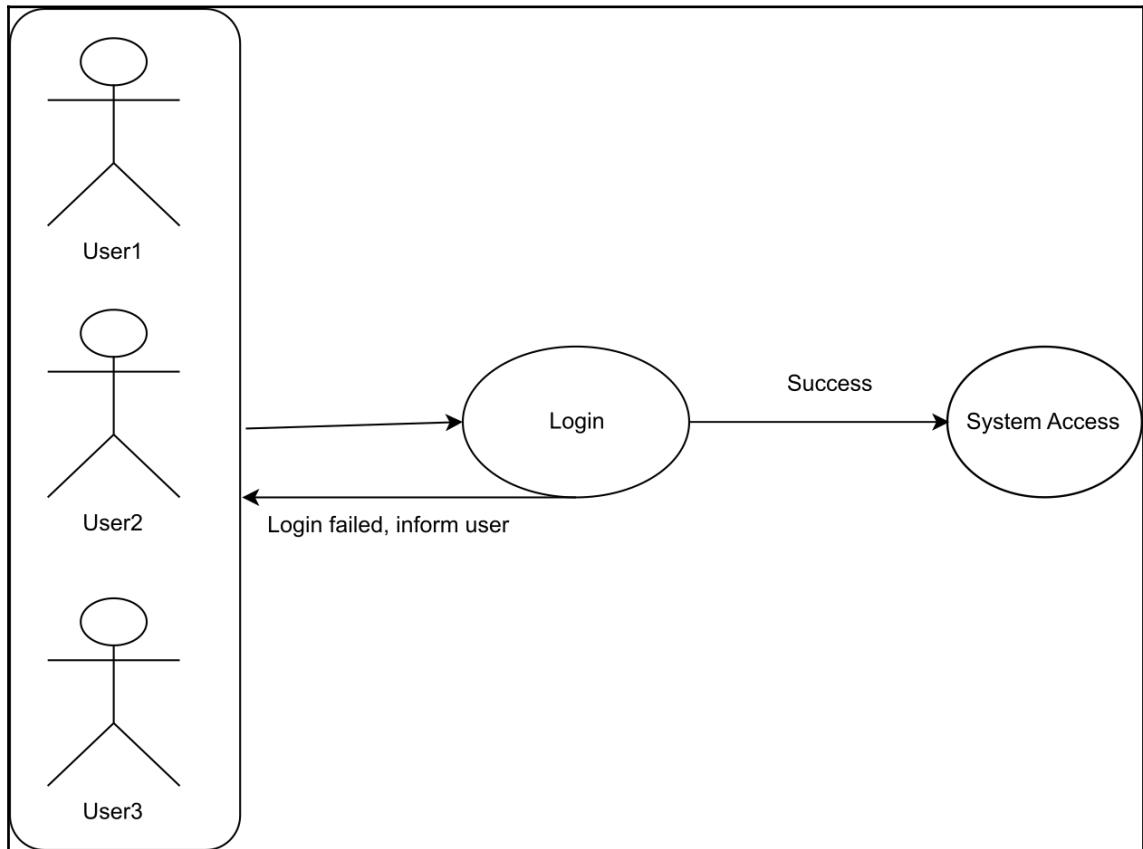




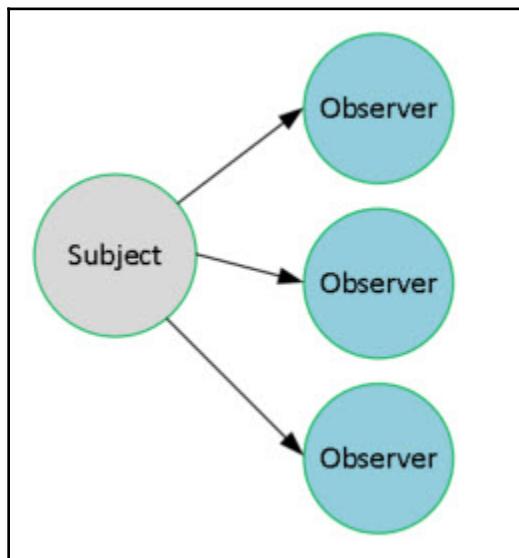




Chapter 13: Miscellaneous Best Practices



Chapter 14: Additional Content_Chapter 2



```
C:\Program Files\dotnet\dotnet.exe
Hello World!
I observer the new quantity value of 12.
I observer the new quantity value of 12.
I observer the new quantity value of 12.
I observer the new quantity value of 17.
I observer the new quantity value of 17.
I observer the new quantity value of 17.
Enter a key to quit.
```