

## Cognizant Academy

### Retail Product Management System

#### FSE – Business Aligned Project Case Study Specification

Version 1.0

	Prepared By / Last Updated By	Reviewed By	Approved By
Name	Nandini Murugesan		
Role	Solution designer		
Signature			
Date			

# Table of Contents

<b>1.0</b>	<b>Important Instructions</b>	<b>3</b>
<b>2.0</b>	<b>Introduction</b>	<b>4</b>
2.1	Purpose of this document	4
2.2	Project Overview	4
2.3	Scope	4
2.4	Hardware and Software Requirement	5
2.5	System Architecture Diagram	6
<b>3.0</b>	<b>Functional Requirements and High Level Design</b>	<b>6</b>
3.1	Individual Components of the System	6
3.1.1	Product Microservice	6
3.1.2	ProceedToBuy Microservice	7
3.1.3	Vendor Microservice	8
3.1.4	Authorization Microservice	9
3.1.5	Swagger	9
3.1.6	e-Commerce Portal (MVC)	10
<b>4.0</b>	<b>Cloud Deployment requirements</b>	<b>10</b>
<b>5.0</b>	<b>Design Considerations</b>	<b>10</b>
<b>6.0</b>	<b>Reference learning</b>	<b>11</b>
<b>7.0</b>	<b>Change Log</b>	<b>11</b>

# 1.0 Important Instructions

1. Associate must adhere to the Design Considerations specific to each Technology Track
2. Associate must not submit project with compile-time or build-time errors
3. Being a Full-Stack Developer Project, you must focus on ALL layers of the application development
4. Unit Testing is Mandatory, and we expect a code coverage of 100%. Use Mocking Frameworks wherever applicable.
5. All the Microservices, Client Application, DB Scripts, have to be packaged together in a single ZIP file. Associate must submit the solution file in ZIP format only
6. If backend has to be set up manually, appropriate DB scripts have to be provided along with the solution ZIP file
7. A READ ME has to be provided with steps to execute the submitted solution, the Launch URLs of the Microservices in cloud must be specified.  
(Importantly, the READ ME should contain the steps to execute DB scripts, the LAUNCH URL of the application)
8. Follow coding best practices while implementing the solution. Use appropriate design patterns wherever applicable
9. You are supposed to use an In-memory database or sessions as specified, for the Microservices that will be deployed in cloud. No Physical database is suggested.

## 2.0 Introduction

### 2.1 Purpose of this document

The purpose of the software requirement document is to systematically capture requirements for the project and the system. "Retail Product Management System" that has to be developed. Both functional and non-functional requirements are captured in this document. It also serves as the input for the project scoping.

The scope of this document is limited to addressing the requirements from a user, quality, and non-functional perspective.

High Level Design considerations are also specified wherever applicable, however the detailed design considerations have to be strictly adhered to during implementation.

### 2.2 Project Overview

A leading Retail Organization wants to strengthen its Middleware by exposing the core logic related to Product Management as Microservices. This middle ware Microservices will be hosted on Cloud so that all the up/downstream applications can get an access to this for performing business transactions.

There will also be a e-Commerce Portal to be developed part of this scope that consumes these Microservices and responses back to customers who are purchasing a product.

### 2.3 Scope

Below are the modules that needs to be developed part of the Project:

Req. No.	Req. Name	Req. Description
REQ_01	Products Module	Products Module is a Middleware Microservice that performs following operations: <ul style="list-style-type: none"><li>• Search Product by Id</li><li>• Search Product by Name</li><li>• Add Rating to Product</li></ul>
REQ_02	ProceedToBuy Module	ProceedToBuy Module is a Middleware Microservice that performs following operations: <ul style="list-style-type: none"><li>• Customer can add Product to wish list – this is when product is out of stock</li><li>• Add Product to cart</li></ul>
REQ_03	Vendor Module	Vendor Module is a Middleware Microservice that performs the following operations:

		<ul style="list-style-type: none"> <li>• Give List Of Vendors for the given Product</li> </ul>
REQ_04	Ecommerce Portal	<p>An e-Commerce Portal that allows a customer to Login and allows to do following operations:</p> <ul style="list-style-type: none"> <li>• Login</li> <li>• Search for a product by id or name.</li> <li>• In the Product page, key in zip code and expected delivery date</li> <li>• Proceed To Buy.</li> <li>• If product is out of stock, add to wish list</li> <li>• Display product details(Id, Name, Price, Vendor details, Delivery charge, Order Total) in checkout page.</li> <li>• Rate a Product</li> </ul>

## 2.4 Hardware and Software Requirement

### 1. Hardware Requirement:

- a. Developer Desktop PC with 8GB RAM

### 2. Software Requirement (Java)

- a. Spring Tool Suite (STS) Or any Latest Eclipse
  - i. Have PMD Plugin, EcEmma Code Coverage Plugin and AWS Code Commit Enabled
  - ii. Configure Maven in Eclipse
- b. Maven
- c. Docker (Optional)
- d. Postman Client in Chrome

### 3. Software Requirement (Dotnet)

- a. Visual studio 2017 enterprise edition
- b. SQL Server 2014
- c. Postman Client in Chrome
- d. Azure cloud access



<b>Trigger</b> – Can be invoked from e-Commerce Portal (local MVC app)
<b>Steps and Actions</b> <ol style="list-style-type: none"> <li>1. Customer Portal will request for searchProductById, searchProductByName operations.</li> <li>2. Authorization has to be performed.</li> <li>3. At any point in time, customer must be able to search a product details from the eCommerce Portal Client. Hence the Product Microservice must expose the Product Details through /searchProductById and /searchProductByName REST End Point.</li> <li>4. Customer must be able to add Rating to a product which must get associated to the Product entity, for the respective product</li> </ol>
<b>Non-Functional Requirement:</b> <ul style="list-style-type: none"> <li>• Only Authorized Customer can access these REST End Points</li> </ul>

### 3.1.2 ProceedToBuy Microservice

Product Management System	ProceedToBuy Microservice
<b>Functional Requirements</b> <p>Can assume that eCommerce Portal App is the only client to this Microservice. An authorized customer can add the product to the cart.</p> <p>An authorized customer can add product to cart and view the product details(Product Price, Delivery Date, Vendor details) in the checkout/cart page.</p> <p>If Product is not available, the customer can add the product to wishlist.</p> <p>Post Authorization, ProceedToBuy Microservice will interact with the Vendor Microservice for the following functionalities:</p> <ul style="list-style-type: none"> <li>• The Microservice will interact with Vendor Module, to get a assign a Vendor for a given order.</li> </ul>	
<b>Entities</b> <ul style="list-style-type: none"> <li>○ <b>Vendor</b> &lt;Vendor ID, Vendor Name, Delivery Charge&gt;</li> <li>○ <b>Cart</b> &lt;CartId, ProductId, zipcode, Delivery Date, Vendor object&gt;</li> <li>○ <b>Vendor Wishlist</b> &lt;Vendor ID, Product ID, Quantity, Date Added to Wishlist etc.&gt;</li> </ul>	
<b>REST End Points</b>	
<b>Cart Microservice</b> <ul style="list-style-type: none"> <li>○ POST: /addProductToCart (Input: Customer_ID, Product_ID, Zip_Code, Expected_Delivery_Date   Output: Cart)</li> <li>○ POST: /addProductToWishlist (Input: Customer_ID, Product_ID   Output:</li> </ul>	

Status)
<b>Trigger</b> – Can be invoked from e-Commerce Portal (local MVC app)
<b>Steps and Actions</b> <ol style="list-style-type: none"> <li>1. Customer Portal will request for ProceedToBuy operations.</li> <li>2. For all the operations, Customer Profile will be verified before interacting with other Microservices.</li> <li>3. In product page customer should key in zip code and select expected delivery date. On click of submit /addProductToCart end point is invoked, the product is added to cart and the user is directed to the cart page. The cart details are displayed to the user there.</li> <li>4. Internally, in the /addProductToCart service method will invoke the /getVendorDetails end point of the Vendor microservice to get the appropriate vendor to fulfill the order. The response returned by Vendor has to be cascaded as the response in this end point.</li> <li>5. Appropriate vendor must be tagged to the Product based on vendor rating.</li> </ol>
<b>Non-Functional Requirement:</b> <ul style="list-style-type: none"> <li>• Only Authorized Customer can access these REST End Points.</li> </ul>

### 3.1.3 Vendor Microservice

Product Management System	Vendor Microservice
<b>Functional Requirements</b> <p>During the order fulfillment cycle, a vendor will be mapped to the order. The vendor will ship the product from their warehouse and deliver it to the customer location.</p> <p>ProceedToBuy Microservice interacts with Vendor Microservice to assign a vendor for the customer's order. Post authorization of request, Vendor Microservice allows the following operations:</p> <ul style="list-style-type: none"> <li>• Choose the Vendor for the given product, who has enough quantities to supply</li> </ul> <p>Perform the below mentioned checks to select a vendor</p> <ul style="list-style-type: none"> <li>○ Vendor should have the given product in stock. If available, retrieve all vendors along with the rating and return to the ProceedToBuy service</li> </ul> <p>Assume that the vendor details will be stored in system automatically, which is out of scope of the system.</p>	
<b>Entities</b> <ul style="list-style-type: none"> <li>○ <b>Vendor</b> &lt;Vendor ID, Vendor Name, Delivery Charge, Rating&gt;</li> <li>○ <b>Vendor Stock</b></li> </ul>	



<Product ID, Vendor ID, Stock In Hand, Expected Stock Replishment Date, etc.>	
<b>REST End Points</b> <b>Vendor Microservice</b> <ul style="list-style-type: none"> <li>GET: /getVendorDetails (Input: Product_ID   Output: Vendor ID, Vendor Name, rating, Delivery Charge)</li> </ul>	
<b>Trigger</b> – Can be invoked from ProceedToBuy Microservice	
<b>Steps and Actions</b> <ol style="list-style-type: none"> <li>Vendor Microservice will have 1 End Points exposed to ProceedToBuy Microservice</li> <li>If /getVendorDetails end point is invoked by ProccedToBuy Microservice, the Vendor Microservice will perform the necessary checks w.r.t stock and will return the response back to ProccedToBuy Microservice.</li> </ol>	
<b>Non-Functional Requirement:</b> <ul style="list-style-type: none"> <li>Only Authorized Customer can access these REST End Points.</li> <li>If there are more than 1 customer adding the product to cart and proceeding to buy, the tasks have to be handled in parallel</li> </ul>	

### 3.1.4 Authorization Microservice

Product Management System	Authorization Microservice
<b>Security Requirements</b> <ul style="list-style-type: none"> <li>Service to Service communication has to happen using JWT</li> <li>Pass End User Context across Microservices</li> <li>Have the token expired after specific amount of time say 15 minutes.</li> <li>Have this service configured in the cloud along with other services</li> </ul>	

### 3.1.5 Swagger

Product Management System	Swagger
<b>Documentation Requirements</b> <ul style="list-style-type: none"> <li>All the Microservices must be configured with Swagger for documentation</li> </ul>	
<b>Java implementation</b>	

- Register the swagger resources in the Swagger Microservice and enable them as REST end points
- Configure this service along with other services in the cloud

### 3.1.6 e-Commerce Portal (MVC)

Product Management System	e-Commerce Portal
<b>e-Commerce Portal Requirements</b> An e-Commerce Portal that allows a customer to Login and allows to do following operations: <ul style="list-style-type: none"> <li>○ Login</li> <li>○ Search for a product by id or name.</li> <li>○ In the Product page, key in zip code and expected delivery date.</li> <li>○ Proceed To Buy.</li> <li>○ Add Product to Wishlist</li> <li>○ Display product details(Id, Name, Description, Price, Vendor details,Delivery charge, Order total) in checkout page.</li> </ul> <p>Each of the above operations will reach out to the middleware Microservices that are hosted in cloud.</p>	

## 4.0 Cloud Deployment requirements

- All the Microservices must be deployed in Cloud
- All the Microservices must be independently deployable. They have to use In-memory database or user sessions wherever applicable
- The Microservices has to be dockerized and these containers must be hosted in Cloud using CI/CD pipelines
- The containers have to be orchestrated using AWS/Azure Kubernetes Services.
- These services must be consumed from an MVC app running in a local environment.

## 5.0 Design Considerations

These design specifications, technology features have to be strictly adhered to.



CDE-Project-Design  
Considerations.pptx

## 6.0 Reference learning

Please go through all of these k-point videos for Microservices deployment into AWS.

<a href="https://cognizant.kpoint.com/app/video/gcc-6e36500f-c1af-42c1-a6c7-ed8aac53ab22">https://cognizant.kpoint.com/app/video/gcc-6e36500f-c1af-42c1-a6c7-ed8aac53ab22</a>
<a href="https://cognizant.kpoint.com/app/video/gcc-92f246c9-024a-40b7-8bfc-96b3ce7c1a39">https://cognizant.kpoint.com/app/video/gcc-92f246c9-024a-40b7-8bfc-96b3ce7c1a39</a>
<a href="https://cognizant.kpoint.com/app/video/gcc-cfedd9c1-e29e-4e3e-b3e2-1960277f72a3">https://cognizant.kpoint.com/app/video/gcc-cfedd9c1-e29e-4e3e-b3e2-1960277f72a3</a>
<a href="https://cognizant.kpoint.com/app/video/gcc-900a7172-43b7-42f3-a6cc-e301bd9cc9b3">https://cognizant.kpoint.com/app/video/gcc-900a7172-43b7-42f3-a6cc-e301bd9cc9b3</a>

### Other References:

Java 8 Parallel Programming	<a href="https://dzone.com/articles/parallel-and-asynchronous-programming-in-java-8">https://dzone.com/articles/parallel-and-asynchronous-programming-in-java-8</a>
Feign client	<a href="https://dzone.com/articles/Microservices-communication-feign-as-rest-client">https://dzone.com/articles/Microservices-communication-feign-as-rest-client</a>
Swagger (Optional)	<a href="https://dzone.com/articles/centralized-documentation-in-Microservice-spring-b">https://dzone.com/articles/centralized-documentation-in-Microservice-spring-b</a>
ECL Emma Code Coverage	<a href="https://www.eclipse.org/community/eclipse_newsletter/2015/august/article1.php">https://www.eclipse.org/community/eclipse_newsletter/2015/august/article1.php</a>
Lombok Logging	<a href="https://javabydeveloper.com/lombok-slf4j-examples/">https://javabydeveloper.com/lombok-slf4j-examples/</a>
Spring Security	<a href="https://dzone.com/articles/spring-boot-security-json-web-tokenjwt-hello-world">https://dzone.com/articles/spring-boot-security-json-web-tokenjwt-hello-world</a>
H2 In-memory Database	<a href="https://dzone.com/articles/spring-data-jpa-with-an-embedded-database-and-spring-boot">https://dzone.com/articles/spring-data-jpa-with-an-embedded-database-and-spring-boot</a> <a href="https://www.baeldung.com/spring-boot-h2-database">https://www.baeldung.com/spring-boot-h2-database</a>
ApplInsights logging	<a href="https://www.codeproject.com/Tips/1044948/Logging-with-ApplicationInsights">https://www.codeproject.com/Tips/1044948/Logging-with-ApplicationInsights</a>
Error response in WebApi	<a href="https://stackoverflow.com/questions/10732644/best-practice-to-return-errors-in-asp-net-web-api">https://stackoverflow.com/questions/10732644/best-practice-to-return-errors-in-asp-net-web-api</a>
Read content from CSV	<a href="https://stackoverflow.com/questions/26790477/read-csv-to-list-of-objects">https://stackoverflow.com/questions/26790477/read-csv-to-list-of-objects</a>
Access app settings key from appSettings.json in .Netcore application	<a href="https://www.c-sharpcorner.com/article/reading-values-from-appsettings-json-in-asp-net-core/">https://www.c-sharpcorner.com/article/reading-values-from-appsettings-json-in-asp-net-core/</a> <a href="https://docs.microsoft.com/en-us/aspnet/core/fundamentals/configuration/?view=aspnetcore-3.1">https://docs.microsoft.com/en-us/aspnet/core/fundamentals/configuration/?view=aspnetcore-3.1</a>

## 7.0 Change Log

	Changes Made
V1.0.0	Initial baseline created on <29-Jul-2020> by <Nandini Murugesan>

	<b>Section No.</b>	<b>Changed By</b>	<b>Effective Date</b>	<b>Changes Effected</b>