

## Cognizant Academy

### Pharmacy medicine supply Management System

### FSE – Business Aligned Project Case Study Specification

### Version 1.0

	Prepared By / Last Updated By	Reviewed By	Approved By
Name	Seshadri M R		
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>System Requirements</b>	<b>6</b>
3.1.1	Functional Requirements – Medicine stock Microservice	6
3.1.2	Functional Requirements – Medical representative schedule Microservice	7
3.1.3	Functional Requirements – Pharmacy medicine supply Microservice	9
3.1.4	Functional Requirements – Authorization Microservice	10
3.1.5	Functional Requirements – Pharmacy Medicine Supply portal	10
<b>4.0</b>	<b>Cloud Deployment requirements</b>	<b>11</b>
<b>5.0</b>	<b>Design Considerations</b>	<b>11</b>
<b>6.0</b>	<b>Reference learning</b>	<b>11</b>
<b>7.0</b>	<b>Change Log</b>	<b>13</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 Unit testing and 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 code level data as specified, for the Microservices that should be deployed in cloud. No Physical database is suggested for Microservice.

## 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 "Pharmacy medicine 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 Pharmaceutical company wants to automate the logic of forming a schedule for their medical representatives to meet the targeted doctors to explain their medicines and its nature for prescription. Based on the response from doctors, the medicine demand will be determined. This should be fed into the system to determine the medicine supply detail to its Pharmacists. The application developed will target this requirement.

### 2.3 Scope

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

Req. No.	Req. Name	Req. Description
REQ_01	Medicine stock module	<p>This module is a Middleware Microservice that performs the following operations:</p> <ul style="list-style-type: none"><li>Provides information on the pharma company medicine stock by the godown area</li></ul>
REQ_02	Medical representative schedule module	<p>This module is a Middleware Microservice that performs following operations:</p> <ul style="list-style-type: none"><li>Creates a schedule to have meetings with doctors.</li><li>The list of doctors that this pharma company is targeting can be stored as a pre-defined information in this Microservice.</li><li>This module should interact with the medicine stock module to find the medicine stock to be explained to the targeted doctors.</li></ul>

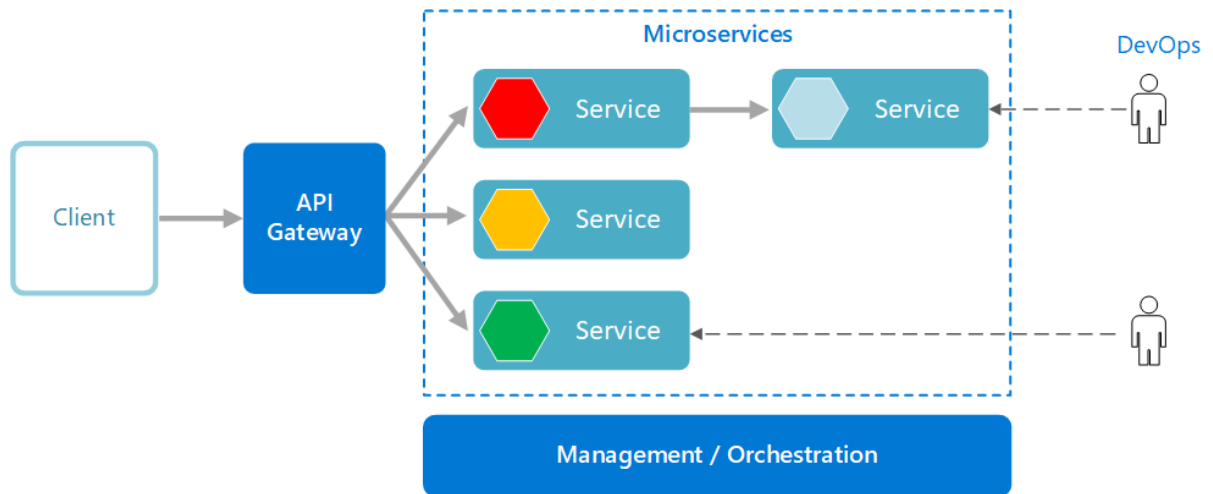
REQ_03	Pharmacy Supply module	<p>This module is a Middleware Microservice that performs the following operations:</p> <ul style="list-style-type: none"> <li>Gets the medicine count as demand as input from web portal.</li> <li>Interacts with the Medicine supply microservice to find the final demand of medicine that can be supplied to its pharmacists.</li> </ul>
REQ_04	Authorization service	This microservice is used with anonymous access to Generate JWT
REQ_05	Pharmacy medicine supply portal	<p>A Web Portal that allows a member to Login and allows to do following operations:</p> <ul style="list-style-type: none"> <li>Login</li> <li>View a schedule for medical representative for doctors meet</li> <li>Provide in the data from medical reps with the medicine demand. Based on this input, the PharmacySupply microservice should provide the medicine count to be distributed to its pharmacists. This detail should be displayed on the UI.</li> <li>Medicine demand and supply to pharmacists should be saved in the database.</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

## 2.5 System Architecture Diagram



## 3.0 System Requirements

### 3.1.1 Functional Requirements – Medicine stock Microservice

Pharmacy medicine supply management System	MedicineStock Microservice
<b>Functional Requirements</b> This Microservice should be invoked from the Pharmacy Supply Microservice. It allows the following operations: <ul style="list-style-type: none"><li>It provides information on the details of the medicines like the medicine name, chemical composition, date of expiry, number of tablets target ailment</li></ul>	
<b>Entity</b>  <b>MedicineStock</b> <ol style="list-style-type: none"><li><b>Name</b> &lt;Medicine name&gt;</li><li><b>ChemicalComposition</b> &lt;Comma separated names of the chemicals used&gt;</li><li><b>TargetAilment</b></li></ol>	

<p>&lt;The ailment for which the medicine is created&gt; Eg: Orthopaedics, General, Gynaecology</p> <p>4. <b>DateOfExpiry</b></p> <p>5. <b>NumberOfTabletsInStock</b></p> <p><b>REST End Points</b></p> <p><b>Claims Microservice</b></p> <p>1. GET: /MedicineStockInformation (Input: None   Output: List of MedicineStock)</p>
<p><b>Trigger</b> – Should be invoked from PharmacySupply Microservice to determine the current medicine stock information.</p>
<p><b>Steps and Actions</b></p> <ul style="list-style-type: none"> <li>○ This microservice should have only 1 endpoint to get the current medicine stock detail. The detail of the medicine along with the godown area that contains the stock.</li> <li>○ The data can be pre-defined in the Microservice or obtained from database</li> </ul>
<p><b>Non-Functional Requirement:</b></p>

### 3.1.2 Functional Requirements – Medical representative schedule Microservice

Pharmacy medicine supply management System	MedicalRepresentativeSchedule Microservice
<p><b>Functional Requirements</b></p> <p>The intent of this Microservice is to provide a doctor meet schedule for the Representatives of the company. Post Authorization using JWT, based on pre-defined doctor information, the microservice should map the available representatives for a period of 5 days. The start date of the schedule should be sent as input from the web portal. Sunday shouldn't be considered for reps meeting.</p> <p><b>Assumption:</b></p> <p>The pharma company has</p> <ul style="list-style-type: none"> <li>• 3 Medical representatives</li> <li>• 5 doctors information who has given one slot(1 PM to 2 PM) everyday</li> </ul> <p>There can be pre-defined Medical representative detail in this Microservice, or can be fetched from database</p> <p>There can be pre-defined Doctor detail in this Microservice stored in a flat file. Doctor detail will contain fields – Name, Contact number, TreatingAilment(Eg: Orthopaedics, General, Gynaecology)</p> <p>This Microservice should interact with the MedicineStock microservice to determine the medicines to</p>	

be used for explanation to Doctors

## Entities

### RepSchedule

1. **Name**  
<Medical rep name>
2. **Doctor name**
3. **Meeting Slot**
4. **Date of meeting**
5. **Doctor Contact number**

## REST End Points

### MedicalRepresentativeSchedule Microservice

- GET: /RepSchedule (Input: scheduleStartDate | Output: List of RepSchedule)

**Trigger** – Should be invoked from Pharmacy medicine supply management (local MVC app)

## Steps and Actions

1. This Microservice should get the schedule start date from the web portal.
2. The doctor details should be loaded into the application from a flat file.
3. Based on the assumption that there would be 3 Medical representatives for the Pharma company, a schedule should be formed in the following format. The schedule will start from the schedule start date sent as input from the web portal, for a period of 5 days. Sundays should not be included. Eg: Schedule start : 30<sup>th</sup> July 2020
4. This microservice should interact with the medicine stock and fetch the medicines that are for the doctor's treating ailment and display them.

Rep name	Doctor name	Treating Ailment	Medicine	Slot	Date	Doctor Contact #
R1	D1	Orthopaedics	Orthoherb, Cholecalciferol	1 PM to 2 PM	30-Jul-2020	9884122113
R2	D2	General	Gaviscon, Dolo-650	1 PM to 2 PM	31-Jul-2020	9884122113
R3	D3	General	Gaviscon, Dolo-650	2 PM to 3 PM	1-Aug-2020	9884122113



	R1	D4	Orthopaedics	Orthoherb, Cholecalciferol	2 PM to 3 PM	3- Aug- 2020	9884122113
	R2	D5	Gynaecology	Cyclopam, Hilact	1 PM to 2 PM	4- Aug- 2020	9884122113

<b>Non-Functional Requirement:</b> <ul style="list-style-type: none"> <li>Only Authorized requests can access these REST End Points</li> </ul>
--

### 3.1.3 Functional Requirements – Pharmacy medicine supply Microservice

Pharmacy medicine supply management System	PharmacyMedicineSupply Microservice
<b>Functional Requirements</b> This should be invoked from the Web portal. It allows the following operations: <ul style="list-style-type: none"> <li>Maintains the list of pharmacies that the company does business with.</li> <li>Receives input from the web portal for the medicine count needed. Upon receiving this, this Microservice should interact with MedicineStock service to get the current stock. This count should be split equally among all the pharmacists and the count should be returned to the web portal. If the stock count is lesser than the demand, then the stock count should be considered.</li> </ul>	
<b>Entities</b>  <b>MedicineDemand</b> <ol style="list-style-type: none"> <li>Medicine</li> <li>DemandCount</li> </ol> <b>PharmacyMedicineSupply</b>	

<ol style="list-style-type: none"> <li>1. Pharmacy name</li> <li>2. Medicine name</li> <li>3. Supply count</li> </ol> <p><b>REST End Points</b></p> <p><b>PensionDisbursement Microservice</b></p> <p>GET: /PharmacySupply (Input: List of MedicineDemand   Output: List of PharmacyMedicineSupply)</p>
<b>Trigger:</b> Should be invoked from Web portal
<p><b>Steps and Actions:</b></p> <ul style="list-style-type: none"> <li>○ The list of pharmacies(name) can be maintained as pre-defined value or can be obtained from database.</li> <li>○ Upon receiving the demand count, the supply count should be determined based on the stock count and this should be returned to the web portal</li> </ul>
<p><b>Non-Functional Requirement:</b></p> <p>Only Authorized requests can access these REST End Points</p>

### 3.1.4 Functional Requirements – Authorization Microservice

Pharmacy medicine supply management System	Authorization Microservice
<p><b>Security Requirements</b></p> <ul style="list-style-type: none"> <li>○ Create JWT</li> <li>○ Have the token expired after specific amount of time say 30 minutes</li> <li>○ Has anonymous access to get the token detail</li> </ul>	

### 3.1.5 Functional Requirements – Pharmacy Medicine Supply portal

Pharmacy medicine supply management System	Pharmacy medicine supply Portal
<p><b>Client Portal Requirements</b></p> <ul style="list-style-type: none"> <li>○ Pharmacy medicine supply Portal must allow a member to Login. Once successfully logged in, the member do the following operations: <ul style="list-style-type: none"> <li>○ Provide a web page to view its medical representatives' schedule for doctor visit on providing a start date. The schedule will be for 5 calendar days, except Sunday.</li> </ul> </li> </ul>	

- Upon receiving doctor's response offline, the application user should provide the medicine demand. The list of medicines should be obtained by invoking the MedicineStock microservice with a textbox against it. The application user will fill the demand count. This should be sent as input to the PharmacyMedicineSupply microservice to get the pharmacy supply count
- The medicine demand and supply should be stored in the database
- Each of the above operations should reach out to the middleware Microservices that are hosted in cloud.

## 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 data in the application 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

Java and Dotnet specific design considerations are attached here. 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 Azure Kubernetes Service.

<a href="#">AzureWithCICD-1</a>
<a href="#">AzureWithCICD-2</a>
<a href="#">AzureWithCICD-3</a>
<a href="#">AzureWithCICD-4</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 <Seshadri M R>			
	Section No.	Changed By	Effective Date	Changes Effected