Drink & Delight

Inventory Management System

Name	Role
Mukul Garg	Product Owner

Table of Contents

1. Introduction

The project Inventory Management System is a complete desktop based application designed on spring framework and angular. The main aim of the project is to develop Inventory Management System Model software in which all the information regarding the stock of the organization will be presented. It is an intranet based desktop application which has an admin component to manage the inventory and maintenance of the inventory system. This desktop application is based on the management of stock of an organization. The application contains general organization profile, sales details, Purchase details and the remaining stock that are presented in the organization. There is a provision of updating the inventory also. This application also provides the remaining balance of the stock as well as the details of the balance of transaction. Each new stock is created and entitled with the name and the entry date of that stock and it can also be updated any time when required as per the transaction or the sales is returned in case.

The project helps to manage both products and raw materials in the inventory.It also provides features so that users can order products and raw materials, admin can than perform the required operation and manage the stock.

2. Overview of project architecture

The projects uses microservice architecture.Based on the type of operation performed and type on which operation is performed 2 core microservice is created

   a) Update Raw Material Stock (for update raw material details)
   b) Display Distributors Details (for Display Distributor Details)

3. Modules

List of Modules
   ● update raw material stock
   ● display distributor details

Update Raw Material Stock: -

In this module the details of the raw material stock will be updated. When the user clicks on the 'Update Raw Material Stock 'Tab, the user sees a form to update the following Raw Material Stock details:

   1. Order Id
   2. Price Per Unit
   3. Quantity Unit
   4. Quality Check

Order id should be present in the database, if the user enters the wrong Order Id which is not in the database, then show the message "Please enter the correct order id".

Total Price is the multiple of the Price Per Unit and the Quantity Unit which is calculated in the backend by itself.

Display Distributors Details: -

When the user clicks on 'Display Distributors Details ' tab, the user need to inset the required supplier Id.
Now here the user has to select the desired supplier Id whose details he would like to see. Here, the values in the dropdowns of Supplier ID is fixed in the angular html Template. This has been handled in the Angular Component.

Finally, he clicks on the Submit button which would send the request to the Rest Controller at the backend.

The table with all the details (Supplier Id, Name, Address, Phone No.) of that particular supplier is displayed.

Sequence Diagrams

Sequence Diagram for Display Distributor Details
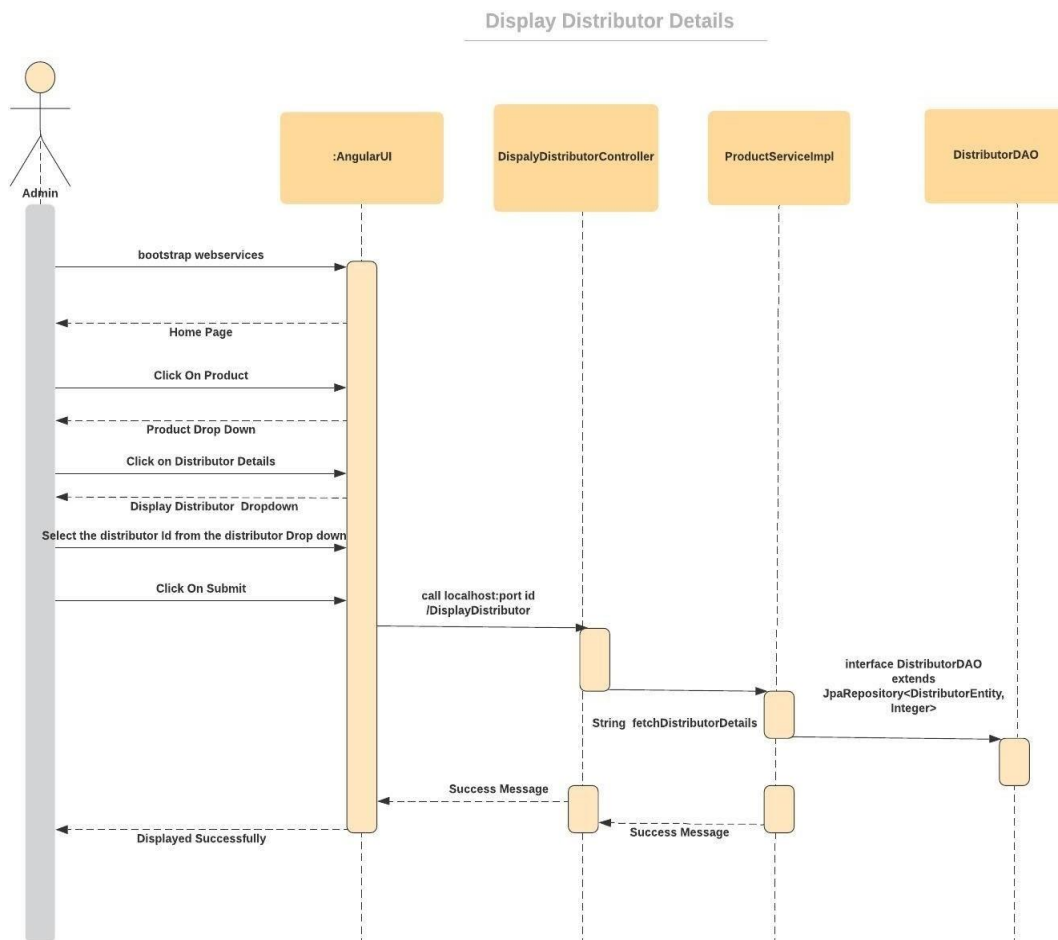
Display Distributor Details

Admin

:AngularUI

DispalyDistributorController

ProductServiceImpl

DistributorDAO

bootstrap webservices

Home Page

Click On Product

Product Drop Down

Click on Distributor Details

Display Distributor Dropdown

Select the distributor Id from the distributor Drop down

Click On Submit

call localhost:port id
/DisplayDistributor

interface DistributorDAO
extends
JpaRepository<DistributorEntity,
Integer>

String fetchDistributorDetails

Success Message

Success Message

Displayed Successfully

Enter the Supplier Id : 201

Submit

Supplier Id	Name	Location	Phone Number
201	Deepak	Delhi pahad ganj	1234567890

Activate Windows
Go to Settings to activate Windows.

localhost:4200 says

Supplier Id is not valid

OK

Enter the Supplier Id : 2010

Submit

Activate Windows
Go to Settings to activate Windows.

Sequence Diagram for Update Raw Material Stock
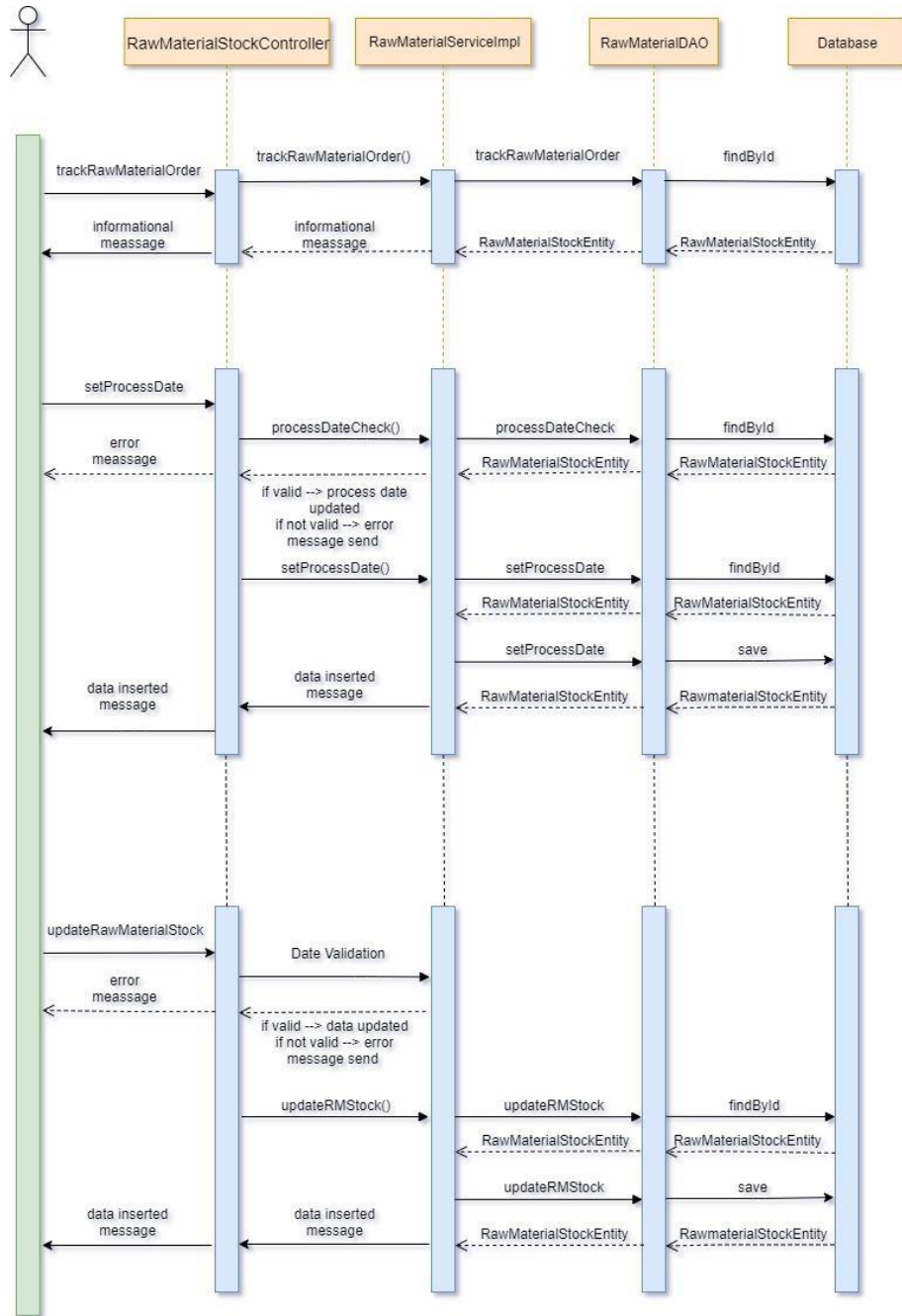
Order Id: [   ]
Price Per Unit: [   ]
Quantity Unit: [   ]
Quality Check: [▾]

[Update]

localhost:4200 says

Data updated successfully

OK

Home   Update Raw Material Stock   Display Distributor Deta   Drink And Delight

Order Id: 120
Price Per Unit: 28
Quantity Unit: 28
Quality Check: Ok
Update

localhost:4200 says

Order Id is not valid

OK

Home   Update Raw Material Stock   Display Distributor Deta   Drink And Delight

Order Id: 1200
Price Per Unit: 28
Quantity Unit: 28
Quality Check: Ok
Update
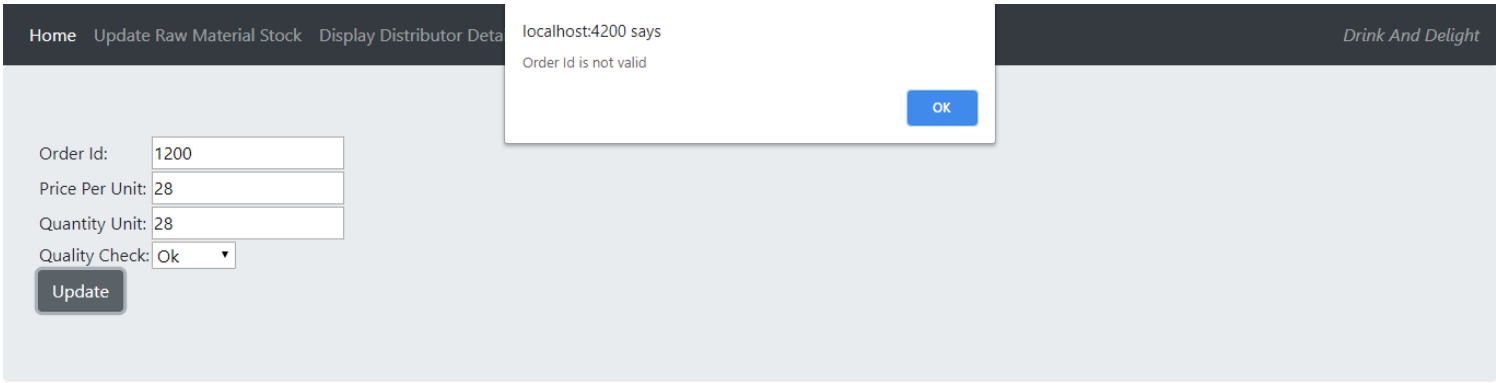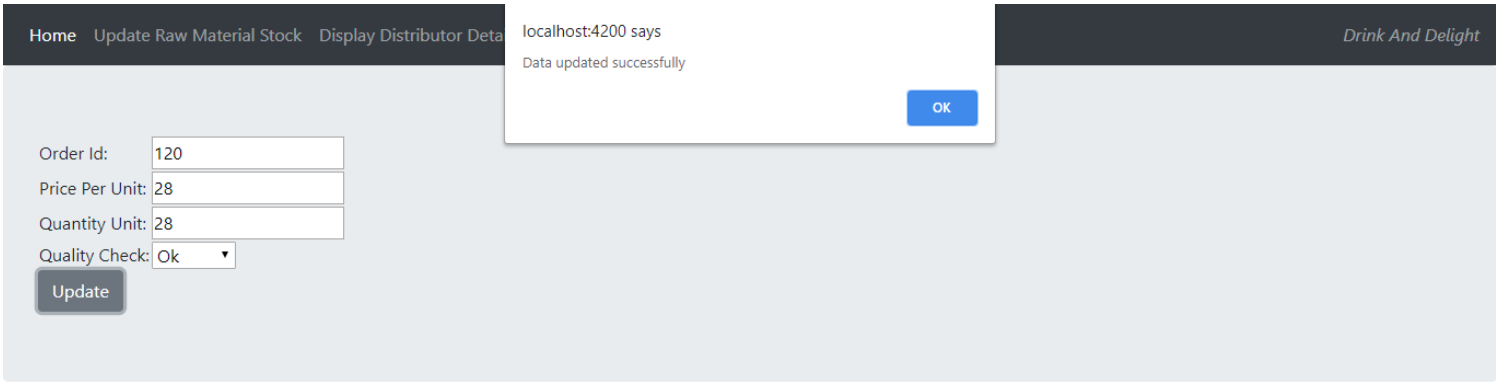
Class Diagram for Raw Material Stock Management:

RawMaterialStock

- orderId: String
- name: String
- price_per_uni: double
- quantityValue: double
- quantityUnit: double
- price: double
- warehouseID: String
- deliveryDate: Date
- manufacturingDate: Dtae
- expiryDate: Date
- qualityCheck: String
- processDate: Date

<<RawMaterialService>>
Interface

- rawMaterialDAO: RawMaterialDAO

+ trackRawMaterialOrder
(RawMaterialStock rawMaterialStock): String
+ doesRawMaterialOrderIdExist(String id): boolean
+ processDateCheck
(RawMaterialStock rawMaterialStock): boolean
+ updateProcessDateinStock
(RawMaterialStock rawMaterialStock): String
+ validateManufacturingDate
(Date manufacturing_date): boolean
+ validateExpiryDate
(Date manufacturing_date, Date expiry_date): boolean
+ updateRawMaterialStock
(RawMaterialStock rawMaterialStock): String
+ doesRawMaterialOrderIdExistInStock
(String orderId): boolean

RawMaterialStockController

- rawMaterialService: RawMaterialService

+ trackProductOrder
(@PathVariable("id") String id): String
+ updateExitDate(@PathVariable("id") String id,
@RequestBody Map<String, Object> paramMap): String
+ updateProductStockDetails(@PathVariable("id") String
@RequestBody Map<String, Object> paramMap): String

<<RawMaterialDAO>>
Interface

+ trackRawMaterialOrder
(RawMaterialStock rawMaterialStock):String
+ processDateCheck
(RawMaterialStock rawMaterialStock): boolean
+ updateProcessDateinStock
(RawMaterialStock rawMaterialStock): String
+ updateRawMaterialStock
(RawMaterialStock rawMaterialStock): String
+ doesRawMaterialOrderIdExistInStock
(String orderId): boolean
+ doesRawMaterialOrderIdExist
(String orderId): boolean