Drink & Delight

Inventory Management System

Team Members:

Team	Role
Atal kumar	Scrum master
Ayushi goyal	member
Harshit verma	member
Mukul Garg	member
Mohit Kumar	member

Capgemini India Technology Services

Plot 72 & 73, EPIP Zone, Vijayanagar, KIADB Export Promotion Industrial Area, Whitefield,
Bengaluru, Karnataka 560066

Table of Contents

1. Introduction
2. Overview
3. Team Member Work Allocation
4. Epic & Stories
5. Use Cases
   - i) Authentication
     - a) Login
     - b) Forgot Password
     - c) Logout
   - 2) Raw Material/ Product Order Management
     - a) Place an Order
     - b) Update an Order
     - c) Track Order
     - d) Display Supplier Details
     - e) Display Distributor Details
   - 3) Stock Management
     - a) Add Stock
     - b) Display Order Details
     - c) Update Stock
   - 4) Network and infrastructure Management
     - a) Add Supplier
     - c) Add warehouse
     - b) Add distributor
6. Class Diagram
7. ER Diagram
8. Test Case Scenario
9. BDD
10. Conclusion

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. Team Member Work Allocation

a) Atal Kumar (Scrum master)
  - Add product stock
  - Update product stock
  - Display Order Details
  - Add supplier
  - Add distributor
  - Add warehouses

b) Harshit Verma
  - Place an Order
  - Update an order

c) Ayushi Goyal
  - Track product order
  - Display supplier details

d) Mukul Garg
  - Add Raw material stock
  - update raw material stock
  - display distributor details

e) Mohit Kumar
  - Track raw material Order
  - Authentication

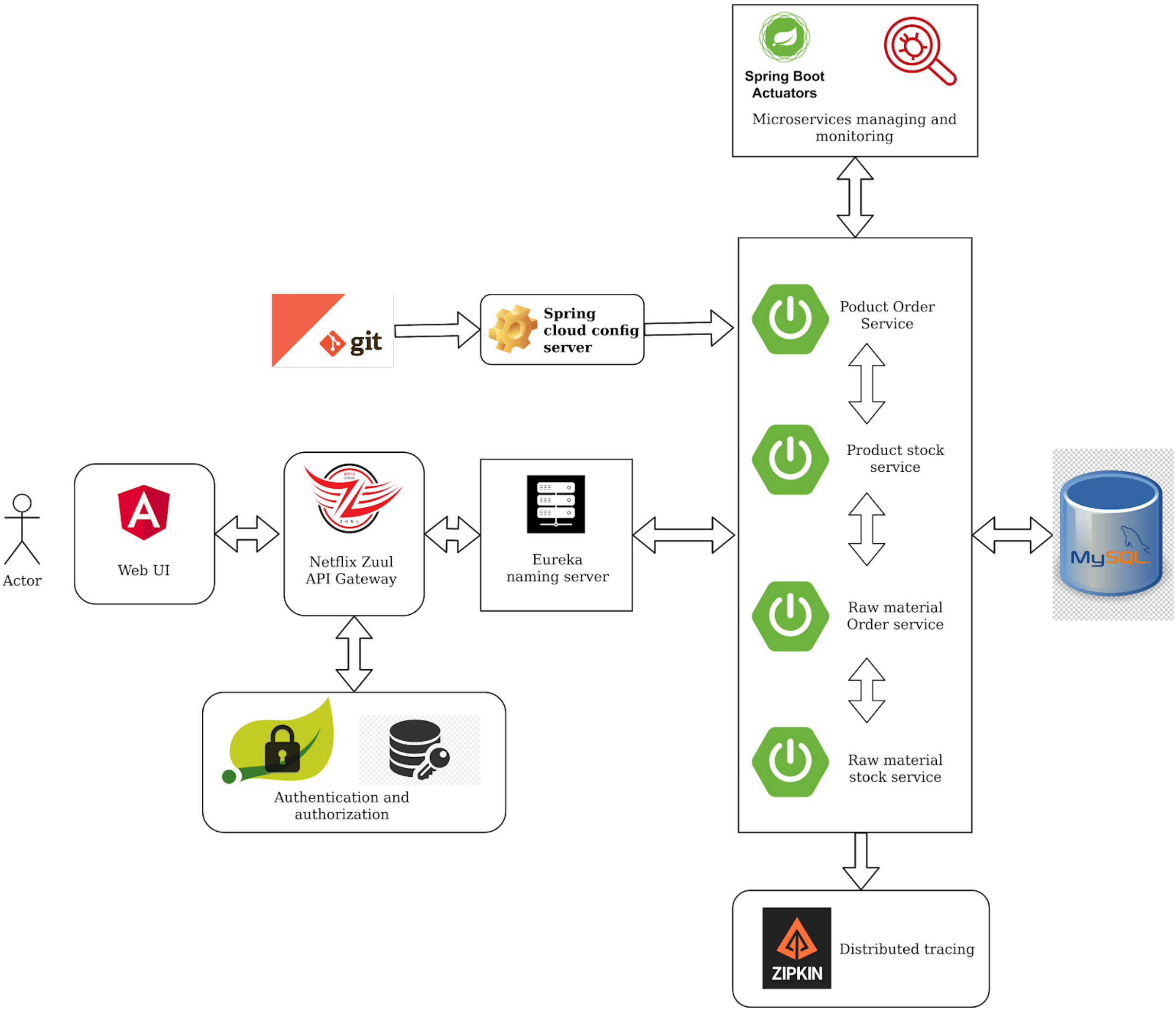
3. Overview of project architecture

The projects uses microservice architecture.Based on the type of operation performed and type on which operation is performed 5 core microservice is created

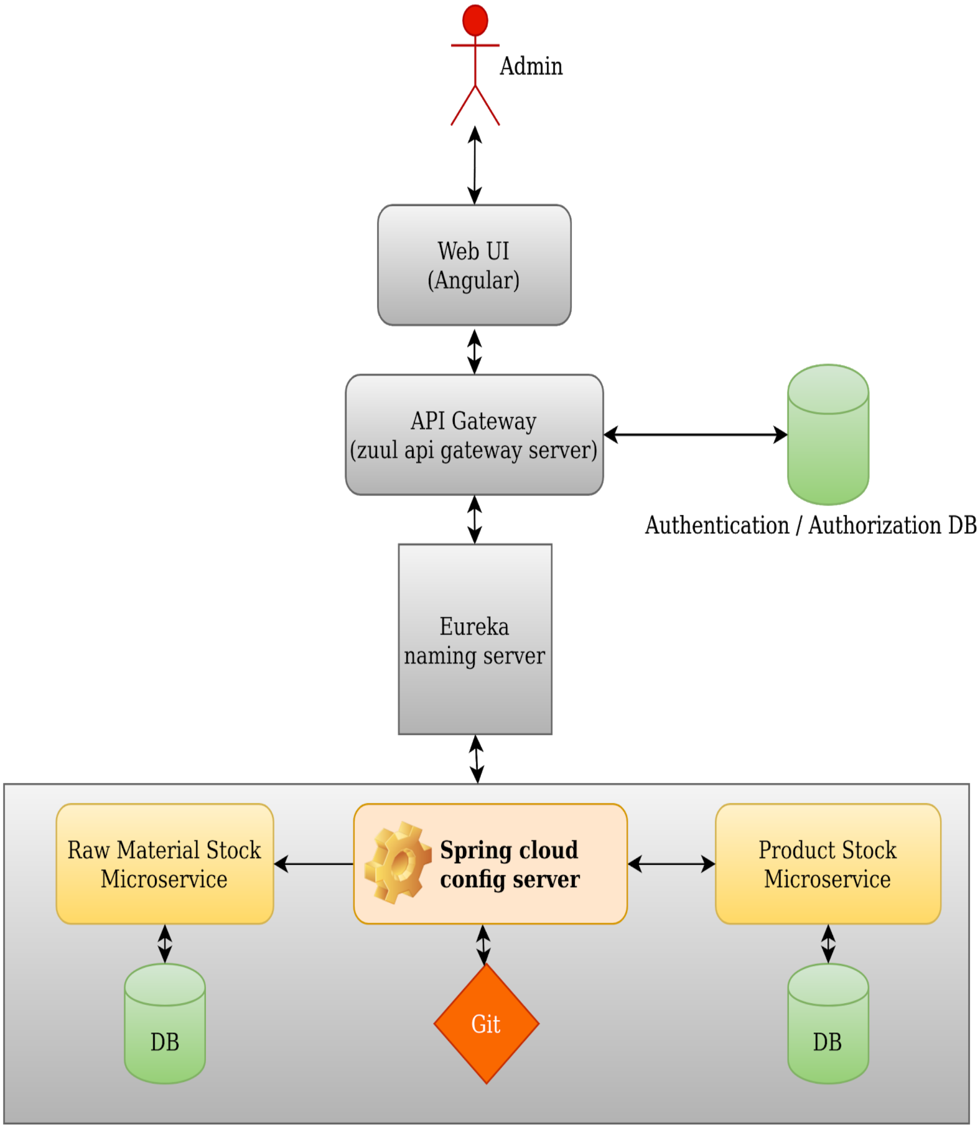
a) product-order (for ordering product by user)
b) product-stock (for stock management of product by admin)
c) raw material-order (for ordering raw material by user)
d) raw material-stock (for stock management of raw material by admin)
e) authentication and authorization

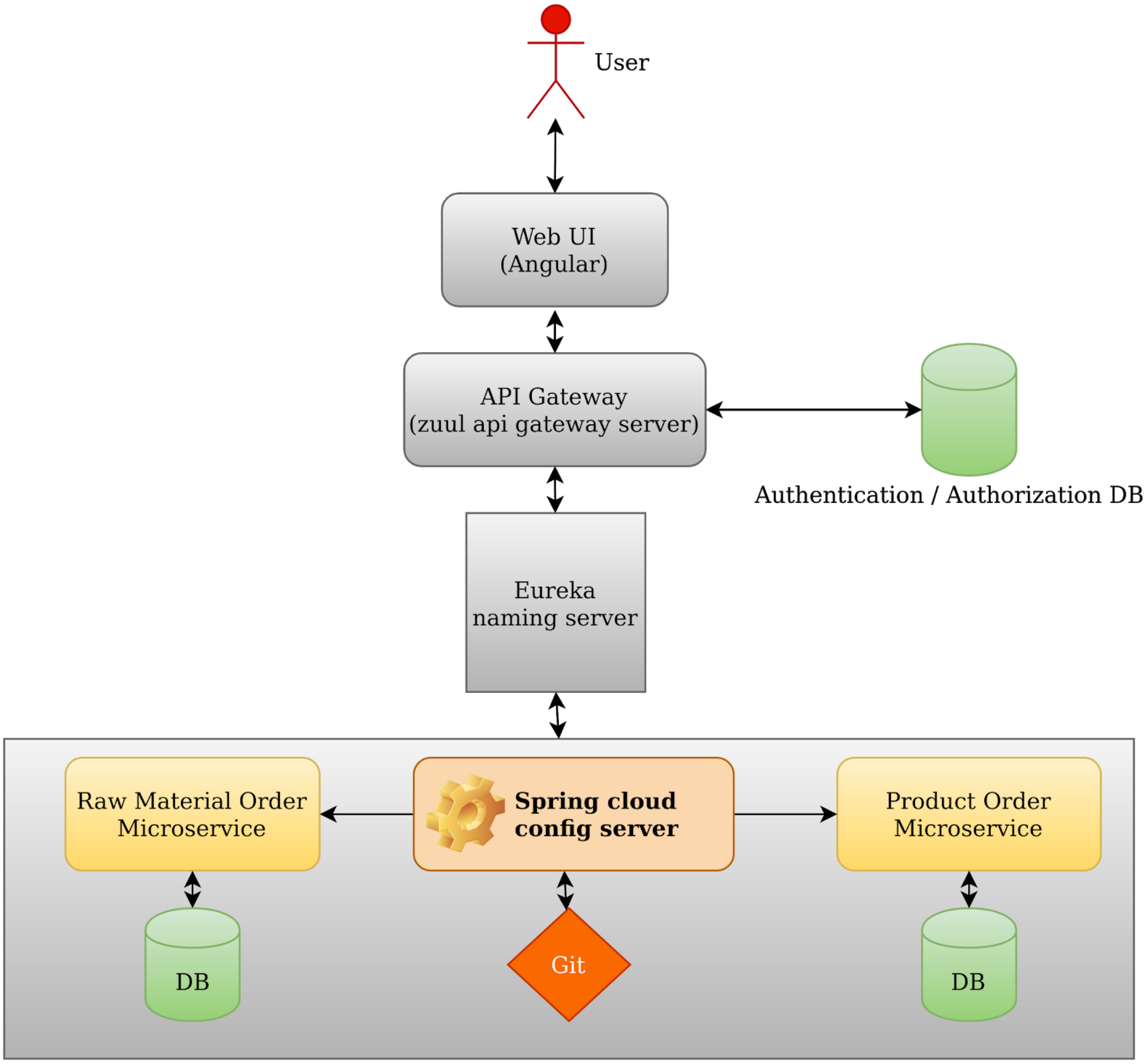
Three microservices 4 are created to facilitate the operation

a) spring configuration server
b) Eureka naming server
c) zuul api gateway server
d) zipkin distributed tracing server



Project Flow




User
Web UI
(Angular)
API Gateway
(zuul api gateway server)
Authentication / Authorization DB
Eureka
naming server
Raw Material Order
Microservice
Spring cloud
config server
Product Order
Microservice
DB
Git
DB

4. Epic& Stories

	s	n	to	t
hentication System			my correct ntials and login	successfully login to the system
			my incorrect credentials and login	not be able to successfully login to the system
	Forgot Password		e my password if I have forgotten my password	uccessfully change my password to a new one
	logout		out of my system	be successfully be able to logout of the system
Raw Material Order			Order Id v material	ll the details about that particular order Id
e Raw Material Stock	ocess Date		RM Order Id and process date for that Id	ss date is inserted in database and data inserted message is shown
	e Manufacturing Date, Expiry Date & QA		RM order Id and set Manufacturing Date, Expiry Date & QA for that Id	tails are inserted in database and data inserted message is shown

Product Order			Order Id duct	ll the details about that particular order Id
Update Product Stock			Order Id duct and exit date for that Id	ate is inserted in database and data inserted message is shown
	e Manufacturing Date, Expiry Date & QA		product Id and nufacturing Date, Expiry Date & QA for that Id	ails are inserted in database and data inserted message is shown
Up date an	Update Delivery Status of Raw Material Order		Raw ial Order Id and new Delivery Status	lelivery status of that Order is set and the same is reflected in the database.
	Update Delivery Status of Product Order		Product Id and new Delivery Status	lelivery status of that Order is set and the same is reflected in the database.
an order	Raw Material Order		all the details for a Raw Material Order	Material Order is placed successfully.
	Product Order		all the details for a Product Order	ct Order is placed successfully.
y order details	y Raw Material Order Details		delivery status, supplier ID, start and end date(optional).	ll the details of all the raw material orders placed.

	y Product Order Details		delivery status, distributor ID, start and end date(optional).	ll the details of all the product orders placed.
y Supplier Details	y Details of Supplier		Supplier	ll the details of the specified Suppliers.
y ɔutor Details	y Details of Distributor		Distributor	ll the details of the specified Distributors.

5.Use Cases

5.1 Authentication System

Overview

A login is a set of credentials used to authenticate a user. Most often, these consist of a username and password. However, a login may include other information, such as a PIN number, passcode, or passphrase. Some logins require a biometric identifier, such as a fingerprint or retina scan.

Logging in is usually used to enter a specific page, website or application, which trespassers cannot see. Once the user is logged in, the login token may be used to track what actions the user has taken while connected to the site. Logging out may be performed explicitly by the user taking some actions, such as entering the appropriate command, or clicking a website link labelled as such. It can also be done implicitly, such as by the user powering off his or her workstation, closing a web browser window, leaving a website, or not refreshing a webpage within a defined period.

Prerequisite

User must login as USER to perform the required functionalities for Raw material and Products for our client Drink and Delight.

Non-Functional Requirement:

- Performance Requirements:

User can login in < 10 seconds

All ad hoc reports should be published in < 5 seconds

Customer Order screen should be able complete submission of customer order within 3 sec including all validations on shipping address.

Customer Address List screen should not take more than 2 sec

- Operations and Reliability:

Up time requirement

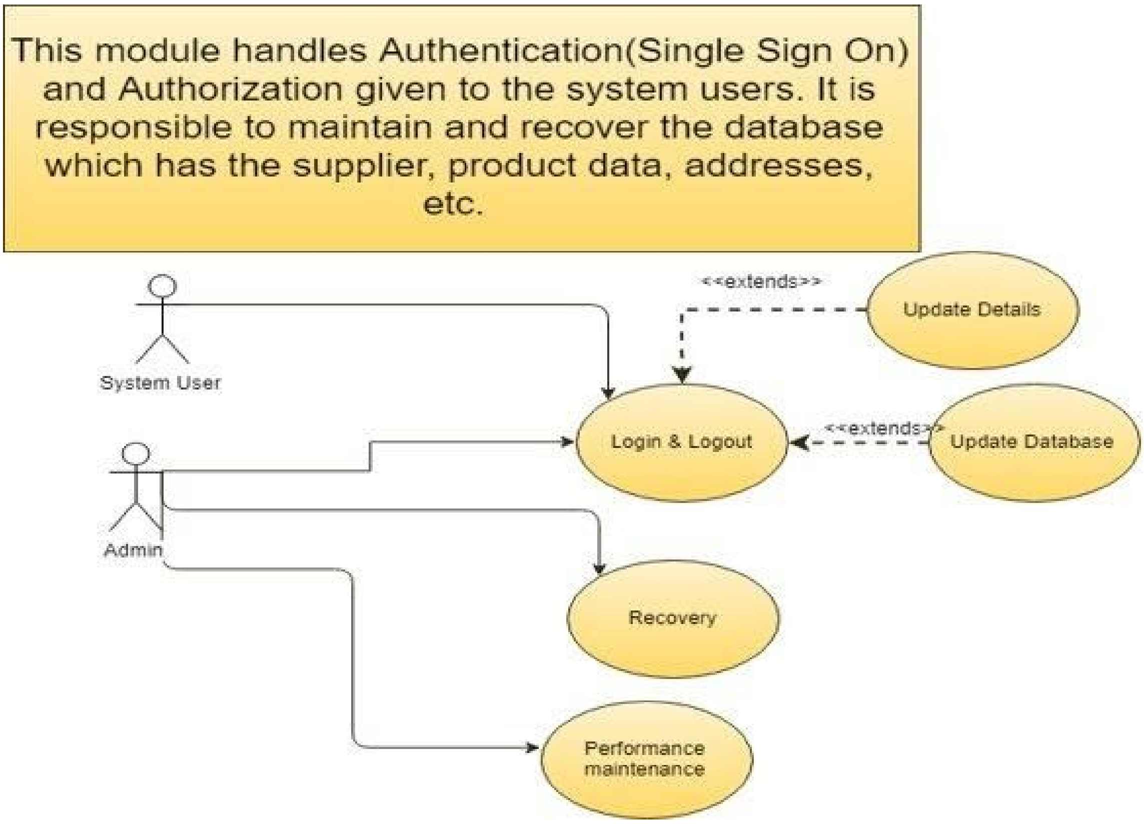
Acceptable data loss

System update

Designed and Implemented By:

Mohit Kumar
Designation: Analyst

Use Case Diagram for Authentication

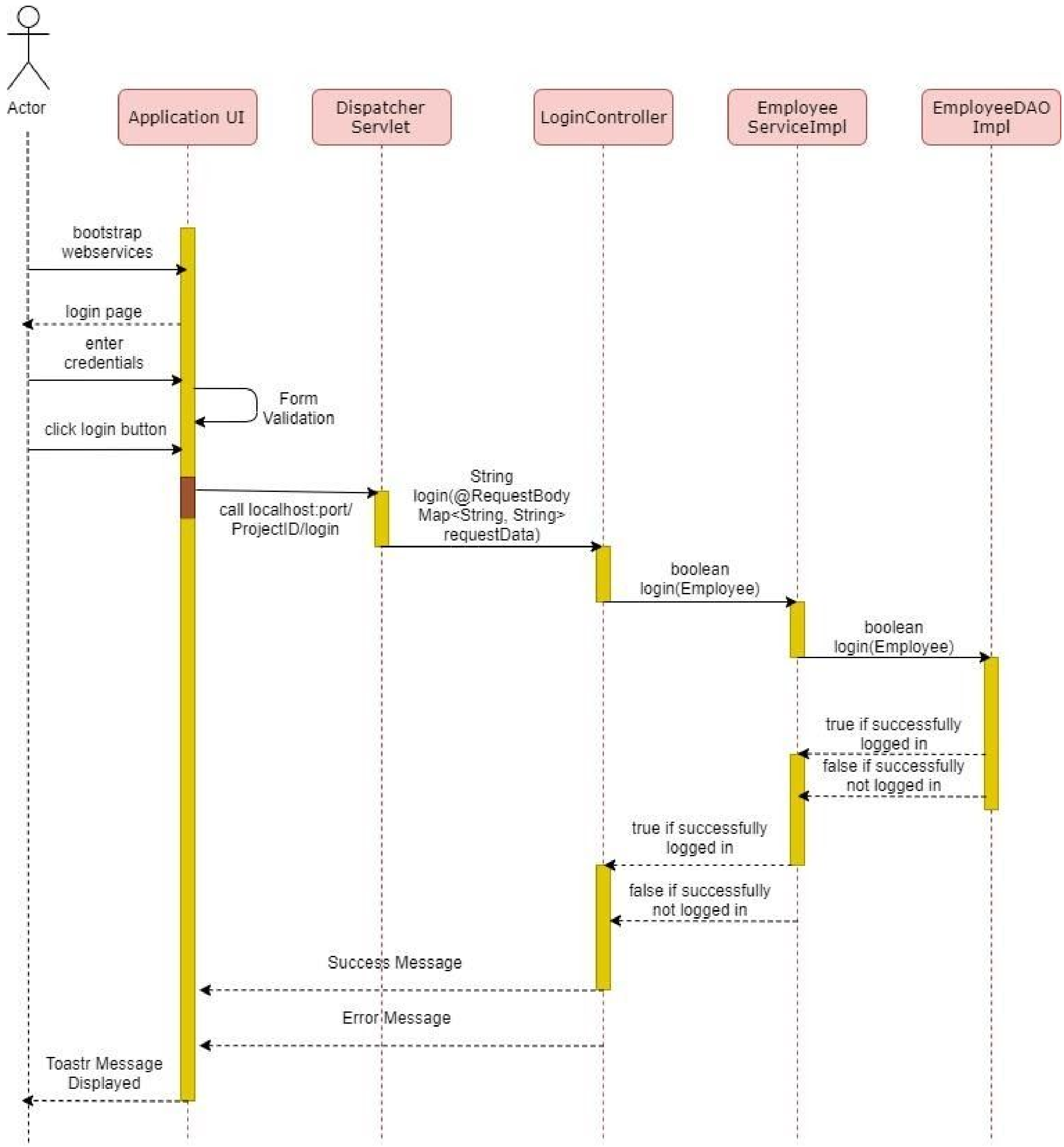


This module handles Authentication(Single Sign On) and Authorization given to the system users. It is responsible to maintain and recover the database which has the supplier, product data, addresses, etc.

a) Login

This module has been designed to login into the database. This can be only accessed by employee with correct credentials.

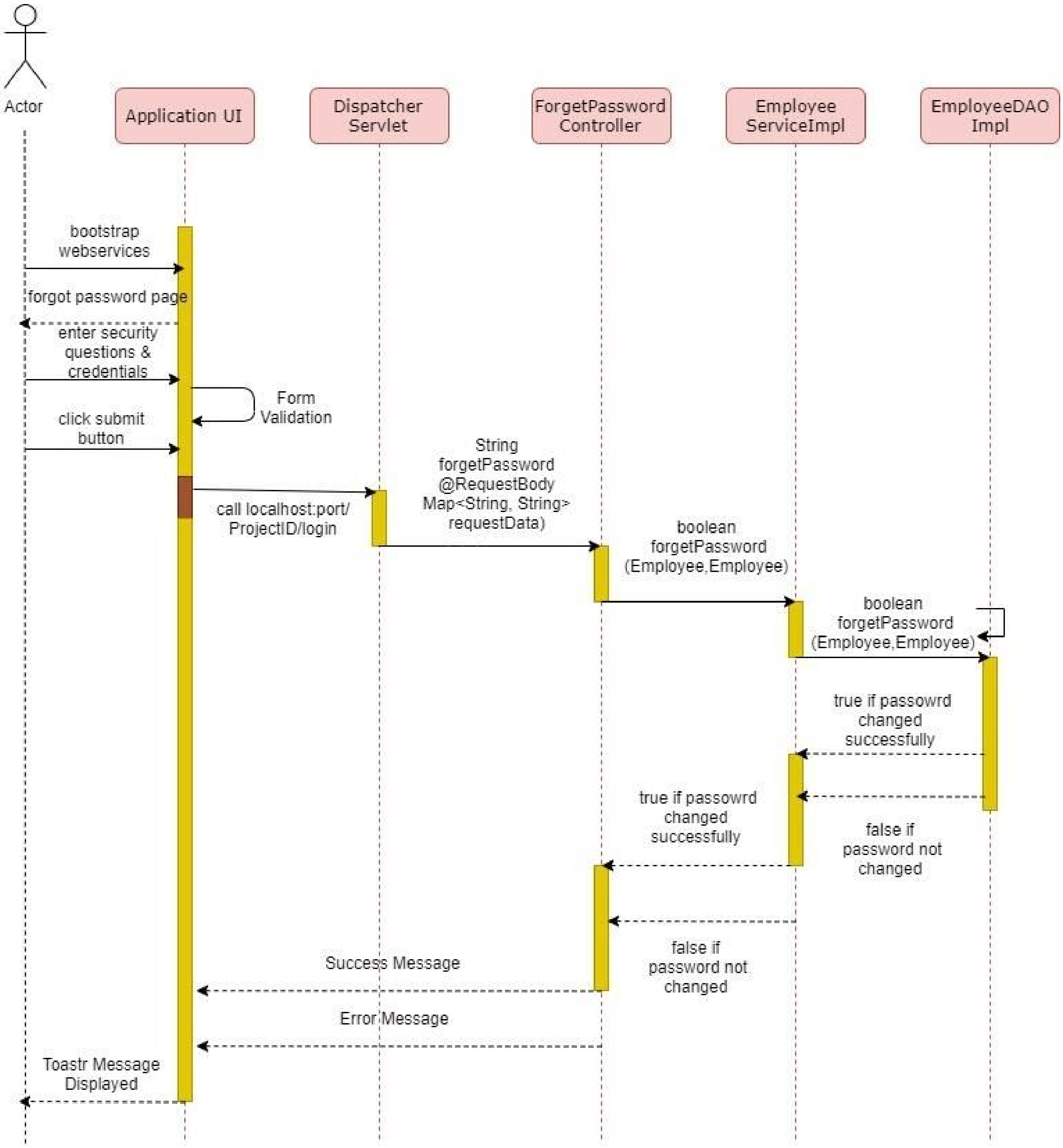
Sequence Diagram for Login



b) ForgetPassword

If the user logins into the system with incorrect credentials then he/she needs to change password the we have a functionality which you can see in the code where we can change the password manually, by giving some security questions and answers that will be specific for each user in order to change their password.

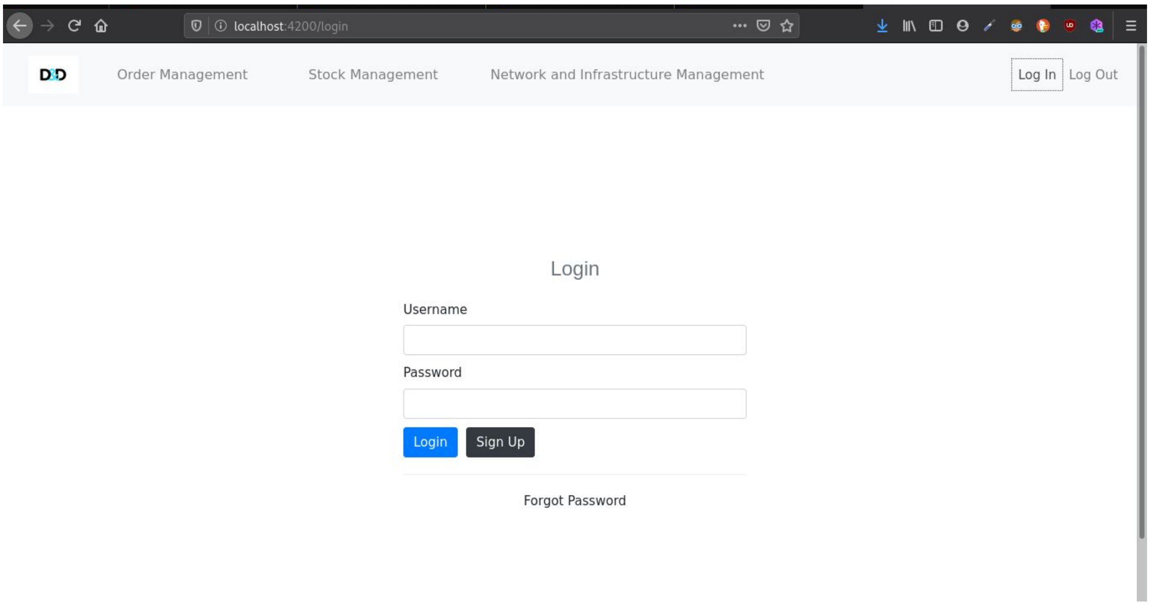
Sequence Diagram for changing the password

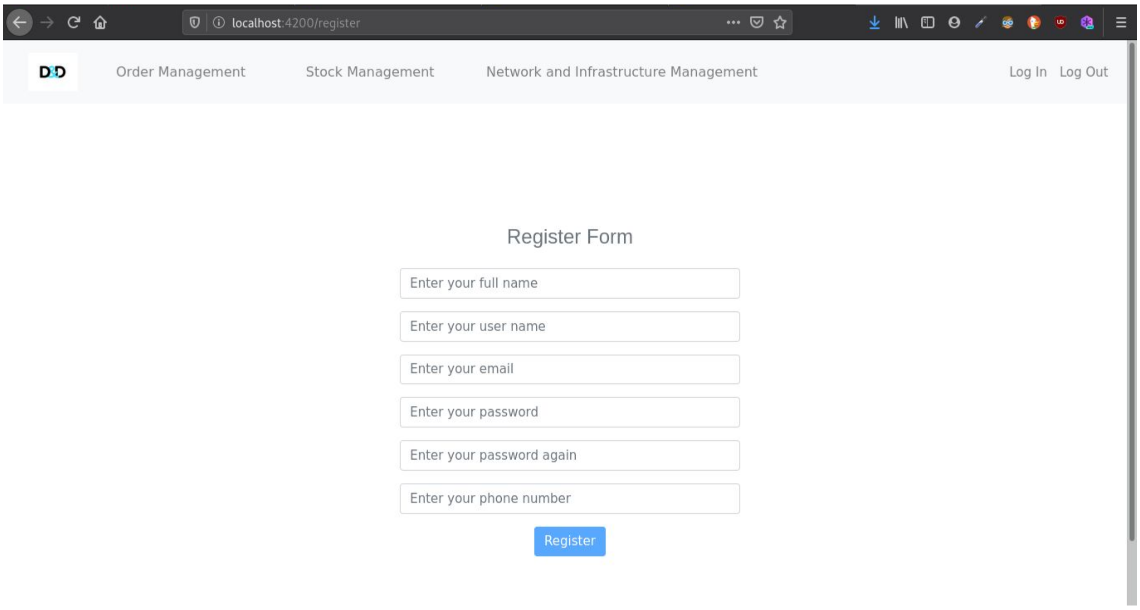

Actor
Application UI
Dispatcher Servlet
ForgetPassword Controller
Employee ServiceImpl
EmployeeDAO Impl
bootstrap webservices
forgot password page
enter security questions & credentials
Form Validation
click submit button
call localhost:port/ ProjectID/login
String forgetPassword @RequestBody Map<String, String> requestData)
boolean forgetPassword (Employee,Employee)
boolean forgetPassword (Employee,Employee)
true if passowrd changed successfully
false if password not changed
true if passowrd changed successfully
false if password not changed
Success Message
Error Message
Toastr Message Displayed

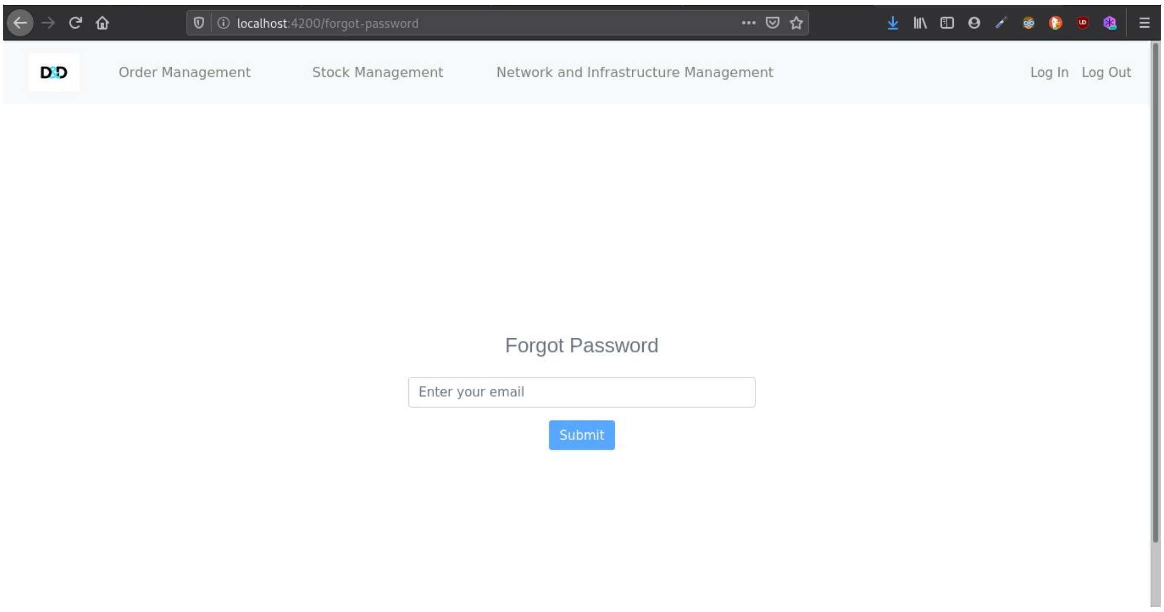
c) Logout

User here can simply logout from the UI as you can see the frontend code part where by using angular we can directly logout.

So, no sequence diagram is required for it.


localhost:4200/login
D2D
Order Management
Stock Management
Network and Infrastructure Management
Log In
Log Out
Login
Username
Password
Login
Sign Up
Forgot Password


localhost:4200/register
D2D
Order Management
Stock Management
Network and Infrastructure Management
Log In
Log Out
Register Form
Enter your full name
Enter your user name
Enter your email
Enter your password
Enter your password again
Enter your phone number
Register


localhost:4200/forgot-password
D3D
Order Management
Stock Management
Network and Infrastructure Management
Log In
Log Out
Forgot Password
Enter your email
Submit

5.2 Raw Material/ Product Order Management

Overview

The life cycle of a Raw material/ Product Order is handled in this case. This includes placing an order, updating the order (if necessary) and displaying the order. The management of supplier and distributor details is also considered.

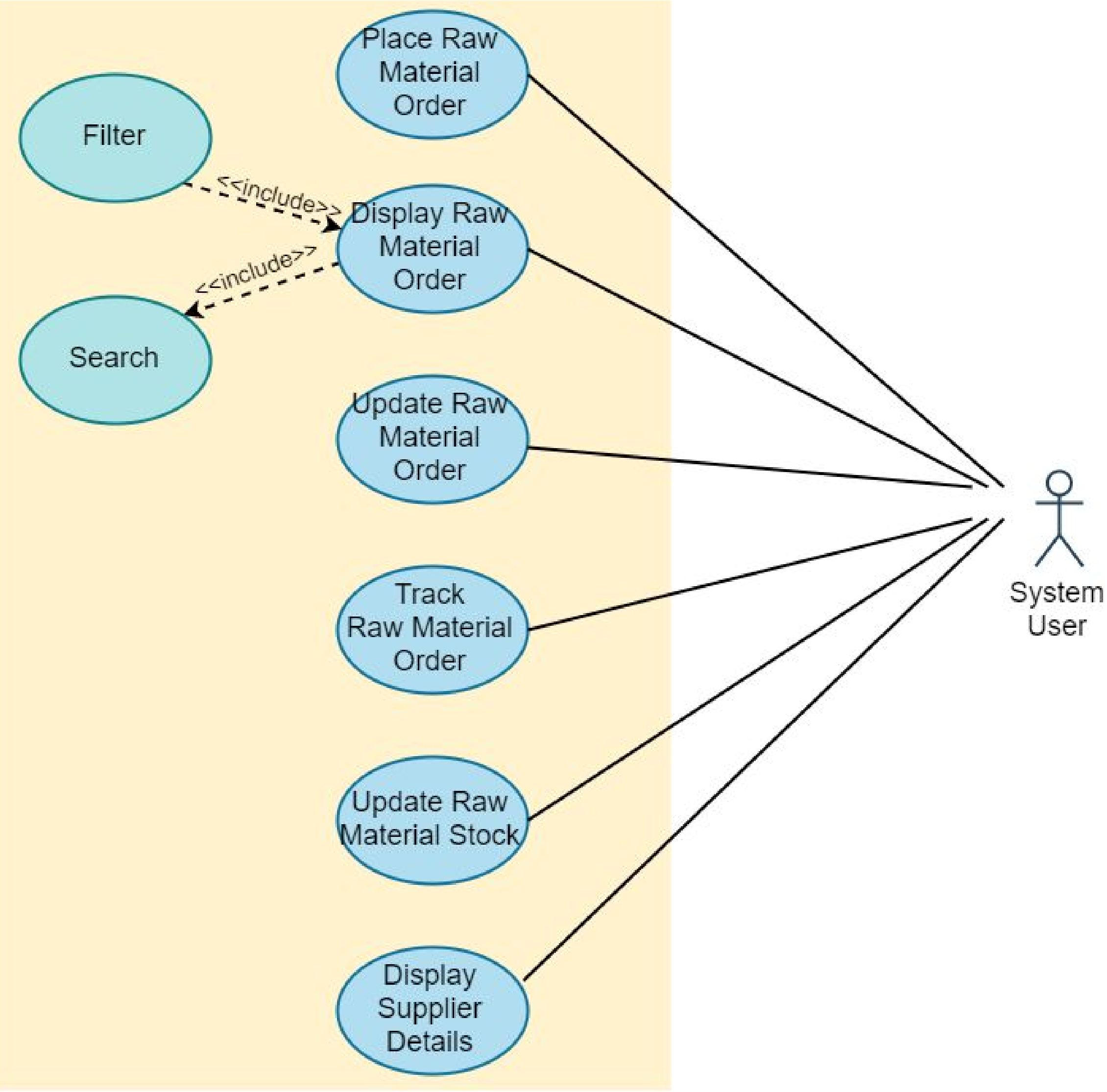
Prerequisite

User must as USER should be ableto perform the required functionalities of Order Management for Raw material and Products for our client Drink and Delight.

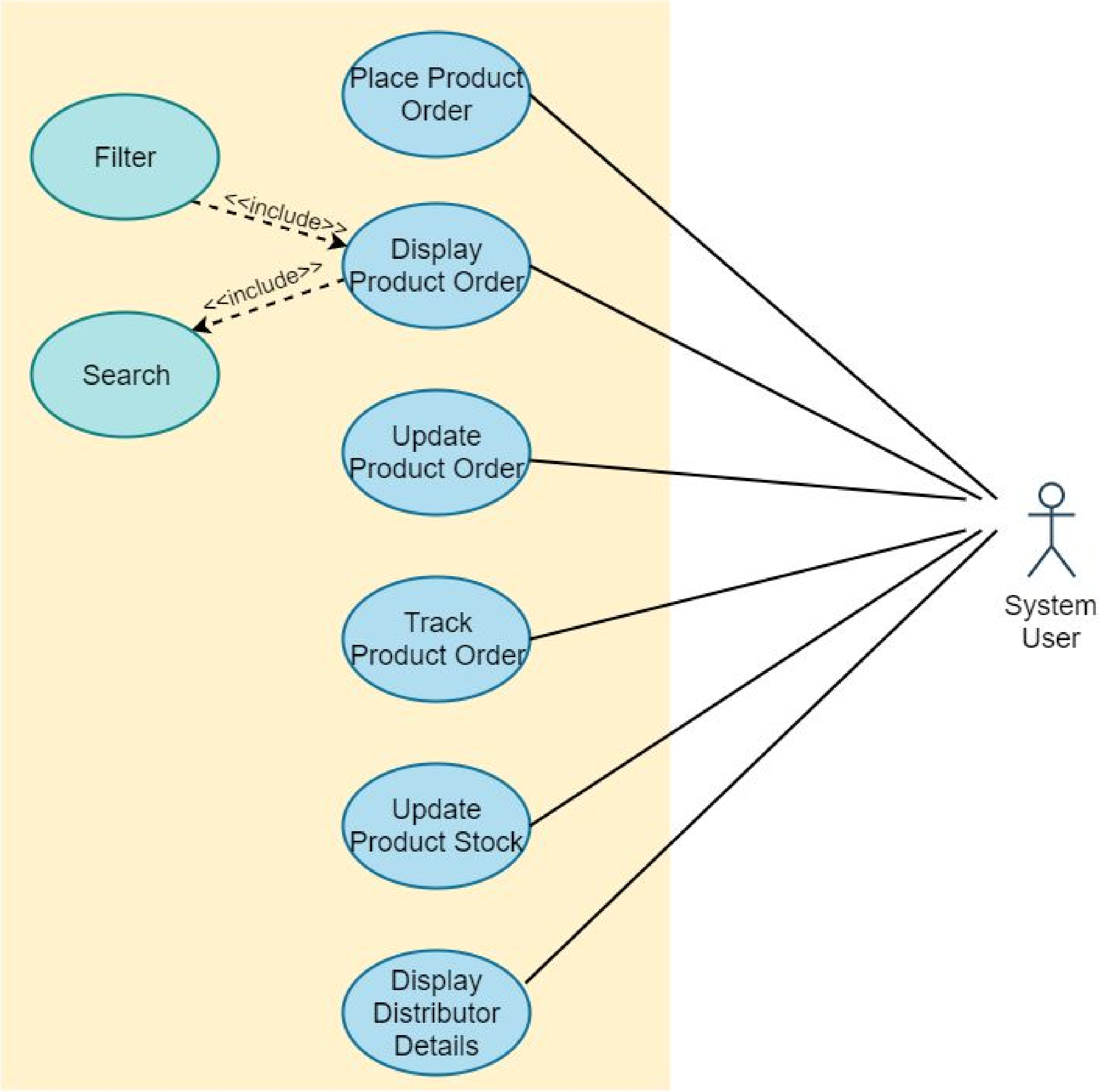
Designed and Implemented By:

Atal Kumar
Ayushi Goyal
Harshit Verma
Mukul Garg
Designation: Analyst

Use Case Diagram for Raw Material Orders/Stocks



Use Case Diagram for Product Orders/Stocks



a) Place an Order

When the user clicks on 'Place an order' tab under 'Raw Material' dropdown, the user sees a form to add following Raw Material order details:

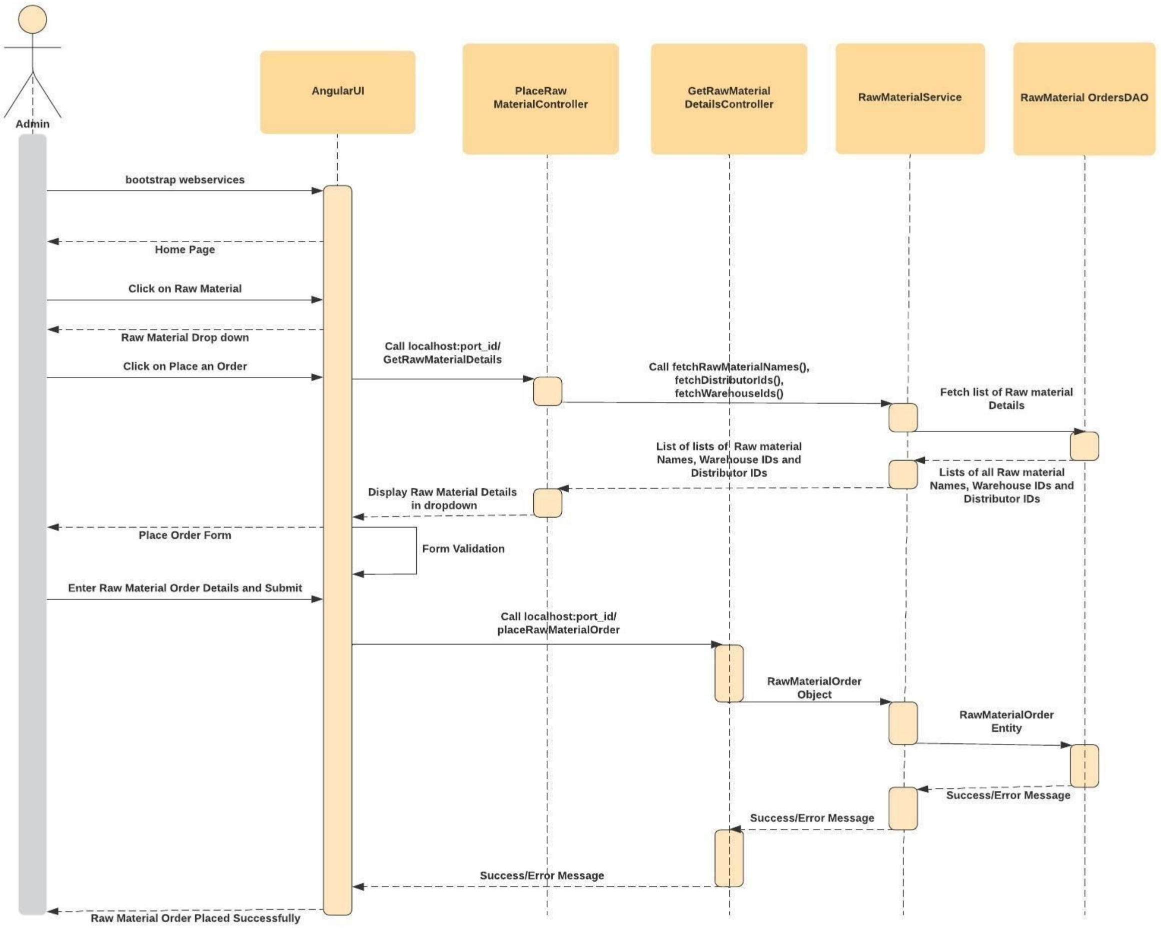
- Raw Material Name
- Supplier ID
- Warehouse ID
- Quantity value and unit
- Price per Unit

- Expected Date of delivery

Now there is a Submit button that the user needs to click which would allow the send the request to the Rest Controller at the backend.

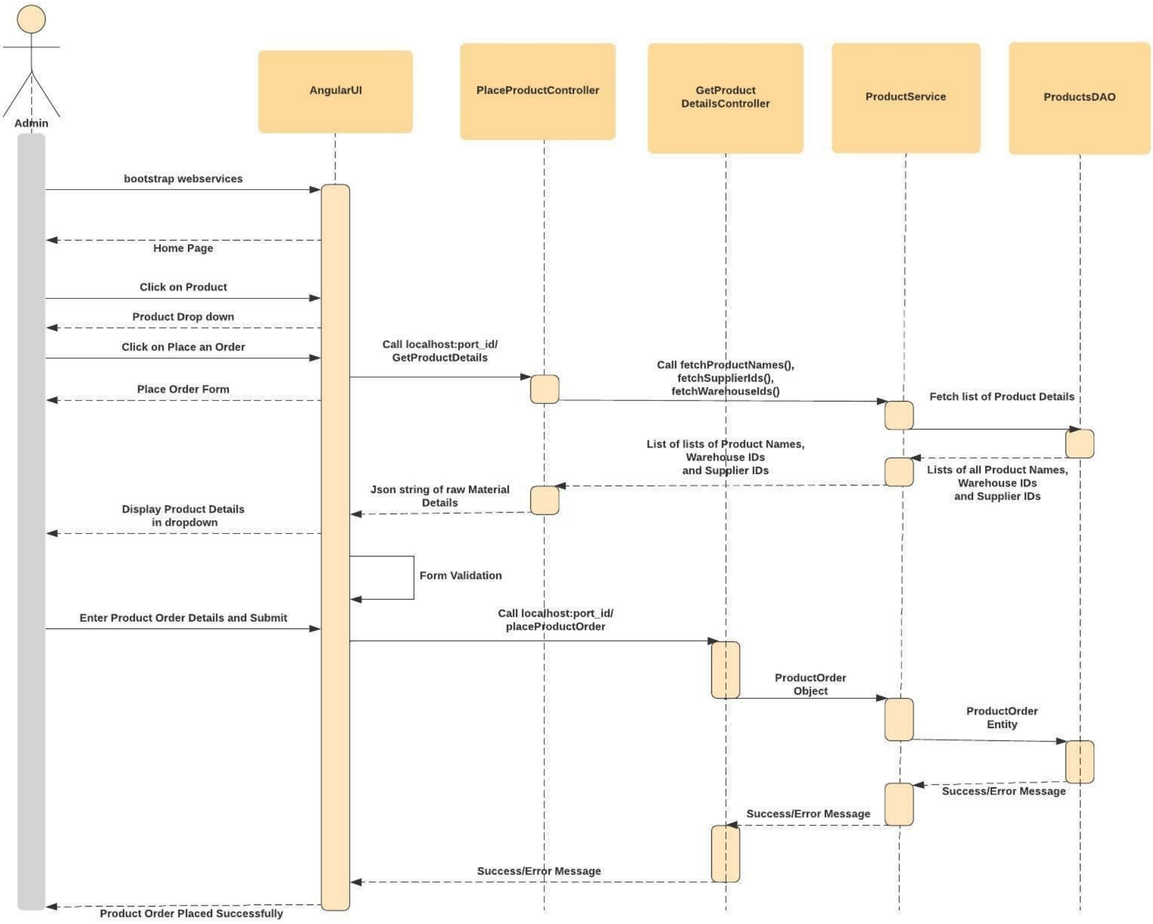
Here, the values in the dropdowns of Raw Material Name, Supplier ID, Warehouse ID are loaded from the backend when the page is loaded. Also, date is restricted in such a way that the user can enter a date before 2 months from the current date. Thus, the user cannot enter a past date or a date after 2 months from the current date. The Quantity value and Price per Unit values need to be number and the field cannot be empty. This has been handled in the Angular Component.

Sequence Diagram for Place Raw Material Order



Admin
AngularUI
PlaceRaw MaterialController
GetRawMaterial DetailsController
RawMaterialService
RawMaterial OrdersDAO
bootstrap webservices
Home Page
Click on Raw Material
Raw Material Drop down
Click on Place an Order
Call localhost:port_id/ GetRawMaterialDetails
Call fetchRawMaterialNames(), fetchDistributorIds(), fetchWarehouseIds()
Fetch list of Raw material Details
Lists of all Raw material Names, Warehouse IDs and Distributor IDs
List of lists of Raw material Names, Warehouse IDs and Distributor IDs
Display Raw Material Details in dropdown
Place Order Form
Form Validation
Enter Raw Material Order Details and Submit
Call localhost:port_id/ placeRawMaterialOrder
RawMaterialOrder Object
RawMaterialOrder Entity
Success/Error Message
Success/Error Message
Success/Error Message
Raw Material Order Placed Successfully

Sequence Diagram for Place Product Order


Admin
AngularUI
PlaceProductController
GetProduct DetailsController
ProductService
ProductsDAO
bootstrap webservices
Home Page
Click on Product
Product Drop down
Click on Place an Order
Place Order Form
Call localhost:port_id/ GetProductDetails
Call fetchProductNames(), fetchSupplierIds(), fetchWarehouseIds()
Fetch list of Product Details
List of lists of Product Names, Warehouse IDs and Supplier IDs
Lists of all Product Names, Warehouse IDs and Supplier IDs
Json string of raw Material Details
Display Product Details in dropdown
Form Validation
Enter Product Order Details and Submit
Call localhost:port_id/ placeProductOrder
ProductOrder Object
ProductOrder Entity
Success/Error Message
Success/Error Message
Success/Error Message
Product Order Placed Successfully

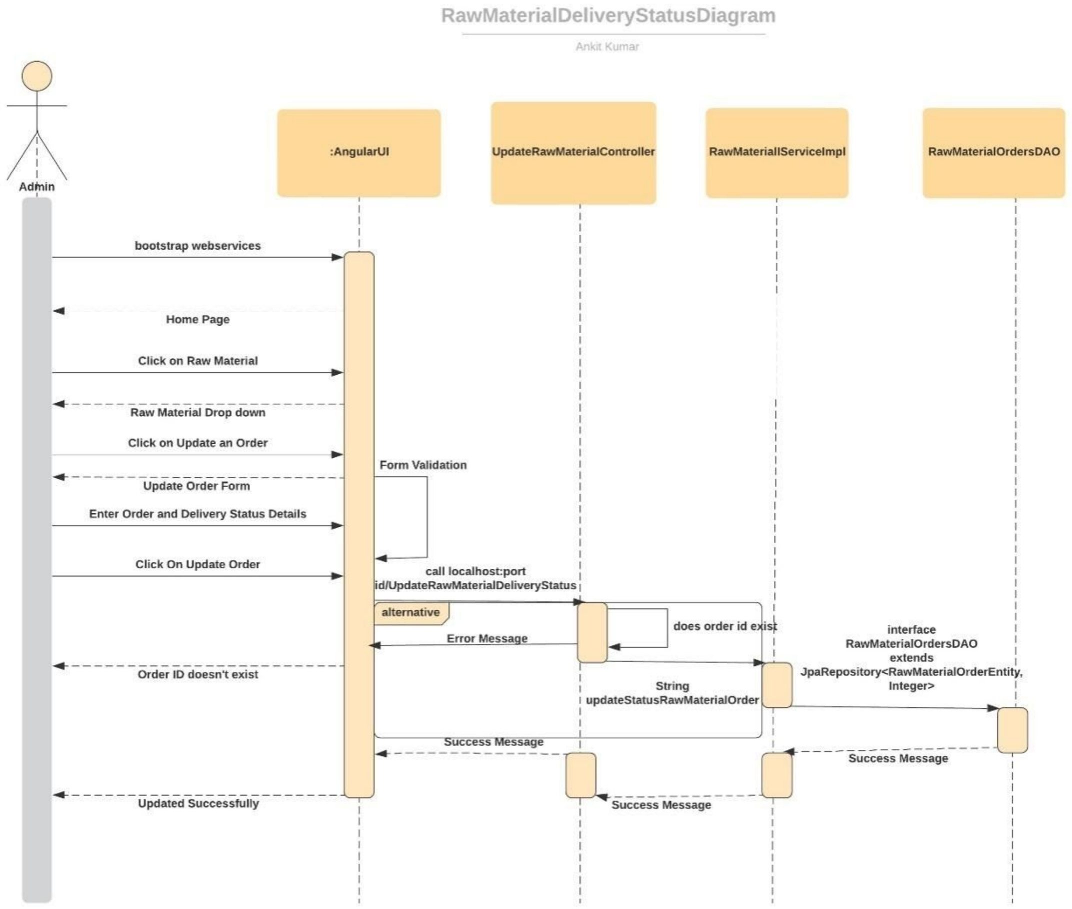
b) Update an Order

When the user clicks on 'Update an order' tab under 'Raw Material' or ‘Product Order dropdown’, the user sees a form to update delivery status of an Order :

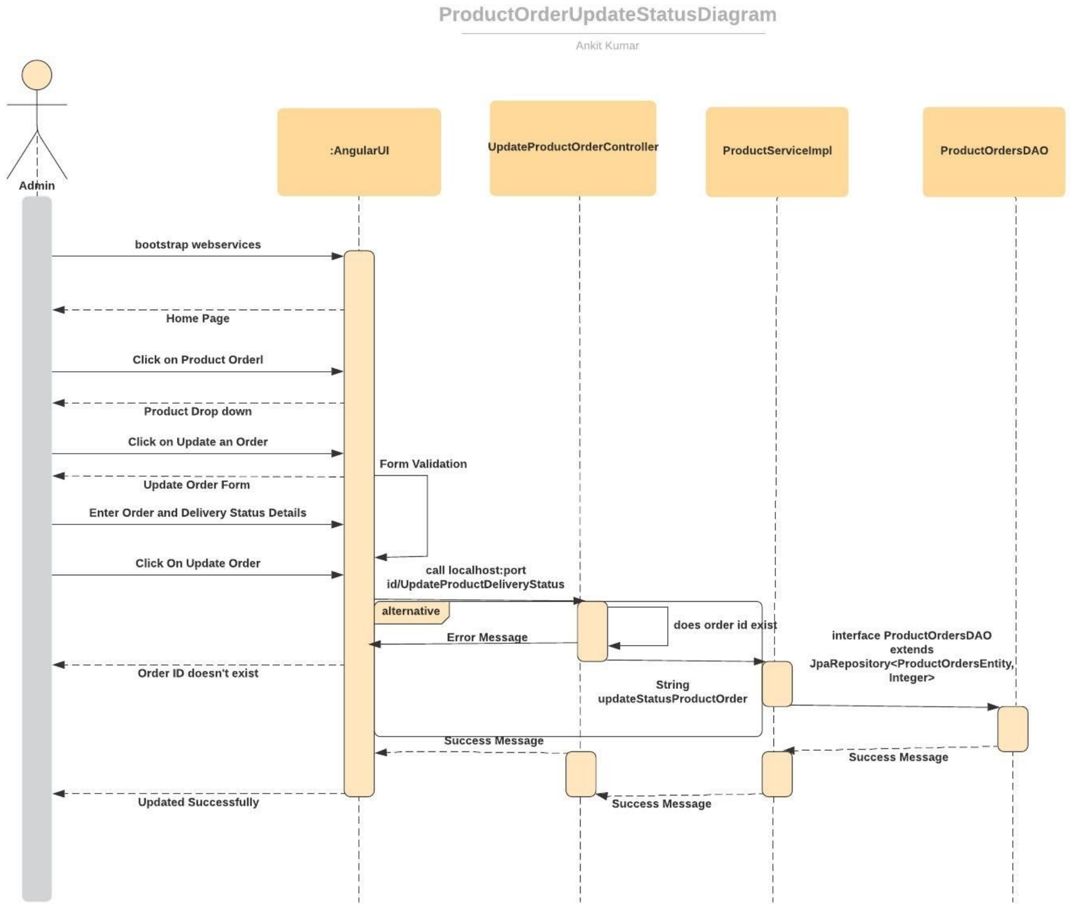
- Raw Material Order Id/ Product Order Id
- Delivery Status

Now there is an Update Order button that the user needs to click which would allow the send the request to the Rest Controller at the backend and change delivery status in database.

Sequence Diagram to Update Delivery Status of Raw Material Order



Sequence Diagram to Update Delivery Status of Product Order

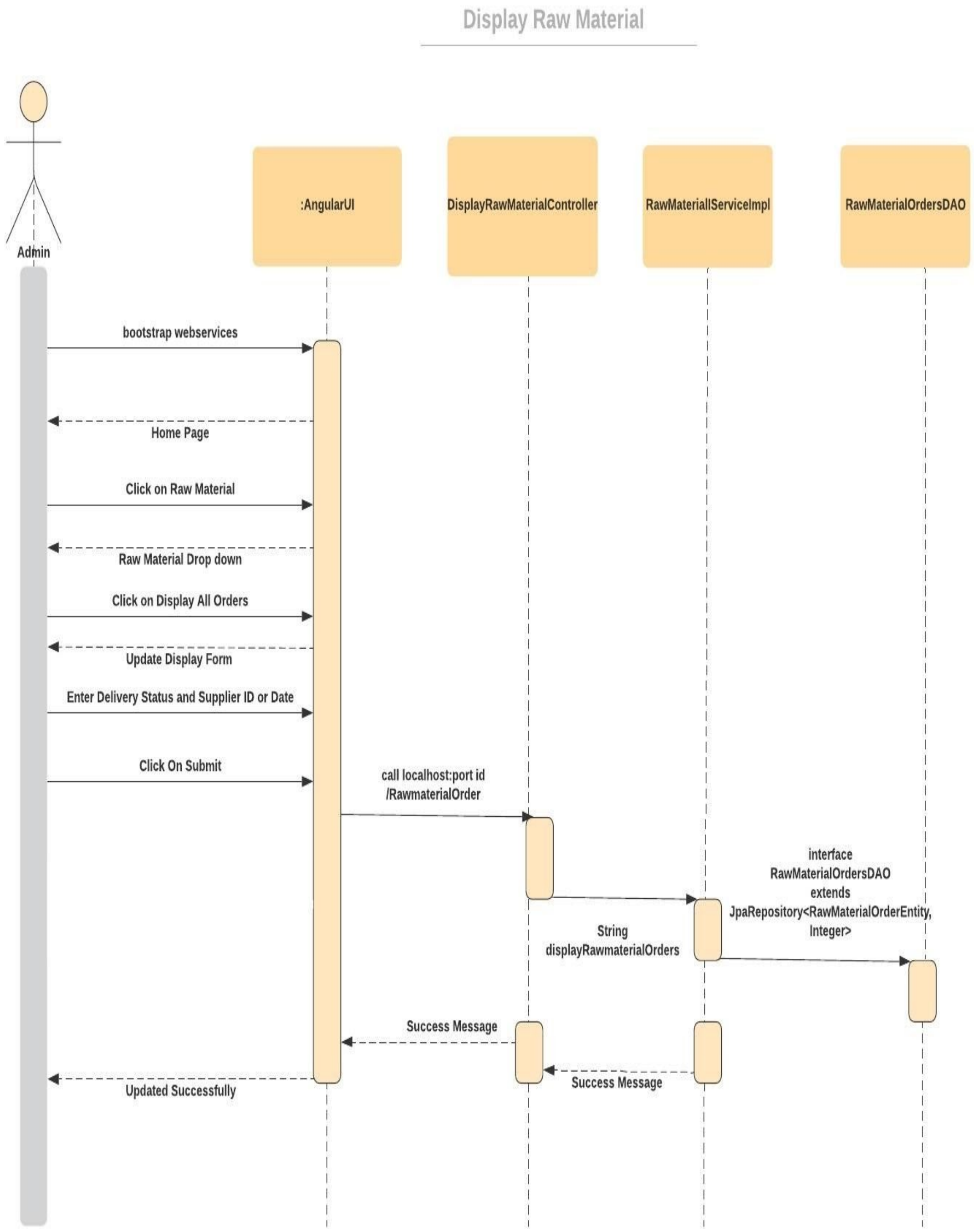


c) Display Order

1.Raw Material

When the admin clicks on 'Display Raw Material order ' tab under 'Raw Material' dropdown, the admin can see all the order he/she has received.

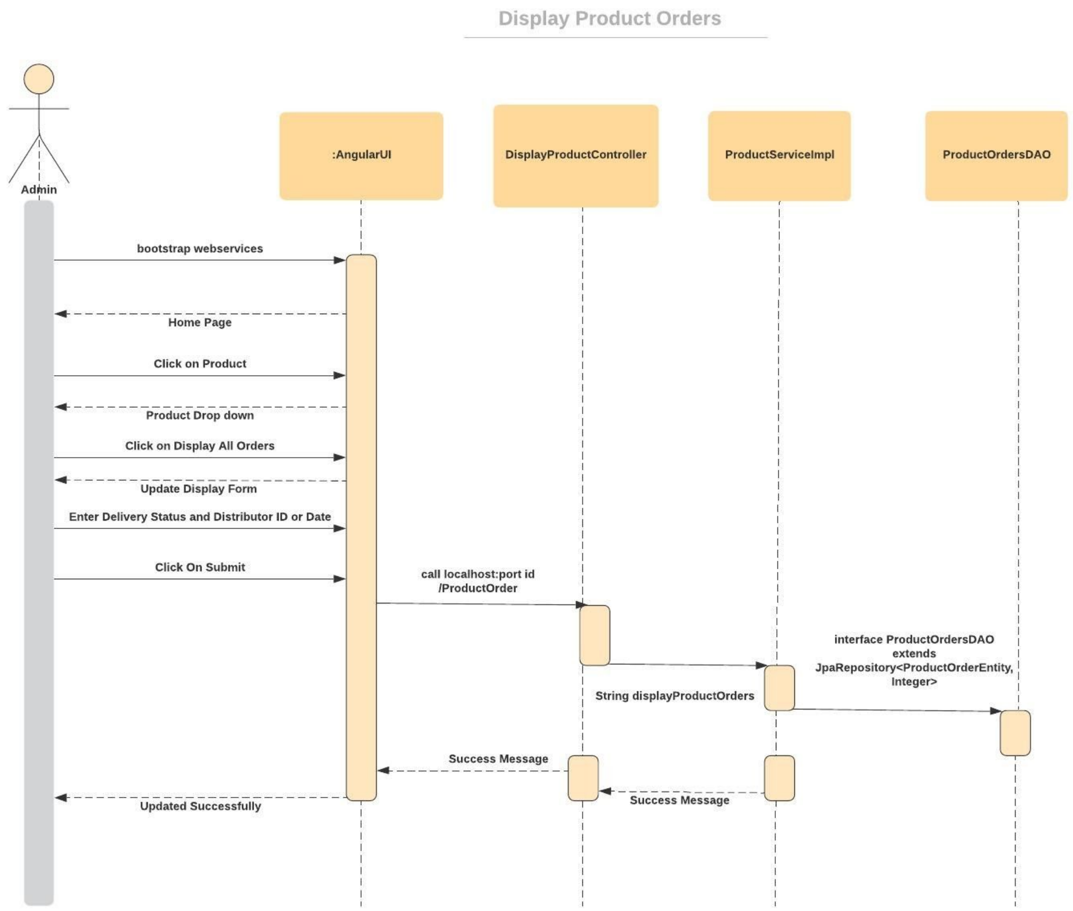
Sequence Diagram for Display Raw Material Orders


Display Raw Material
Admin
:AngularUI
DisplayRawMaterialController
RawMateriallServiceImpl
RawMaterialOrdersDAO
bootstrap webservices
Home Page
Click on Raw Material
Raw Material Drop down
Click on Display All Orders
Update Display Form
Enter Delivery Status and Supplier ID or Date
Click On Submit
call localhost:port id /RawmaterialOrder
String displayRawmaterialOrders
interface RawMaterialOrdersDAO extends JpaRepository<RawMaterialOrderEntity, Integer>
Success Message
Success Message
Updated Successfully

2.Product

When the user clicks on 'Display Product ' tab under ' Product ' dropdown, the user see all the product orders he/she has received.

Sequence Diagram for Display Product Orders

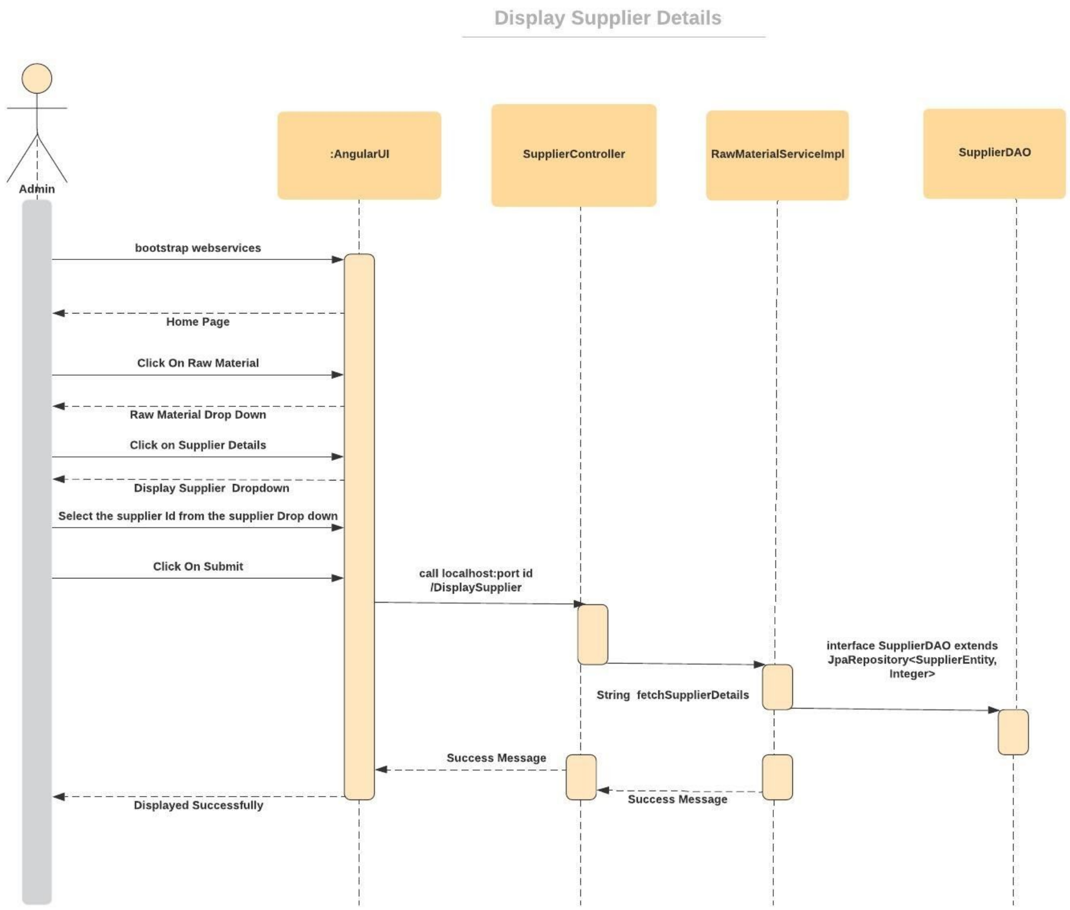


d) Display Supplier Details

When the user clicks on 'Display Supplier 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 Diagram for Display Supplier Details



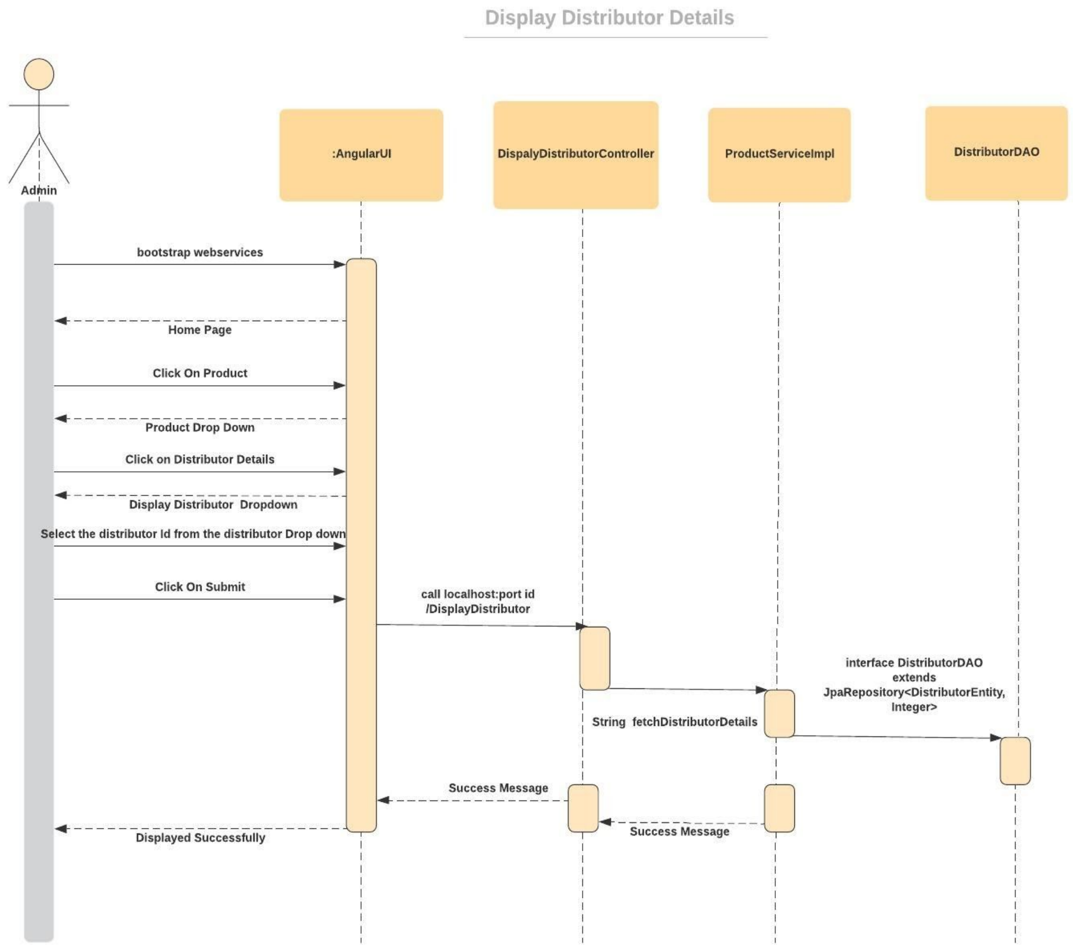
Display Supplier Details
Admin
:AngularUI
SupplierController
RawMaterialServiceImpl
SupplierDAO
bootstrap webservices
Home Page
Click On Raw Material
Raw Material Drop Down
Click on Supplier Details
Display Supplier Dropdown
Select the supplier Id from the supplier Drop down
Click On Submit
call localhost:port id /DisplaySupplier
String fetchSupplierDetails
interface SupplierDAO extends JpaRepository<SupplierEntity, Integer>
Success Message
Success Message
Displayed Successfully

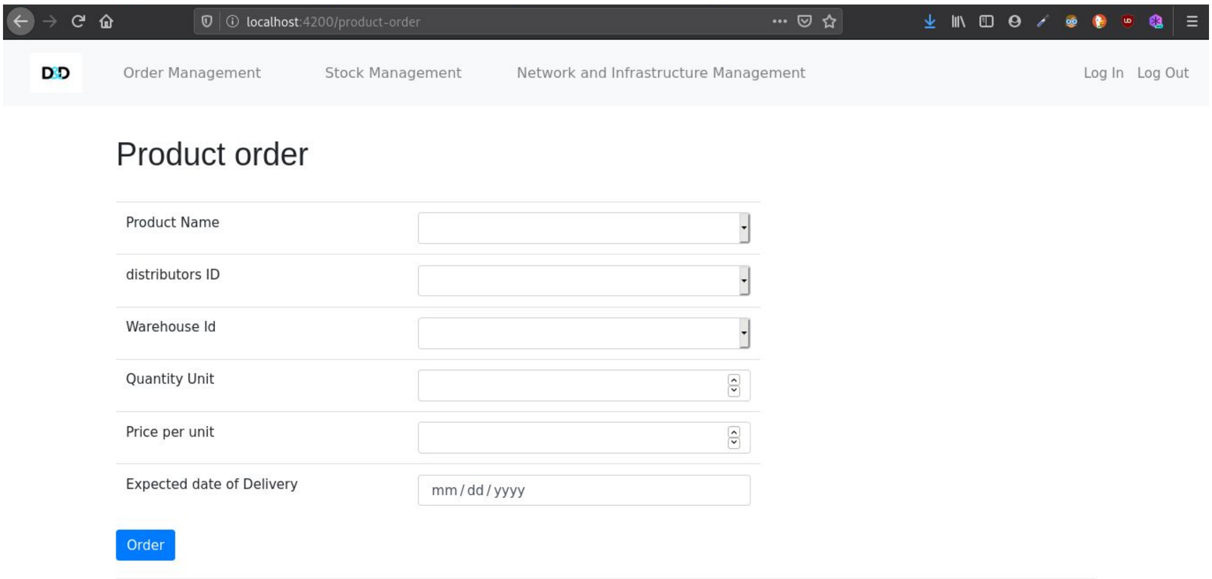
e) Display Distributor Details

When the user clicks on 'Display Distributor which would send the request to the Rest Controller at the backend.

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

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
String fetchDistributorDetails
interface DistributorDAO extends JpaRepository<DistributorEntity, Integer>
Success Message
Success Message
Displayed Successfully


localhost:4200/product-order
D3D
Order Management
Stock Management
Network and Infrastructure Management
Log In
Log Out

Product order

Product Name

distributors ID

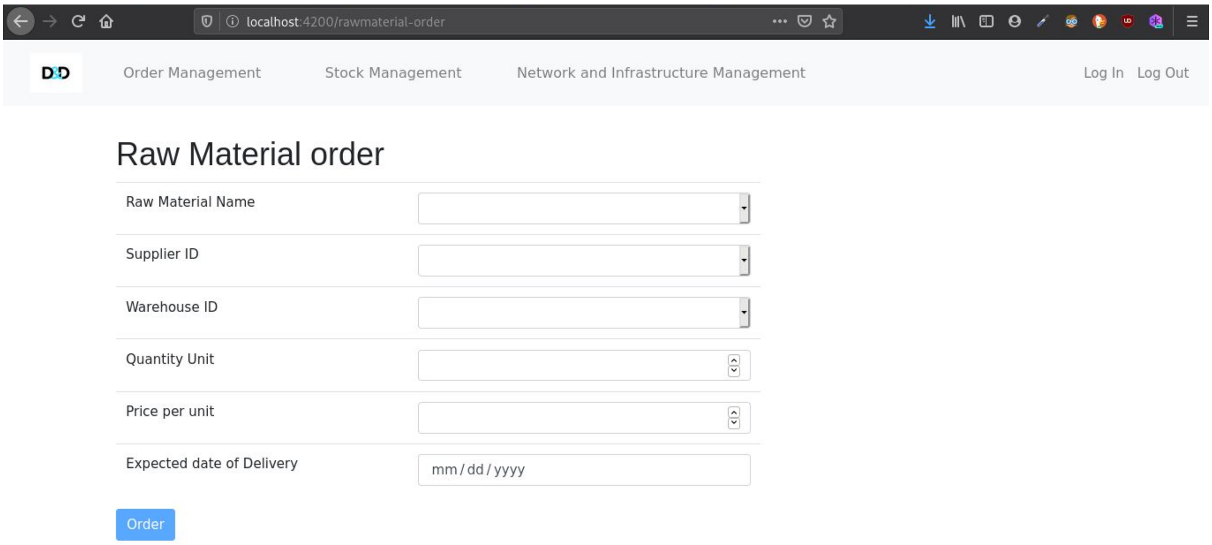
Warehouse Id

Quantity Unit

Price per unit

Expected date of Delivery

mm / dd / yyyy


localhost:4200/rawmaterial-order
D3D
Order Management
Stock Management
Network and Infrastructure Management
Log In
Log Out

Raw Material order

Raw Material Name

Supplier ID

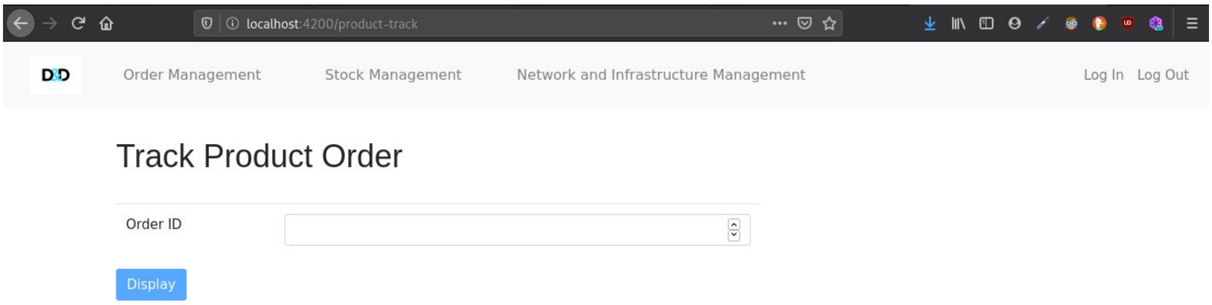
Warehouse ID

Quantity Unit

Price per unit

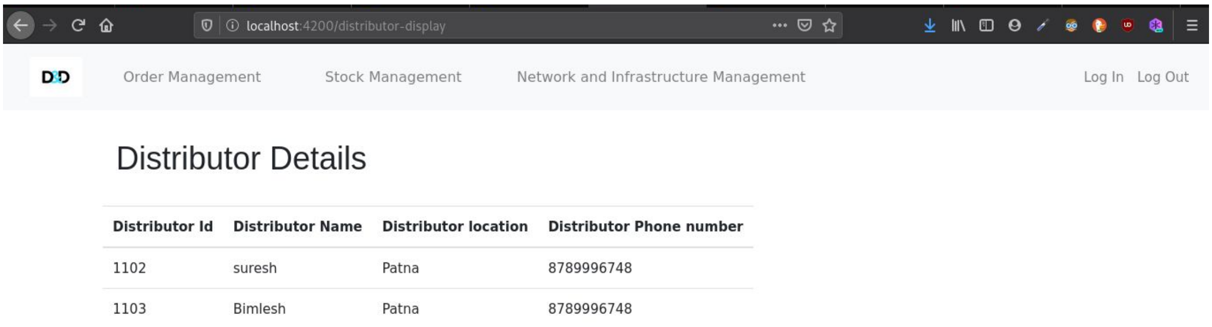
Expected date of Delivery

mm / dd / yyyy


localhost:4200/product-track
D&D
Order Management
Stock Management
Network and Infrastructure Management
Log In
Log Out

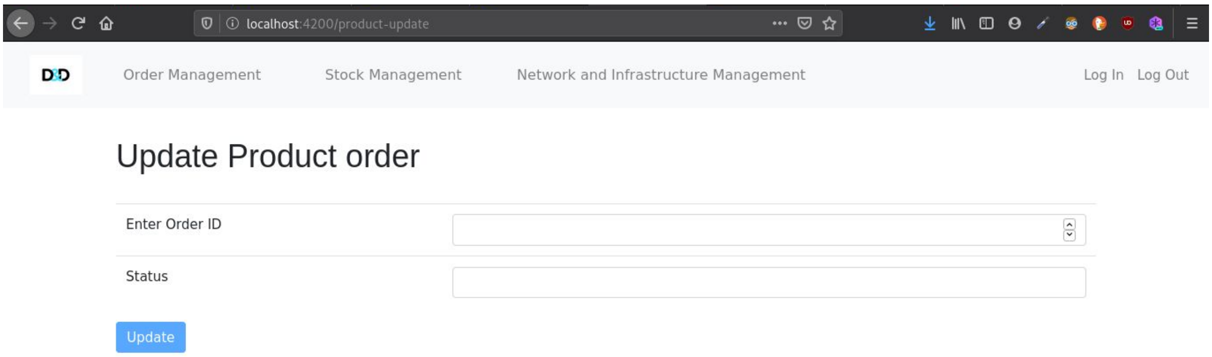
Track Product Order

Order ID


localhost:4200/distributor-display
D&D
Order Management
Stock Management
Network and Infrastructure Management
Log In
Log Out

Distributor Details

Distributor Id	Distributor Name	Distributor location	Distributor Phone number
1102	suresh	Patna	8789996748
1103	Bimlesh	Patna	8789996748

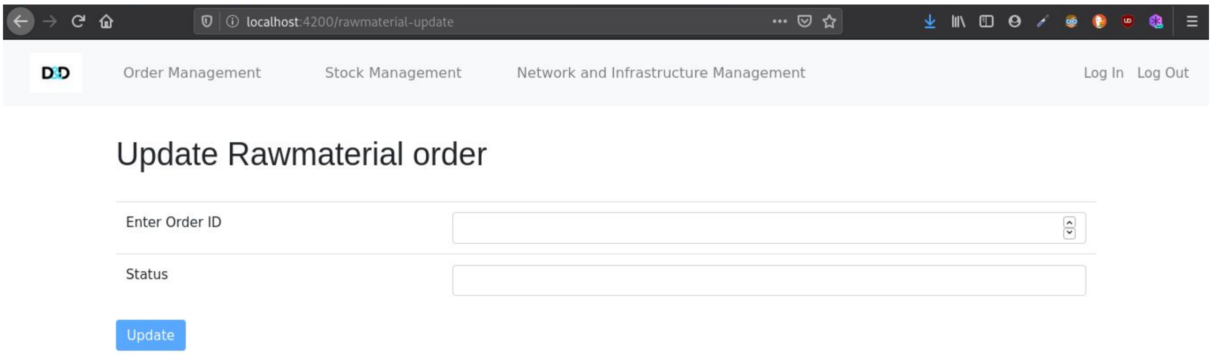


localhost:4200/product-update

Update Product order

Enter Order ID

Status

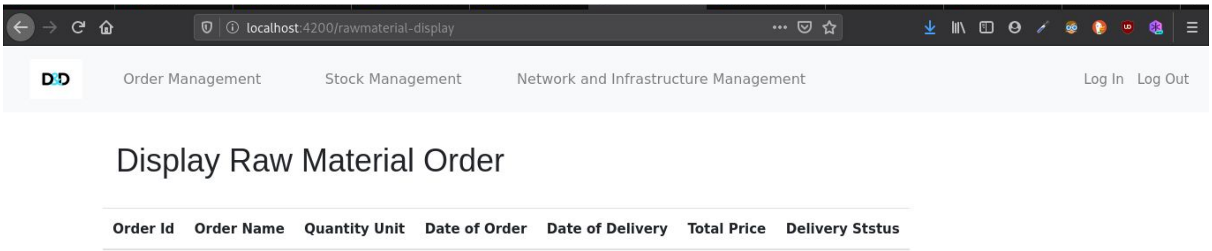


localhost:4200/rawmaterial-update

Update Rawmaterial order

Enter Order ID

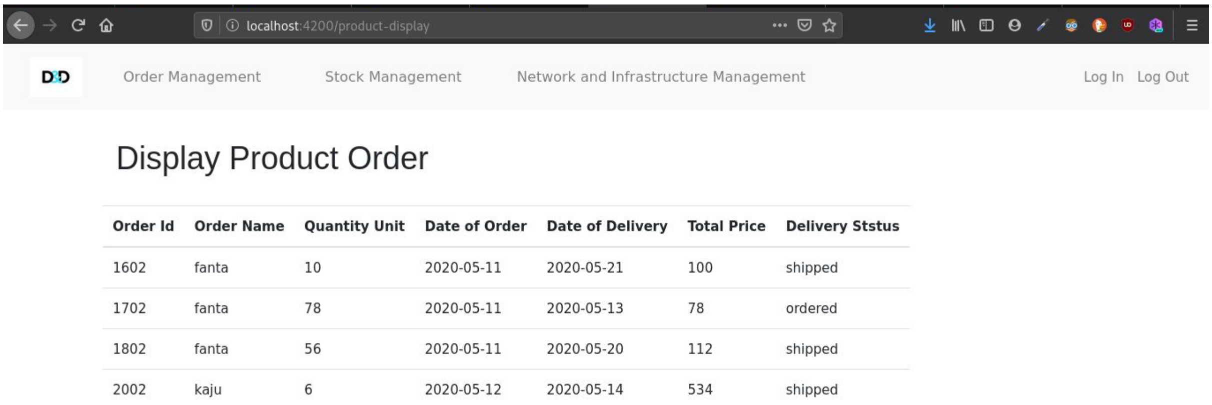
Status



localhost:4200/rawmaterial-display
D&D
Order Management
Stock Management
Network and Infrastructure Management
Log In
Log Out

Display Raw Material Order

Order Id	Order Name	Quantity Unit	Date of Order	Date of Delivery	Total Price	Delivery Ststus



localhost:4200/product-display
D&D
Order Management
Stock Management
Network and Infrastructure Management
Log In
Log Out

Display Product Order

Order Id	Order Name	Quantity Unit	Date of Order	Date of Delivery	Total Price	Delivery Ststus
1602	fanta	10	2020-05-11	2020-05-21	100	shipped
1702	fanta	78	2020-05-11	2020-05-13	78	ordered
1802	fanta	56	2020-05-11	2020-05-20	112	shipped
2002	kaju	6	2020-05-12	2020-05-14	534	shipped

5.3 Raw Material/ Product Stock Management

Overview

The life cycle of a Raw material/ Product Stock is handled in this case. Updating of Stock inventory according to the received order of raw material/product is done. Functions include set process date for raw material, set exit date for product, update other stock details (manufacturing date, expiry date and quality status) for both raw material and product. Track time and location of raw material and product is also a function.

Prerequisite

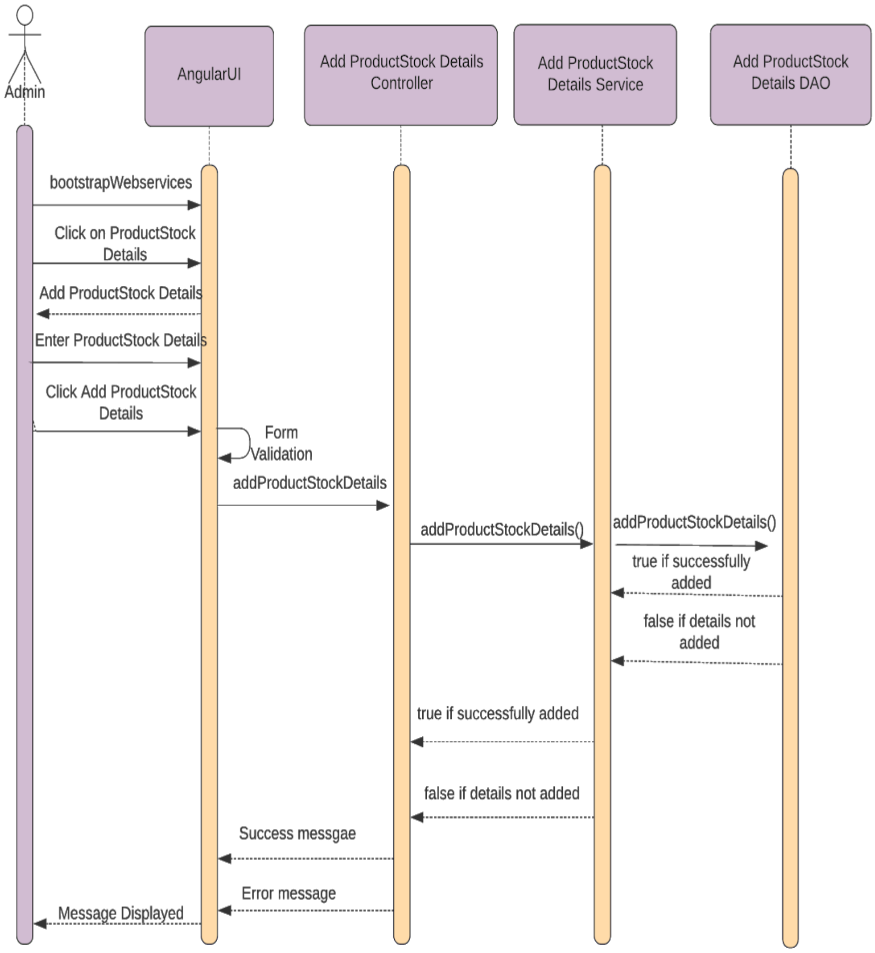
User must as USER should be ableto perform the required functionalities of Stock Management for Raw material and Products for our client Drink and Delight.

Designed and Implemented By:

Atal Kumar
Designation: Analyst

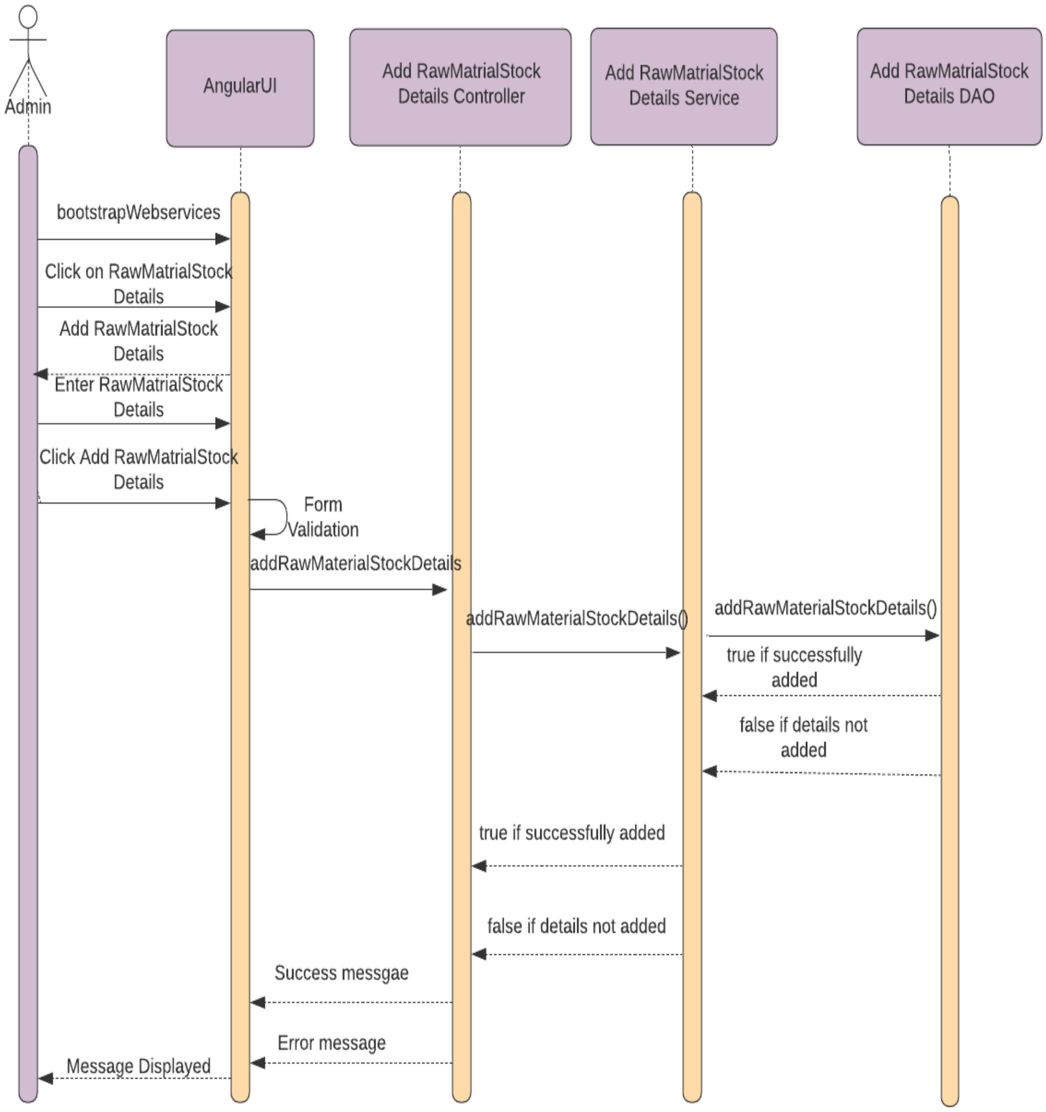
a) Add product Stock

Sequence Diagram for add material stock


Admin
AngularUI
Add ProductStock Details Controller
Add ProductStock Details Service
Add ProductStock Details DAO
bootstrapWebservices
Click on ProductStock Details
Add ProductStock Details
Enter ProductStock Details
Click Add ProductStock Details
Form Validation
addProductStockDetails
addProductStockDetails()
addProductStockDetails()
true if successfully added
false if details not added
true if successfully added
false if details not added
Success messgae
Error message
Message Displayed

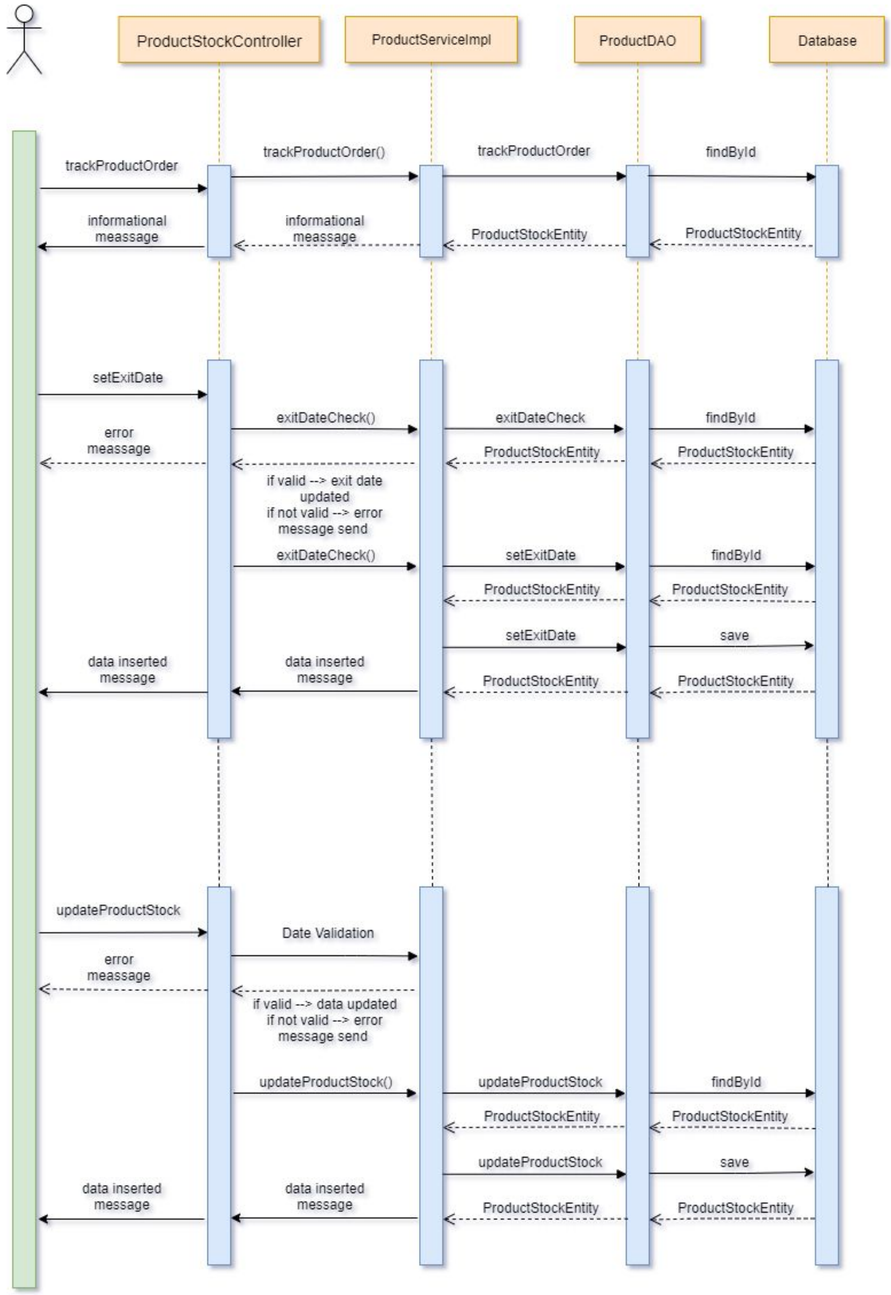
b) Add Raw Material Stock

Sequence Diagram for add material stock


Admin
AngularUI
Add RawMatrialStock Details Controller
Add RawMatrialStock Details Service
Add RawMatrialStock Details DAO
bootstrapWebservices
Click on RawMatrialStock Details
Add RawMatrialStock Details
Enter RawMatrialStock Details
Click Add RawMatrialStock Details
Form Validation
addRawMaterialStockDetails
addRawMaterialStockDetails()
addRawMaterialStockDetails()
true if successfully added
false if details not added
true if successfully added
false if details not added
Success messgae
Error message
Message Displayed

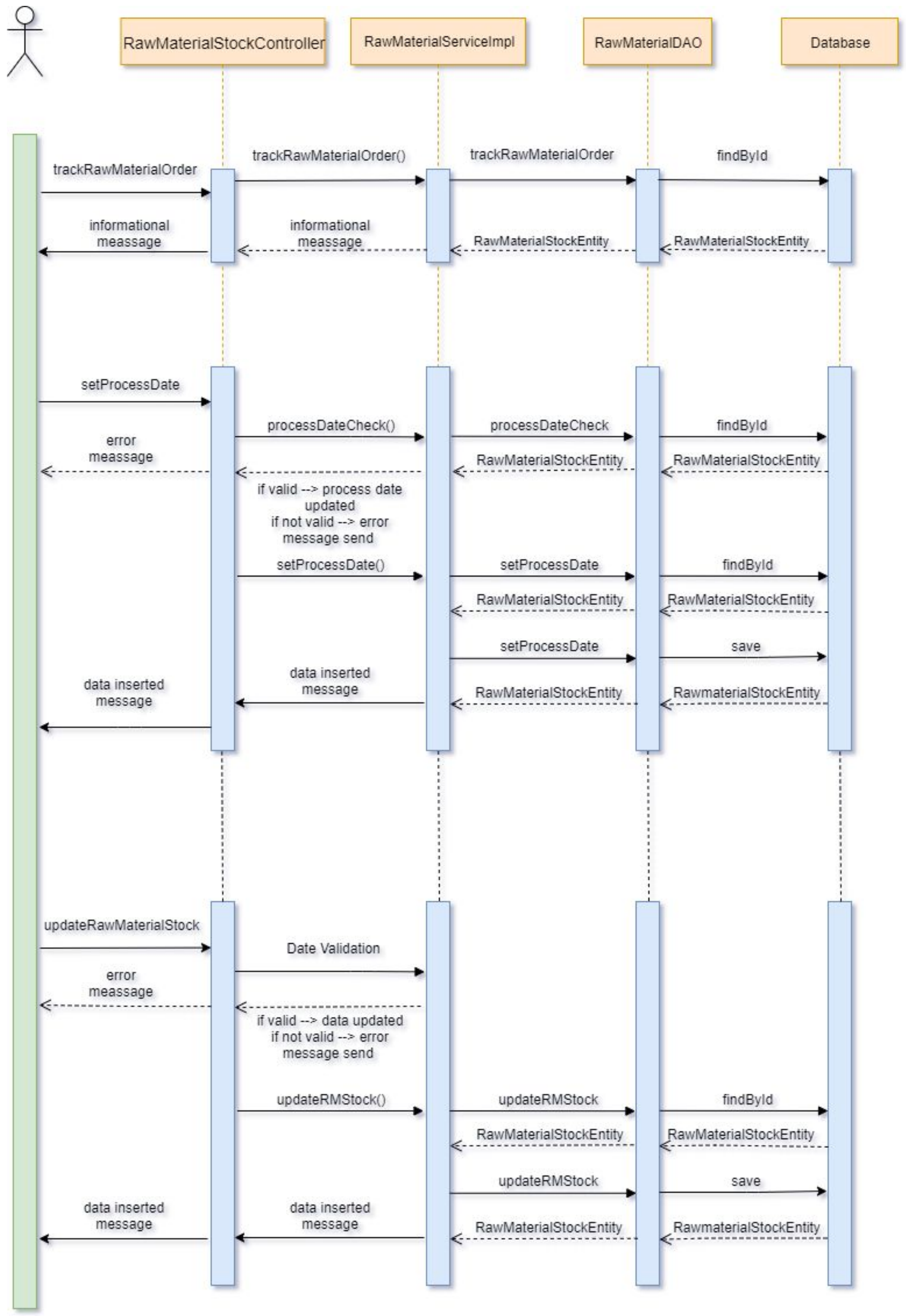
c) Update product Stock

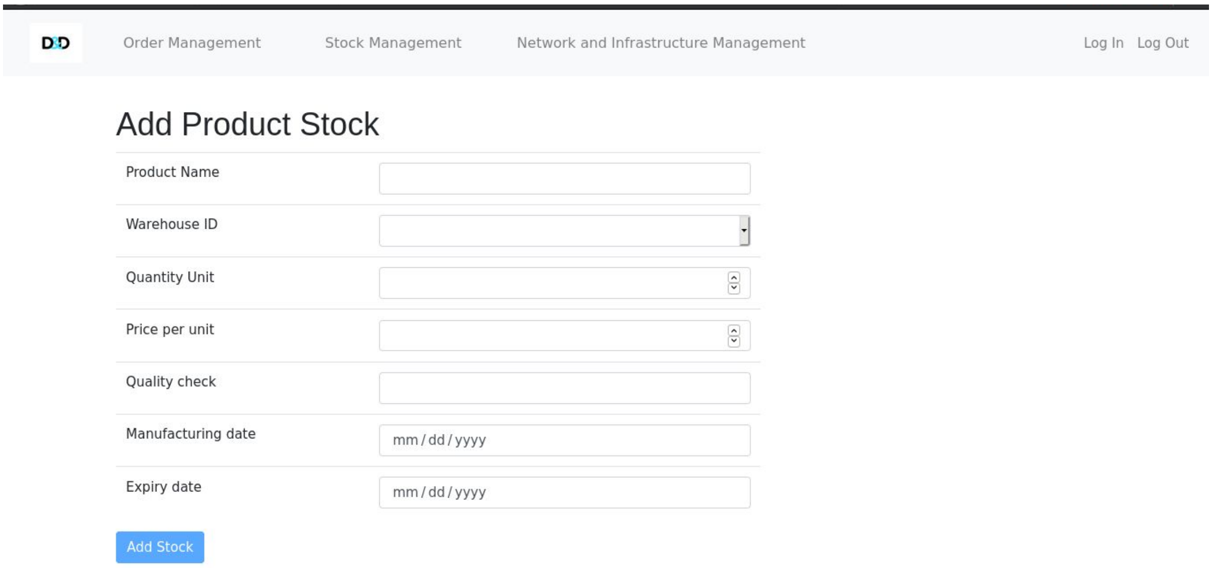
Sequence Diagram for Track Product Order and Update Product Stock



d) Update Raw Material Stock

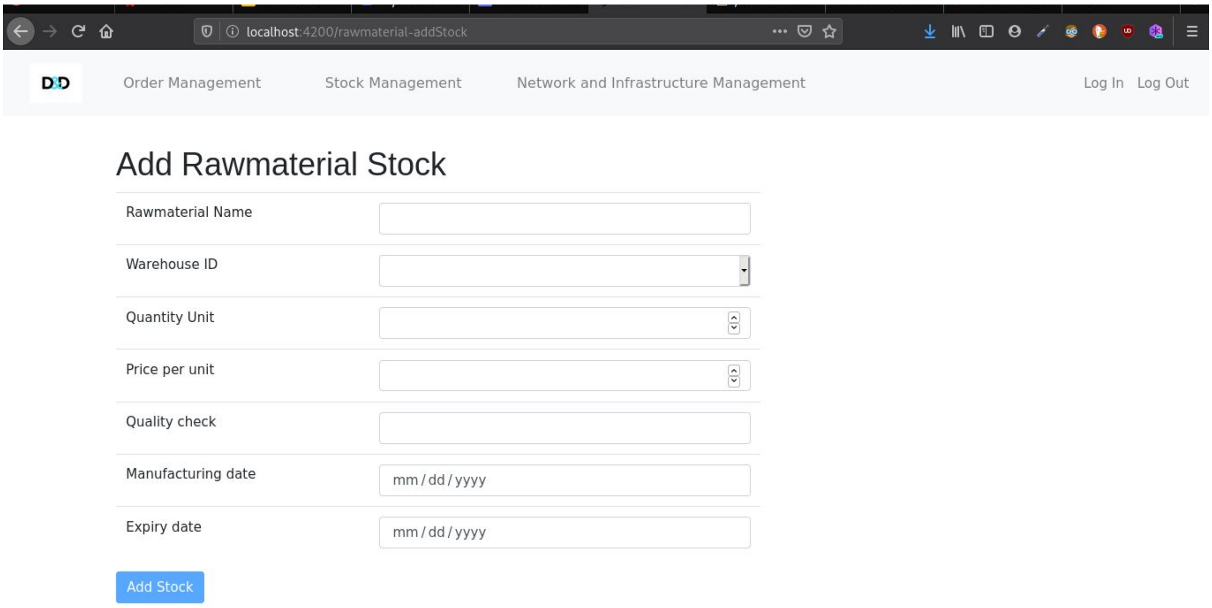
Sequence Diagram for Track Product Order and Update Product Stock





Add Product Stock

Product Name	
Warehouse ID	
Quantity Unit	
Price per unit	
Quality check	
Manufacturing date	mm / dd / yyyy
Expiry date	mm / dd / yyyy

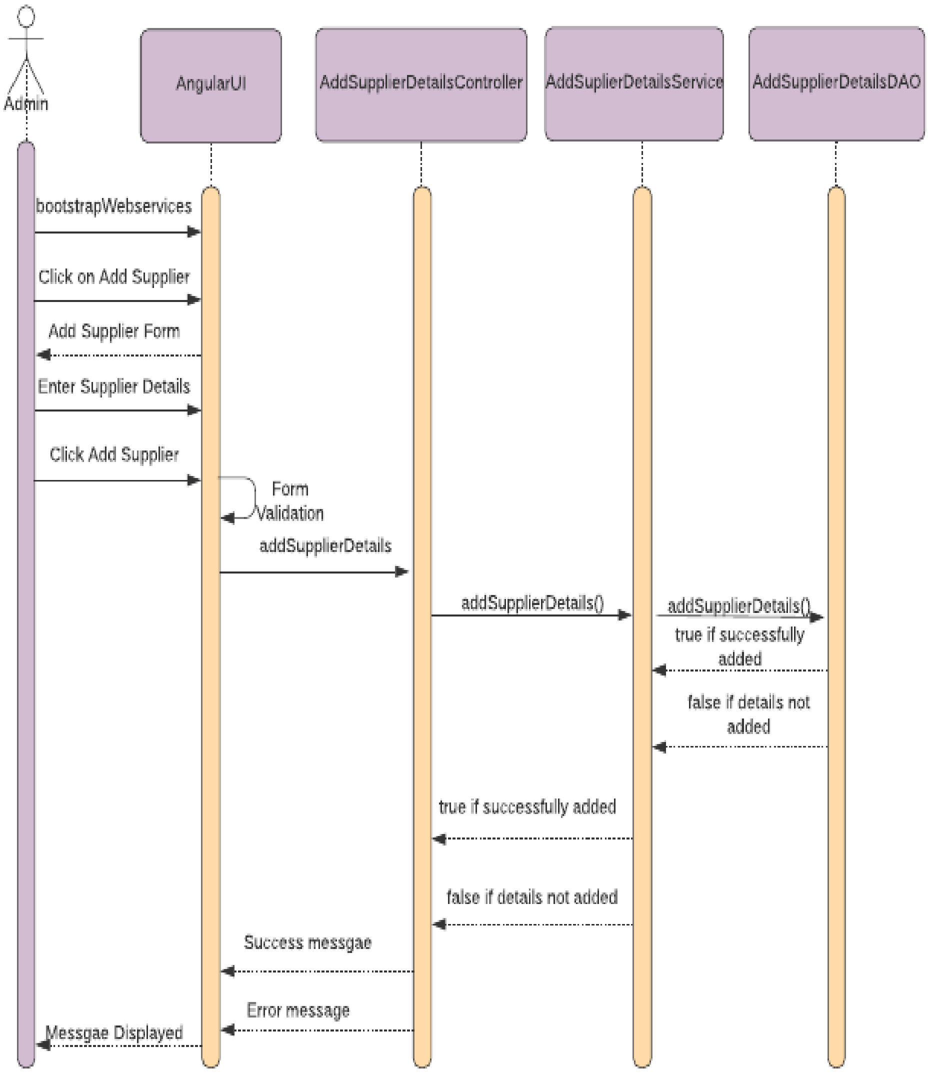

localhost:4200/rawmaterial-addStock
D:D
Order Management
Stock Management
Network and Infrastructure Management
Log In
Log Out

Add Rawmaterial Stock

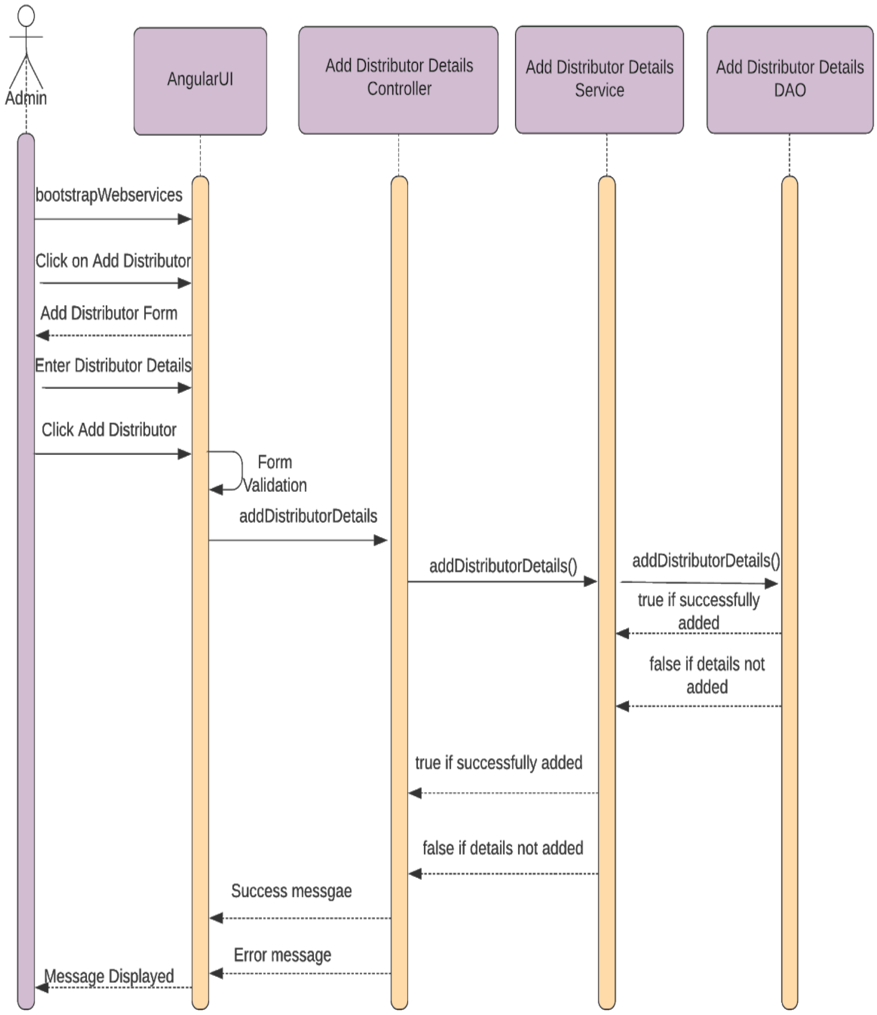
Rawmaterial Name	
Warehouse ID	
Quantity Unit	
Price per unit	
Quality check	
Manufacturing date	mm / dd / yyyy
Expiry date	mm / dd / yyyy

5.4 Network and infrastructure management

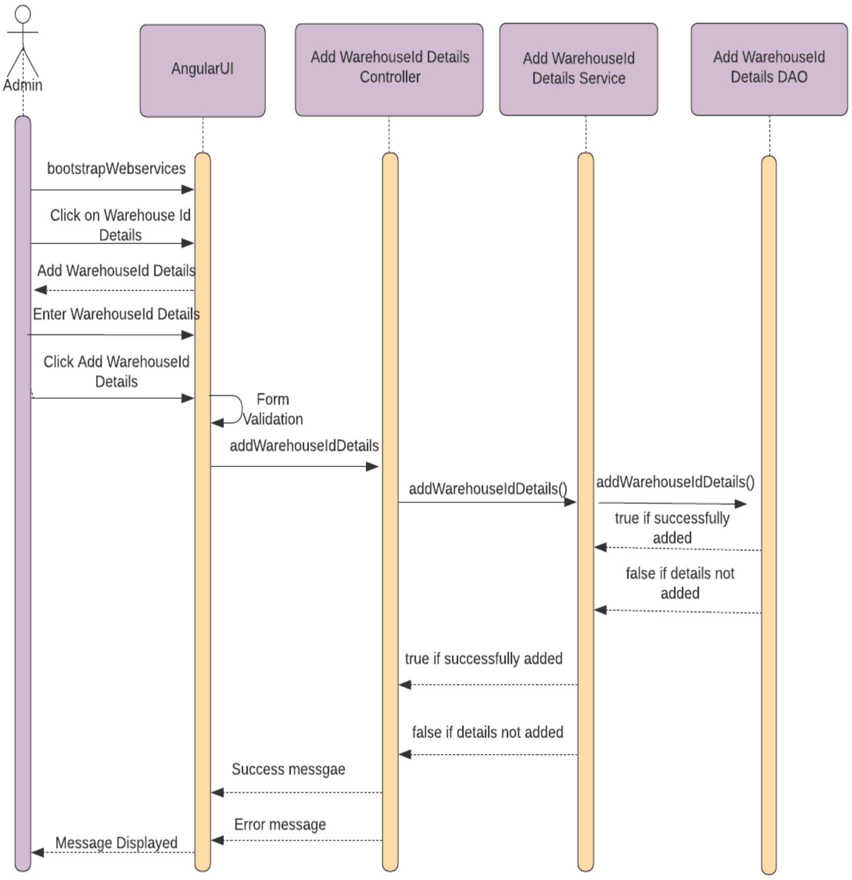
a) Add supplier

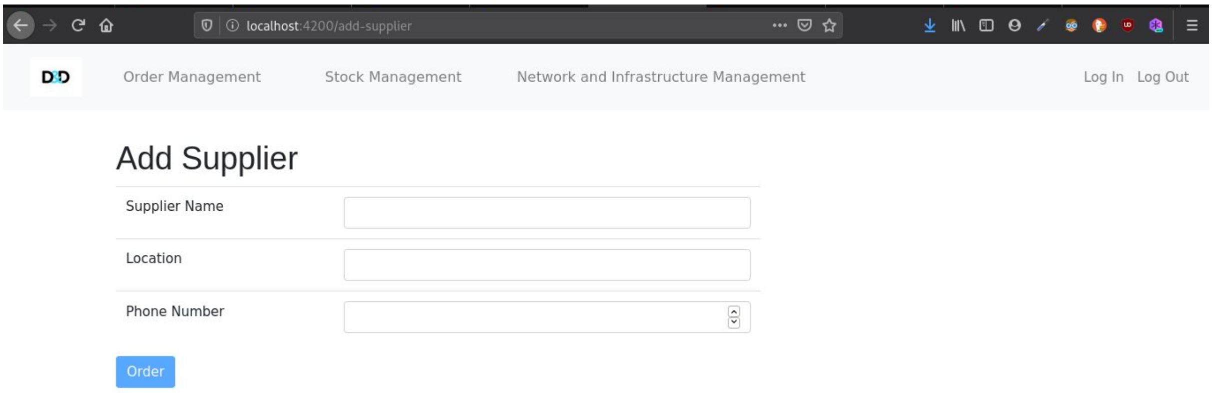


b) Add distributors



c)Add warehouses


Admin
AngularUI
Add WarehouseId Details Controller
Add WarehouseId Details Service
Add WarehouseId Details DAO
bootstrapWebservices
Click on Warehouse Id Details
Add WarehouseId Details
Enter WarehouseId Details
Click Add WarehouseId Details
Form Validation
addWarehouseIdDetails
addWarehouseIdDetails()
addWarehouseIdDetails()
true if successfully added
false if details not added
true if successfully added
false if details not added
Success messgae
Error message
Message Displayed

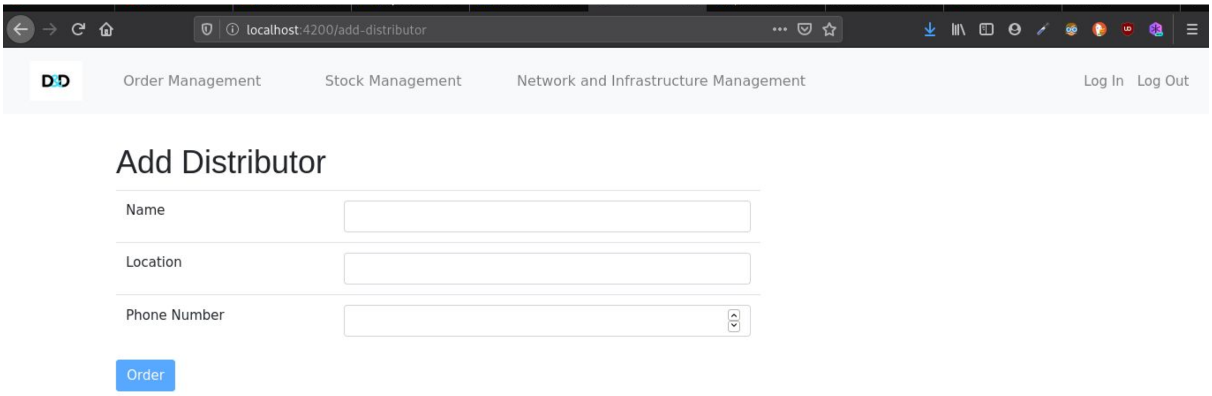

localhost:4200/add-supplier
D&D
Order Management
Stock Management
Network and Infrastructure Management
Log In
Log Out
Order

Add Supplier

Supplier Name

Location

Phone Number

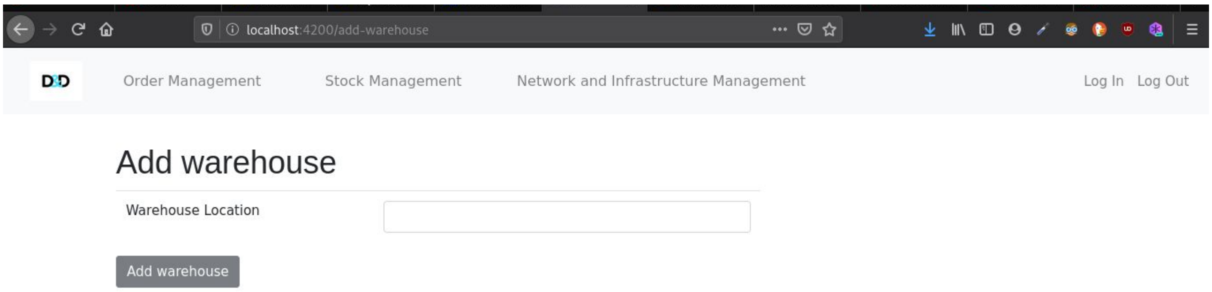

localhost:4200/add-distributor
D&D
Order Management
Stock Management
Network and Infrastructure Management
Log In
Log Out
Order

Add Distributor

Name

Location

Phone Number

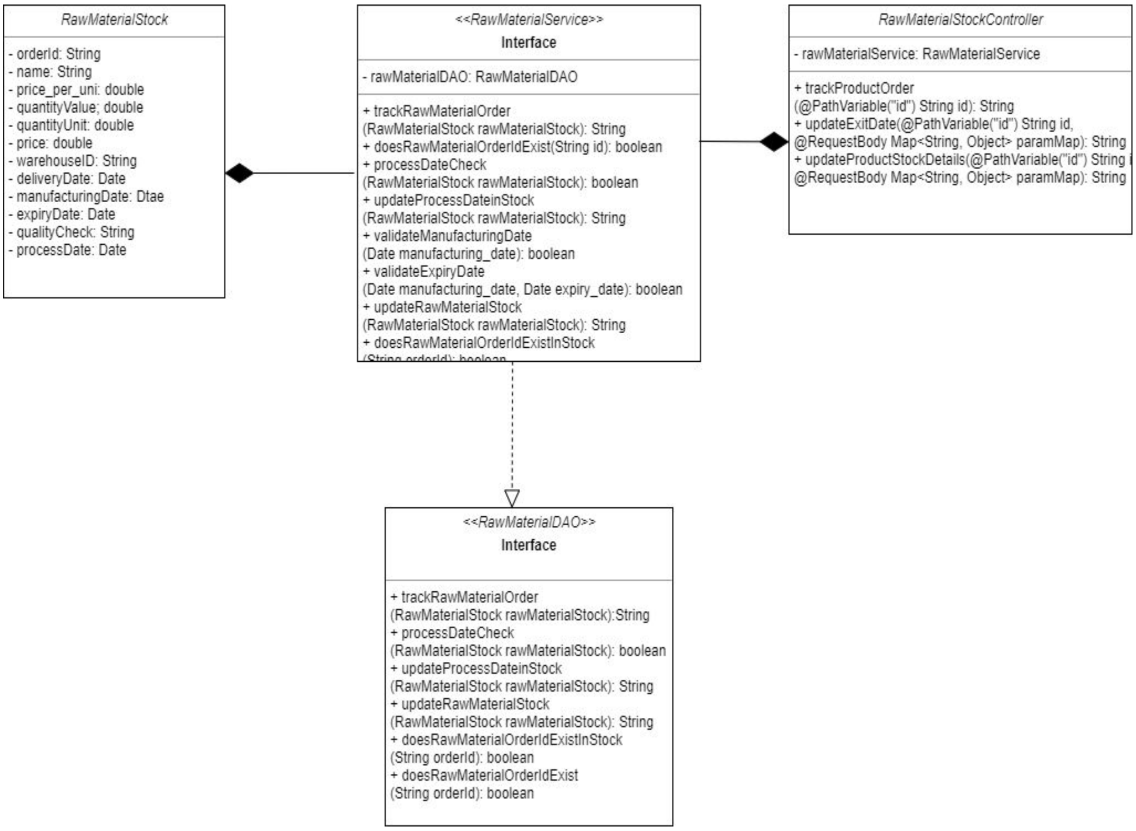

localhost:4200/add-warehouse
D3D
Order Management
Stock Management
Network and Infrastructure Management
Log In
Log Out

Add warehouse

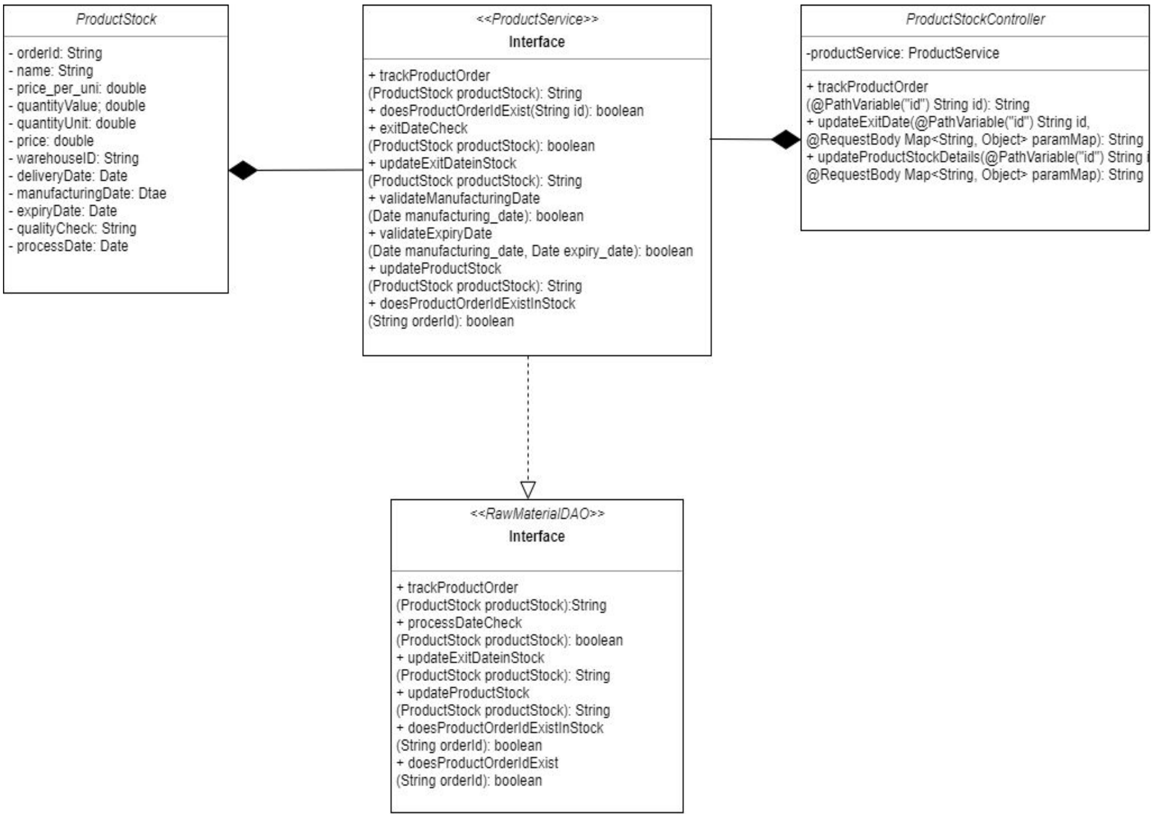
Warehouse Location

Add warehouse

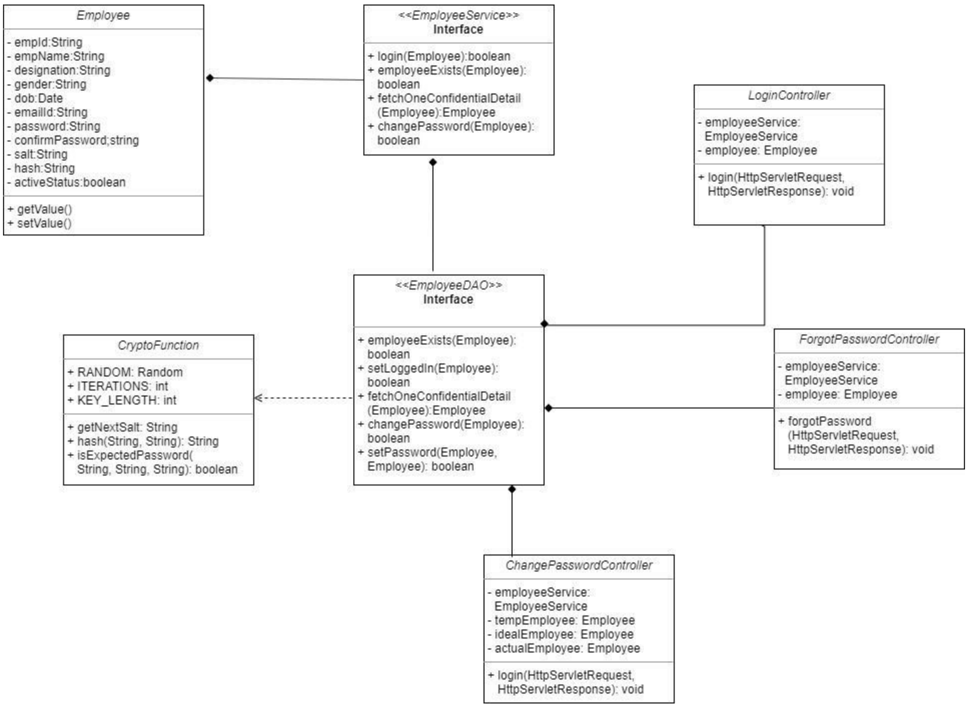
Class Diagram for Raw Material Stock Management:


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 i
@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

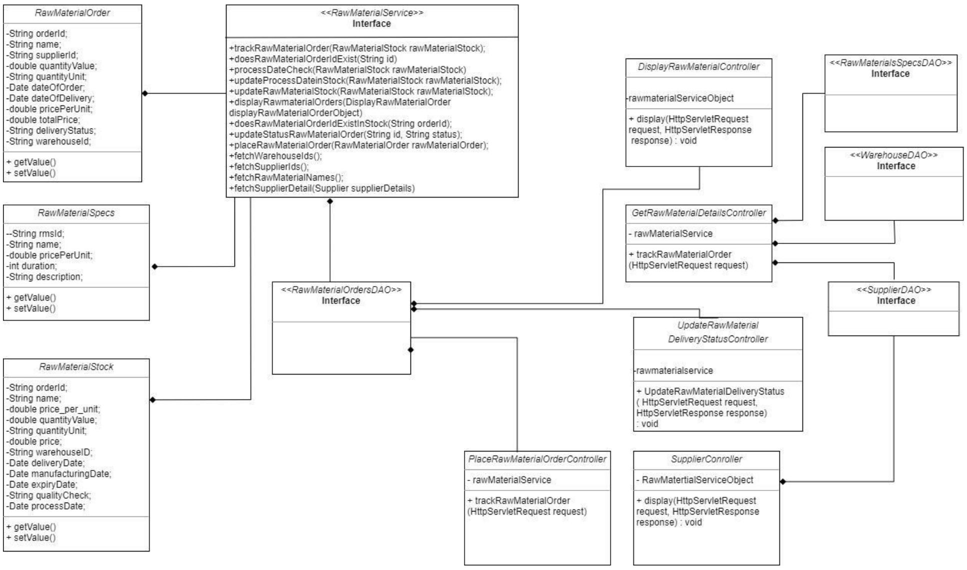
Class Diagram for Product Stock Management:


ProductStock
- 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
<<ProductService>>
Interface
+ trackProductOrder
(ProductStock productStock): String
+ doesProductOrderIdExist(String id): boolean
+ exitDateCheck
(ProductStock productStock): boolean
+ updateExitDateinStock
(ProductStock productStock): String
+ validateManufacturingDate
(Date manufacturing_date): boolean
+ validateExpiryDate
(Date manufacturing_date, Date expiry_date): boolean
+ updateProductStock
(ProductStock productStock): String
+ doesProductOrderIdExistInStock
(String orderId): boolean
ProductStockController
-productService: ProductService
+ trackProductOrder
(@PathVariable("id") String id): String
+ updateExitDate(@PathVariable("id") String id,
@RequestBody Map<String, Object> paramMap): String
+ updateProductStockDetails(@PathVariable("id") String i
@RequestBody Map<String, Object> paramMap): String
<<RawMaterialDAO>>
Interface
+ trackProductOrder
(ProductStock productStock):String
+ processDateCheck
(ProductStock productStock): boolean
+ updateExitDateinStock
(ProductStock productStock): String
+ updateProductStock
(ProductStock productStock): String
+ doesProductOrderIdExistInStock
(String orderId): boolean
+ doesProductOrderIdExist
(String orderId): boolean

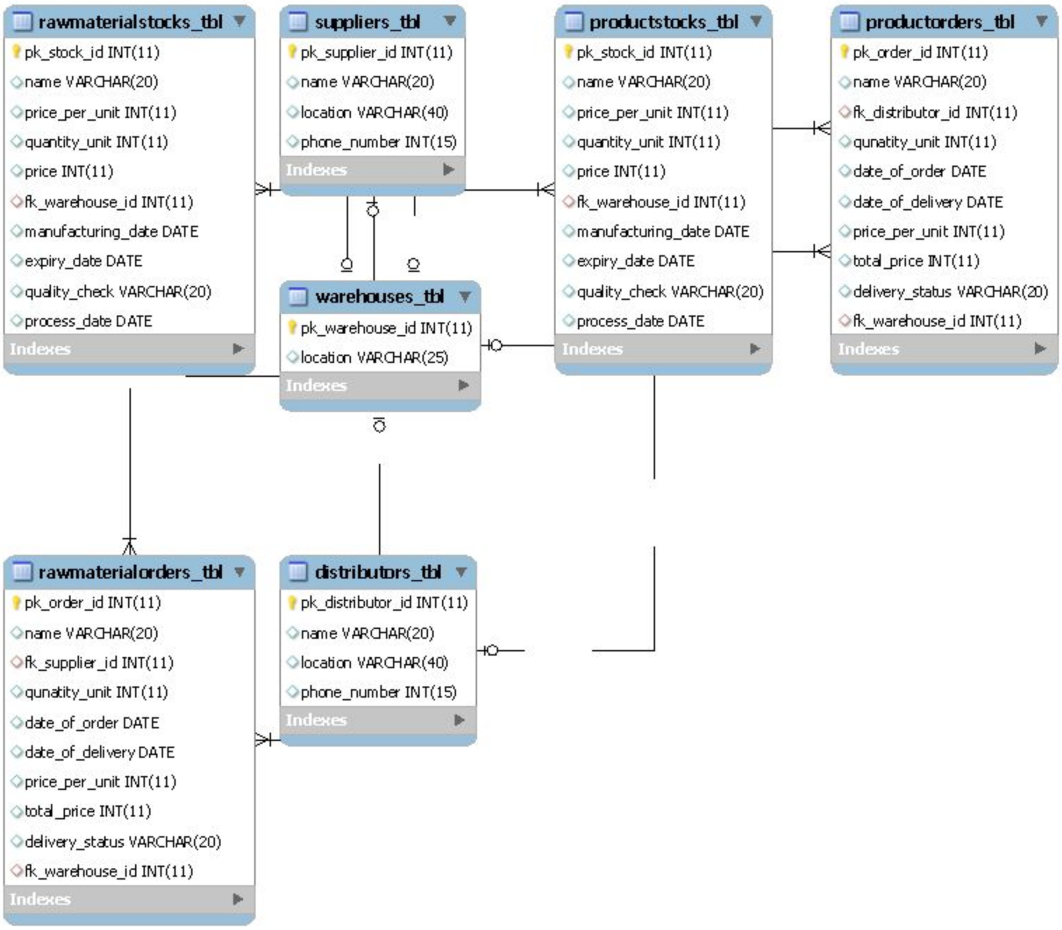
Class Diagram of Login Functionality:

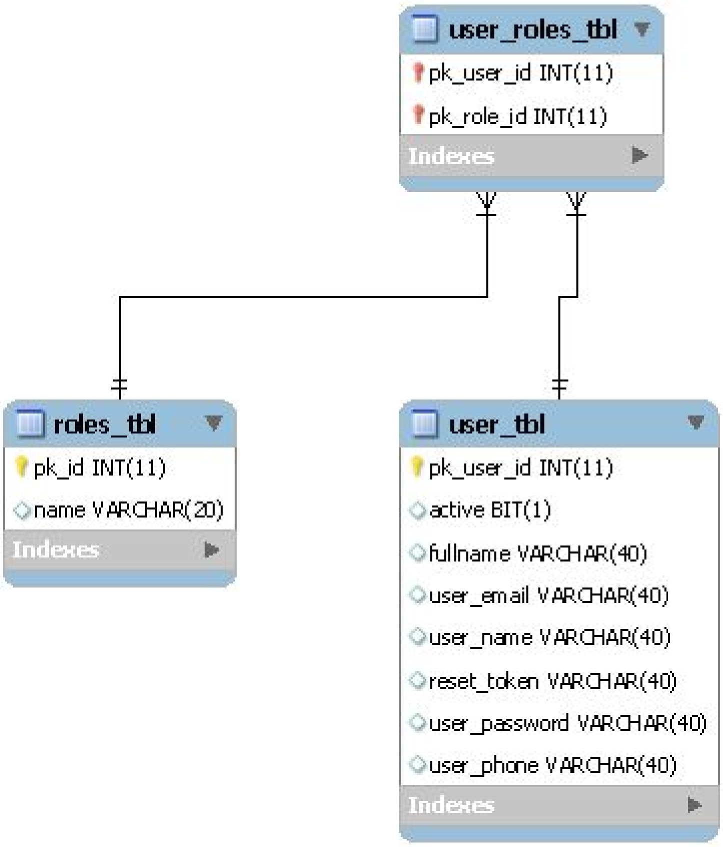

Employee
- empId:String
- empName:String
- designation:String
- gender:String
- dob:Date
- emailId:String
- password:String
- confirmPassword;string
- salt:String
- hash:String
- activeStatus:boolean
+ getValue()
+ setValue()
<<EmployeeService>>
Interface
+ login(Employee):boolean
+ employeeExists(Employee): boolean
+ fetchOneConfidentialDetail (Employee):Employee
+ changePassword(Employee): boolean
LoginController
- employeeService: EmployeeService
- employee: Employee
+ login(HttpServletRequest, HttpServletResponse): void
<<EmployeeDAO>>
Interface
+ employeeExists(Employee): boolean
+ setLoggedIn(Employee): boolean
+ fetchOneConfidentialDetail (Employee):Employee
+ changePassword(Employee): boolean
+ setPassword(Employee, Employee): boolean
CryptoFunction
+ RANDOM: Random
+ ITERATIONS: int
+ KEY_LENGTH: int
+ getNextSalt: String
+ hash(String, String): String
+ isExpectedPassword(String, String, String): boolean
ForgotPasswordController
- employeeService: EmployeeService
- employee: Employee
+ forgotPassword (HttpServletRequest, HttpServletResponse): void
ChangePasswordController
- employeeService: EmployeeService
- tempEmployee: Employee
- idealEmployee: Employee
- actualEmployee: Employee
+ login(HttpServletRequest, HttpServletResponse): void

Class Diagram of Raw Material Order management:


RawMaterialOrder
-String orderId;
-String name;
-String supplierId;
-double quantityValue;
-String quantityUnit;
-Date dateOfOrder;
-Date dateOfDelivery;
-double pricePerUnit;
-double totalPrice;
-String deliveryStatus;
-String warehouseId;
+ getValue()
+ setValue()
<<RawMaterialService>>
Interface
+trackRawMaterialOrder(RawMaterialStock rawMaterialStock);
+doesRawMaterialOrderIdExist(String id)
+processDateCheck(RawMaterialStock rawMaterialStock)
+updateProcessDateinStock(RawMaterialStock rawMaterialStock);
+updateRawMaterialStock(RawMaterialStock rawMaterialStock);
+displayRawmaterialOrders(DisplayRawMaterialOrder displayRawMaterialOrderObject)
+doesRawMaterialOrderIdExistInStock(String orderId);
+updateStatusRawMaterialOrder(String id, String status);
+placeRawMaterialOrder(RawMaterialOrder rawMaterialOrder);
+fetchWarehouseIds();
+fetchSupplierIds();
+fetchRawMaterialNames();
+fetchSupplierDetail(Supplier supplierDetails)
DisplayRawMaterialController
-rawmaterialServiceObject
+ display(HttpServletRequest request, HttpServletResponse response) : void
<<RawMaterialsSpecsDAO>>
Interface
<<WarehouseDAO>>
Interface
RawMaterialSpecs
--String rmsId;
-String name;
-double pricePerUnit;
-int duration;
-String description;
+ getValue()
+ setValue()
GetRawMaterialDetailsController
- rawMaterialService
+ trackRawMaterialOrder (HttpServletRequest request)
<<RawMaterialOrdersDAO>>
Interface
<<SupplierDAO>>
Interface
UpdateRawMaterial DeliveryStatusController
-rawmaterialservice
+ UpdateRawMaterialDeliveryStatus (HttpServletRequest request, HttpServletResponse response) : void
RawMaterialStock
-String orderId;
-String name;
-double price_per_unit;
-double quantityValue;
-String quantityUnit;
-double price;
-String warehouseID;
-Date deliveryDate;
-Date manufacturingDate;
-Date expiryDate;
-String qualityCheck;
-Date processDate;
+ getValue()
+ setValue()
PlaceRawMaterialOrderController
- rawMaterialService
+ trackRawMaterialOrder (HttpServletRequest request)
SupplierConroller
- RawMatertialServiceObject
+ display(HttpServletRequest request, HttpServletResponse response) : void

Database entity Diagram


rawmaterialstocks_tbl
pk_stock_id INT(11)
name VARCHAR(20)
price_per_unit INT(11)
quantity_unit INT(11)
price INT(11)
fk_warehouse_id INT(11)
manufacturing_date DATE
expiry_date DATE
quality_check VARCHAR(20)
process_date DATE
Indexes
suppliers_tbl
pk_supplier_id INT(11)
name VARCHAR(20)
location VARCHAR(40)
phone_number INT(15)
Indexes
productstocks_tbl
pk_stock_id INT(11)
name VARCHAR(20)
price_per_unit INT(11)
quantity_unit INT(11)
price INT(11)
fk_warehouse_id INT(11)
manufacturing_date DATE
expiry_date DATE
quality_check VARCHAR(20)
process_date DATE
Indexes
productorders_tbl
pk_order_id INT(11)
name VARCHAR(20)
fk_distributor_id INT(11)
qunatity_unit INT(11)
date_of_order DATE
date_of_delivery DATE
price_per_unit INT(11)
total_price INT(11)
delivery_status VARCHAR(20)
fk_warehouse_id INT(11)
Indexes
warehouses_tbl
pk_warehouse_id INT(11)
location VARCHAR(25)
Indexes
rawmaterialorders_tbl
pk_order_id INT(11)
name VARCHAR(20)
fk_supplier_id INT(11)
qunatity_unit INT(11)
date_of_order DATE
date_of_delivery DATE
price_per_unit INT(11)
total_price INT(11)
delivery_status VARCHAR(20)
fk_warehouse_id INT(11)
Indexes
distributors_tbl
pk_distributor_id INT(11)
name VARCHAR(20)
location VARCHAR(40)
phone_number INT(15)
Indexes



Project Progress Status (Product Management System)

#	Assigned	:	rk
1	ng UML Diagrams, Defining Test Cases	nd Stories written. Use Case Diagram	e the sequence diagram, include the life block in

	and Sequence Diagrams	and Sequence Diagram defined	sequence diagram. Develop Knowledge about micro services architecture
2	ment the test cases using Junit, Implement the modules with core java implementation. Use Java Collection API for data storage (non-persistence)	es Implemented using Java. Junit test cases are written and successfully tested. nce Diagrams are modified according to the previous sprint feedback	r architecture is not properly designed. Write more test case scenario. Properly comment the code. Code Convention is not upto industry standard. Presentation layer is not implemented
3	ment 3-tier architecture. Link the business logic with Database using JDBC connection	ase is designed as per ER am. Service and Presentation layer is properly implanted. All Validations are done in presentation layer. s properly commented. Inline comments and redundant codes are removed. JDBC connections are done. r is being implemented.	

4	n the front end with Html , CSS, Bootstrap	end pages are designed with HTML, CSS and Bootstrap. JSP and servlet are being used to connect the front end with backend Java modules.	are not made responsive for mobile. Proper use of bootstrap is missing. Unity in colour scheme is missing. onal advise : To make the software more dynamic. Toaster message should be implemented
5	ce the presentation layer with Angular client App and write the BDD test Cases using cucumber.	ar app is designed. Pages are made mobile responsive according to the previous sprint feedback. r scheme is done uniform across all pages. project is made dynamic. Toastrmessege have been implemented as well as Servlet Technology is being explored to link the front-end with backend	r Scrum Model is not followed. Every Individual is being asked to assigned a single module. JDBC connections are not closed properly. Proper Documentation is missing.
6 & Sprint 7	ce JDBC connection with JPA Hibernate. Implement Sprint MVC	s being replaced with JPA Hibernate API. Connection open and close is being managed by session and transaction management of Hibernate.	r Documentation required. Add proper validation for product id and product image. Write Spring test cases.

		Singleton design is being done for session and transaction management is done using Spring bean. Dependency injection is implemented using Auto wiring. Started working on updated documentation. ve properly divided the modules.	onal Advise: plement audit trail
8	g Boot Implementation	g Boot implemented	ed to use RestTemplate