**Drink and Delight**

****

**Team Members: -**

|  |  |
| --- | --- |
| Name | Role |
| Gagandeep Singh | Scrum Master |
| Palak Khandelwal | Member |
| Ajinkya Hase | Member |
| Rushikesh Walke | Member |
| Shubham Sangle | Product Owner |

Table of Contents

1. Introduction ------------------------------------------------------------------------------------
2. Overview ----------------------------------------------------------------------------------------
3. Epic & Stories ----------------------------------------------------------------------------------
4. Object Diagram -------------------------------------------------------------------------
5. Class Diagram ----------------------------------------------------------------------------------
6. Use Cases ---------------------------------------------------------------------------------------
7. Database Tables
8. System Requirements

1. Introduction

Drink and Delight is an inventory management system. In this system we can add/buy raw materials and products from suppliers and distributors respectively, place orders for them and update their stock and delivery status in the warehouse.

2. Overview 

Drink and Delight is a Spring Boot Application where the frontend is designed using Angular js and MySQL database is used.

In our system there are two actors: Admin and User. Admin has the functionalities to add users, raw materials, products, suppliers and distributors.

The whole system is divided into 5 services namely : Auth service, Stock management service, Supplier service, Product order service & Raw material order service.

Users can perform operations like place order for both products and raw materials and update their delivery status.

3. Epic & Stories

Auth service :-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epic | Stories | As a/an | I want to | So that… |
| Login | Registration | Admin | Sign up as new user | A new user will be registered |
| Login | Admin, User | Validate the entered username and password | Check whether the entered password and username is correct or not |
| Change Password | User, Admin | Change existing password | A new password will be created for a particular Account |
| Update User | User | Modify User details | Employee can modify user details |

Raw Material Order service :-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epic | Stories | As a/an | I want to | So that... |
| Raw Material order | Place Raw Material Order | User | Enter all the details for a Raw Material Order | Raw Material Order is placed successfully. |
| Fetch All Orders | User | Give command to fetch all orders | All Orders will be displayed |
| Fetch an Order by its ID | User | Give the order ID to be displayed | Get all the information related to that particular raw material order |
| Update order’s delivery status | User | Select an order to change it’s delivery status | Just update the delivery status from processing to delivered/ cancelled |

Product Order service :-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epic | Stories | As a/an | I want to | So that... |
| Product order | Place Product Order | User | Enter all the details for a Product Order | Product Order is placed successfully. |
| Fetch All Product Orders | User | Give command to fetch all Product orders | All Orders will be displayed |
| Fetch an Order by its ID | User | Give the order ID to be displayed | Get all the information related to that particular product order |
| Update order’s delivery status | User | Select an order to change it’s delivery status | Just update the delivery status from processing to delivered/ cancelled |

Inventory Stock Management Service :-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epic | Stories | As a/an | I want to | So that... |
| Manage Stock(Add , Update , Delete Product and Raw Material) | Add Product/Raw Material | User | Enter all the details for Product/Raw Material | Product/Raw Material is added successfully |
| Fetch All Products/Raw Material | User | Give command to fetch all Products/Raw Material | All Products/Raw Materials will be displayed |
| Fetch Product/Raw Material by id | User | Give the Product/Raw Material ID to be displayed | Get all the information related to that particular Product/Raw Material |
| Update Product/Raw Material Stock | User | Send the selected update Product/Raw Material Stock request | Just update the available quantity of Product/Raw Material. |

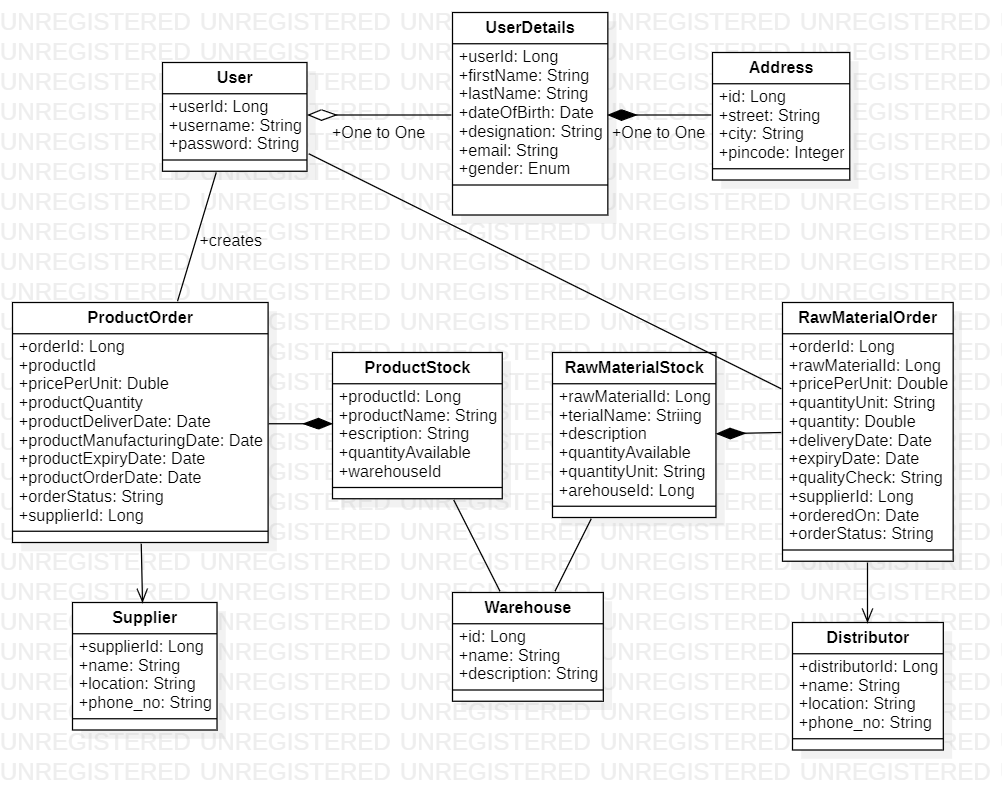
Inventory-Supplier**-**Service

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epic | Stories | As a/an | I want to | So that... |
| Supplier | Add Supplier | Admin | Enter all the details for a supplier | Supplier is added successfully |
| Fetch All Suppliers | Admin | Give command to fetch all suppliers | All suppliers will be displayed |
| Update Supplier details | Admin | Give the supplier details to be updated | Supplier record will get updated. |

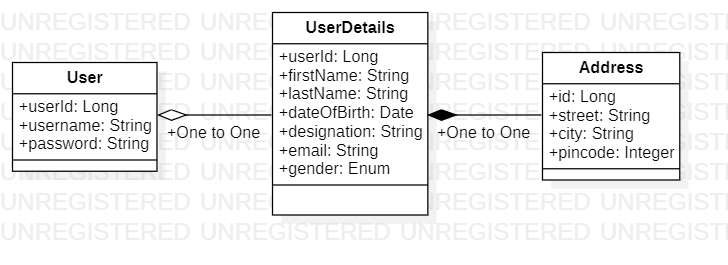
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epic | Stories | As a/an | I want to | So that... |
| Distributor | Add Distributor | Admin | Enter all the details for a distributor | Distributor is added successfully |
| Fetch All Distributors | Admin | Give command to fetch all distributors | All distributors will be displayed |
| Update Distributor details | Admin | Give the distributor details to be updated | Distributor record will get updated. |



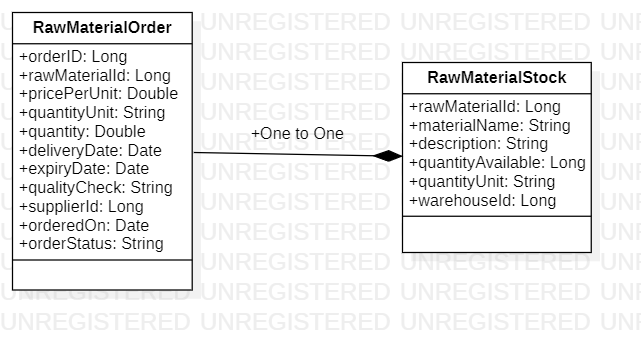
4. Object Diagram



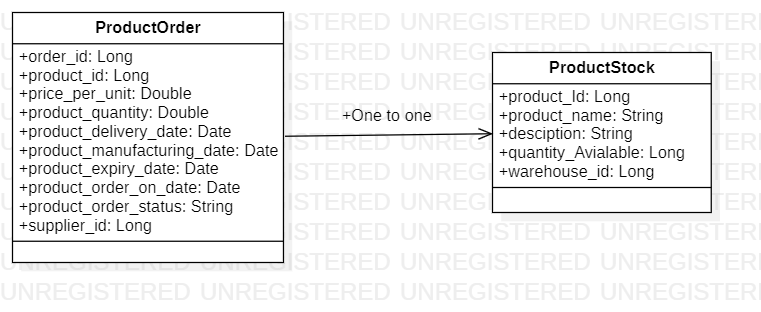
4.1 Authentication

****

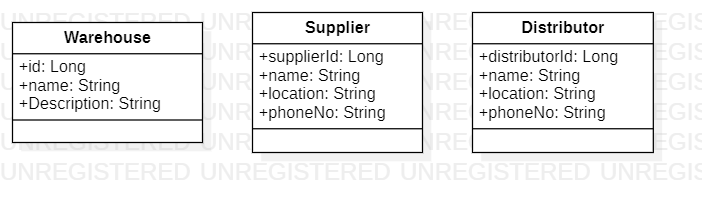
4.2 Raw Material



4.3 Products



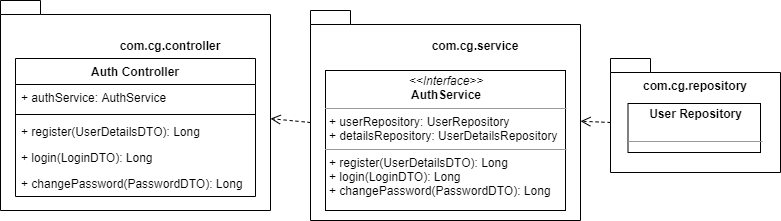
4.4 Supplier



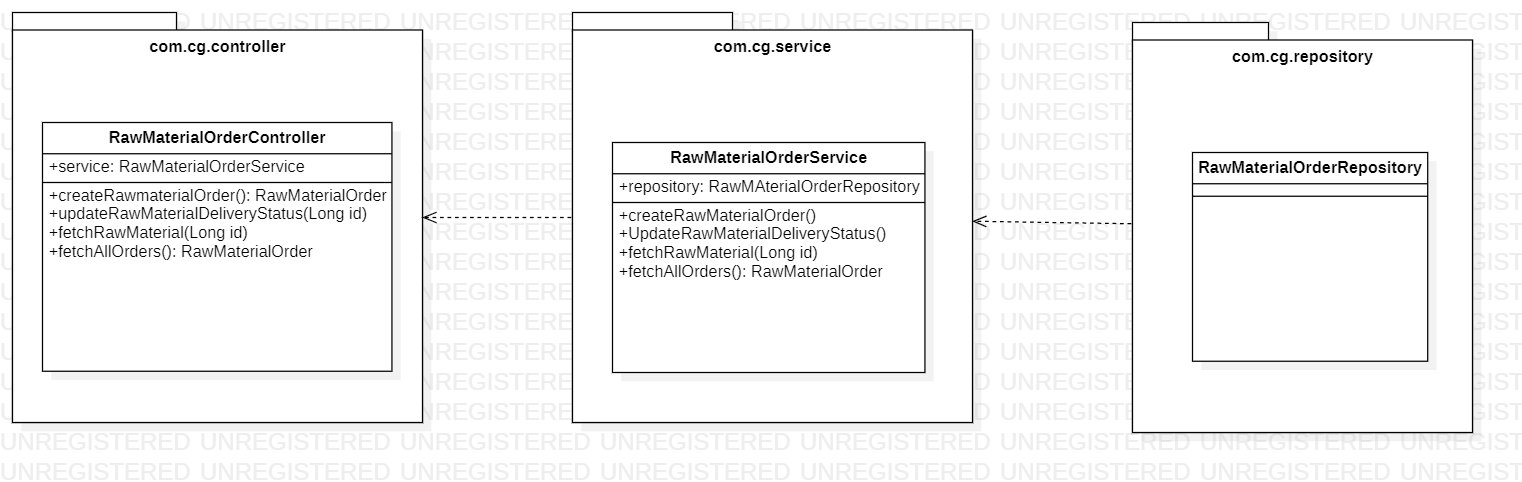


5. Class Diagram

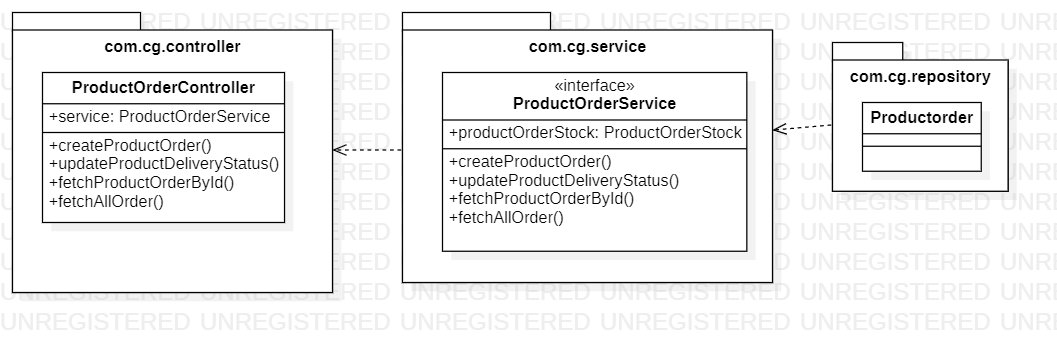
5.1 Authentication

****

5.2 Raw Material



5.3 Products



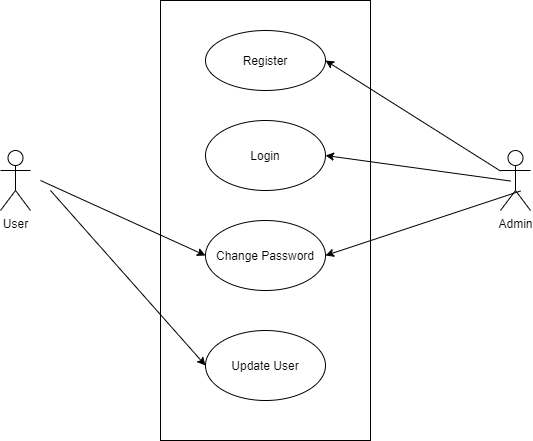
5.5 Supplier





6. Use Case Diagram

6.1 Auth Service

****

* 1. Register

A new user can be registered by the admin and he will be assigned a username and password.

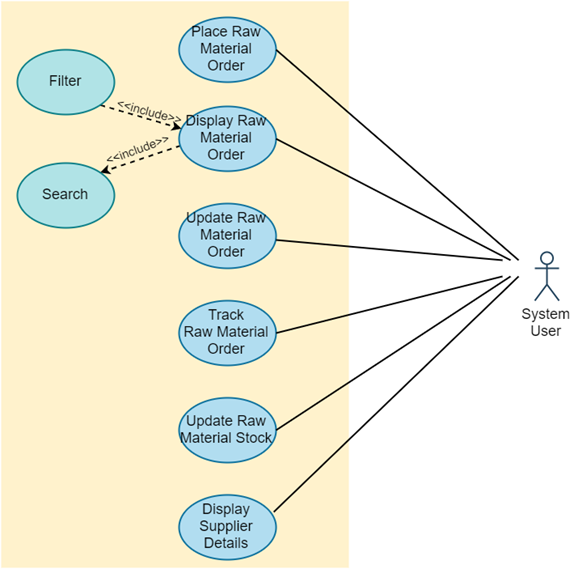
* 1. Login

The user entered username and password is validated.

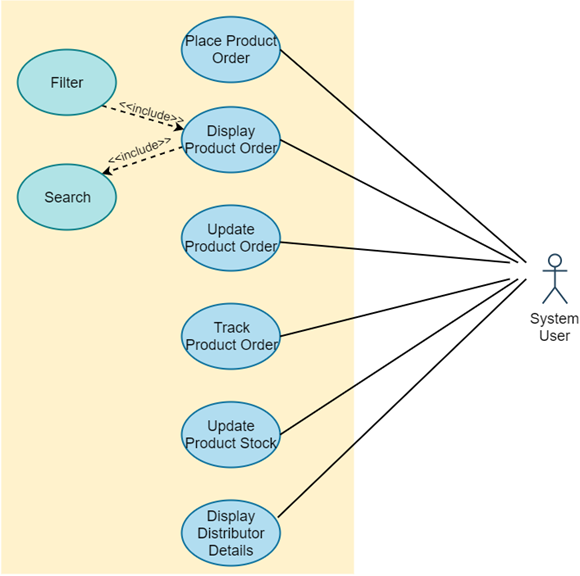
* 1. Change password

The existing password can be changed, and a new password will be created.

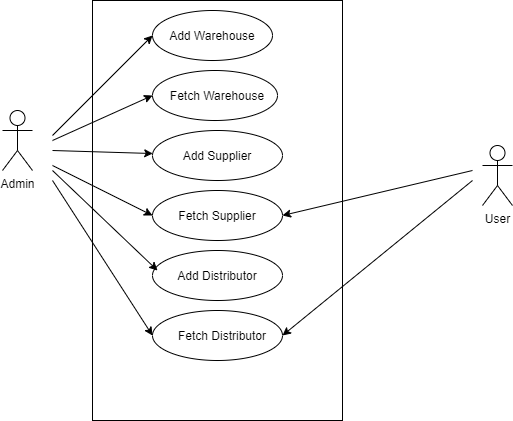
6.2 Raw Material and Stock Management



6.3 Products and Stocks Management



6.4 Supplier Service





7. Database Tables

1. **User Credential table**

|  |  |  |  |
| --- | --- | --- | --- |
| Database Table | user\_master |  | Stores User Credentials |
| user\_id | NUMBER | 6 | Primary Key Id |
| username | VARCHAR | 25 | Username |
| password | VARCHAR | 25 | Password |
| role | NUMBER | 6 | Role table mapping |

1. **User details table**

|  |  |  |  |
| --- | --- | --- | --- |
| Database table | user\_details |  | To store additional details of user |
| user\_id(FK) | NUMBER | 6 | Mapping to user table |
| first\_name | VARCHAR | 25 | First name |
| last\_name | VARCHAR | 25 | Last name |
| DOB | DATE | DD-MM-YYYY | Date of birth (age 18-58) |
| email | VARCHAR | 20 | Email |
| phoneno | VARCHAR | 10 | Phone No |
| gender | VARCHAR | 1 | Gender (M, F) |
| designation | VARCHAR | 25 | Designation |

1. **Address Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Database table | address |  | To store addresses |
| user\_id (FK) | NUMBER | 6 | User foreign Key |
| city | VARCHAR | 20 | City |
| street | VARCHAR | 20 | State |
| pincode | VARCHAR | 6 | Pin code |

1. **Product Stock**

|  |  |  |  |
| --- | --- | --- | --- |
| **ProductStock** | **DONE** |  |  |
| **column name** | **type** | **length** | **desc** |
| **productId** | **Long** | **10** | **primary key** |
| **materialName** | **String** | **30** |  |
| **description** | **String** | **50** |  |
| **quantityAvailable** | **Long** | **6** | **Stock amount** |
| **quantityUnit** | **String** | **5** | **\*\*Units will be an Enum** |
| **warehoueId** | **Long** | **10** | **Foreign Key** |

1. **RawMaterial Stock**

|  |  |  |  |
| --- | --- | --- | --- |
| RawMaterialStock | DONE |  |  |
| column name | type | length | desc |
| productId | Long | 10 | primary key |
| materialName | String | 30 |  |
| description | String | 50 |  |
| quantityAvailable | Long | 6 | Stock amount |
| quantityUnit | String | 5 | \*\*Units will be an Enum |
| warehoueId | Long | 10 | Foreign Key |

1. **Product Order**

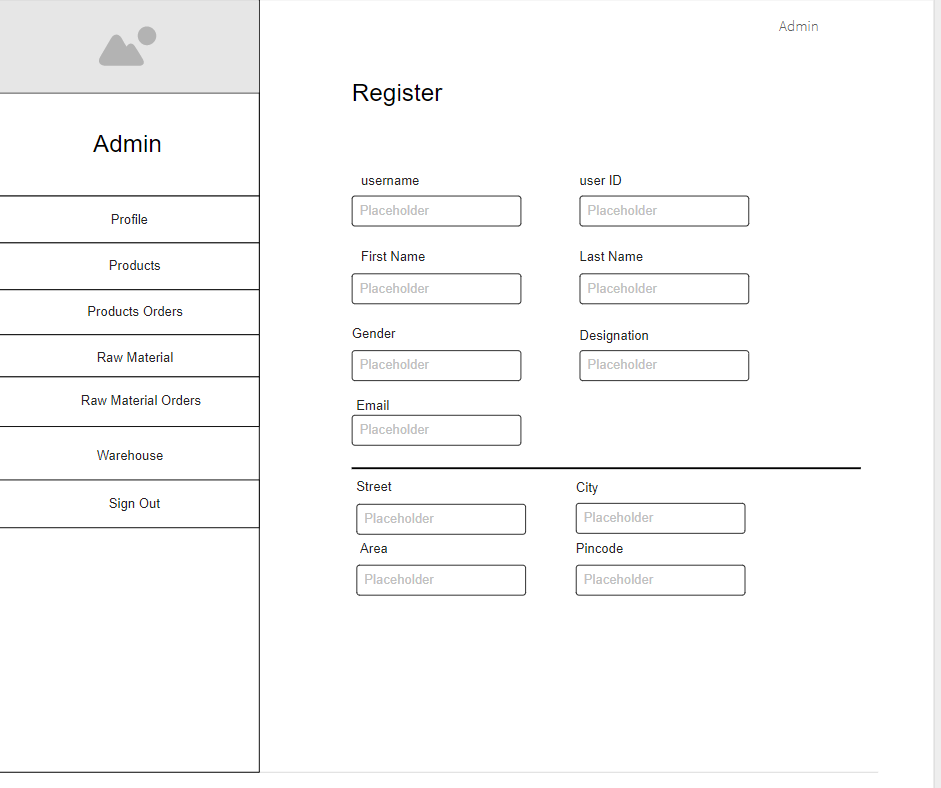
|  |  |  |  |
| --- | --- | --- | --- |
| RawMaterialOrder | DONE |  |  |
| column name | type | length | desc |
| orderId | Long | 10 | primary Key |
| rowMaterialId | Long | 10 | foreign key |
| pericePerUnit | Double | 6 | Price |
| quantityUnit | String |  | \*\*Units will be an Enum |
| quantity | Double | 10 |  |
| dilveryDate | Date | Date | Estimated date |
| expiryDate | Date | Date | last date |
| qualityCheck | String | 20 | All passed |
| supplierId | Long | 10 | Foreign Key |
| orderedOn | Date | Date | created on |
| orderStatus | String | 20 | Values: Processing, Delivered, Cancelled |

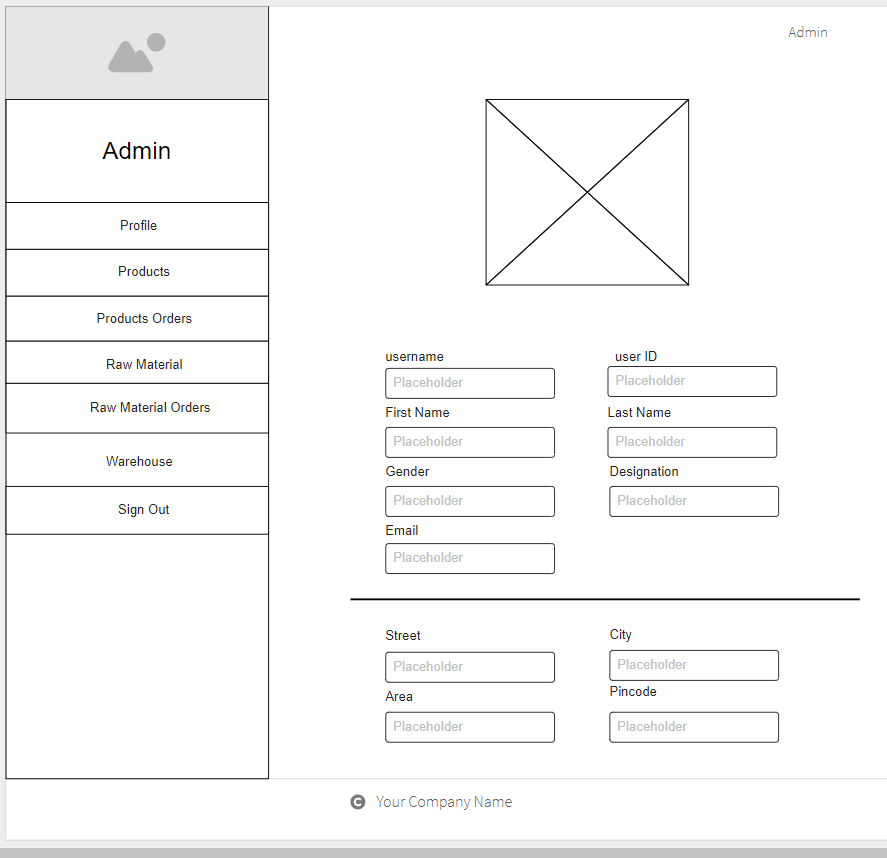
1. **RawMaterial Order**

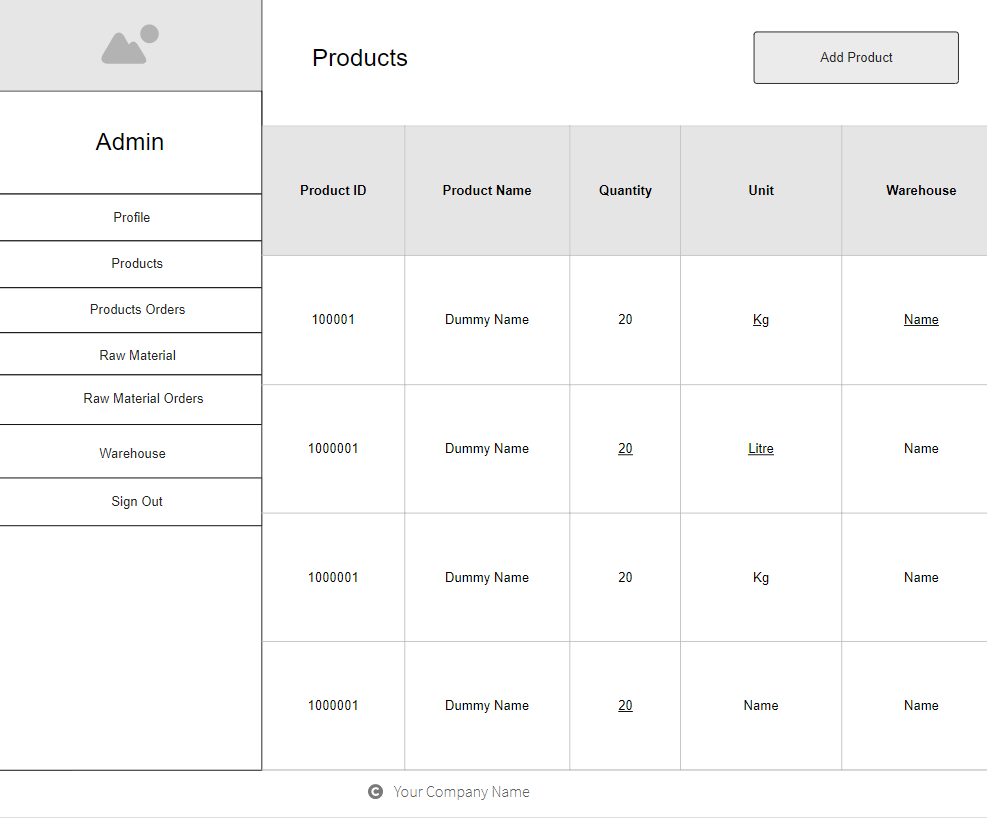
|  |  |  |  |
| --- | --- | --- | --- |
| ProductOrder | DONE |  |  |
| column name | type | length | Description |
| orderId | Long | 10 | primary Key |
| productId | Long | 10 | foreign key |
| pericePerUnit | Double | 6 | Price |
| quantityUnit | String |  | \*\*Units will be an Enum |
| quantity | Double | 10 |  |
| dilveryDate | Date | Date | estimated date |
| manufacturingDate | Date | Date | created on |
| expiryDate | Date | Date | last date |
| supplierId | Long | 10 | foreign key |
| qualityCheck | String | 20 | All passed |
| orderedOn | Date | Date |  |
| orderStatus | String | 20 | Values: Processing, Delivered, Cancelled |

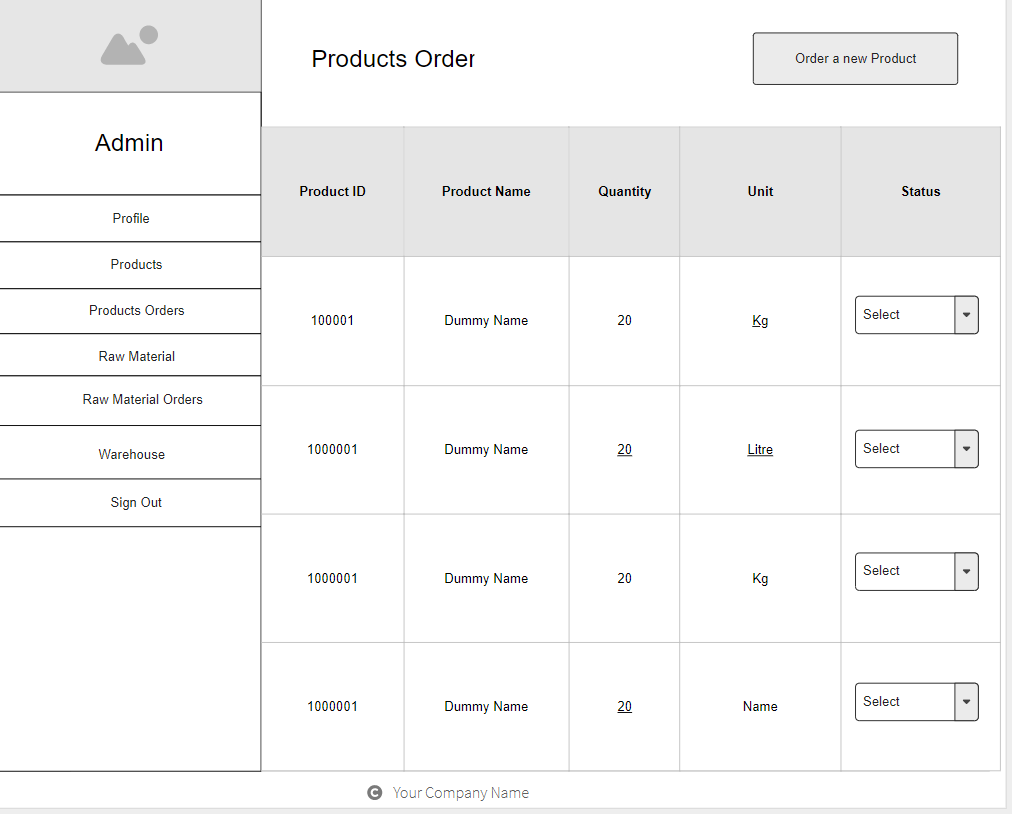
8. Wireframe Diagrams

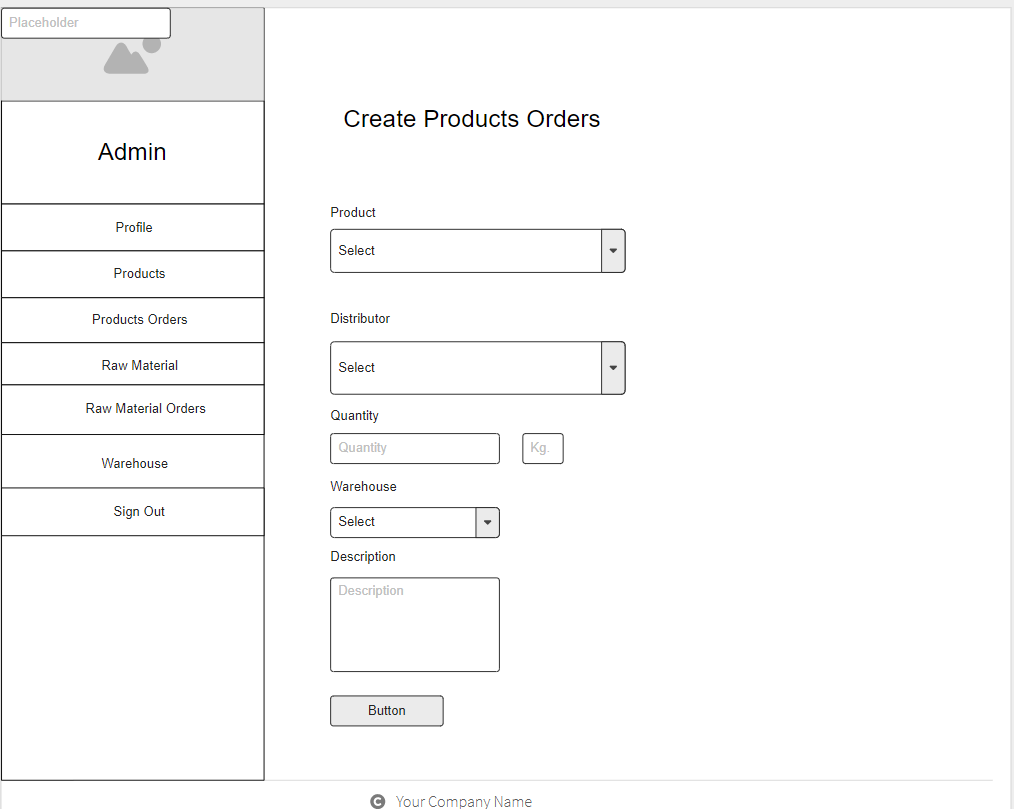
****

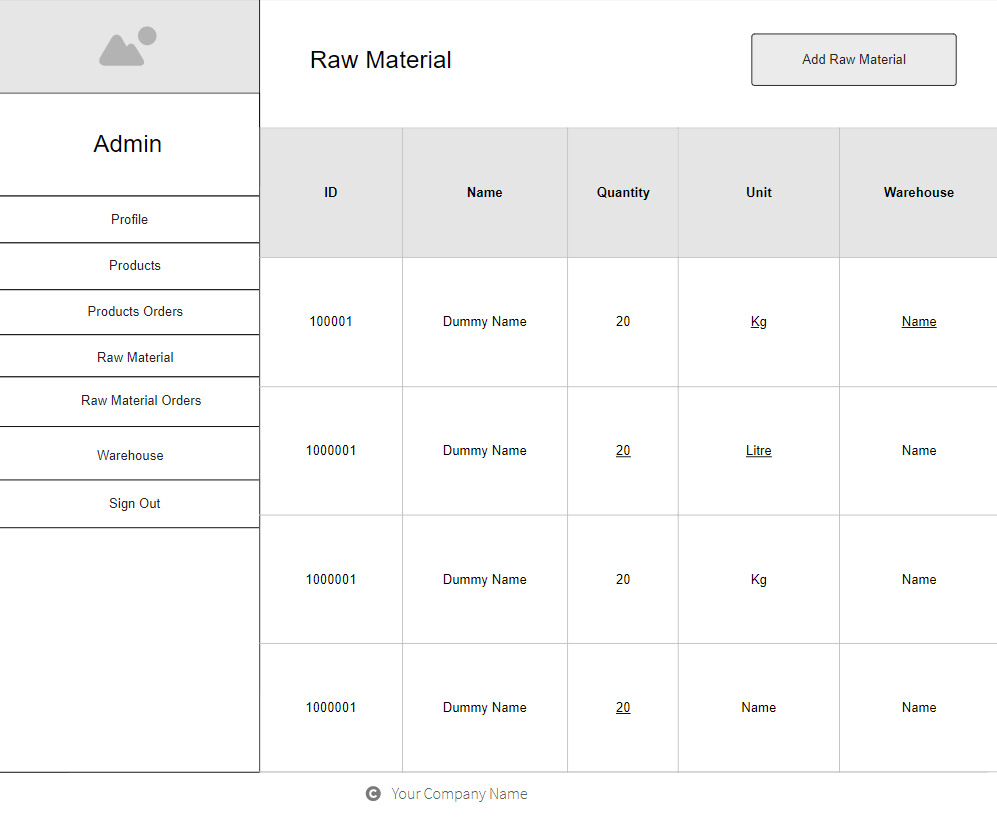
****

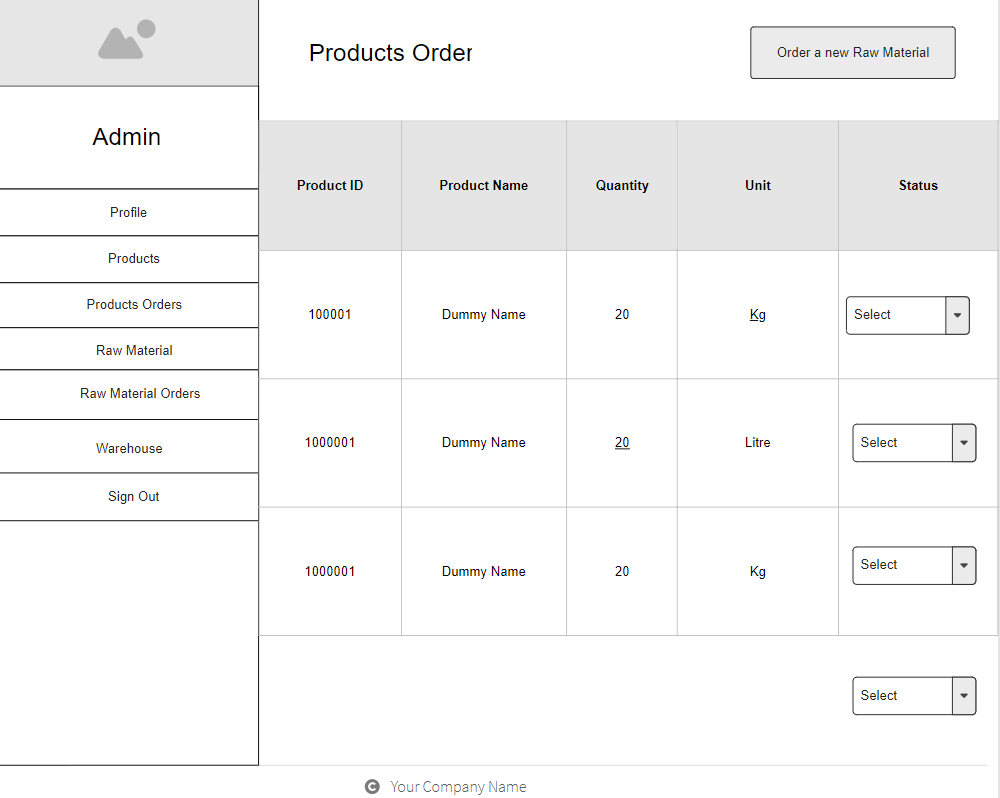
****

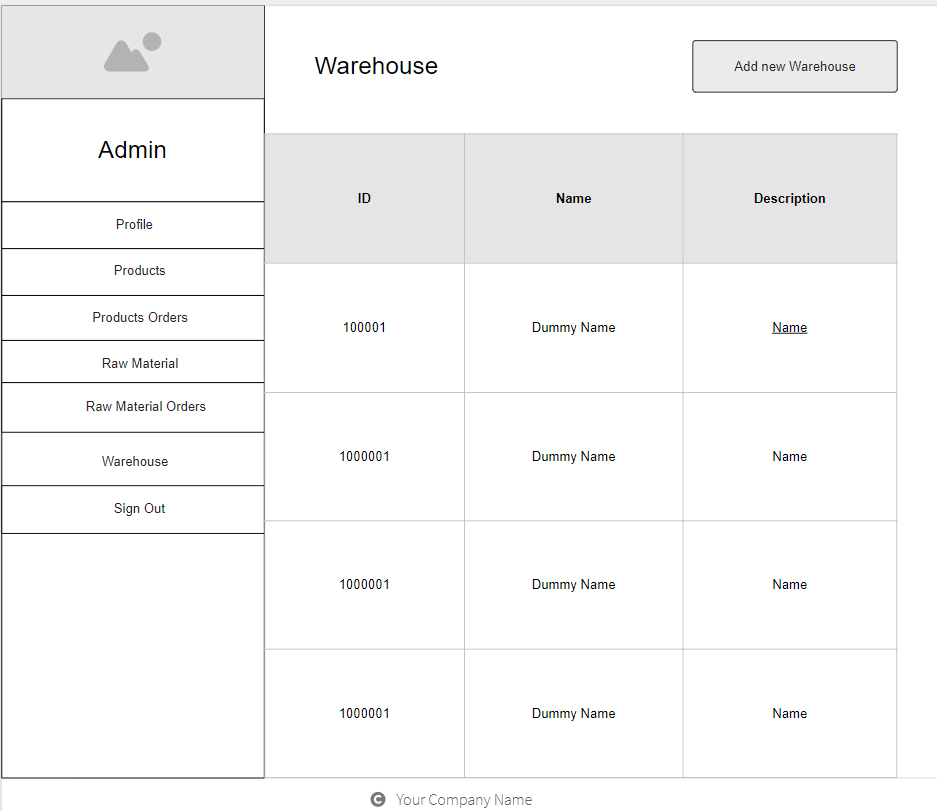
****

****

****

****

****

****

9. Deployment

**Client**

* FrontEnd Angular code is deployed at AWS S3

**Server**

* Backend Code is Deployed in Google Kubernetes Engine
* VM Specs - 4 CPU, 8GB RAM
* Flow - Load Balancer -> Gateway -> Microservices Pods

**CI/CD**

* Continuous integration is performed using **GitHub**
* **Github Actions** are used for Code Quality Review
* GH Pages Workflow to build, test code and Create a Docker image and push to **Docker** Hub on push to dev branch
* Roll Out deployment in Kubernetes
* GH Pages workflow to build static Angular code and **Deploy on S3**
* GH Page workflow to build Angular Code Docker image using (Nginx) and push to docker hub
* **AWS RDS** (MySQL) Database used as Database

Deployment Diagram

10. System Requirements

Below is a list of the minimum Hardware and Software requirements

**Operating System:**

* Windows 7 and above.
* Mac OSX 10.8, 10.9, 10.10 or 10.11
* Any OS that supports Chrome browser

**Hardware:**

* Processor (CPU) with 2 gigahertz (GHz) frequency or above
* A minimum of 4 GB of RAM
* Monitor Resolution 1024 X 768 or higher (For better view)
* A minimum of 5 GB of available space on the hard disk
* Internet Connection Broadband (high-speed) Internet connection with a speed of 2 Mbps or higher
* Keyboard and a Mouse or some other compatible pointing device

**Browsers:**

* Chrome\* 58+
* Microsoft Edge\* 20+
* Mozilla Firefox 40+
* Internet Explorer 11+ (Windows only)
* Safari 6+ (MacOS only)*Users using unsupported browsers may experience issues.*

**Browser Configuration:**

Your browser must be configured as follows:

* JavaScript must be enabled
* CORS must be configured properly
* Cookies must be enabled.
* Pop-up windows must be enabled.

**Software:**

* Java — to view and interact with all available blackboard applications.
* Spring Tool Suite — STS workbench was used to run JDK (write, compile and run the code).
* Visual Studio Code — for writing codes for frontend using Angular, VS Code was used as a workbench.
* Apache Tomcat — it was used as a server for hosting the website.

**DevOps Tools:**

* GitHub – Collaboration for Project development
* Docker – Project is deployed in the form of docker container

**Framework:**

* Swagger2 Docs
* Spring Sleuth for Logging
* Eureka Discovery Server
* Netflix Zuul Gateway