**Application overview:**

This application is designed for back tracing the food products from customer to manufacturer via retailer and distributor. In the scenario members are modeled using Blockchain technology with the following attributes:

In this, data will be updated at four stages from manufacturer to customer. When the customer queries for product details with respect to the invoice and then invoice number will be act as key for retrieving the corresponding retailer data. From the retailer data, distributor-retailer transaction will be figured out and the actual expiry date will be traced. Then, the corresponding manufacturer data will be traced from distributor key and the real manufacturer date will be retrieved and freshness of the product will be evaluated.

| **Stage** | **Data needs to be updated** |
| --- | --- |
| Manfacturer-Distributor | Manfid,Distid,Itemid,Quality |
| Distributor-Retailer | Distid,Manid,Retid,PurchDate,SellDate |
| Retailer-Customer | Retid,Invo,Item,Quantity,Cost |
|  |  |
|  |  |

The demonstration allows a view of the ledger that stores all the interactions that the above stages have has with the respective member. The ledger view shows the parent every transaction that has occurred showing who tried to do what at what time and to which member. The ledger view also allows the user to see transactions that they were involved with as well as showing the interactions with the members e.g. they can see when it was created.

**Application scenario:**

The scenario goes through the product cycle of any item which has the following stages:

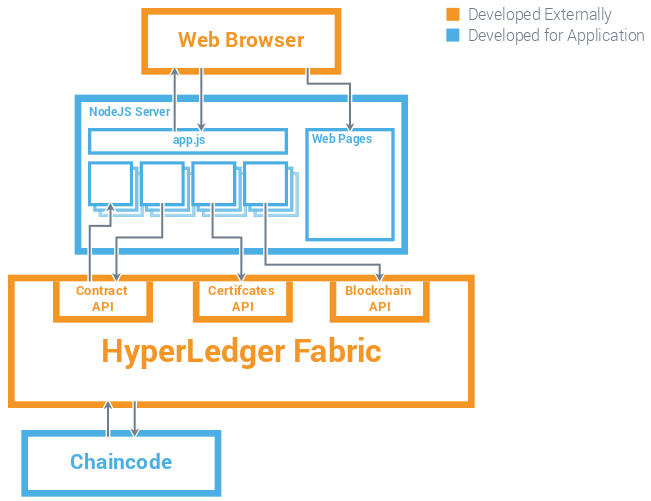
Manufacturer Distributor Retailer Customer

**Stages:**

1. Once the product is moved from Manufacture to Distributor, the corresponding value will be put in blockchain database with key of concatenated string of Manid,Distid,Item,Selldate
2. Then, when the product is reached out to retailer from the distributor, retailer data will be updated with unique key of Retid,Distid,Item,Selldate
3. Finally, once the product is sold, customer level data will be recorded with the key of Invoice no
4. While back tracing the product details from the customer with invoice number key, we can retrieve all levels of data with unique keys at each stage

**Component model:**

The demo will be built using a 3 tier architecture. The user interacts with the demo using a [web front end](https://github.com/IBM-Blockchain/car-lease-demo/blob/master/Documentation/Client%20Side.md) that is provided by the NodeJS server in the middle tier. This web front end uses JavaScript to make HTTP requests to the NodeJS server which has an API ([defined here](Documentation/API Methods.md)) which in turn makes calls via HTTP to the HyperLedger fabric to get details about the blockchain and also interact with the [chaincode](https://github.com/IBM-Blockchain/car-lease-demo/blob/master/Chaincode/src/vehicle_code/vehicles.go).

[](https://github.com/IBM-Blockchain/car-lease-demo/blob/master/Images/Technical_Component_Model.png)

Note: The demo currently does not include the ability to validate the application from user interfaces. But it has the entire flow for retrieving the data from manufacturer, distributor, retailer and customer and evaluation of expiry date and product freshness

The code is developed in go language and is written to handle the above mentioned scenarios.

The URL to the GIThub repository:

|  |  |
| --- | --- |
|  |  |
| [**https://github.com/Salai123/FreshFoodApp/blob/master/FreshFood\_Chaincode.go**](https://github.com/Salai123/FreshFoodApp/blob/master/FreshFood_Chaincode.go) |  |