**Overview**

GlobalMart - Scalability issue, robust backend - code reuse

* regions - North America, Germany, USA, SA

- each region has 100s of shops

- each shop has 100s of items

- Z axis scaling for performance, DB partition can be done based on region

* Service oriented app

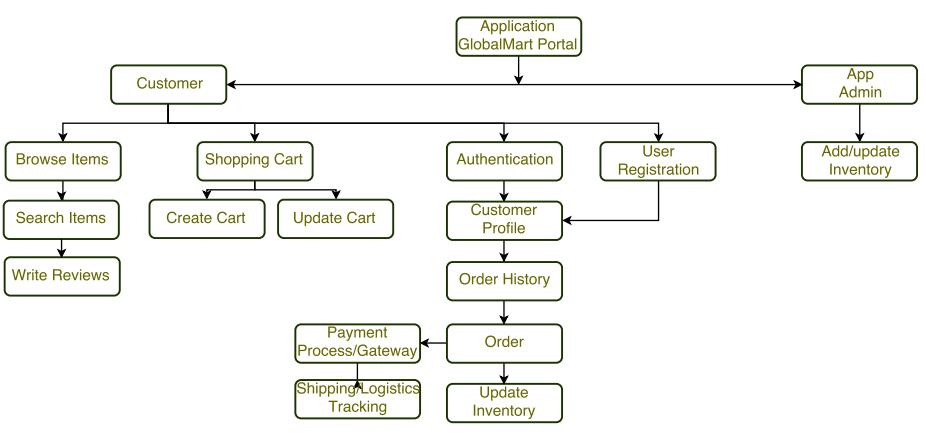
functionalities - inventory management, customer-browse, shop, find store location

- y axis scaling by implementing micro services for each functionality

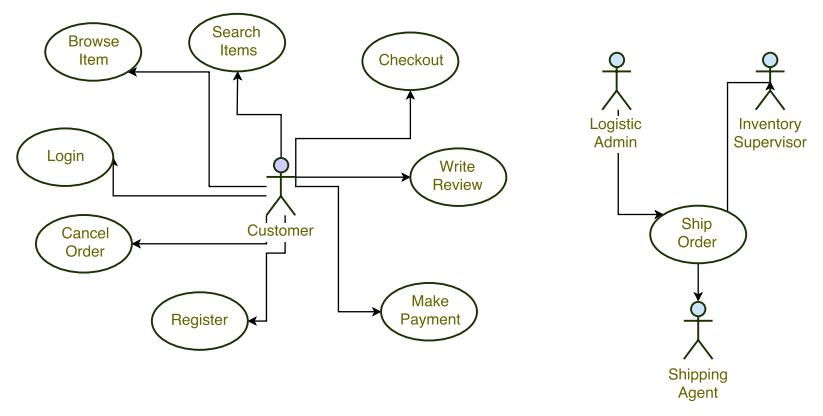
- transactional - spring tx management will be used

* vendor lock in issue

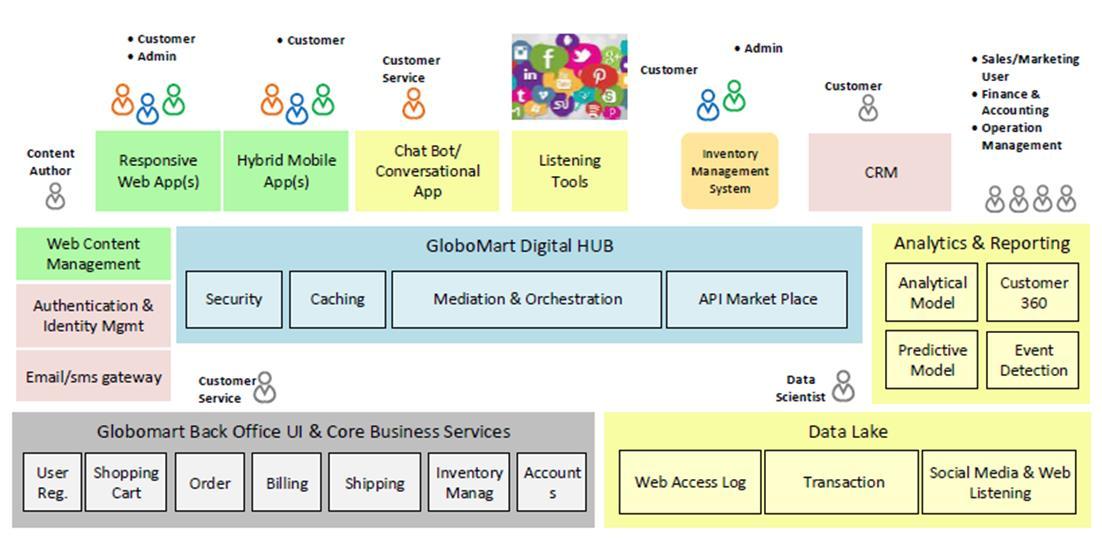
Need to go through each provider policy in detail, and support for data movement and migration. Chose a provider, who has pledged to support emerging industry standards

**Functional decomposition**

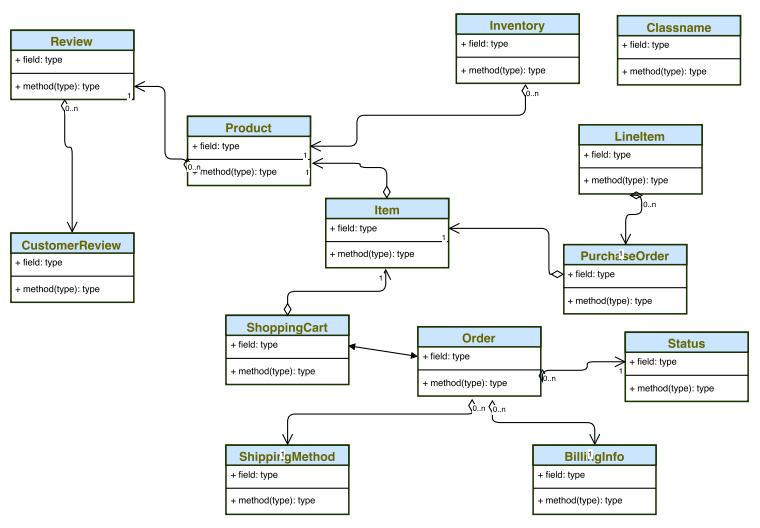
**Use Case Model**

****

**Layered Architecture**

****

**Object Model**

****

**Assumptions**

* Single application instance to serve multiple geo location
* Geo location specific inventory management or database
* Geo location specific logistics integration
* It is not to be a market place for third party sellers. It is only for GloboMart specific inventory
* Geo specific product inventory to store prices in local currency. Payment gateway to handle if payment made in other currency
* AWS Cloud would be preferred choice for implementation and to handle vertical auto scaling
* ASP to be determined for a time period based on no. of unit sold and revenue

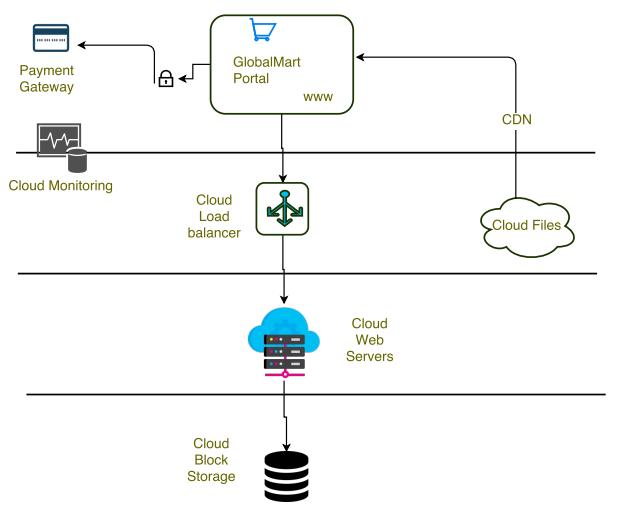
**Content Management**

* Channels (Web, Digital etc.) to be decoupled from the backend platform. All communication to happen via common set of micro-services using restful calls. This will ensure same content versions for all channels
* Updates to catalog/products information should follow standard CMS workflow like Creator  Editor  Publisher etc.

**Business Risks**

* Possibilities of refund and reorder
* Logistic cost overhead on refunds
* Security risk like customer data (PII), price overriding via web proxies may impact the credibility of the business

**Deployment Model**



**Solution to be provided...**

Product Catalogue Service

Microservice that provides the functionality to

* Add a product
* Retrieve the list of products based on simple search criteria e.g. product type
* Remove a product from the catalogue.

**Implemented Solution**

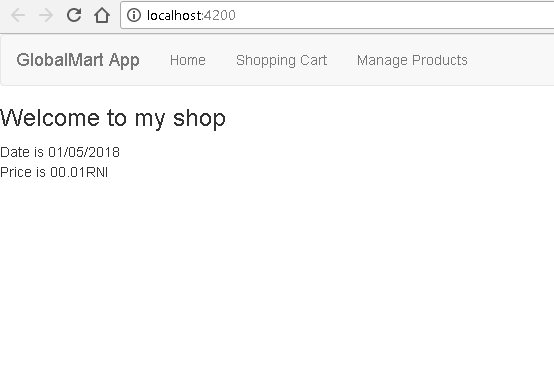
- Checkout the UI app from https://github.com/sanketjavaj2ee/gmc-ngcart-ui and run by executing following commands from project folder location

> **npm install**

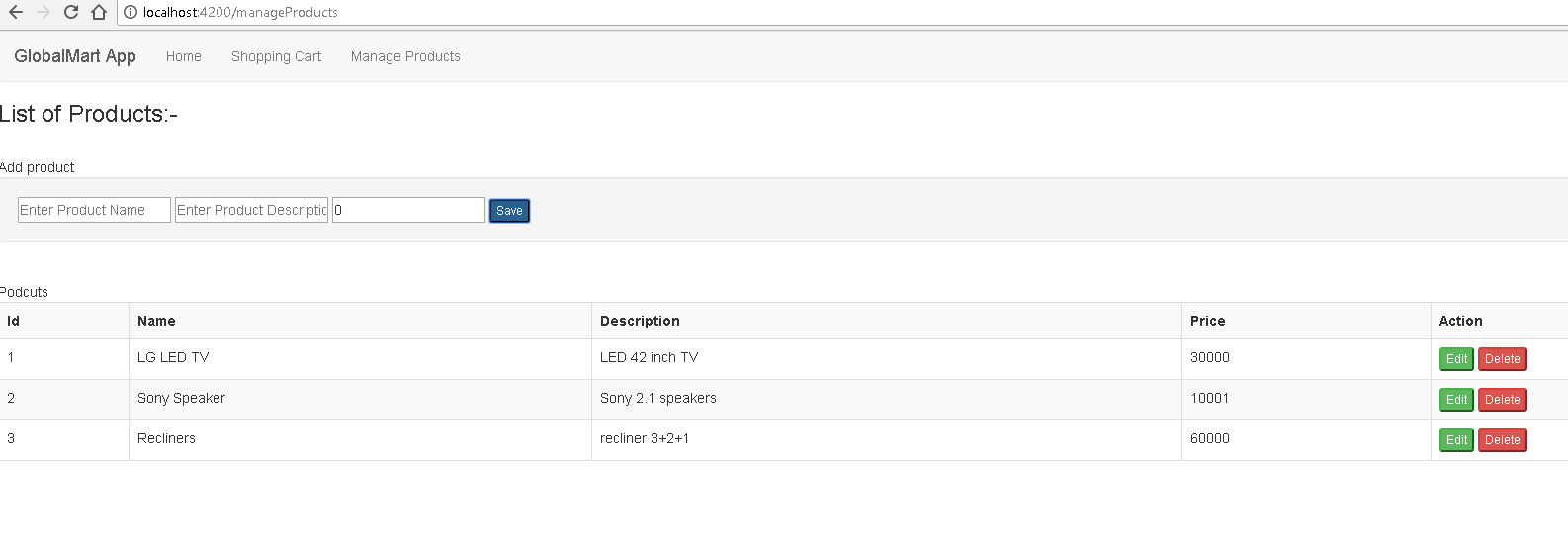
**> ng serve --portno**

- Checkout the rest service server side from https://github.com/sanketjavaj2ee/gmc-cart-restservice, build and run as spring boot application (as of now not included steps to make it a jar, running from sts eclipse only)

**- open browser at localhost:<portno>, following screen should be visible**

****

**Click on manage products link, and can add/remove/update products**

****

**As of now only 3 fields taken to keep this simple**

**issues to be fixed**

- on refresh from /manage producst page the existing products not rendered

**Enhancement Pending**

- Add search product functionality

-- initial plan is to pass the search string and get all products names matching the string, will do it.