## **E- Boutique Shop High Level Requirements**

1. A public facing Shopping site/Cart website with below features:
   * Customers can browse through products by category or by searching
   * Customers can place orders using CreditCard or Cash On Delivery
   * Customers can login and view order history
   * Customers will receive confirmation emails when order is placed or order status changed
2. An Administration website which has the following features:
   * Ability to manage Categories and Products
   * Ability to create one or more users for Administration website
   * Ability to restrict access to certain features based on roles
   * Ability to view customer orders and update the status
3. A mobile (Android/iOS) application for Shoppinng site/Cart
   * Not an immediate requirement, but in future I'll work on this polepole.

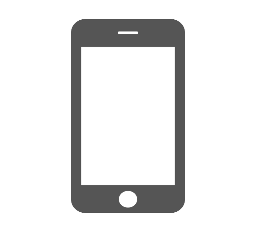
### Overview of System Architecture

Shopping Site/Cart

Website

Database

Administration website

Android APP

REST API

These are very high-level requirements and design only.

**E-Boutique Shop Requirements Analysis**

For building my Boutique shop application I'll develop two web applications

* one for Shopping Site/Cart- i.e the user site;
* and another one for Administration.

**Shopping Site Requirements**

1. **Home Page:** This page shows list of categories and few products in each category. From this screen customers can click on any Category name to see all the products in that particular category or can add a product to cart.
2. **Category Page:** This page displays all the products in the selected category with pagination. Each product will be displayed along with it an Add To Cart button.
3. **Product Page:** This page displays detailed information of the product along with one or more images of the product and shows Add To Cart button.
4. **Search Results Page:** In the header section I'll provide a search box where customers can search for products. The search results will be displayed with pagination. Each product will be displayed along with an Add To Cart button.
5. **Login/Registration Pages:** Customers should be able to login/register with the system so that they can view their profile and order history.
6. **Cart:** In header section there will be Cart Icon showing the no. of items currently added to cart. When Cart icon is clicked the cart page will be shown with all the Items details. Customers should be able to update product quantities or remove products and the cart totals should be updated automatically.
7. **Billing & Delivery Page:** This page will be displayed once the customer clicked Checkout button in Cart page. If the customer is already registered with system he should login otherwise a registration form will be displayed. Customer should register by providing Email and Password. After successfully login, customer should provide Delivery Address, Payment details like CreditCard number, CVV, Expiry Date etc.
8. **Order Confirmation:** Once the customer provided all the valid details and clicked on Submit button an Order should be placed and should display the Order Confirmation page with details including the Products, Delivery Address etc. An order confirmation email should be sent to customer with estimated delivery time(to be determine later)
9. **My Account Page:** Customer should be able to login into the system and see his profile details, order history, outstanding order details.

**Administration Site Requirements**

**Security:** I will provide Roles and Permission based security for the Administration website.  
**Privilege Management:** Ability to configure and manage all the privileges within the system. Each privilege represent the ability to perform an action.  
For Example: **MANAGE\_CATEGORIES** represents the ability to create new Categories, update existing categories.

**Role Management:** Ability to create various roles with one or more assigned privileges.  
For ex:  
**ROLE\_SUPER\_ADMIN** – Users with this role can do anything within the system.  
**ROLE\_ADMIN** – Users with this role can create other users (except other SUPER\_ADMIN or ADMIN), and can do Category Management, Product Management etc.  
**ROLE\_CMS\_ADMIN**: Users with this role can manage Categories, Products etc.

**Category Management:** A category represent a logical grouping of related products.  
Category management includes:

* Create new categories
* Update existing categories
* Disable/enable categories

**Product Management:**In the system product can be anything normally sold in a boutique shop e.g Shoes and diffrnt cataehgories of clothes.  
Product management includes

* Create new products
* Update existing products
* Delete products

**Order Management:**Authorised users can see the list of order, update status or cancel orders.  
If an order is cancelled then refund should be triggered and a cancellation email should be sent to customer.

**Customer Management:**Admin users should be able to view the customer details like their addresses, email ids etc for contacting them.

* List all the customers
* Search customers by email or name
* View customer profile
* Customer order history

## **Development Tools**

Tools I'll be using in developing Online Boutique Shop will include:

1. **Version Control System**:  - Git VCS and i will continiously host my code on BitBucket and Github. Collaboration and task management: Ill use Trello and Slack or (\*\*\*----KabanBoard and Confuluence----\*\*\*)
2. **Build Tool:** I understand that Gradle especially is so hot these days on mobile base apps but I am happy with Maven. There4 i'll use apache maven for my application.
3. **IDE:** As I will be using Maven which is IDE agnostic I can use any of my favourite IDE i.e Netbeans for db modeling, STS for spring initialization and spring technologies, Intellij for any other general coding
4. **Server:** I will use Tomcat as embedded server but ill also implement Glassfish server support later for redundancy purposes
5. **Database:** I will use MySQL and Oracle. But ideally the app shoul be able to use other databases like Postgres and MariaDB or even InMemory database like H2 or HSQL.
6. **Tools:**Ill use Jenkins for CI, and SonarQube for code quality checking.
7. **Production:** When deploying my application I will chose between Oracle WebLogic or Glassfish as production servers and Apache or Ngnix as my load balancer. I wiill also try to to use some application monitoring tools like Zabbix

NB: I'll not use any container technologies like Docker bcoz my app is going to run on only one application server.

## **Application Release Plan**

This app develpment will follow the interation model. I will plan for #N interations where in each i will implement some usecases.

**Iteration#1:**

* Initial code setup for Admin webapp
* Domain Modelling and Database Designing
* Creating JPA Entities
* Email Service SetUp
* Admin UI Layout SetUp
* Admin Login
* Admin Forgot Password
* Admin Reset Password
* Configuring HTTPS/SSL/TLS
* Setting up Jenkins/SonarQube ------I'll skip Jenkins for CI until the app is complete

My Domain JPA Entities

1. Product
2. Category
3. Customer
4. Order
5. OrderItem
6. Cart
7. Address
8. User
9. Role
10. Permission

Important reference

Database – DataSources

* <https://docs.spring.io/spring-boot/docs/current/reference/html/boot-features-sql.html>
* <https://docs.spring.io/spring-boot/docs/current/reference/html/howto-data-access.html#howto-configure-a-datasource>
* <https://docs.spring.io/spring-boot/docs/current/reference/html/howto-database-initialization.html>
* <http://www.thymeleaf.org/doc/articles/springmail.html>

Added Jenkins for Continous Code Integrations

No direct polling from localhost becoz of the reason explain on the link below

<https://stackoverflow.com/questions/30576881/jenkins-build-when-a-change-is-pushed-to-github-option-is-not-working?answertab=active#tab-top>

Added Sonar Qube for Code Quality Check

<https://docs.sonarqube.org/display/SONAR/Concepts>

**Iteration #2**

Create service classes

Services.

Add product

Delete product

List products

Add to cart

Calculate total

Bill customer

Update orders, update payments

View orders

List products at the homepage

List product

Add items to cart