**Shop Management System**

**Detailed Design document created for Deutsche Bank**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Details** |
| 1.0 | 10-June-17 | Malay Desai | Initial version  Created Shop Management APIs |

1. **Project Overview**

Create Shop management REST APIs with the help of Java Spring Boot (<https://spring.io/guides/gs/spring-boot/>)

* 1. **Objective**

The Objective of this project is to create two Restful web services for shop management.

1. Web service to store shops in the retail management system. This service will be called by Retail manager to add all shops into the system.

It will be a POST call with shop details and the shop will be stored in memory.

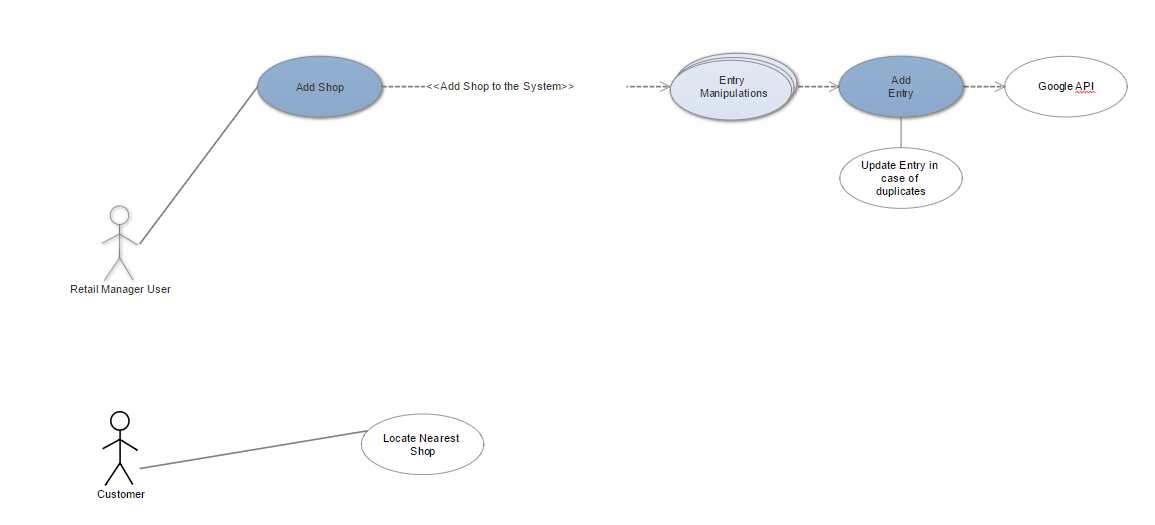
1. Web service GET call to locate closest shop.

This service will be used by customers to locate closest shop from their current position

* 1. **Risks & Out of Scope items**

Below are the risks and open items identified with the system.

1. Multiple users can use the application at same point of time and concurrency should be taken care of.
2. At the time of retrieval, the list may not be up-to-date with all changes due to concurrent modifications
3. No security aspects taken care of in this project
4. Objects are stored in memory and not in Database, all changes will be lost once the application is restarted.
5. The project is using developer APP\_KEY for Google API. It should be used for test only purposes and not suitable for deployment to higher environments.
6. Restrictions of max number of concurrent calls and max number of calls per day applicable as per Google policy.
7. **Technical Design Diagrams**
   1. Usecase diagram



1. Details of Service Design
   1. Add Shops

This is a POST call to store shops in system.

The shops will be added to a static HashMap with key shopName.

If Shop already exists, the shop will be updated with new details.

Google Maps API will be called to locate the shop using Address.

The Geocoding API will return latitude and longitude of the shop, the same will be stored in the HashMap as shop address.

Java Client for Google Map services is used to access Google Maps API.

The Developer Key is hard coded in the project for testing purpose.

**Request**:

URL : localhost:8080/shops

POST call with below JSON

{

"shopName": "Tesco Express5",

"shopAddress": {

"number" : "2",

"postCode" : "LU2"

}

}

**Response**:

If new shop:

{

"result": " New shop successfully added to the system",

"shop": null

}

If Duplicate :

{

"result": "This shop already exists in the system, data replaced by new values. Old shop details shown here",

"shop": {

"shopName": "Tesco Express5",

"shopAddress": {

"number": "2",

"postCode": "LU2",

"latitude": 0,

"longitude": 0

}

}

}

* 1. GetShops

Customers will use this REST API to retrieve a shop closest to their location.

The API takes latitude and longitude of customer and calculates distance from all shops.

The closest shop is picked and returned.

If there are no shops in the system, an error message will be returned.

Request URL : localhost:8080/shops?latitude=2&longitude=2

Response:

{

"result": "Below is closest shop from your location",

"shop": {

"shopName": "Tesco Express5",

"shopAddress": {

"number": "2",

"postCode": "LU2",

"latitude": 15.34343,

"longitude": 10.3434

}

}

}