**Objectives:**

The objectives for this increment are to be able to collect and store required data set in database, Implement REST service so that the client application can be able to access and update data, to be able to use REST services and represent the obtained data in the client application, a customer should be able to select items and add them to cart, use local database (SQLite) to store the cart items, giving customer the facility to delete and modify items in the cart.

**Import Existing Services/API:**

For this increment, we haven’t used any existing services. But while developing the client application we referred the following links and sample codes to get to know how to setup action bar, consuming REST services in client application, parsing JSON objects , how to use local database to store items from cart and few small concepts like arrayadapter, listview, toasts etc. Coming to API, we have developed our own API which provides data like prices and description of Fruits, Vegetables, Fastfoods and Restaurant menu.

<https://developer.android.com/training/basics/actionbar/setting-up.html>

<http://www.tutorialspoint.com/android/android_json_parser.htm>

<http://www.androidhive.info/2012/01/android-json-parsing-tutorial/>

<http://developer.android.com/guide/topics/ui/layout/listview.html>

<http://developer.android.com/reference/android/widget/ArrayAdapter.html>

<http://developer.android.com/guide/topics/ui/notifiers/toasts.html>

<http://stackoverflow.com/questions/2265661/how-to-use-arrayadaptermyclass>

<http://developer.android.com/training/basics/data-storage/databases.html>

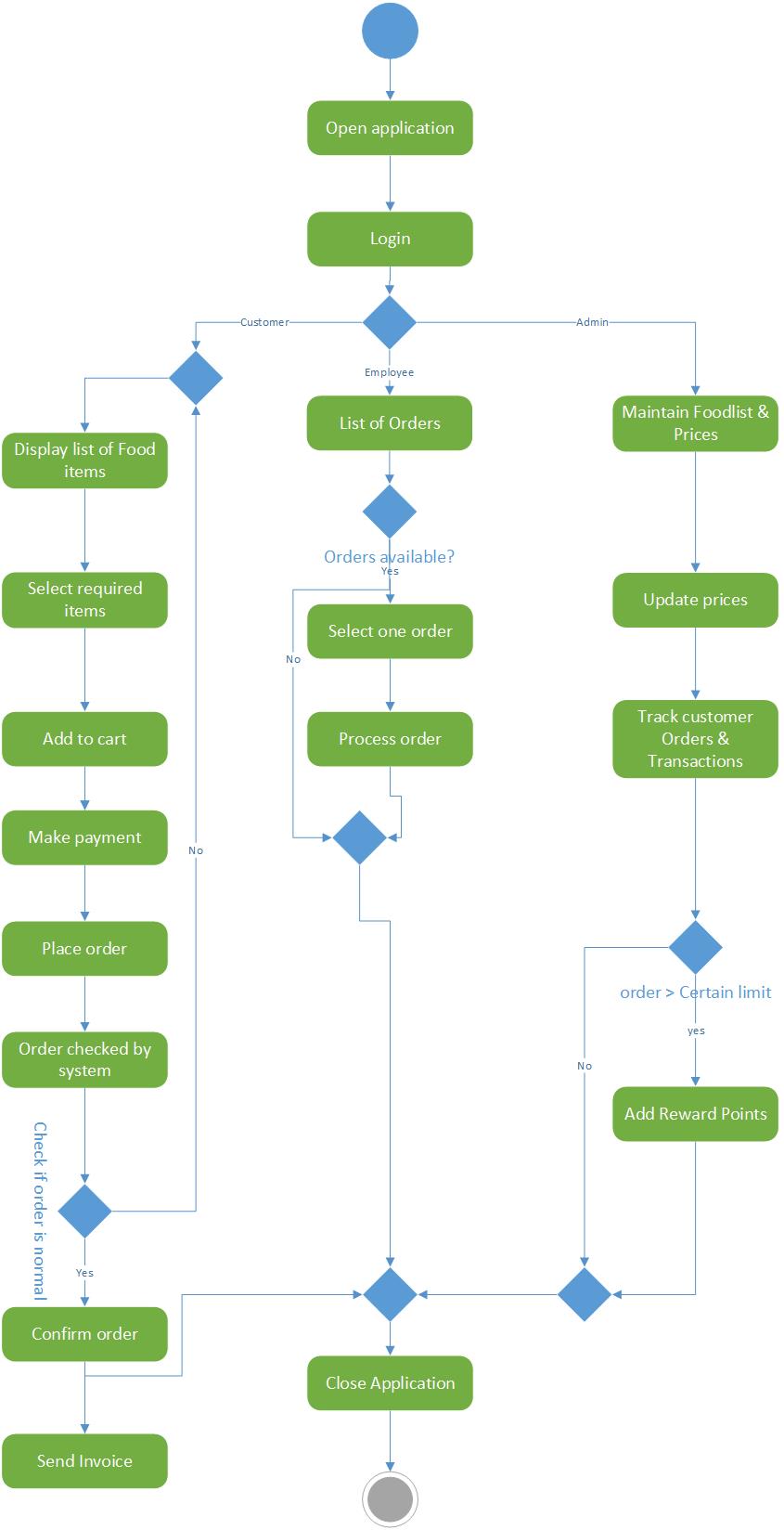
<http://www.tutorialspoint.com/android/android_sqlite_database.htm>

**Detail Design of Services:**

User Stories:



Activity diagram:



Services:

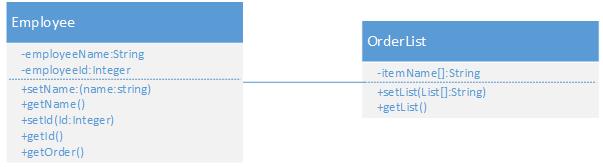
We have developed the registration and login service in first increment. For this increment, we developed the customer service where customer will be able to select items and add them to cart. We kept more effort on obtaining required data, maintaining database, implementing REST services, writing queries and parsing the JSON objects obtained through REST services and representing on mobile client.

So, only the Customer service is covered in this iteration. More of the services required for this project will be developed in 3rd iteration.

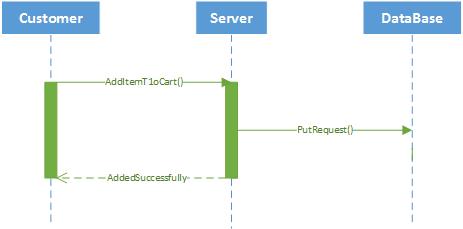
Customer service:

Customer will be able to select required items, add them to cart, modify items in the cart.

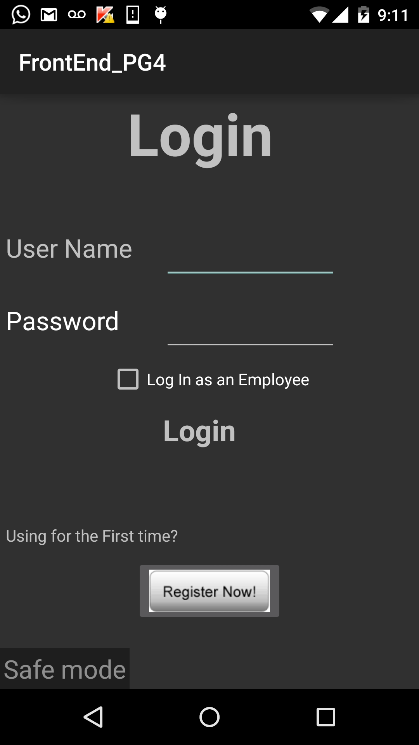
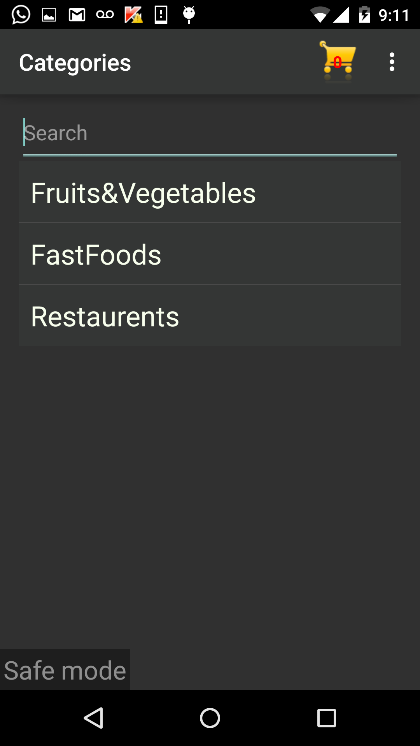
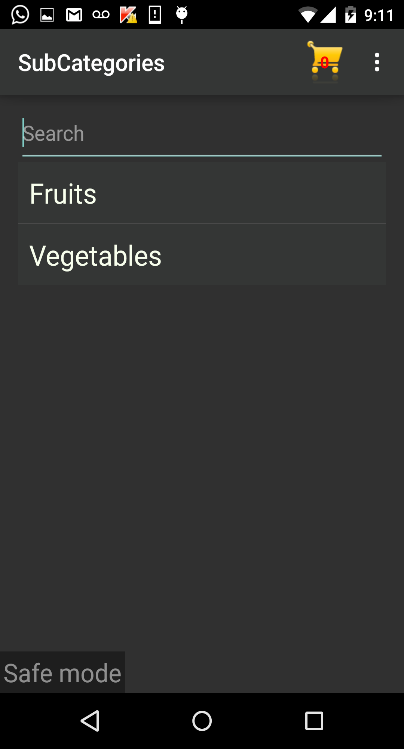
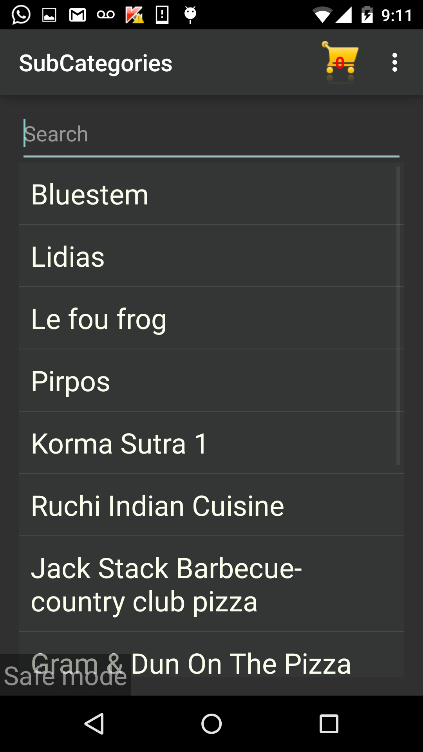
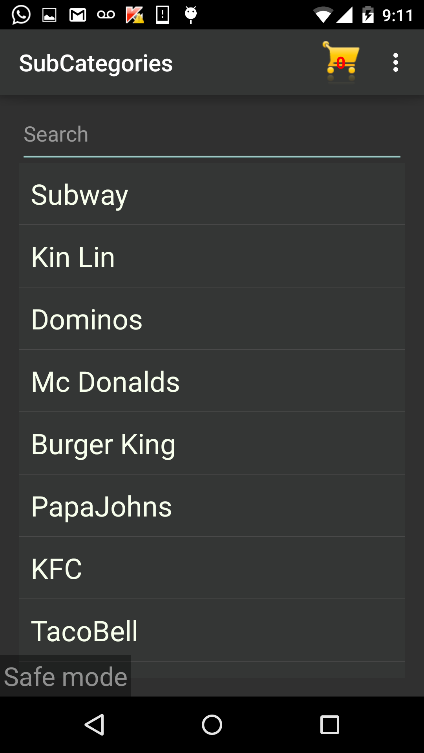
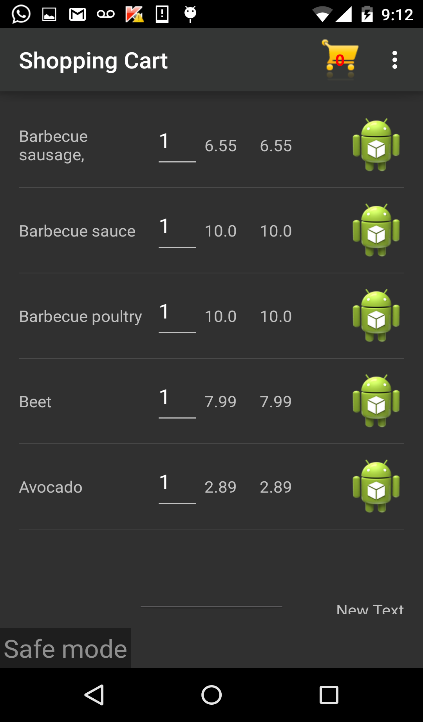
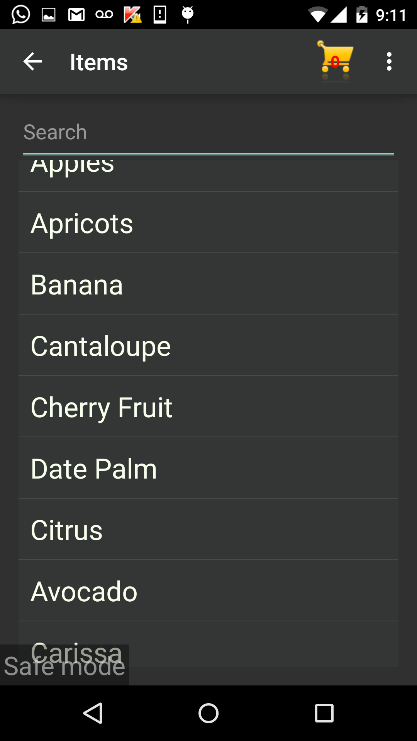
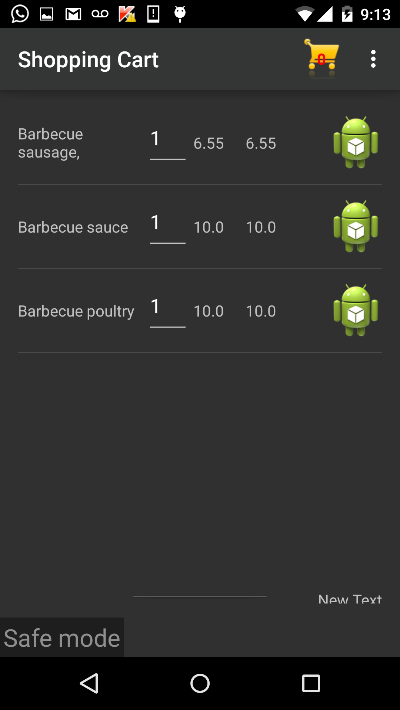
Class diagram:



Sequence diagram:



Design of Mobile Client Interface

**Implementation:**

**Implementation of REST services:**

The REST service we developed provides all the data and information about prices and description of fruits, vegetables, fast foods, restaurant menu etc.

We used MySQL as database for our project. We developed and created REST services using Apache CXF and Spring dependency injection. For mapping Java classes to database tables, we used Hibernate.

Apache CXF is open source and multi featured web services framework. It takes care of all the http requests. Every request from the mobile client goes to CXF servlet.

Spring dependency injection maintains the independency of classes and at the same time gluing them together. This helps while doing unit testing and increases the possibility of reusing the classes.

We divided the backend part into a total of four modules.

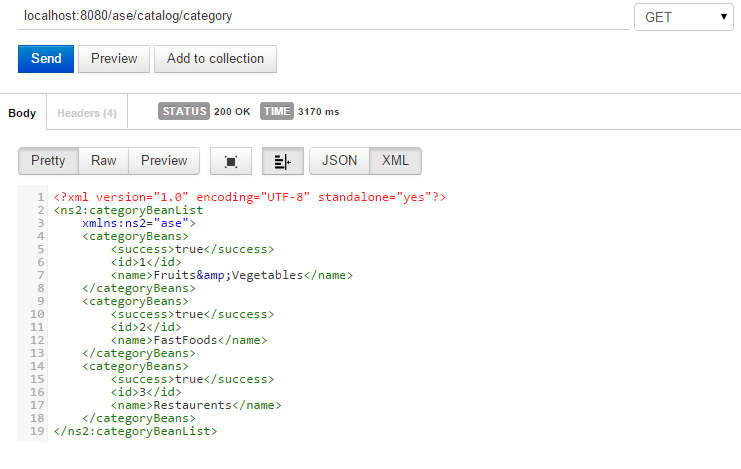
core, handler, service-spec, web

**Testing:**

We tested the REST service using POSTMAN (an extension of Google Chrome)

Testing of customer service:

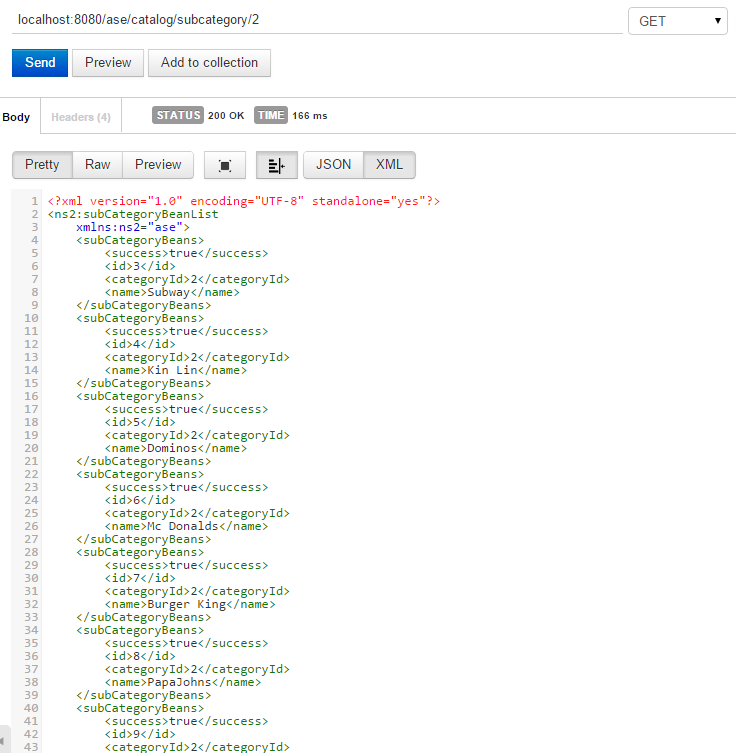
Category Response:



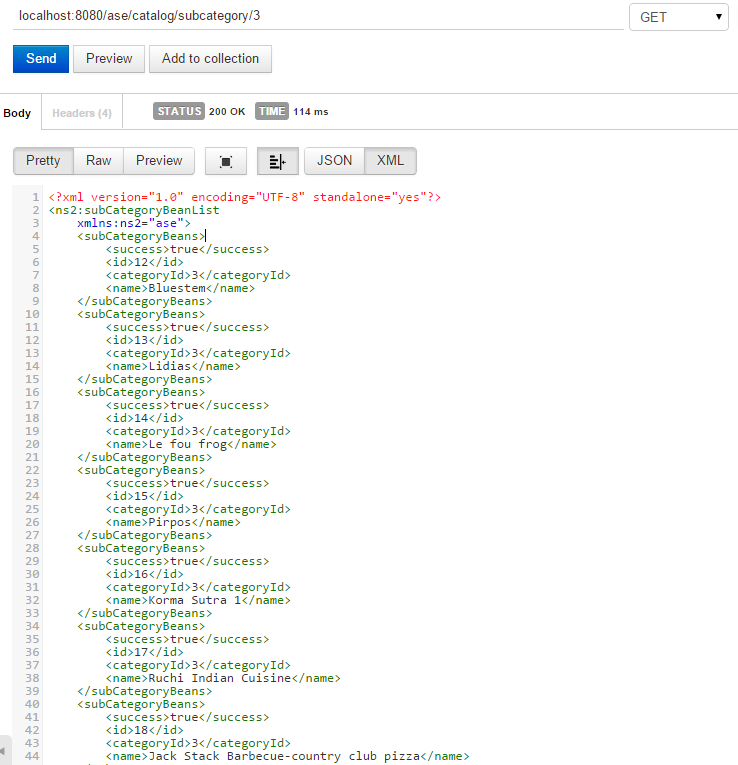
Subcategory Response (category id 1):



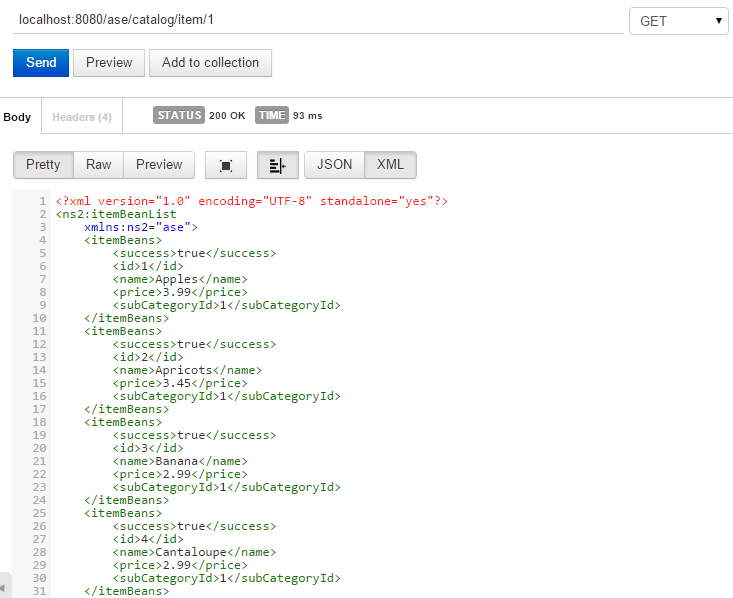
Subcategory Response (category id 2):



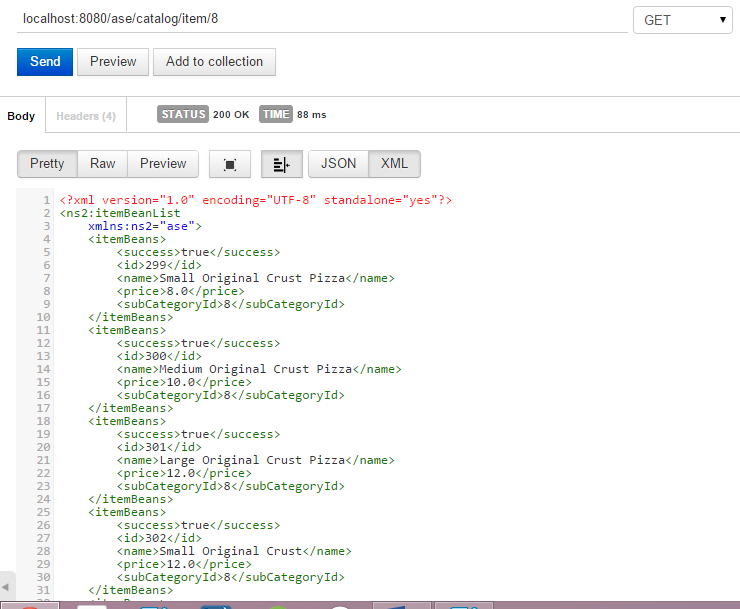
Subcategory Response (category id 3):



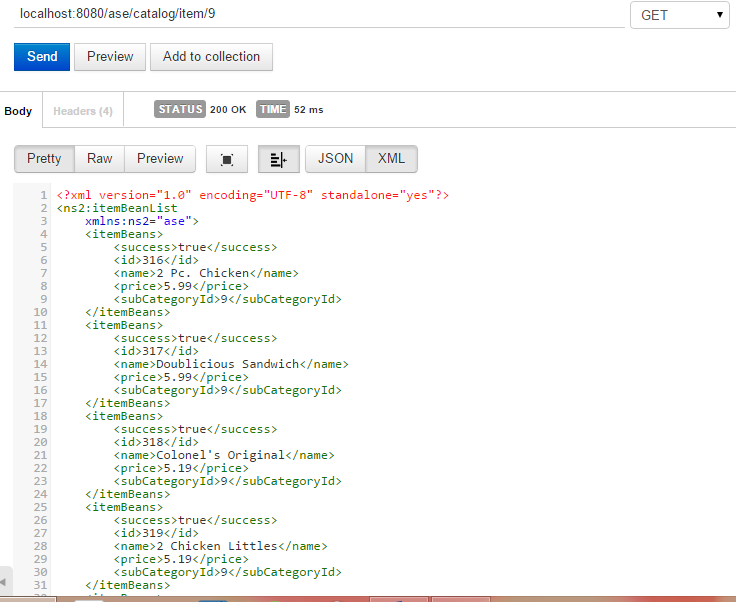
Items Response (subcategory id 1):



Items Response (subcategory id 8):



Items Response (subcategory id 9):



**Deployment:**

ScrumDo link:

<http://www.scrumdo.com/projects/project/alpha8/iteration/121755/board>

UMKC VM:

Since we developed backend in JAVA, we cannot use UMKC VM (only support .NET)

We are using jetty server plugin in the backend project (maven).

localhost:8080/ase/catalog/categories

localhost:8080/ase/catalog/subcategory/1

localhost:8080/ase/catalog/subcategory/2

localhost:8080/ase/catalog/items/1

localhost:8080/ase/login/register?firstname=saiteja&lastname=saranam&email=saiteja.saranam@hotmail.com&password=saiteja123

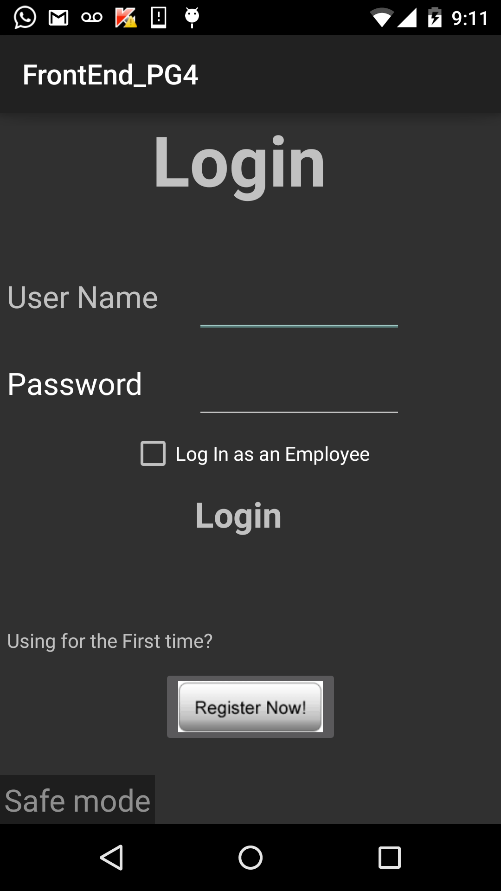
GitHub url:

<https://github.com/saiteja10/ASE_SecondIncrement>

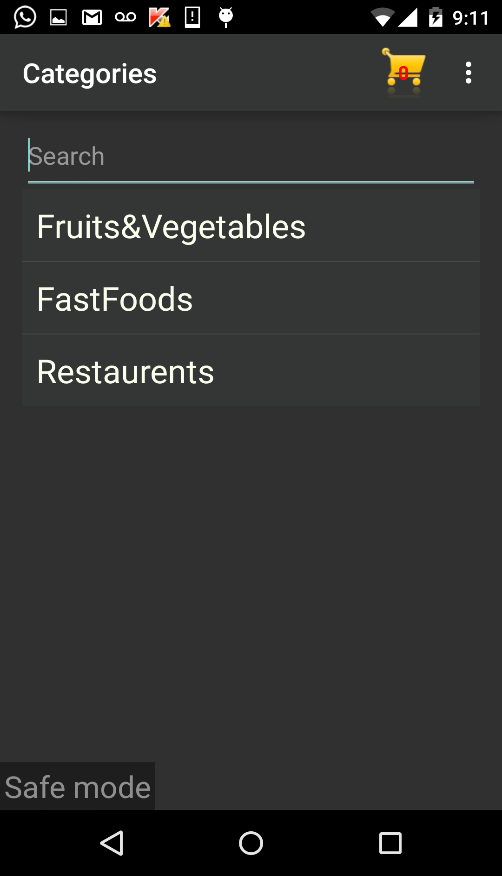
**Report:**

Mobile client Screenshots:

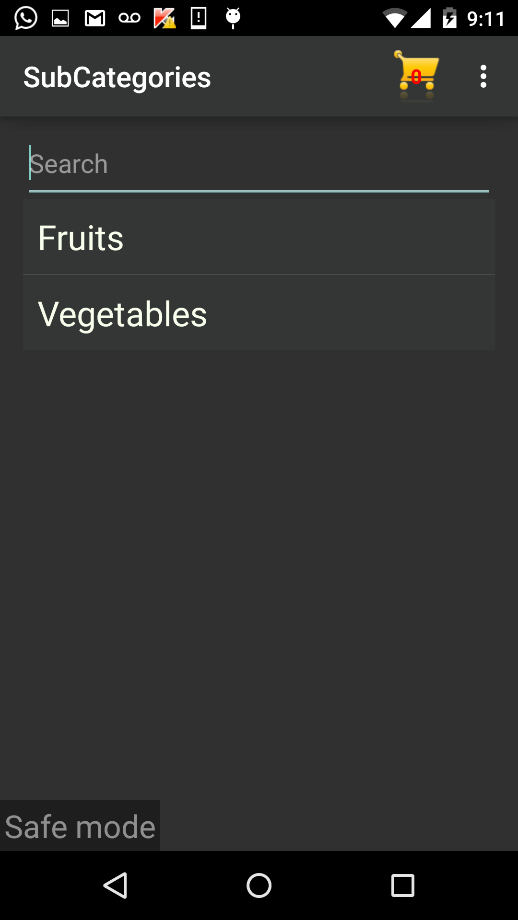
Login page:



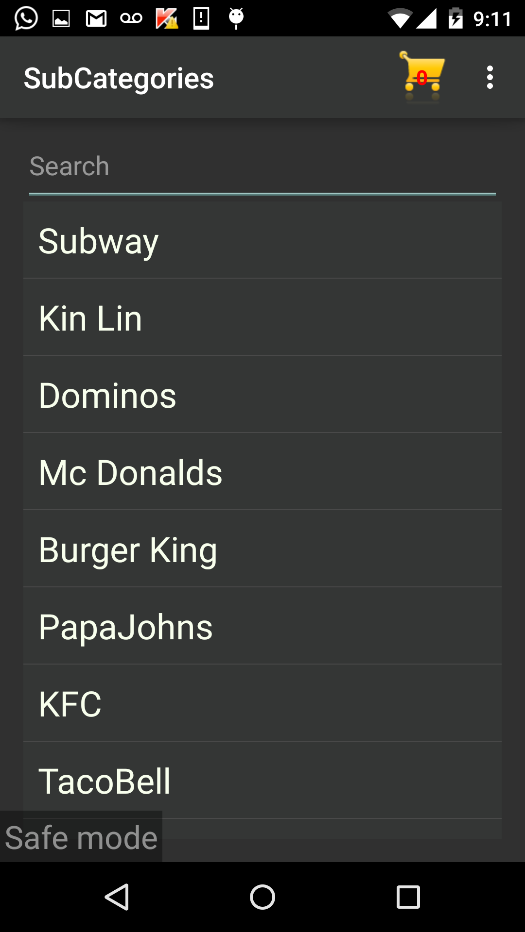
Categories page:



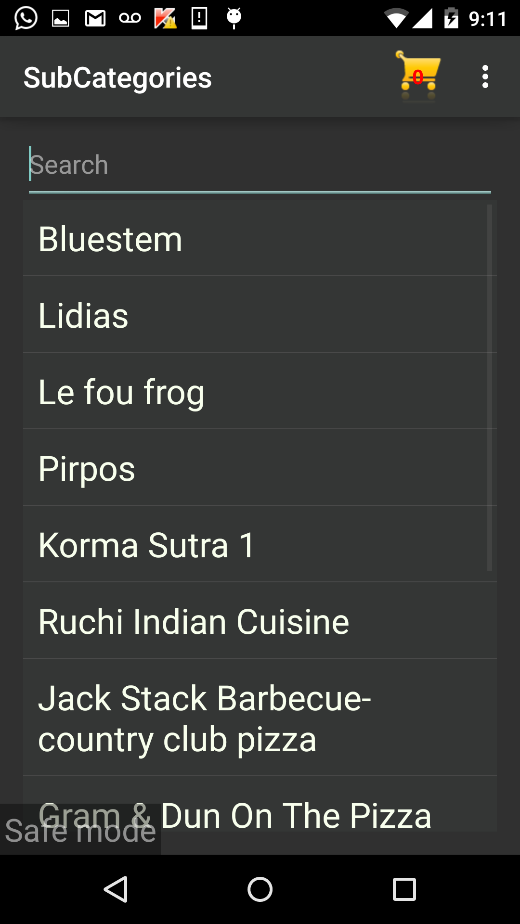
Subcategory page(on tapping fruits and vegetables from category page)



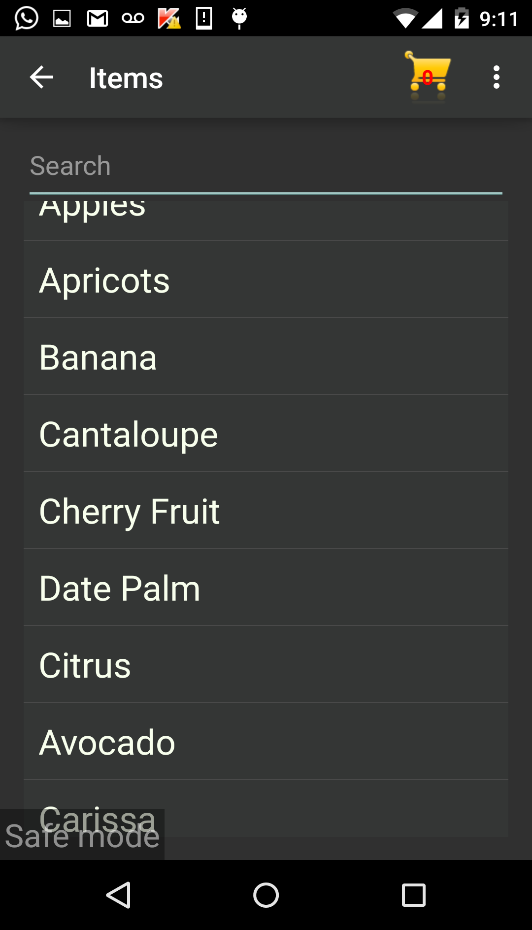
Subcategory page(on tapping Fastfoods from category page)



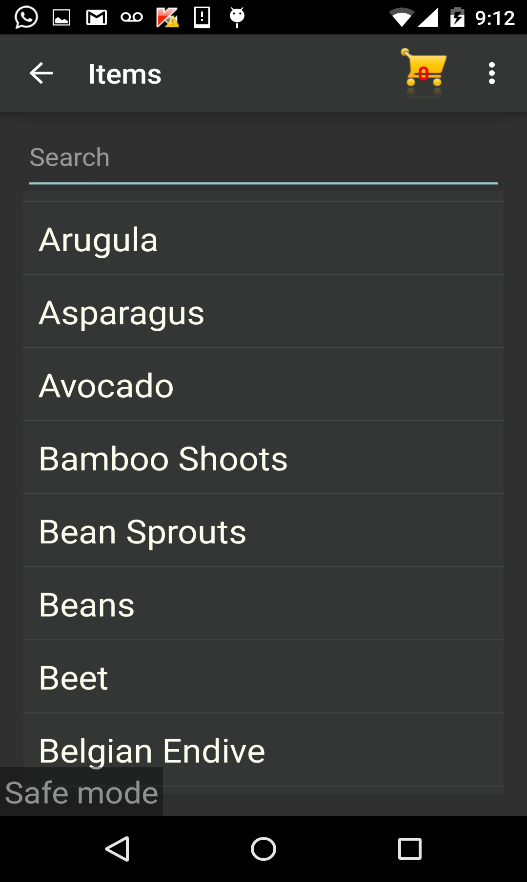
Subcategory page(on tapping Restaurents from category page)



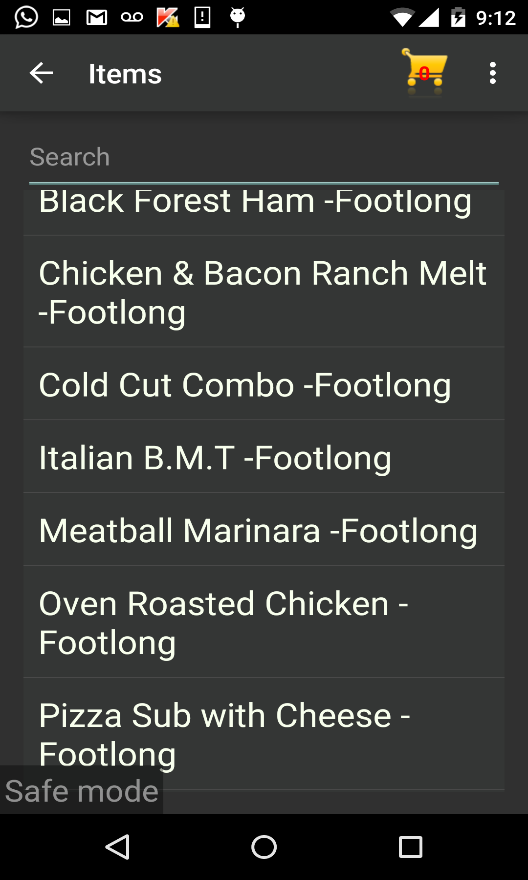
Items page(on tapping fruits from subcategory page)



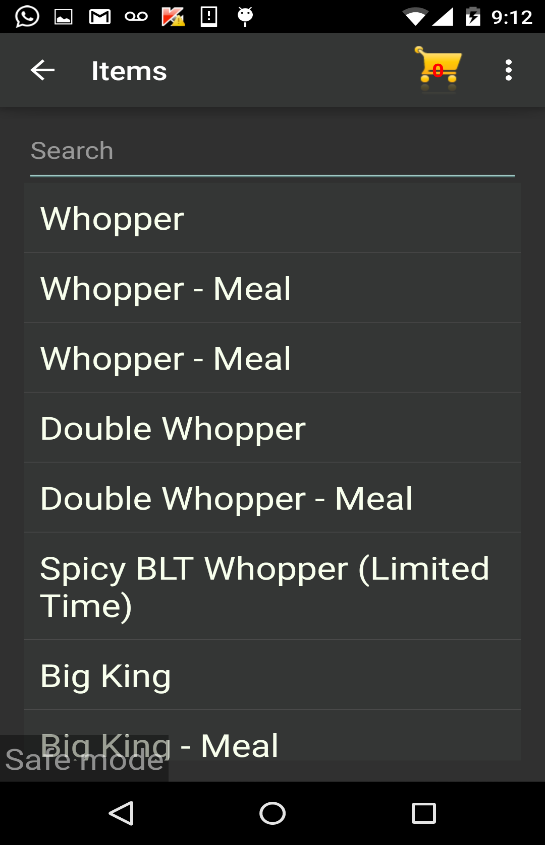
Items page(Vegetables):



Items page(Subway menu):



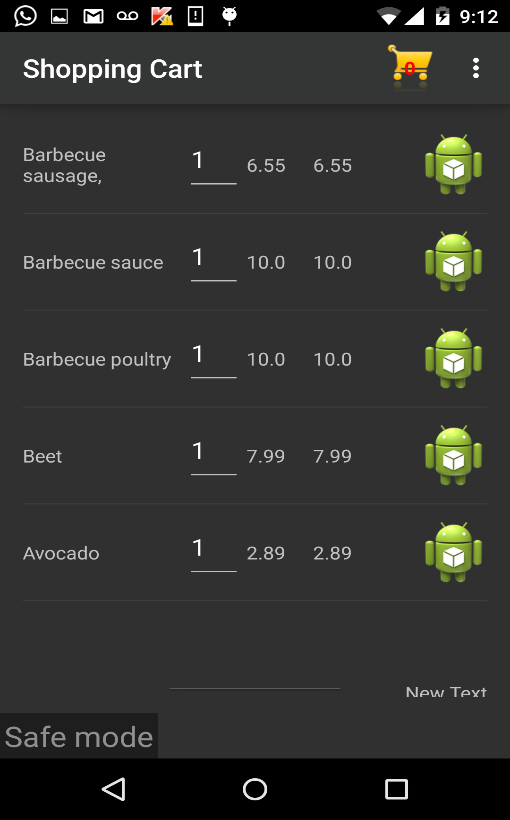
Items page(Burger King menu)

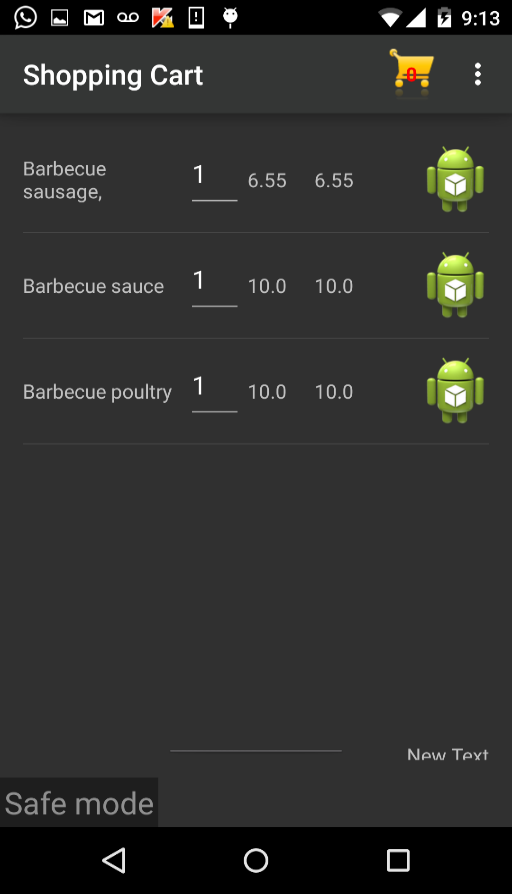


Items page(Ruchi restaurant menu)



Shopping Cart page(The visible android logos are kept to remove items from cart (temporary only)):



On removing few items from cart:  


**Project Management:**

**Scrumdo link:**

<http://www.scrumdo.com/projects/project/alpha8/iteration/121755/board>

**Implementation status report:**

**Work completed:**

In this iteration, we have completed obtaining all the data set required for client application, completed writing query’s, implemented REST services, consume rest services through mobile client, parsing obtained data and represented in client application, used local database (SQLite) to store the items in the cart, gave functionality to edit the cart.

|  |  |
| --- | --- |
| **Name** | **Class id** |
| Sai Teja Saranam | 29 |
| Srikanth Nadendla | 23 |
| Nagender Goud | 31 |
| Ram Nikhil | 35 |

|  |  |
| --- | --- |
| **Task** | **Person** |
| Obtaining required data set | Srikanth Nadendla , Ram Nikhil, Nagender Goud |
| Wrote queries for the newly added tables | Sai Teja Saranam |
| Functionality of mobile client | Sai Teja Saranam |
| Call rest services from android | Sai Teja Saranam |
| Parsing and representing data | Sai Teja Saranam, Srikanth Nadendla |
| Layout of mobile client | Srikanth Nadendla |
| Creation of local database to store items in cart | Sai Teja Saranam, Ram Nikhil |
| Implementation of REST services (Apache CXF) | Sai Teja Saranam |
| Hibernate (mapping Java classes to database tables) | Nagender Goud, Srikanth Nadendla |
| Spring Dependency Injection | Nagender Goud, Ram Nikhil |
| Testing REST services | Nagender Goud, Ram Nikhil, Srikanth Nadendla |
|  |  |
| **Contributions** |  |
| **Member** |  |
| Sai Teja Saranam | **Percentage** |
| Srikanth Nadendla | 30 |
| Nagender Goud | 30 |
| Ram Nikhil | 20 |
|  | 20 |