**Project Plan**

I. **Introduction**

Purchasing food items and items required to prepare food involves going to multiple stores which is a hectic task. The motivation of the project is to create a platform where users can easily order and purchase Vegetables, Fruits and any kind of food items from the nearby stores. The items in the order will be purchased by one of our available employees who is near by the customer’s location and delivers to him in promised time.

**II. Project Goal and Objectives**

* Overall goal:

The goal of this project is to create a platform, where one can order multiple food items from different stores without having to go to stores.

* Specific objectives:

The objective of this application is to reduce the effort and time of the customers in going to different stores for purchasing the food items.

* Significance:

The main significance of this application is that it reduces the wastage of customer’s time. It will be useful for people who works full day in their office and who finds it difficult to find some time for purchasing the Vegetables, Fruits and any other food items. This application comes in very handy to the people who are very new to the location by eliminating their effort of searching for stores. This will be very useful when a person who don’t have any food items at his home orders while he starts at his /her office

**III. Project Background and Related Work**

There is no platform, where one can order items like vegetables, fruits, groceries, Burgers etc., from different stores at the same time. This gave us the idea to develop an application where one can order different items from different stores.

We searched if there are any related applications which does this task. It is fortunate that we haven’t found any such existing applications which gave us the opportunity to develop it.

**IV. Requirement Specification**

* **Functional Requirements:**

1. A registered customer must be able to login
2. User must be able to see list of items
3. User must be able to select required items and add them to cart
4. User should be able to make payment for the items in the cart to complete his order
5. User should be able to track his order
6. Employee must be able to login
7. Employee must be able to see list of open orders
8. Employee must be able to select and accept an order
9. Admin must be able to login
10. Admin should be able to update the items and their prices
11. Admin must be able to track the customer transactions and provide special privileges to the regular customers.

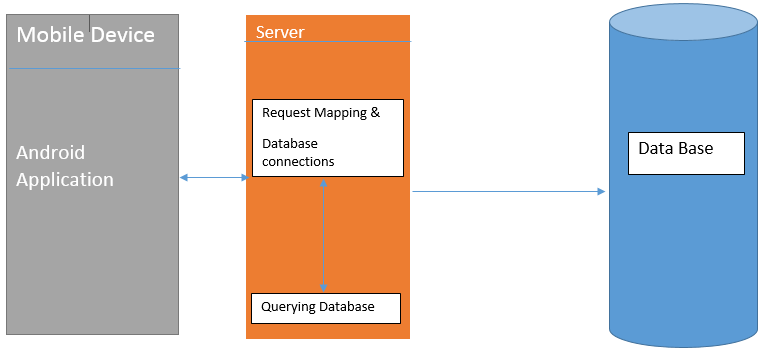
* **Nonfunctional Requirements**

1. Our design should allow us to add additional features in future.
2. Backup of database is required in case of emergencies.
3. Performance of the system should not be hindered by the amount of users.

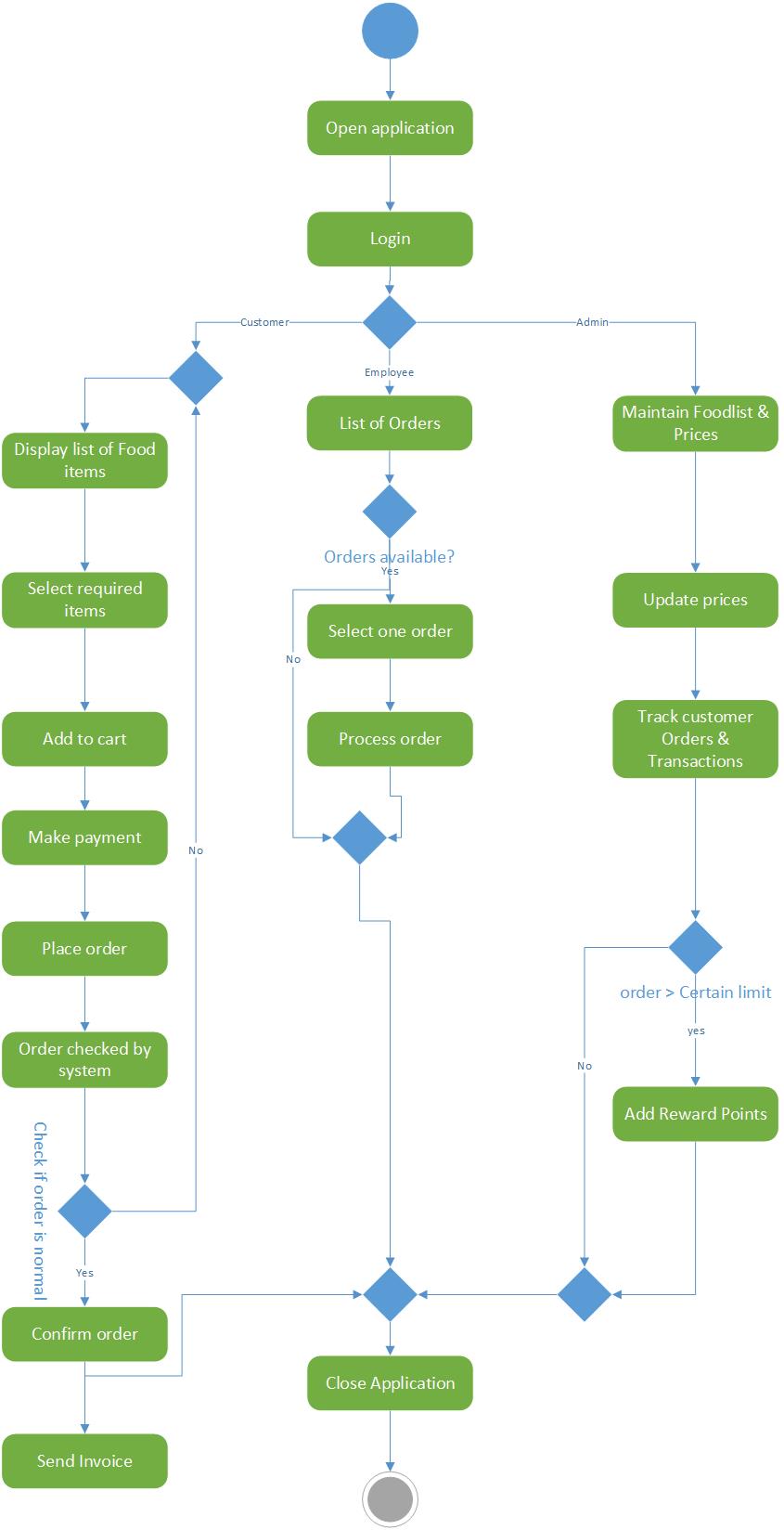
* **Technological and Architectural requirements**

1. API using Apache-CXF ( JAX-RS)
2. Android
3. MySQL
4. Spring Dependency Injection
5. Hibernate framework
6. IntelliJ IDEA ( IDE)
7. JETTY server

**V. System Architecture Diagram:**

****

**UML Activity Diagram**

****

**VI. System Specifications**

**Existing Services:**

* Name**:** Farmers’ Market API

URL: <http://search.ams.usda.gov/FarmersMarkets/v1/data.svc?wsdl>

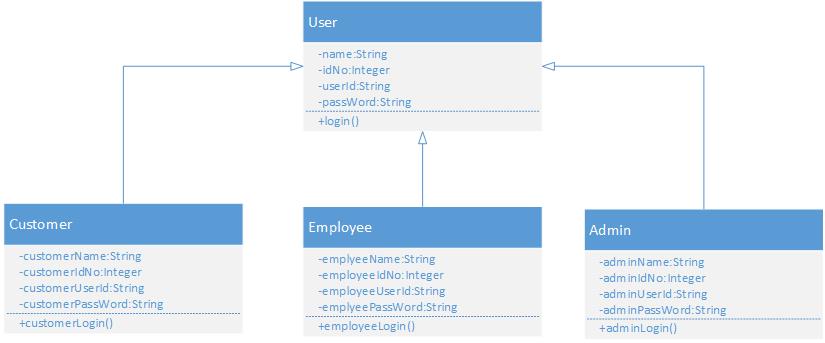
Description:

Used to find the nearby markets, items sold in market and available timings of market.

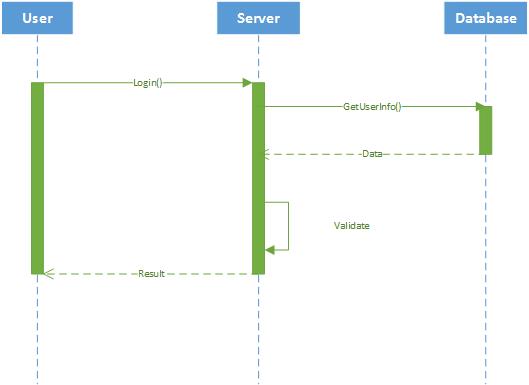
**New Services to be built:**

* **Signup & Login:**

**Class Diagram**

****

**Sequence Diagram:**

****

**Input/output**

The registered users will be able to sign in using the user id and password assigned to him at the time of registration.

Input is his User id and password

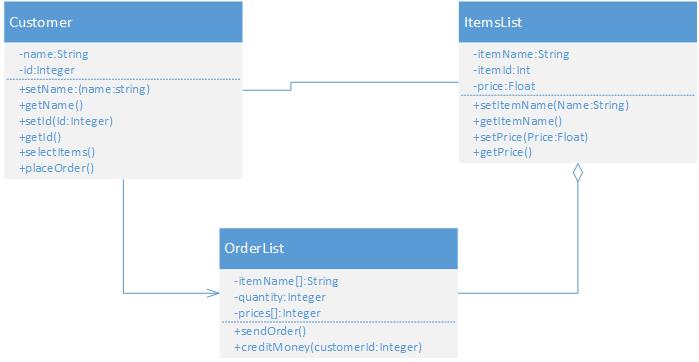
Output is application main page

**Constraints/Exceptions**

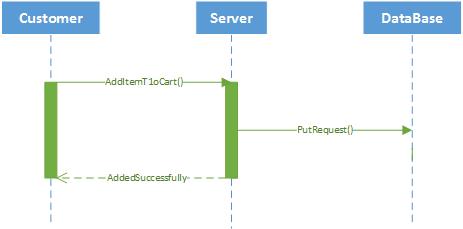
When user enters wrong credentials, he wont be able to log in . An exception will be raised

* **Customer Service**

**Class Diagram**

****

**Sequence Diagram**

****

**Input/ Output**

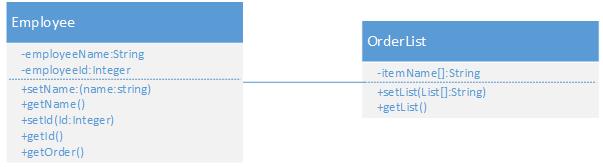
Adds items to the cart as an input and he will be able to see in his orders list

**Operational description**

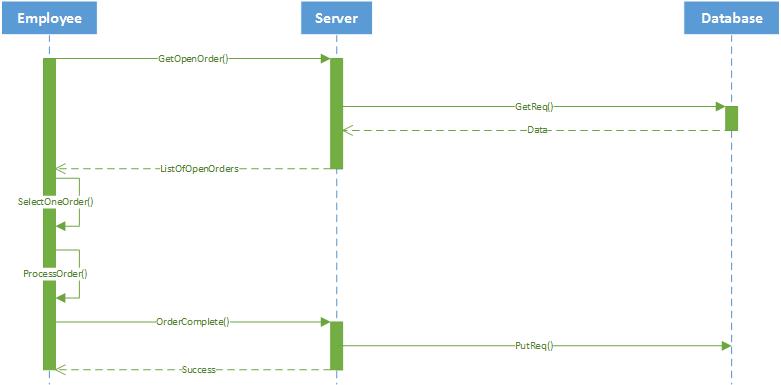
Customer selects and adds items to the cart and makes payment to complete his order.

* **Employee Service**

**Class Diagram**

****

**Sequence Diagram**

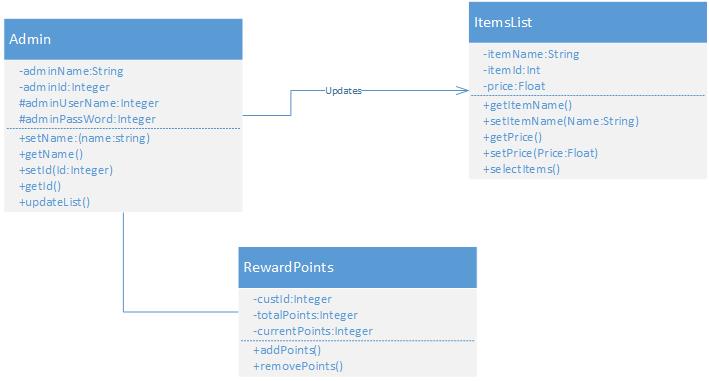


**Operational description**

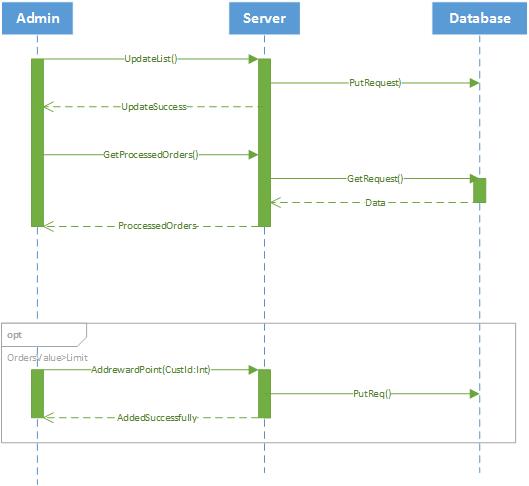
An employee should be able to see list of open orders and select one of orders and processes the order.

* **Admin Service**

**Class Diagram**

****

**Sequence Diagram**

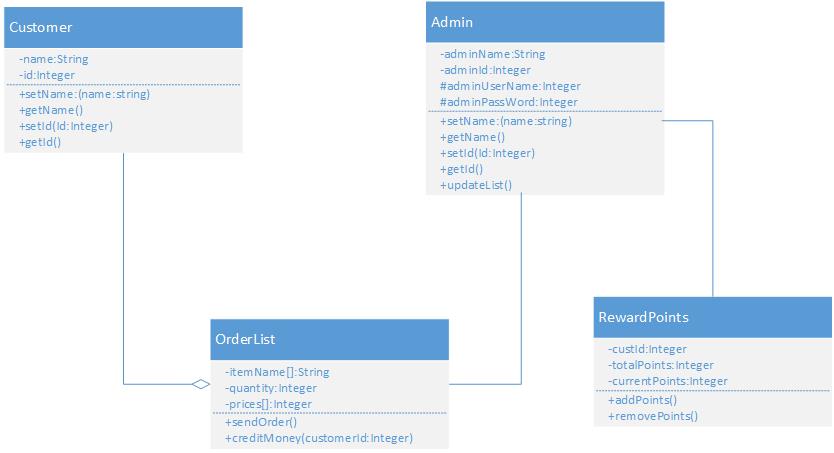


**Operational description:**

Admin should be able to update the prices of the items and track transactions.

* **Loyalty Service**

**Class Diagram**

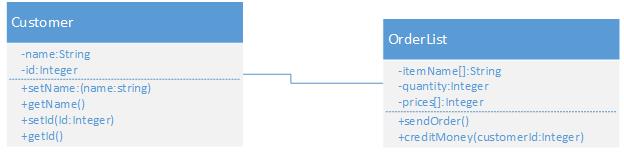
****

**Operational description:**

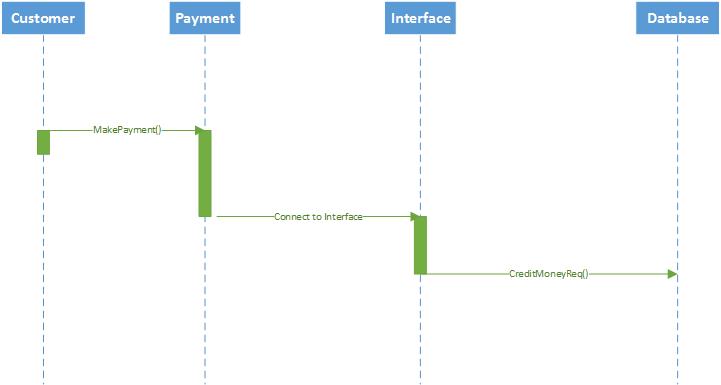
Reward points will be awarded to the customer by admin, when customer order value is more than certain limit.

* **Payment Transaction**

**Class Diagram**

****

**Sequence Diagram**

****

**Operational description:**

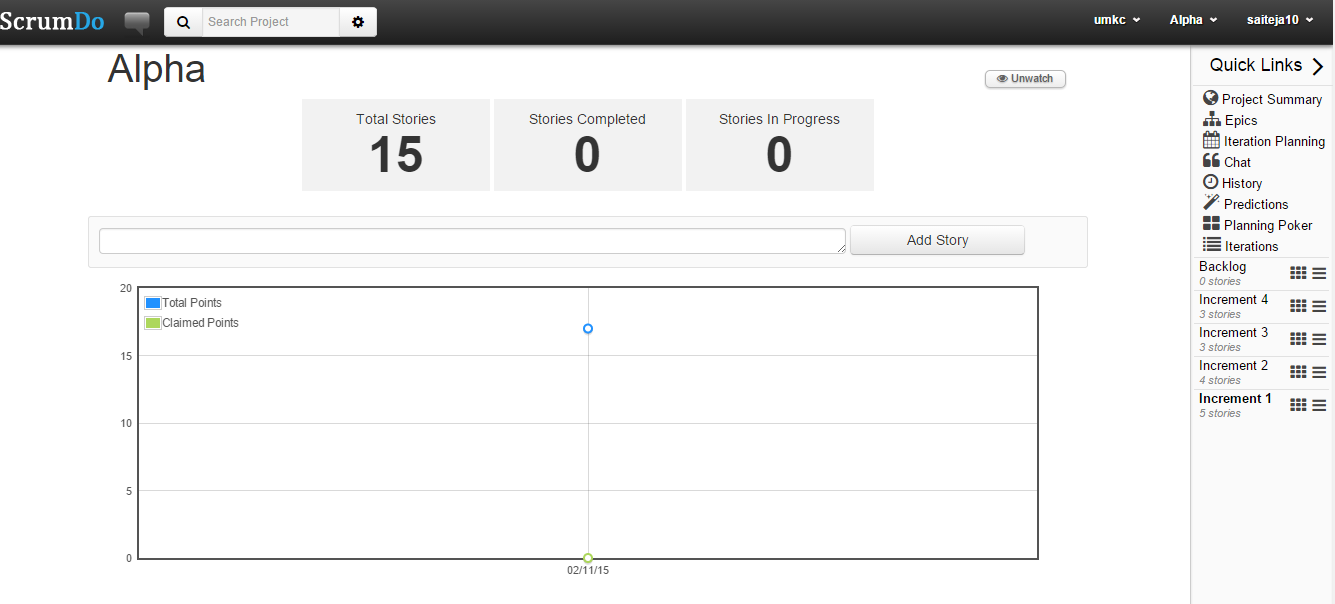
When customer selects items and adds to cart, he will be prompted to pay inorder to complete his order.

**Constraints/exceptions**

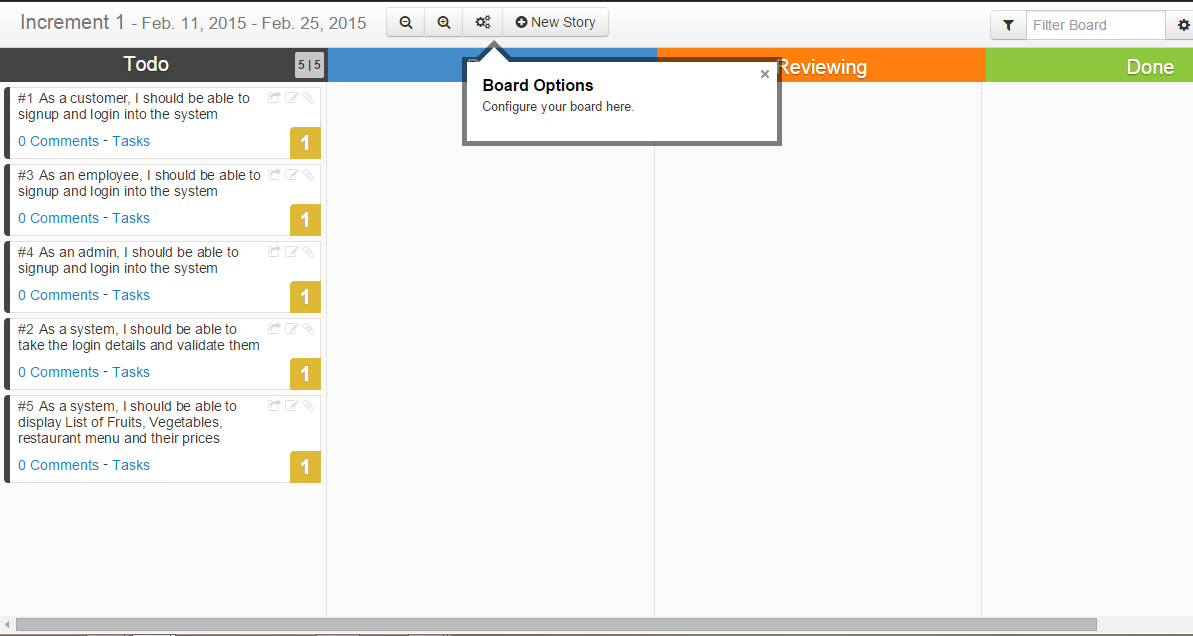
When Customer enters wrong payment details, an exception will be raised and will be prompted to enter his details again.

**VII. Plan by Services**

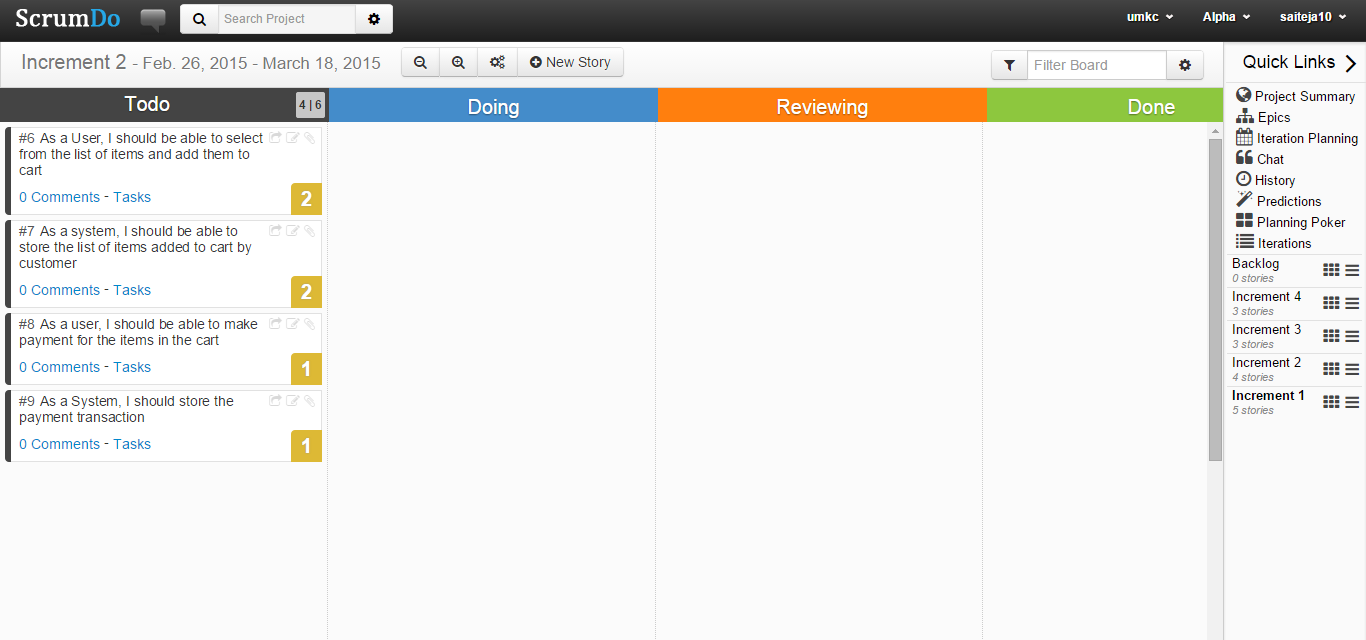
**Scrum Do Screen Shots:**

****

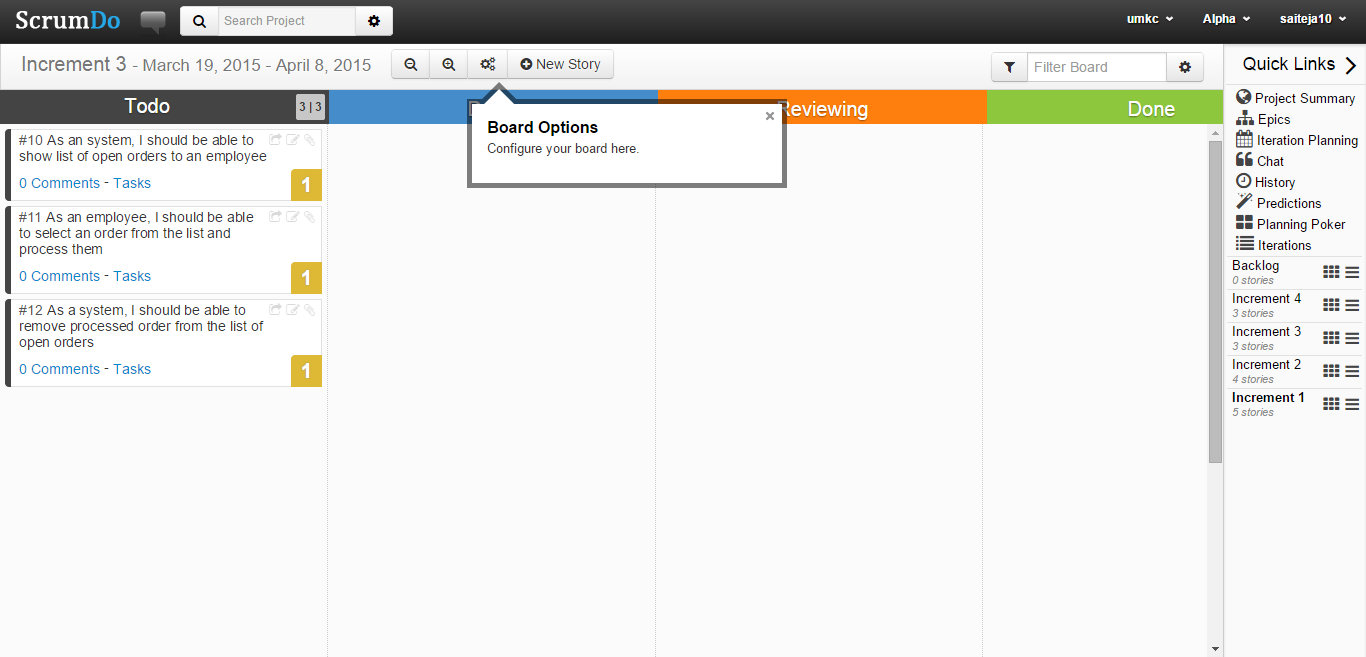
**Increment 1**

****

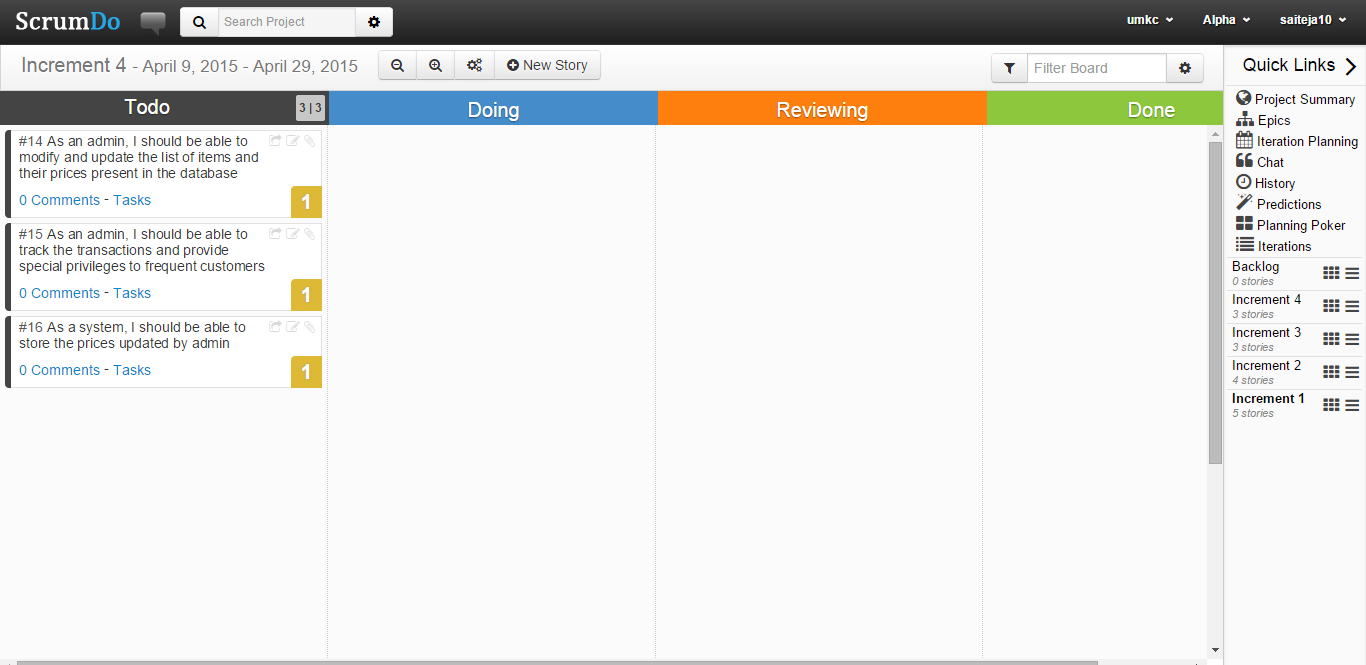
**Increment 2**

****

**Increment 3**

****

**Increment 4**

****

**All Iterations**

****

**GitHub url:**

<https://github.com/saiteja10/ASE-project-plan>

**VIII. Bibliography**

1. [**http://developer.android.com/index.html**](http://developer.android.com/index.html)
2. [**http://search.ams.usda.gov/mnsearch/mnsearch.aspx**](http://search.ams.usda.gov/mnsearch/mnsearch.aspx)
3. [**http://cxf.apache.org/**](http://cxf.apache.org/)
4. [https://www.**wikipedia**.org/](https://www.wikipedia.org/)

**PG4:**

SAI TEJA SARANAM (Class ID: 29)

Srikanth Nadendla (Class ID:23)

Nagender Goud Teegala (Class ID: 31)

Ramnikhil Voora (Class ID: 35)