San José State
UNIVERSITY

CMPE239: Web and Data Mining (Group - 13)

Project Report

Submitted by:
Ekta Vaswani (010021804)
Sumit Gerela (010021596)
Nipun Ahuja (010129899)
Rohil Shah (010015551)

Department of Software Engineering

Submitted to:
Prof. Chandrasekar Vuppalapati

ABSTRACT

Market basket analysis is an effective method to find patterns of customer's buying habits in a store. Store's transactional database can be used to find this pattern. Over the years, this approach has attracted a large number of researchers because of different use cases of this technique. Data gathered from this technique is used in marketing, service, generating strategies, sales, providing recommendations etc.

In this project, we have developed an iOS app for grocery outlets that uses customer's buying habits, location of the customer and then provides them with recommendation and discounts using these parameters.

Customer's buying habits were analyzed using Association rule mining, Apriori algorithm and then appropriate products were recommended to them based upon the results. Also, Clearance offers on products were provided based upon this analysis.

Acknowledgement

This project included a lot of research and dedication work. And the implementation would not have been possible without the contribution of many individuals. So we would like to thank them for their help.

We would like to thank Prof. Chandrasekar Vuppalapati for providing his expertise and technical support during the implementation. Without his input, the project would lack the quality that it has now.

Group Members

Name	**Responsibilities**
Ekta Vaswani	Wrote R & MongoDB Scripts, design & integrate REST APIs, Project Report
Sumit Gerela	Develop iOS App, Consume Web Services, Project Report, MongoDB
Rohil Shah	Design REST APIs, Project Presentations, Dataset Management, MongoDB
Nipun Ahuja	Develop iOS App, Project Report, R script, MongoDB

Table of Contents

Table of Figures

1. Introduction

In the 1990s, Connection to internet began to proliferate in consumer and enterprise markets, but it was limited in its use because of the low performance of the network interconnect. In the 2000s internet connectivity became the norm for many applications. Today, internet has become a necessity for almost every person however; most of the devices that are connected to the internet require human interaction and monitoring through interfaces and apps. Enter the phrase: "Internet of things".

The Internet of Things (IoT) is basically when everyday objects are connected to the internet and participating together to form an integrated system that makes the world we live in more intelligent and smarter. Till date, the world has deployed about 5 billion "smart" connected things. There are predictions which say that there would be around 50 billion connected devices by 2020 and during the course of our life we will definitely experience a trillion node networks. Since almost everything from cars to washing machines to refrigerators is connected to the internet, every "Thing" generates data which is already amassing huge amounts. The value of the Data isn't in simply amassing huge amounts, but in extracting actionable insights through deep analysis of that data. The use of Data Analytics will become a key basis of competition and growth for individual firms.

Grocery outlets such as Safeway, Walmart etc. are coming under great pressure as other large chains are expanding their market reach. Also, some independents are coming up in the market trying to match their larger rivals by leveraging power of data to understand the behavior of the customer. In brief, the field of data analytics has become an imperative field irrespective of the fact whether the outlet belongs to a large chain or it is individually run. Grocery outlets can use data analytics in providing recommendation which can drastically improve the total revenue of the outlet.

Recommendation systems aggregate the behavior of the customer to find trends and patterns and provide with recommendations based upon them. Building Recommendation Systems involves mathematically calculating how one person's actions relate to another's. As a means of this project we are analyzing transactions of a grocery outlet and then provide recommendations through iBeacon on a customer's phone.

iBeacon are pocket sized computers which leverage the power of Bluetooth low energy. iBeacon are being extensively used at airports, shopping malls, etc. Basically as soon as a Bluetooth low energy device comes into the field of iBeacon they do a task that is assigned to them. This task can be sending the location of the beacon on the cloud, giving a notification to the person who has the Bluetooth device etc.

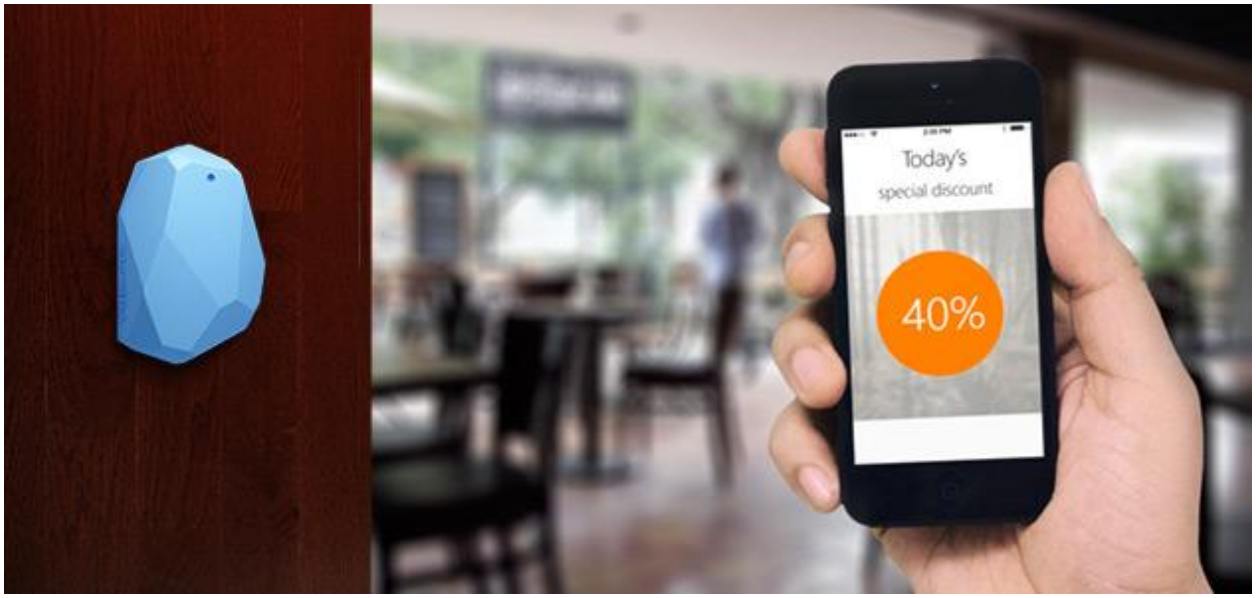
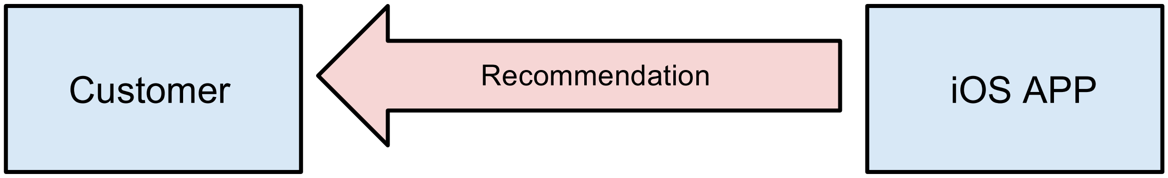


Figure 1: iBeacon in Action

In this project we are building an iOS app, which would basically provide recommendation to a particular customer. We would be comparing a customer's purchase history to the transactions data set and then provide him/her with the products that appear in similar transactions. The recommendations would be provided to the customer using iBeacon. We are using two iBeacon for this project, so as soon as person comes under the field of a particular iBeacon recommendation would be made to the person based upon the products that are associated with that particular iBeacon.



2. Requirements

2.1 Software Requirement

Following are the software requirements of our project.

- R framework
- STS IDE
- Apple developer program for iOS
- MongoDB instance
- Amazon EC2 to host the business tier

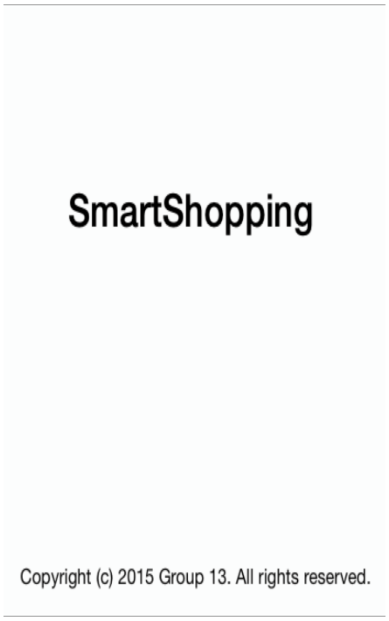
2.2 Hardware Requirement

Following are the hardware requirements of our project.

- iBeacon
- iOS devices to run the app

3. UI Design Principle – Storyboard and Wireframes

Below are some representations of what we visualized our User Interface to be like.



SmartShopping

Copyright (c) 2015 Group 13. All rights reserved.

Figure 2: App loading page

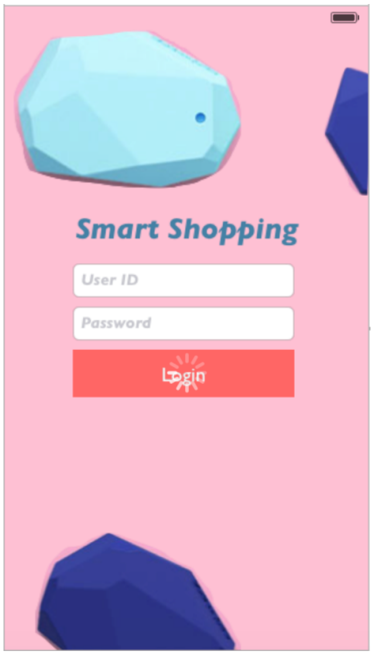
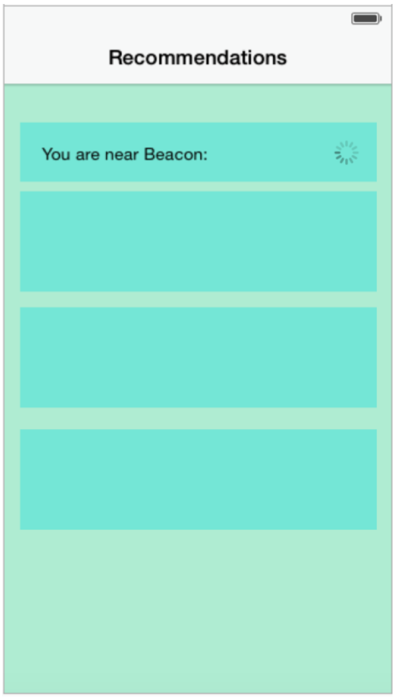


Figure 3: Login Page



Recommendations

You are near Beacon:

Figure 4: Recommendation Page

4. High Level Architecture Design

The architecture for our recommendation system is as follows:

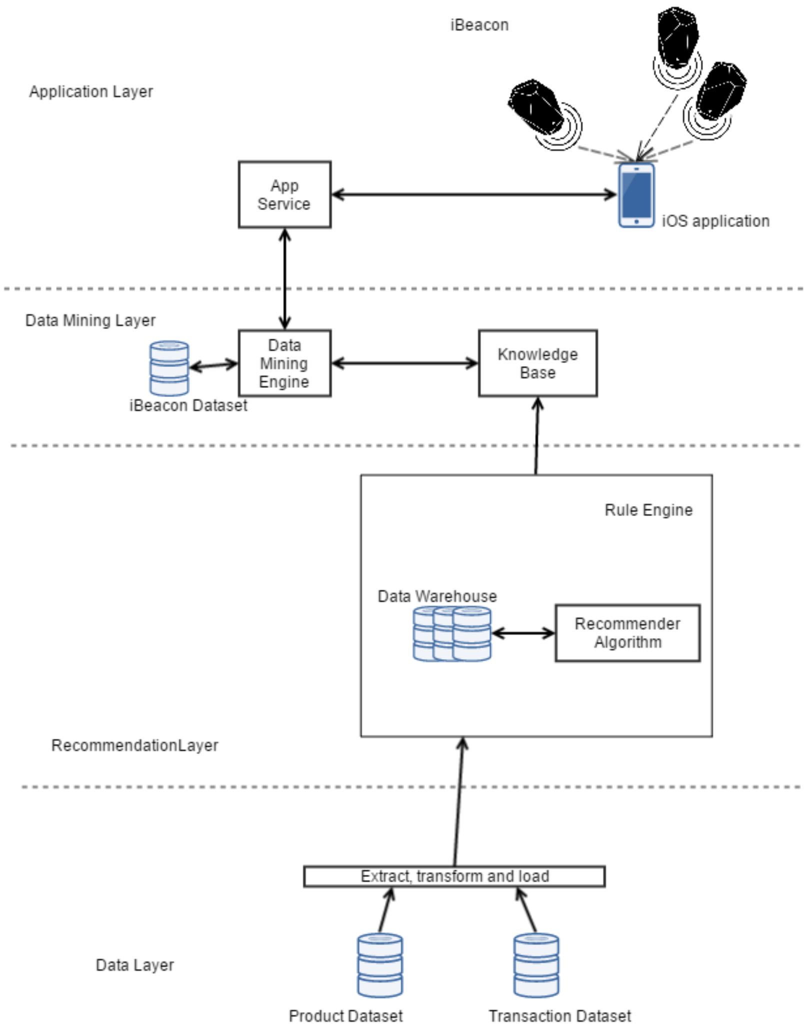


Figure 5: Architecture diagram

Our project architecture is divided into four layers namely Application layer, Data Mining layer, Recommendation layer and Data layer. Detailed description of these layers is given below.

1. Application layer: At this layer, we have different iBeacon broadcasting their information in a specific format. This format contains its UUID, Major id and Minor id. iBeacon in different stores have different major and minor ids. This way the app knows which iBeacon it is

communicating with and displays the recommendations accordingly. Once the app gets this id, it triggers the recommendation system and passes this id to it along with the customer id.

2. Data mining layer: This layer contains Data mining engine, iBeacon dataset and Knowledge base. This is the most important layer of our project. The major id of iBeacon it receives is mapped with the database to get the list of products around that beacon. It then retrieves rules from knowledge base and also gets customers history using customer's id. After checking and matching these rules and customer's history, it gets rules only for that customer and send it to the application layer where it is displayed to the customer.
3. Recommendation layer: This layer is responsible for generating rules that are used for recommendation. We have used Apriori algorithm for our project. It finds frequently bought items from a dataset and finds relation between them. This layer contains Data warehouse and Apriori algorithm. It calculates the rules using this and passes on to the data mining layer.
4. Data layer: This layer contains all the datasets required for the project. Data is extracted from datasets present in this layer and pushed into the data warehouse. ETL process can be configured whether to be run daily, weekly, or so on.

5. Datasets and Data Pattern

There are different datasets used in this process. The list of these datasets is given below along with a sample data from each of this dataset.

- beacons - This dataset contains UUID, major id and minor id of the iBeacons along with the list of products around a particular beacon

```
{
  "_id": {
    "$oid": "5538aba9e4b07a82907025c0"
  },
  "uuid": "B9407F30-F5F8-466E-AFF9-25556B57FE6D",
  "major_id": "43853",
  "minor_id": "11904",
  "products": [baby cosmetics”, "baby food", "rolls/buns", "brown bread","white bread","semi-finished bread","liver loaf","zwieback","long life bakery product","instant coffee","fruit/vegetable juice","bottled water","coffee","misc. beverages","soda","beverages","cocoa drinks", "tea","detergent","cleaner", "dish cleaner","house keeping products","decalcifier","bathroom cleaner","abrasive cleaner",“softener","toilet cleaner", "sauces", "sweet spreads", "mustard", "syrup", "ketchup", "honey",”jam",”herbs",”whole milk",”butter",”butter milk","yogurt",”curd cheese",”processed cheese",”curd",”condensed milk",”hard cheese","cream cheese ","UHT-milk","margarine","whipped/sour cream","sliced cheese","specialty cheese","spread cheese","soft cheese","mayonnaise","cream"]
}
```

- customer - This dataset contains the customer information for signing into the iPhone application i.e. username and password

```
{
  "_id": {
    "$oid": "55376540e4b0a6a3fc2d2703"
  },
  "cust_id": "1",
  "username": "Ekta",
  "password": "ekta1313"
}
```

- rules - This contains list of association rules generated from the list of transactions we have. These rules will be used to recommend products to the customers. The data for this dataset is generated using apriori algorithm in R.

```
{
  "_id": {
    "$oid": "5538be5335e34ba1f3520952"
  },
  "support": "0.009252669",
  "confidence": "0.5229885",
  "lhs": "baking powder ",
  "rhs": " whole milk"
}
```

- transactions - This contains the list of transactions of a grocery store. Each transaction is mapped to a customer using his/her customer id and the date if transaction.

```
{
  "_id": {
    "$oid": "55374bbb35e34ba1f350c347"
  },
  "cust_id": 7,
  "date": "3-Apr-14",
  "product": ["sausage","citrus fruit","tropical fruit","root vegetables","other vegetables","whole milk","yogurt","whipped/sour cream","processed cheese","rolls/buns","pastry","coffee","cake bar","newspapers"]
}
```

- history or map_reduce_history - This dataset contains list of products that a particular customer has bought till date. This dataset is generated from transactions dataset by applying mapreduce technique.

```
{
  "_id": {
    "$oid": "5538b213e4b07a829070263e"
  },
  "cust_id": 1,
  "products": ["tropical fruit","other vegetables","white bread","bottled water","brown
bread","chocolate","citrus fruit","tropical fruit","pip fruit","grapes","onions","other
vegetables","sliced cheese","cream cheese","mayonnaise","ice cream","frozen
dessert","margarine","mustard","soups","pickled vegetables","coffee","long life bakery
product","chocolate","seasonal products","female sanitary products","hygiene articles","cling
film/bags"]
}
```

- map_reduce_output2 - This dataset contains the count of each product bought till date i.e. number of times a product has been bought. This dataset is generated from transactions dataset by applying mapreduce technique.

```
{
  "_id": "UHT-milk",
  "value": 329
}
```

- discount - This dataset contains the discount percentage that can be provided for a particular product.

```
{
  "_id": {
    "$oid": "553ee55335e34ba1f35ce256"
  },
  "product": "other vegetables",
  "discount": 45
}
```

6. Data flow diagram and Architecture

The data flow of our project is as follows:-

1. There are iBeacon placed across different aisle. Each has its own UUID, major id and minor id. As soon as a device comes in the beacon's range, the beacon sends its details to the app.
2. Customer has to sign in first so that the system can know about customer's id. The app then sends this data along with customer's id to the recommendation component.
3. At this layer, data received from the app is used along with rules data, customer history data and recommendations are filtered for the customer.
4. After filtering the rules, it finally sends the recommendation rules to the app for products near that beacon; where it is displayed for the user.
5. Similarly, recommendation component also sends trending products based on max count in map_reduce_output2 dataset.
6. Also, recommendation component sends discount on product that has less market demand to increase its sale.

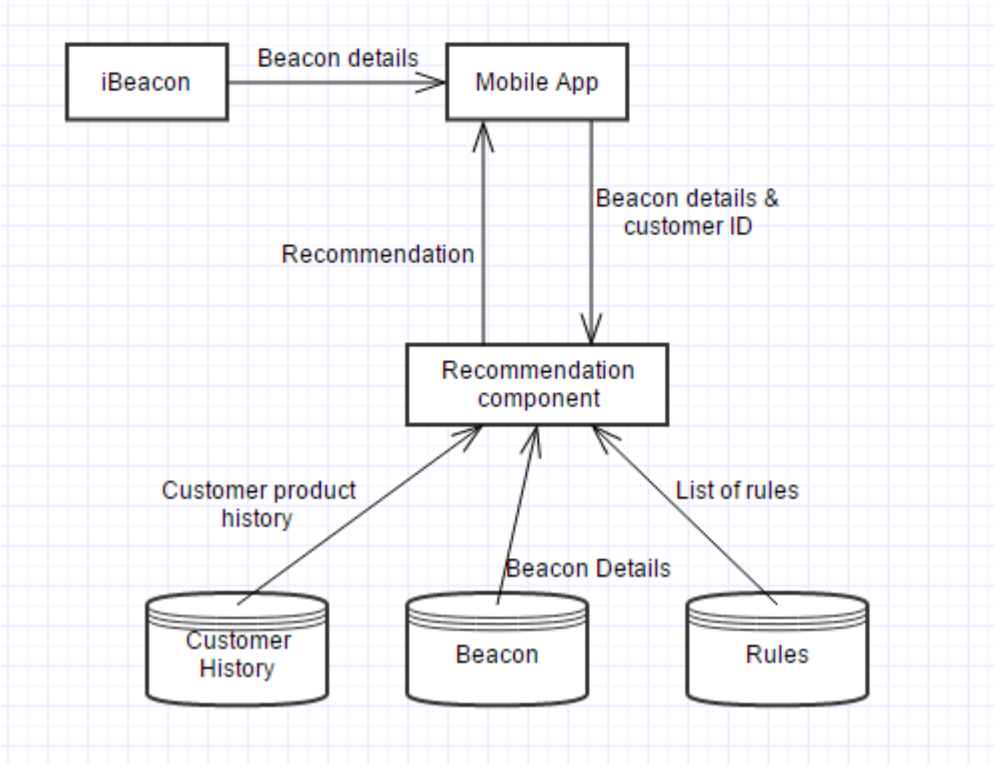


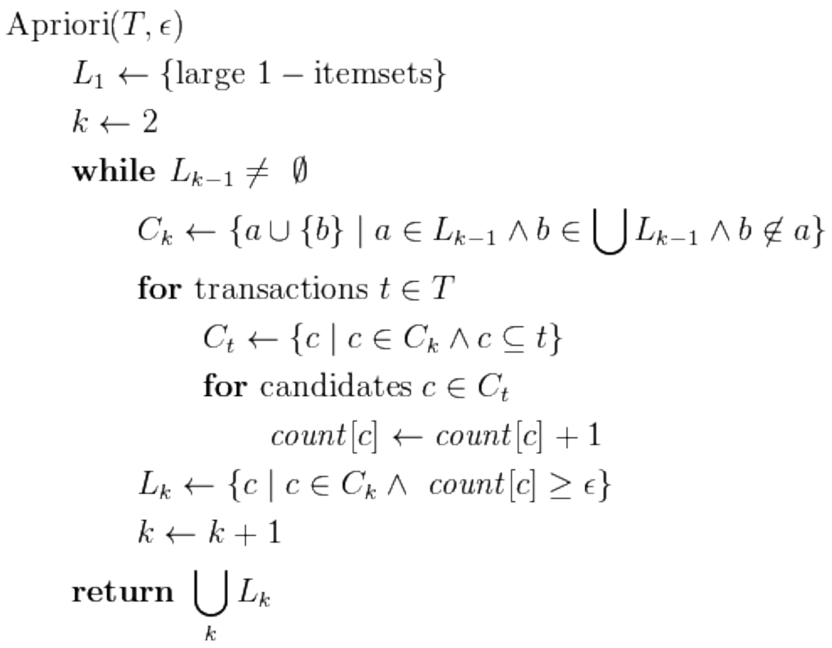
Figure 6: Data Flow diagram

7. Data Mining Principle and Algorithm

- **Apriori Algorithm**

Apriori algorithm is an algorithm which is used to find out frequent itemset mining over a transactional database. Apriori algorithm uses bottom up approach where frequent itemsets are enhanced one by one and groups of items are tested with respect to the data. This process continues till no more successful relations are found.

Pseudo code for apriori algorithm is given below where T is the transactional database and ϵ is the minimum support threshold.



```
Apriori($T, \epsilon$)
    $L_1 \leftarrow \{\text{large } 1 - \text{itemsets}\}$
    $k \leftarrow 2$
    while $L_{k-1} \neq \emptyset$
        $C_k \leftarrow \{a \cup \{b\} \mid a \in L_{k-1} \wedge b \in \bigcup L_{k-1} \wedge b \notin a\}$
        for transactions $t \in T$
            $C_t \leftarrow \{c \mid c \in C_k \wedge c \subseteq t\}$
            for candidates $c \in C_t$
                $count[c] \leftarrow count[c] + 1$
        $L_k \leftarrow \{c \mid c \in C_k \wedge count[c] \geq \epsilon\}$
        $k \leftarrow k + 1$
    return $\bigcup_k L_k$
```

Worst case complexity of this algorithm is O(N2M). Since most of the users have less purchases, in practice the complexity of this algorithm is O(NM).

8. KDD Principles

Data Mining is basically transforming humungous amounts of data into actionable information by the use of various mathematical techniques. Data Mining and Knowledge Discovery in Databases (KDD) are interchangeably used.

8.1 KDD Steps

The KDD process has the following steps:

1) Selection
2) Preprocessing
3) Transformation
4) Data Mining
5) Interpretation/Evaluation

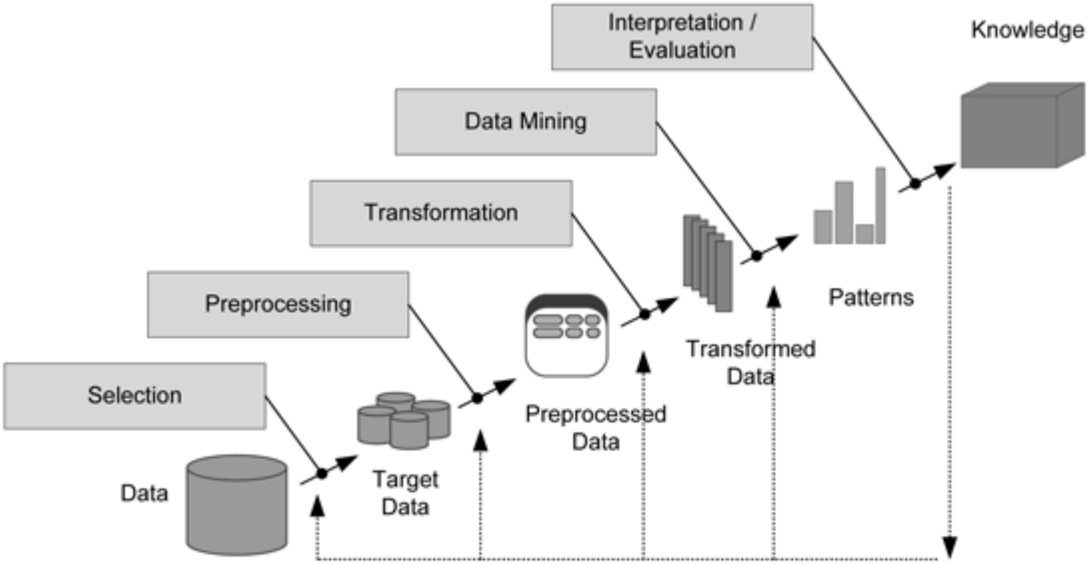


Figure 7: KDD process

8.1.1 Selection

In this step an Understanding of the application domain, goals of the end-user etc. are developed. After this step a target dataset is obtained by selecting a dataset, or focusing on subset of variables, or data samples, on which discovery is to be performed.

8.1.2 Preprocessing

The Second step in the KDD process is the preprocessing step. In this step outliers or noise is removed. Also necessary information to model for noise is collected and strategies for handling missing data fields are thought off.

8.1.3 Transformation

The third step in the KDD process is the Transformation step. In this step one basically finds useful features to represent the data that depends on the goal of the task. Also, dimensionality reduction techniques such as PCA are used to reduce the number of variables under consideration. It should be noted that transformation step is an iterative process.

8.1.4 Data Mining

Before the Data mining two steps are performed: 1) Choosing the data mining algorithm 2) Choosing the data mining task. In the first step, we choose the appropriate algorithm that is to be used for searching trends in the data. Also, we decide on the models and parameters that would be used in data mining.

Now we choose the data mining task and decide whether the goal of the KDD process is regression, classification clustering, etc. After doing all this we do the actual data mining and search for patterns or trends in a particular representation form.

8.1.5 Interpretation/Evaluation

This is final step in which trends and patterns are evaluated or interpreted so that one can take an action based upon the evaluation.

8.2 KDD Process with respect to Recommendations using iBeacon.

During the course of our project we have used the KDD process extensively. Below is the explanation of the KDD process with respect to our project:

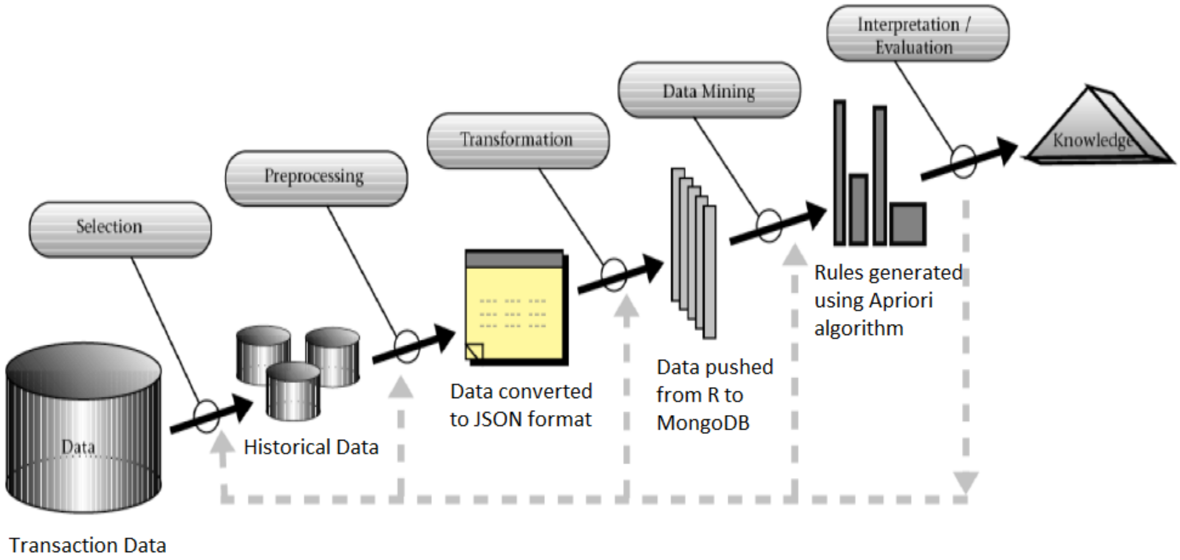

Selection
Preprocessing
Transformation
Data Mining
Interpretation / Evaluation
Knowledge
Data
Transaction Data
Historical Data
Data converted to JSON format
Data pushed from R to MongoDB
Rules generated using Apriori algorithm

Figure 8: KDD Process with respect to Recommendations using iBeacon

8.2.1 Selection

The first step in the KDD process is the selection step. In our project we have extracted customer's historical data from the transaction dataset. Here, the target dataset is the customer's historical data, which would be used to generate recommendations based upon this target data.

8.2.2 Preprocessing.

The target dataset is now preprocessed and converted to JSON format. JSON has many advantages over other formats as JSON format is faster i.e. it has simple syntax which ultimately makes it easier to parse.

8.2.3 Transformation.

Next, Data is pushed into MongoDB so that it can reference during data mining. MongoDB is a NoSQL database which is extensively used in applications involving huge amounts of data as it can scale out easily.

8.2.4 Data Mining.

At this stage Association rule mining (ARM) and algorithms like Apriori algorithm are used to generate association rules so that the generated rules can used to generate personalized recommendations. These rules are then saved in MongoDB.

8.2.5 Interpretation/Evaluation.

In this step, the generated rules are compared to customers historical data and based upon this comparison appropriate products are recommended to the customer on their phones.

9. Data Tools

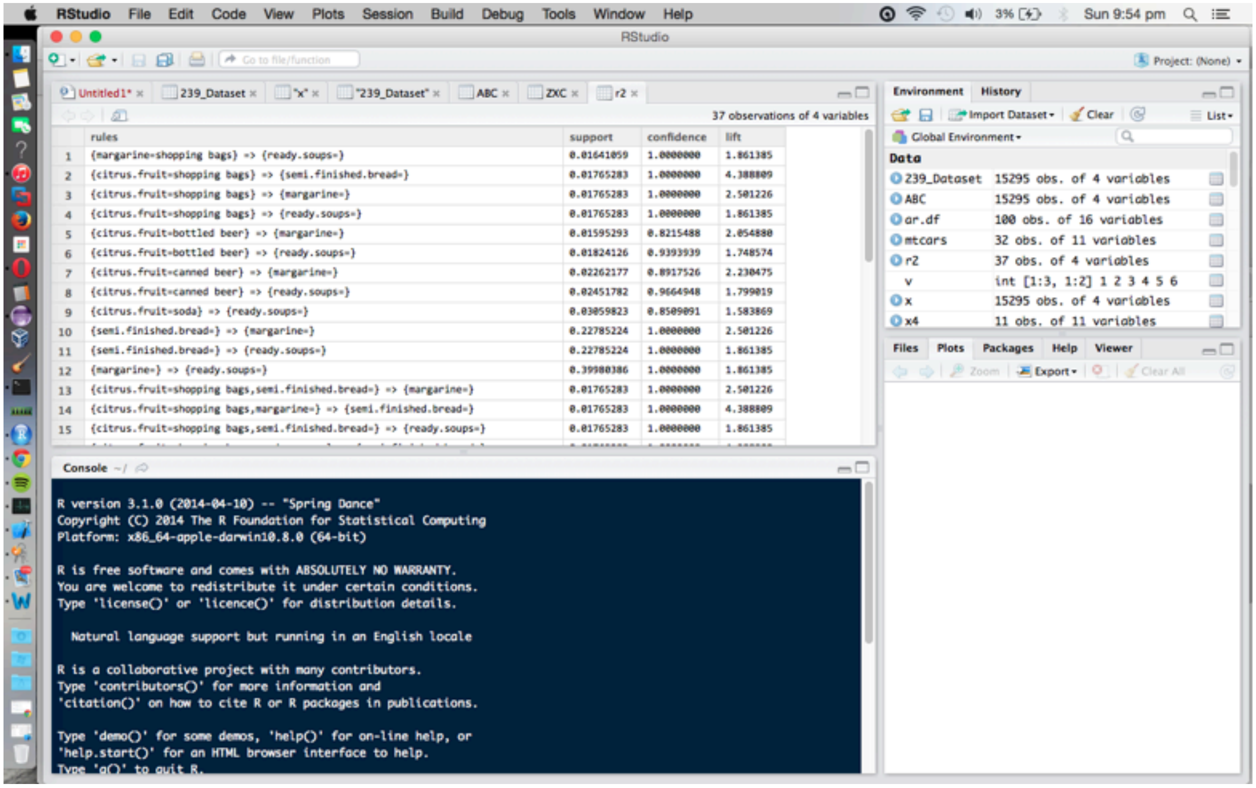
9.1 R

R is a language that is most commonly used for statistical computing and graphics. Various machine learning and graphical techniques are provided by R. Some of these include clustering, linear and non-linear modeling, classification, etc. It has many in-built functions for statistical and machine learning. Some of them are:-

- Data Extraction
- Data Cleaning
- Data Loading
- Data Transformation
- Data Visualization
- Predictive modeling

R can also connect to various data stores like MySQL, Hadoop, MongoDB. Data Analytics like regression, classification, clustering, recommendation and text mining can also be done using R. It is an integrated suite of software facilities. It is used for data manipulation and graphical display. It also includes

- Data handling and data storage facilities
- Different operators for calculation on arrays
- Intermediate tools used for analysis of data
- Graphical representation for data analysis



	rules	support	confidence	lift
1	{margarine=shopping bags} => {ready.soups=}	0.01641059	1.0000000	1.861385
2	{citrus.fruit=shopping bags} => {semi.finished.bread=}	0.01765283	1.0000000	4.388809
3	{citrus.fruit=shopping bags} => {margarine=}	0.01765283	1.0000000	2.501226
4	{citrus.fruit=shopping bags} => {ready.soups=}	0.01765283	1.0000000	1.861385
5	{citrus.fruit=bottled beer} => {margarine=}	0.01595293	0.8215488	2.054880
6	{citrus.fruit=bottled beer} => {ready.soups=}	0.01824126	0.9393939	1.748574
7	{citrus.fruit=canned beer} => {margarine=}	0.02262177	0.8917526	2.230475
8	{citrus.fruit=canned beer} => {ready.soups=}	0.02451782	0.9664948	1.799019
9	{citrus.fruit=soda} => {ready.soups=}	0.03059823	0.8509091	1.583869
10	{semi.finished.bread=} => {margarine=}	0.22785224	1.0000000	2.501226
11	{semi.finished.bread=} => {ready.soups=}	0.22785224	1.0000000	1.861385
12	{margarine=} => {ready.soups=}	0.39980386	1.0000000	1.861385
13	{citrus.fruit=shopping bags,semi.finished.bread=} => {margarine=}	0.01765283	1.0000000	2.501226
14	{citrus.fruit=shopping bags,margarine=} => {semi.finished.bread=}	0.01765283	1.0000000	4.388809
15	{citrus.fruit=shopping bags,semi.finished.bread=} => {ready.soups=}	0.01765283	1.0000000	1.861385

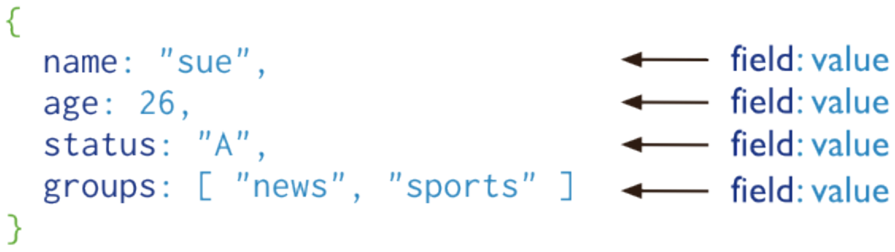
Data	
239_Dataset	15295 obs. of 4 variables
ABC	15295 obs. of 4 variables
ar.df	100 obs. of 16 variables
mtcars	32 obs. of 11 variables
r2	37 obs. of 4 variables
v	int [1:3, 1:2] 1 2 3 4 5 6
x	15295 obs. of 4 variables
x4	11 obs. of 11 variables

```
R version 3.1.0 (2014-04-10) -- "Spring Dance"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin10.8.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
```

Figure 9: RStudio

9.2 MongoDB

MongoDB is an open source document database. Every record in the database is called document. This document contains records as field and value pair. The values of field can also include other document, an array or an array of different documents.



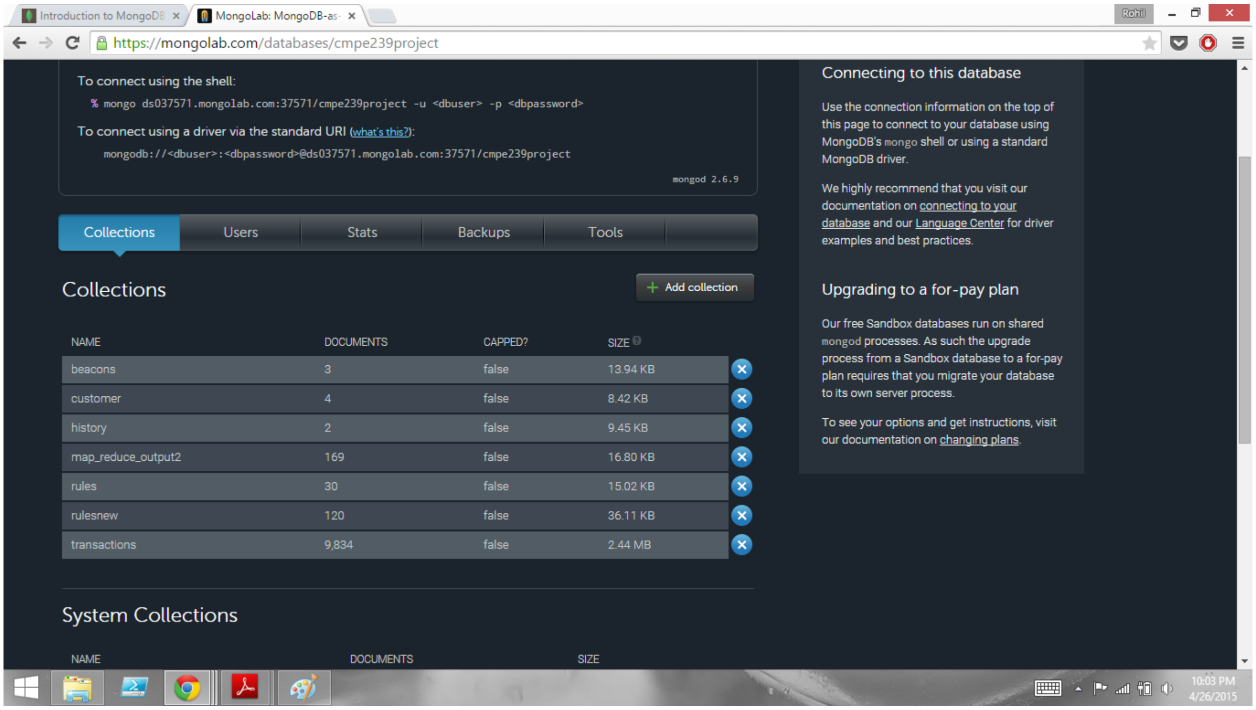
```
{
   name: "sue",
   age: 26,
   status: "A",
   groups: [ "news", "sports" ]
}
```

Figure 10: Field-Value pair example

Key features of MongoDB:-

1. High Performance
2. High Availability

3. Automatic Scaling



To connect using the shell:

```
% mongo ds037571.mongolab.com:37571/cmpe239project -u <dbuser> -p <dbpassword>
```

To connect using a driver via the standard URI (what's this?):

```
mongodb://<dbuser>:<dbpassword>@ds037571.mongolab.com:37571/cmpe239project
```

mongod 2.6.9

Collections

NAME	DOCUMENTS	CAPPED?	SIZE
beacons	3	false	13.94 KB
customer	4	false	8.42 KB
history	2	false	9.45 KB
map_reduce_output2	169	false	16.80 KB
rules	30	false	15.02 KB
rulesnew	120	false	36.11 KB
transactions	9,834	false	2.44 MB

System Collections

NAME	DOCUMENTS	SIZE

Connecting to this database

Use the connection information on the top of this page to connect to your database using MongoDB's mongo shell or using a standard MongoDB driver.

We highly recommend that you visit our documentation on connecting to your database and our Language Center for driver examples and best practices.

Upgrading to a for-pay plan

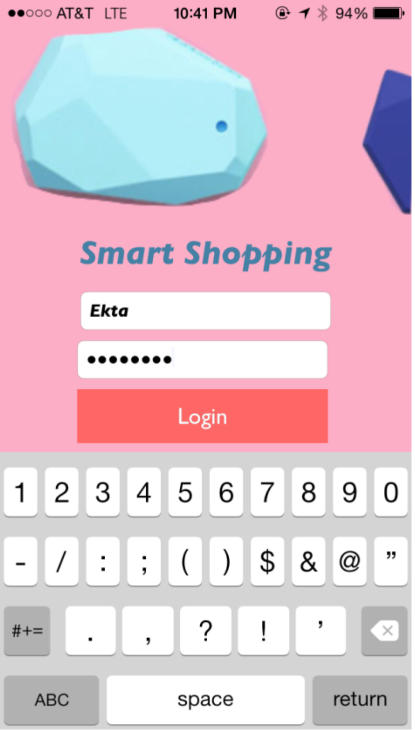
Our free Sandbox databases run on shared mongod processes. As such the upgrade process from a Sandbox database to a for-pay plan requires that you migrate your database to its own server process.

To see your options and get instructions, visit our documentation on changing plans.

Figure 11: MongoLab

10. Client Side Design

Customer must login to the app so that the app can send customer specific recommendations.



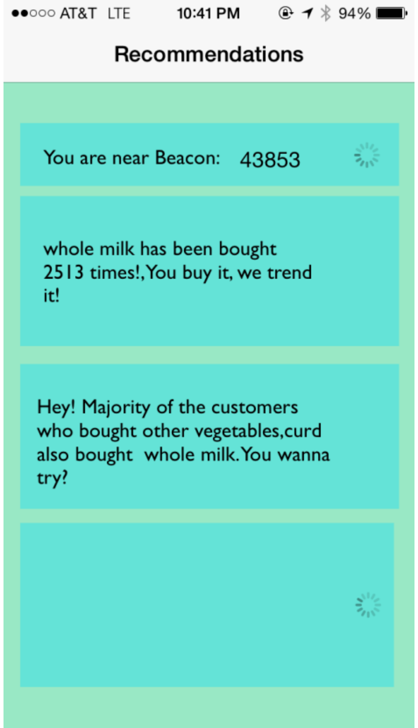
Ekta

••••••••

Figure 12: Login Page

The figure above shows the Login Page. Once the user enters the Credentials and press Login Button, corresponding REST API is called to validate the credentials. Once user is authenticated, this user id is used for future recommendations.

The Figure below shows the different recommendations shown to the customer. The very first information is the Beacon Major ID the user is near to. Once that is determined, they app issues various API requests with this major id and other relevant parameters to trigger the recommendation system.

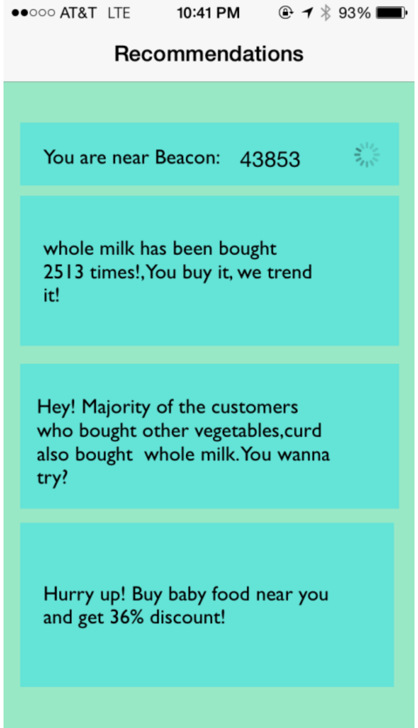


You are near Beacon: 43853

whole milk has been bought 2513 times!,You buy it, we trend it!

Hey! Majority of the customers who bought other vegetables,curd also bought whole milk.You wanna try?

Figure 13: Loading Recommendations



You are near Beacon: 43853

whole milk has been bought 2513 times!,You buy it, we trend it!

Hey! Majority of the customers who bought other vegetables,curd also bought whole milk.You wanna try?

Hurry up! Buy baby food near you and get 36% discount!

Figure 14: Recommendation Loaded

11. Testing

We have a test data which includes list of transaction at a grocery store. We were successful in applying Apriori algorithm on this data to generate rules. For testing purpose, we have used history data of 2 customers which is stored in a database. Depending on the customer id and the rules generated earlier, we were successfully able to recommend product to the customer through the mobile app.

12. Design Pattern Used

12.1 Front End

Delegate Design Pattern

We have used Delegate design pattern to handle the web service communication from various controller. Delegate design pattern is basically used when a particular tasks needs to be delegated to someone else who then calls a method once that task is finished. So in this project, We have two View Controllers (Login and Recommendation). We have written custom delegate which needs to be confirmed by every controller that needs to call the web service. Web service calls can be made in an asynchronous mode.

Below example shows the delegate pattern n action:

```
@protocol RESTWebServiceHandlerDelegate
@required
- (void) serverDidFinishOperation:(NSData *) data ;
@optional
-(void) serverDidFailOperation:(NSError *) error;
@end
```

As per the rules, all the methods under @required keyword needs to be implemented by the View Controllers. For instance, Login View Controller conforms to this view controller by implemented the necessary methods. Once the web service finished the request, it then calls the relevant methods on delegate.

This design pattern helped in developing a single module within the project that can handle all the REST calls from the iPhone Application.

Target-Action Pattern

Target-Action design pattern is used to when a particular object holds the information, and when certain event on that object occurs, that information can be processed. This pattern has two parts, IBOutlet and IBAction. IBOutlet can be defined as the reference of the object under consideration.

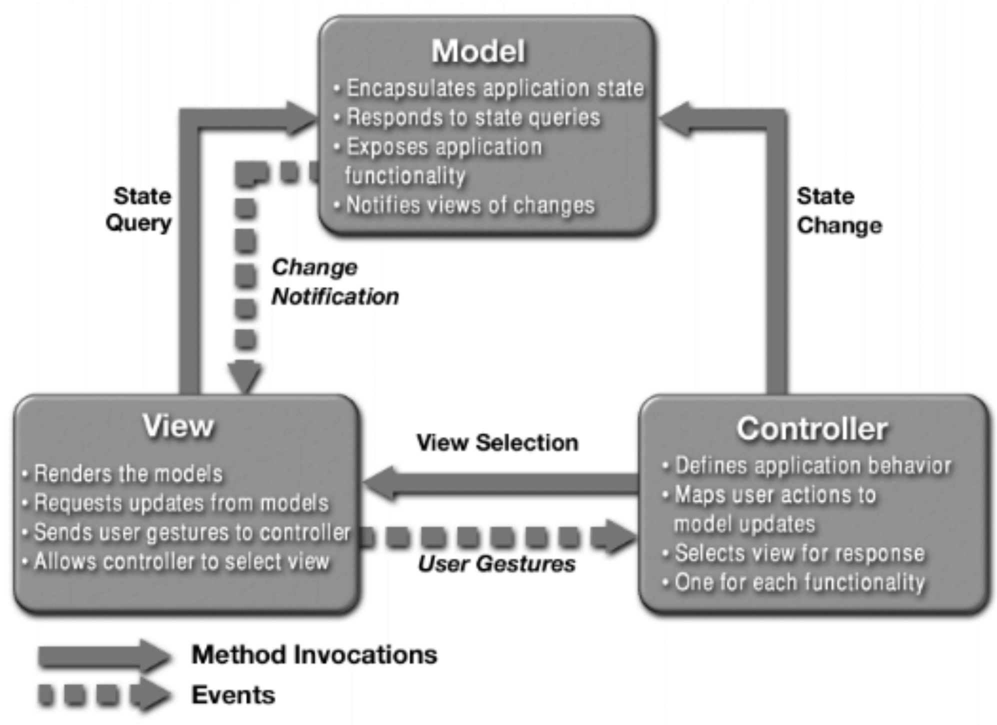
This is the pattern that has been used almost everywhere in the application to handle the UI elements. We have many UI components like Button, Activity Indicator, UILabels. For every UI element, if necessary, we have the IBOutlets and IBActions.

12.2 Middle Tier

Model view controller

MVC is a design pattern widely used in software applications in order to divide the responsibilities using a layered approach. It is a popular pattern adopted across many languages and frameworks.The purpose of MVC is to decouple data from the view. Model represents the business data and logic. View represents visual part of the data. In our system, it is the iPhone application. Controller is the link between model and view. It handles user actions that change the data or view.

SpringMVC is a web framework based on MVC design pattern. It is a request driven framework built on top of JEE servlets. Servlets on certain ports will listen for incoming request and each request triggers a process to serve the request.



Model
• Encapsulates application state
• Responds to state queries
• Exposes application functionality
• Notifies views of changes
State Query
Change Notification
State Change
View
• Renders the models
• Requests updates from models
• Sends user gestures to controller
• Allows controller to select view
View Selection
User Gestures
Controller
• Defines application behavior
• Maps user actions to model updates
• Selects view for response
• One for each functionality
Method Invocations
Events

Figure 15: MVC Architecture

Advantages of MVC fall into following three categories:

- Separation of Concerns
- Developer focus
- Parallel development

Separation of concern:

This allows reuse of business logic across various applications. It lets you develop user interface without affecting business logic. It minimizes copy and pasting of repetitive code.

Developer focus:

User interface developers can focus on UI logic and not be concerned with business logic. Similarly, business logic developers can focus on business logic and not be concerned with UI.

Parallel Development:

Business logic developers can expose stub classes that UI developers can use for UI development. Thus UI and business logic can be simultaneously developed and tested.

12.3 Data Store

MapReduce

MapReduce is a programming model which is extensively used for processing large datasets. In this programming model we have a map function and a reduce function. Map function takes a set of key value pairs and generates an intermediate key value pair. This intermediate key-value pairs is the input of the reduce function which basically aggregates all of the intermediate data to give the final output. A typical MapReduce setting can have many Mappers and many Reducers. It is to be noted at an enterprise scale these Mappers and reducers are distributed among different nodes. This programming model follows the new ACID properties i.e. A - Associative, C - Communicative, I - Idempotent and D- Distributive.

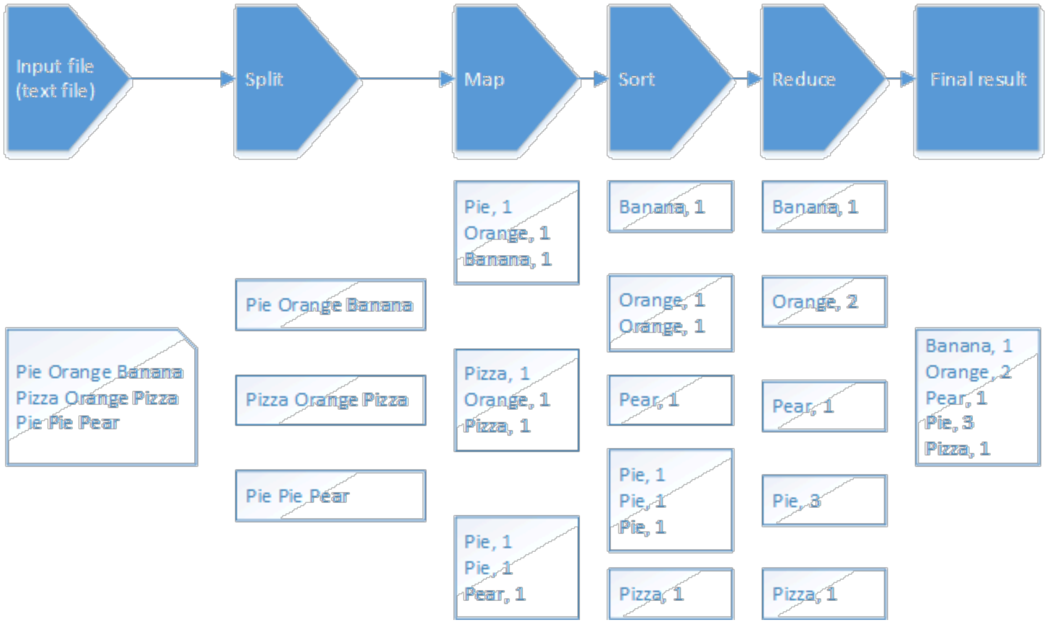


Figure 16: MapReduce Word Count

In our project the MapReduce programming model is used to find out the trending product that is selling in the store. We basically count the frequency of every item in the transaction dataset and declare the item having highest count as the trending item.

13. Project Cost

- Cost of Estimote iBeacon - $126
- Apple developers program for iOS - $99

14. References

[1] Developer.apple.com, 'Apple Developer', 2015. [Online]. Available: http://developer.apple.com. [Accessed: 28- Apr- 2015].

[2] 2015. [Online]. Available: https://ist.berkeley.edu/as-ag/pub/pdf/mvc-seminar.pdf. [Accessed: 28- Apr- 2015].

[3] C. Matheus, P. Chan and G. Piatetsky-Shapiro, 'Systems for knowledge discovery in databases', *IEEE Transactions on Knowledge and Data Engineering*, vol. 5, no. 6, pp. 903-913, 1993.

[4] S. Hilmi Ozguc, 'iBeacons and the future of retail shopping: Consumers are ready, but are retailers?', *Venturebeat.com*, 2014. [Online]. Available: http://venturebeat.com/2014/01/15/ibeacon-retailers/. [Accessed: 28- Apr- 2015].

[5] Y. Fan, Y. Shen and J. Mai, 'Study of the Model of E- commerce Personalized Recommendation System Based on Data Mining', 2008 International Symposium on Electronic Commerce and Security, 2008.