Mini Project On

Grocery Recommendation System

By

Mayur Dhevale (2021510012)
Dipesh Ghag (2021510016)

Under the guidance of

Internal Supervisor

Prof. Pooja Roundale

Department of Master Of Computer Application
Sardar Patel Institute of Technology
Autonomous Institute Affiliated to Mumbai University

2021-22

CERTIFICATE OF APPROVAL

This is to certify that the following students

Mayur Dhevale (2021510012)
Dipesh Ghag (2021510016)

Have satisfactorily carried out work on the project entitled

"Grocery Recommendation System"

Towards the fulfilment of project, as laid down
by
Sardar Patel Institute of Technology
during year
2021-22.

Project Guide:
Prof. Pooja Roundale

PROJECT APPROVAL CERTIFICATE

This is to certify that the following students

Mayur Dhevale (2021510012)
Dipesh Ghag (2021510016)

Have successfully completed the Project report on

"Grocery Recommendation System",

which is found to be satisfactory and is approved

at

SARDAR PATEL INSTITUTE OF TECHNOLOGY,
ANDHERI (W), MUMBAI

INTERNAL EXAMINER EXTERNAL EXAMINER

HEAD OF DEPARTMENT PRINCIPAL

Contents

Abstract

Now a days there is a lot of trend of online shopping. Almost all the products which are used in day to day life are available online especially grocery. So there are many projects going on online grocery shopping systems. And most of them have recommendation systems in them. Recommendations are used for making the work of the customer more easy and fast.

This reduces their valuable time and also the efforts. For this the recommendations given to the customer should be exact and should be fast. And most importantly they should none irritate the customer.

These recommendations are mostly given based on their necessity and their interest. Therefore the customer's necessity can be predicted from their purchase history and customer's interest can be predicted from the people have interest same as that customer.

Objectives

The Android based Application "Grocery App" is used

- To find out which products the customers might like to purchase based on his/her previous purchase history.
- To provide recommendations based on recorded information on the users' preferences.
- Recommendations are used for making the work of the customer easier and faster.
- To provide a platform to easily managed daily needs.

List of Figures

List of Tables

1 Introduction

1.1 Problem Definition

The purpose of this project is to introduce the concept and the idea behind creating a grocery shopping site. This grocery shopping website will analyze the data of customers based on their activities and recommend products using a machine learning model based on user activity.

1.2 Objectives and Scope

1.2.1 Objectives

The Android based application "Grocery App" is

- To find out which products the customers might like to purchase based on his/her previous purchase history.
- To provide recommendations based on recorded information on the users' preferences.
- Recommendations are used for making the work of the customer easier and faster.
- To provide a platform to easily managed daily needs.

1.2.2 Scope

The customer can provide his/her details in the profile and view for various grocery options on the app.

In the application the customer/user must enter his/her address required details in the profile section.

Our System is being made for reducing the information loss and smoothening the grocery shoping experience.

1.3 System Requirements

- Hardware Requirements on Server Side

Table 1.5.1: Hardware Requirements on Server Side

Processor	Dual Core Processor or Above
RAM	Minimum 4 GB RAM
Storage	Minimum 10 GB Hard Disk Space for smooth run

- Hardware Requirements on Client Side

Table 1.5.2: Hardware Requirements on Client Side

Device	Android Device with Touch Screen minimum 5” inch Display
Processor	Dual Core Processor or Above
RAM	Minimum 2 GB RAM
Storage	Minimum 250 MB Storage Space

- Software Requirements on Server Side

Table 1.5.3: Software Requirements on Server Side

Operating System	OS Independent
Database	Firestore

- Software Requirements on Client Side

Table 1.5.3: Software Requirements on Client Side

Operating System	Android/IOS Smartphone
Server	Not Required

2 Software Requirement Specification (SRS) and Design

2.1 Purpose

The purpose of our project is to develop an application that can help customers to easily choose their grocery needs and to keep all the required information handy.
This can save lots of efforts of customers s and it will be easier to store the each and every single and small detail regarding each grocery. This app will keep track of all the information regarding the grocery recommendations.

2.2 Definition

To build a Grocery Recommendation System so the customers can have an easy to go grocery selections.

2.3 Overall Description

2.3.1 Product Functions

The product function includes:

1. Authentication: Users are required to Sign-up and Log-in . Users will get a verification email for successful registration.
2. Profile: This will contain information regarding Customer.
3. Grocery: This will contain information regarding Grocery.
4. View Grocery Information: Customer can view the Grocery information.
5. Add grocery list : Customer can also add the grocery to the list.

2.3.2 User Characteristics

There are one types of users:

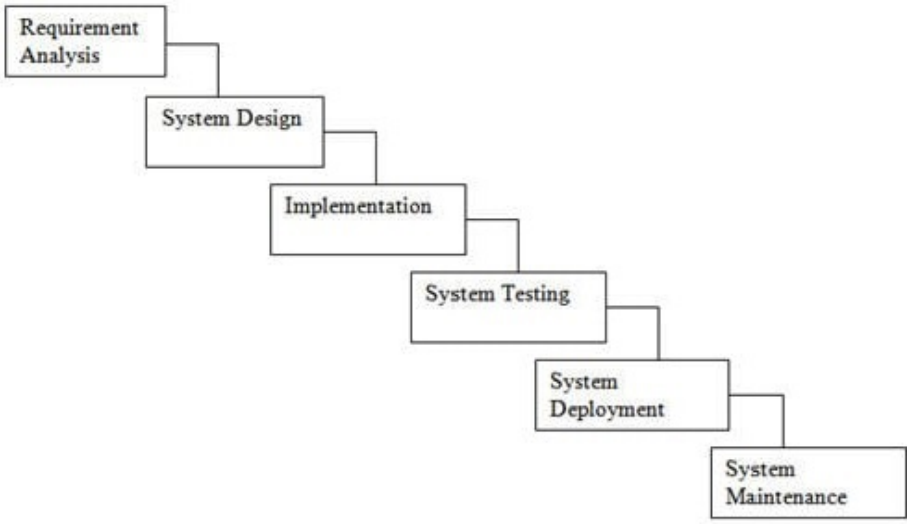
- Daily Customer's: Once customer's email id is encountered via login, customer is set on in database and he/she gets all the recommendations.

3 Project Analysis and Design

3.1 Methodologies Adapted

In Waterfall model, very less customer interaction is involved during the development of the product. Once the product is ready then only it can be demonstrated to the end users.

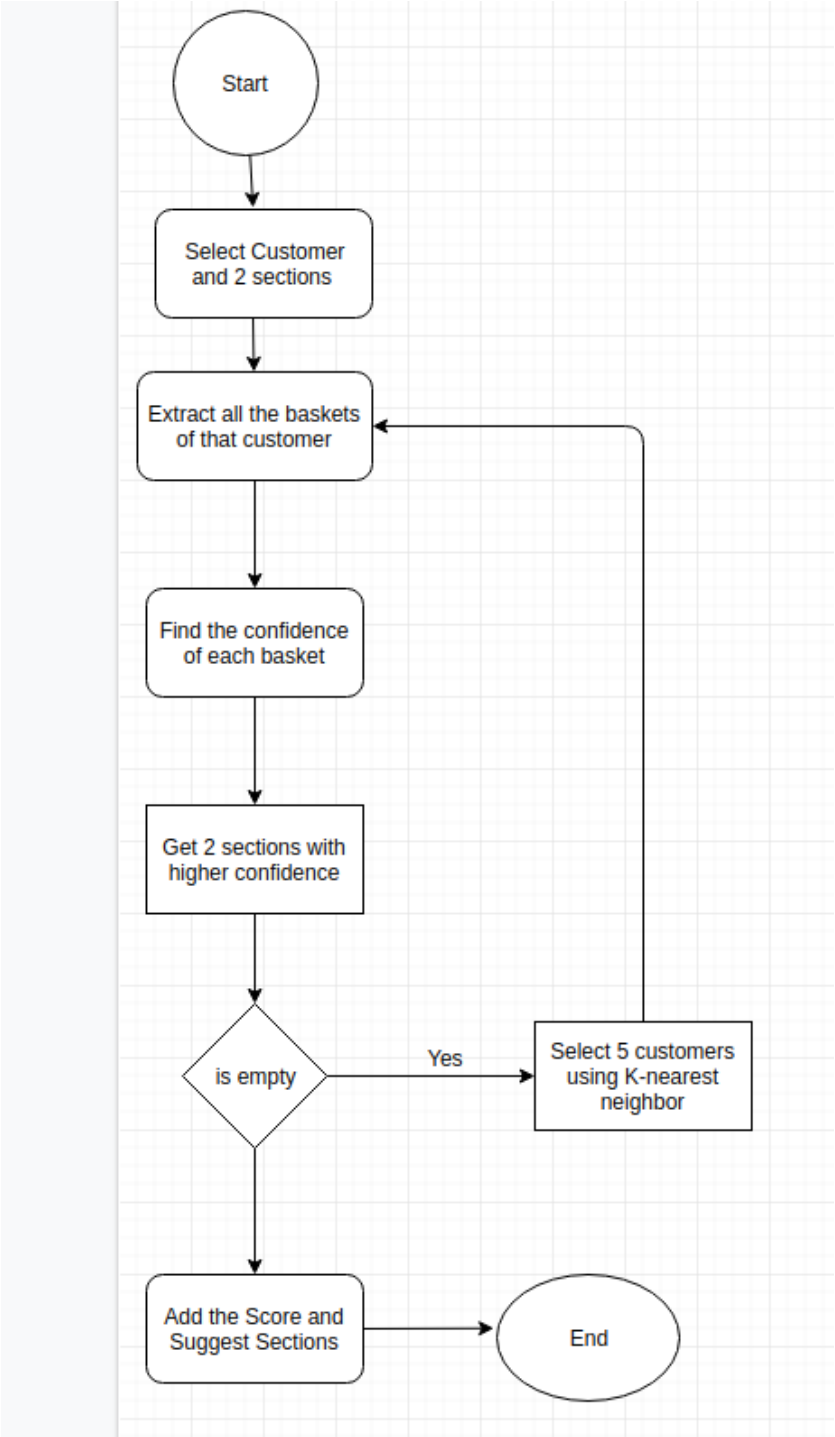
Once the product is developed and if any failure occurs then the cost of such issues is very high, because we need to update everything from document till the logic.



3.1.1: Diagrammatic Representation of Waterfall Model

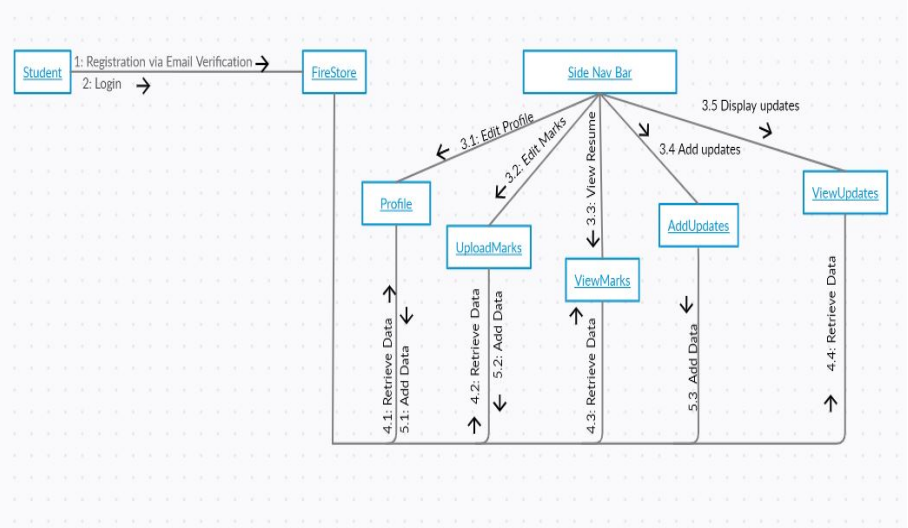
3.2 Modules

3.2.1 Activity diagram



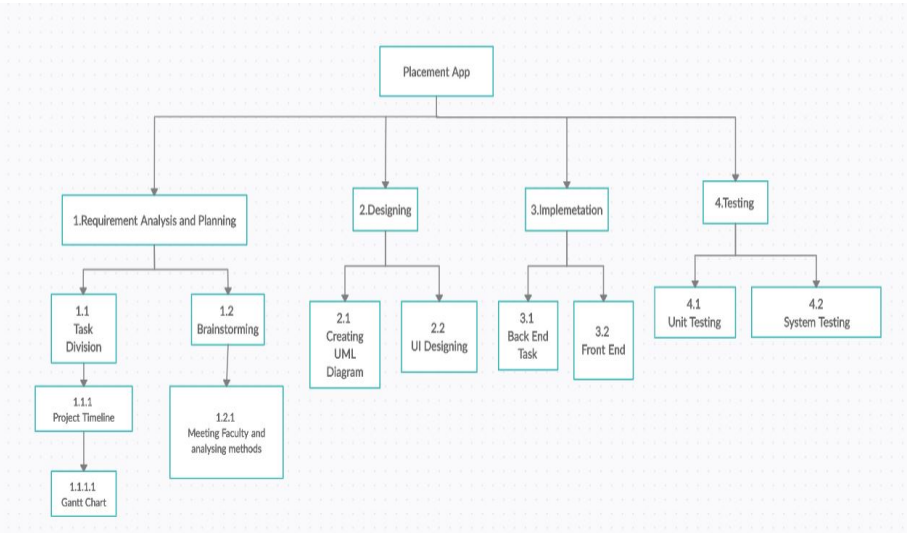
3.2.1: Activity Diagram

3.2.2 Communication Design



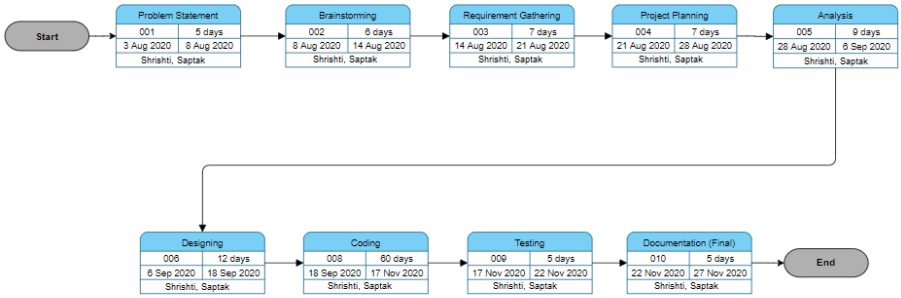
3.2.2: Communication Diagram

3.2.3 Work Breakdown Structure



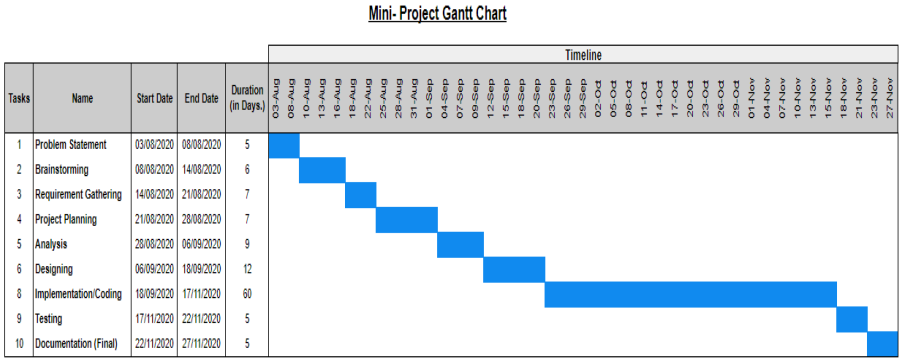
3.2.3: Work Breakdown Structure

3.2.4 PERT Chart



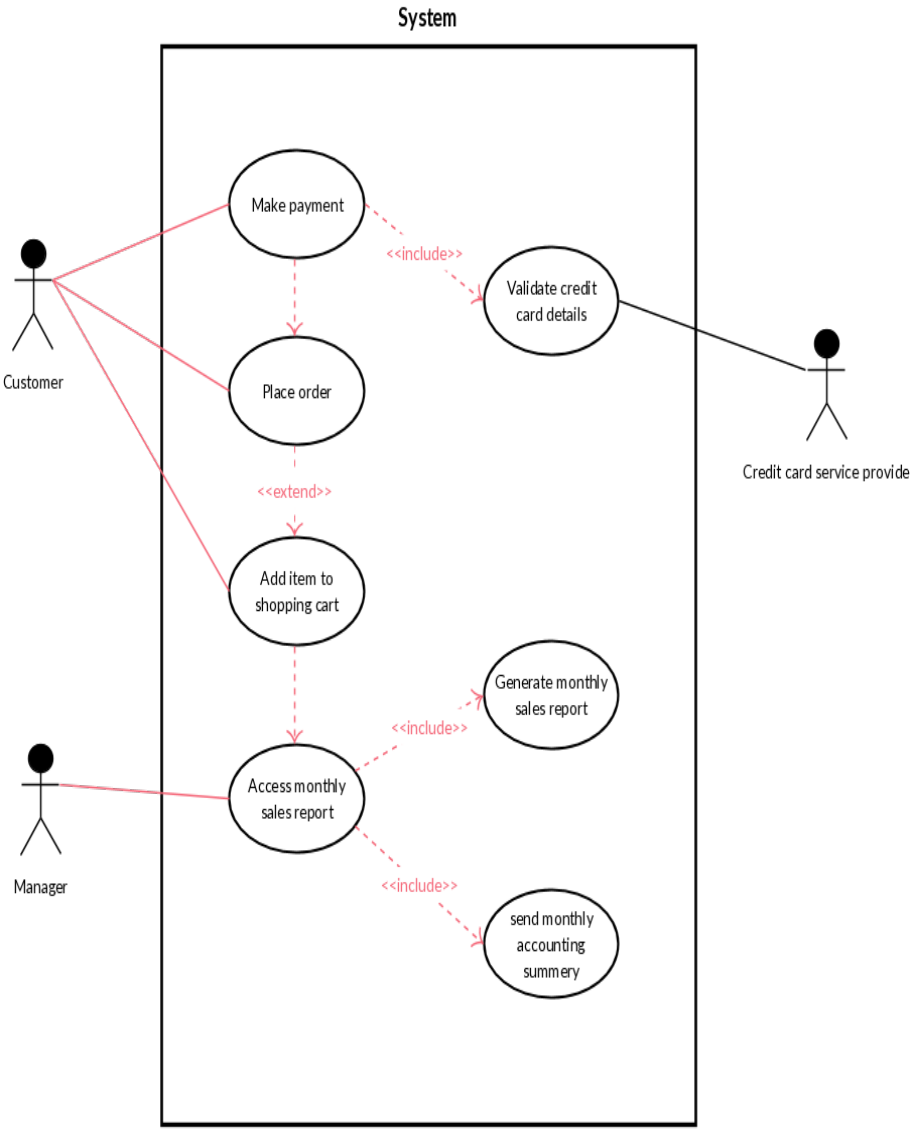
3.2.4: PERT Chart

3.2.5 Gantt Chart



3.2.5: Gantt Chart

3.2.6 Use-Case



3.2.6: Use-Case Diagram

Use Cases:

1. Register
2. Login
3. Update Profile
4. Update Grocery List
5. View Grocery List
6. Add Items
7. View Items

Table 4.2.1: Use Case Table - Register

Use Case ID	1
Use Case Name	Register
Actor	Customer
Pre-Condition	They must register themselves first
Post-Condition	User can login
Flow of events	Login,Register or Edit address details .

Table 4.2.3: Use Case Table - Update Profile

Use Case ID	2
Use Case Name	Update Profile
Actor	Customer
Pre-Condition	Login
Post-Condition	User can view or edit the profile

Table 4.2.4: Use Case Table - Update Grocery List

Use Case ID	3
Use Case Name	Favourite Items
Actor	Customer
Pre-Condition	Login
Post-Condition	User can view and edit the Favourite Items

Table 4.2.5: Use Case Table - View Grocery

Use Case ID	4
Use Case Name	View Grocery List
Actor	Customer
Pre-Condition	Login
Post-Condition	Can view Grocery List

Table 4.2.6: Use Case Table - Add Items

Use Case ID	5
Use Case Name	Add Items
Actor	Customer
Pre-Condition	Login
Post-Condition	Can add Items

Table 4.2.7: Use Case Table - View Items

Use Case ID	6
Use Case Name	View Grocery List
Actor	TPC
Pre-Condition	Login
Post-Condition	Can view Groceries as per categories or all together.

4 Project Implementation and Testing

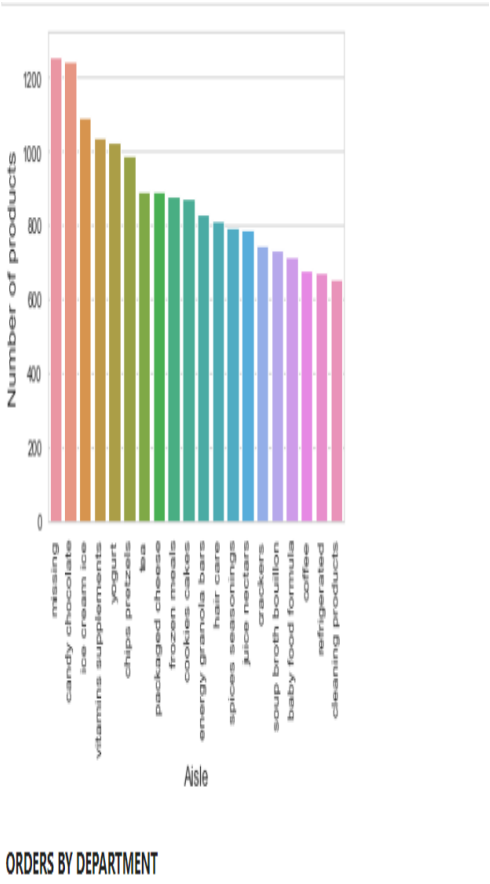
4.1 Code

[58]:

	product_name	add_to_cart_order
0	Pappardelle Nests Pasta	1.0
1	Flax Oil, Organic, Omega-3, Original Formula	1.0
2	Rosa Mosqueta Rose Hip Seed Oil	1.0
3	Prenatal Nutrients	1.0
4	Indian Wells Merlot	1.0
5	Lndbrg White Quinoa 16 Z	1.0
6	Indoor & Outdoor Allergies, Allergy & Congesti...	1.0
7	Chocolate Peppermint Tart	1.0
8	Easter Basket	1.0
9	Vanilla Flavor Multi-Symptom Relief	1.0

4.1.1

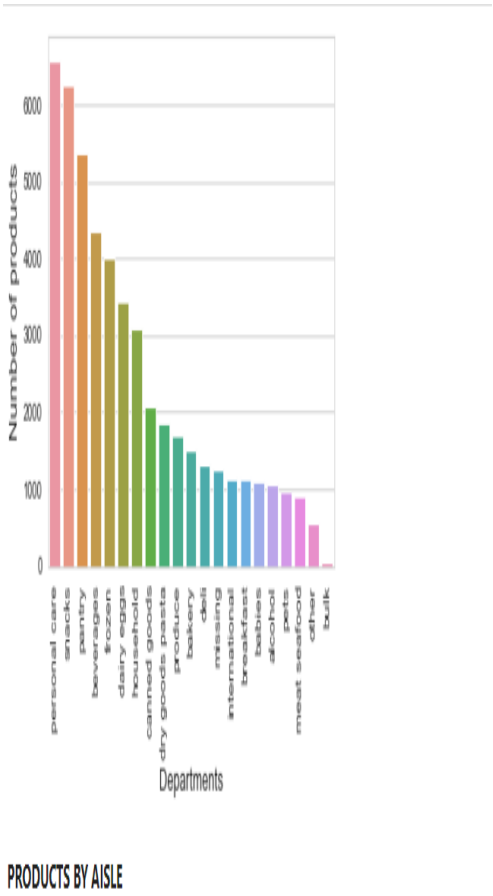
4.2



ORDERS BY DEPARTMENT

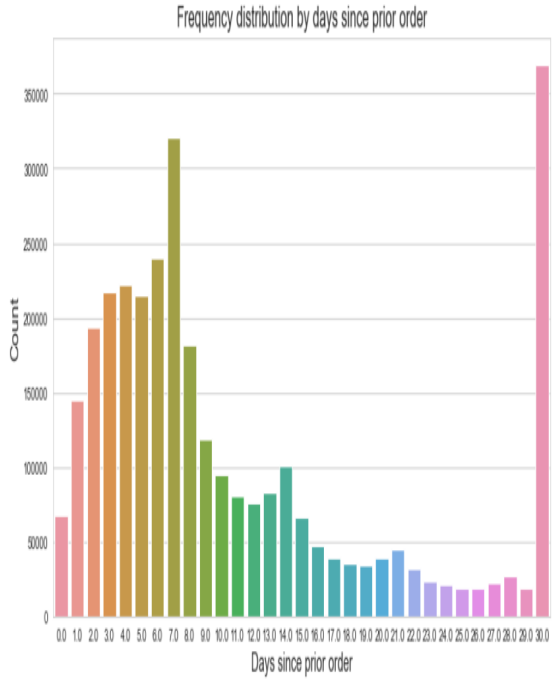
4.2.1

4.3


Number of products
0
1000
2000
3000
4000
5000
6000
personal care
snacks
pantry
beverages
frozen
dairy eggs
household
canned goods
dry goods pasta
produce
bakery
deli
missing
international
breakfast
babies
alcohol
pets
meat seafood
other
bulk
Departments
PRODUCTS BY AISLE

4.3.1

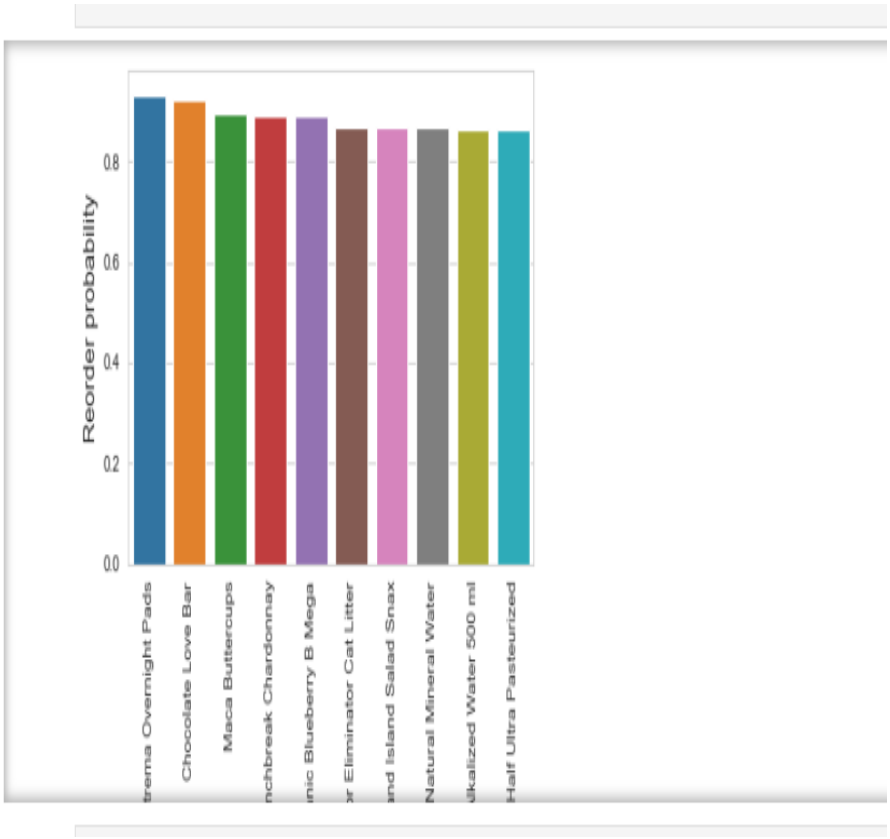
4.4



From this plot we can see that 7th day is where we have a spike, and then a relative small peak at days 14,21 and 28 which indicates that every 7 days or weekly is the order frequency. And then again there's a huge peak at the end of the month indicating that there's a monthly peak.

4.4.1

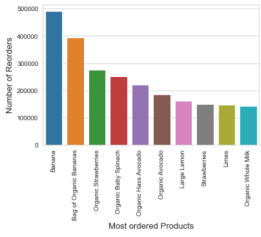
4.5


Reorder probability
0.0
0.2
0.4
0.6
0.8
trema Overnight Pads
Chocolate Love Bar
Maca Buttercups
nchbreak Chardonnay
nic Blueberry B Mega
r Eliminator Cat Litter
nd Island Salad Snax
Natural Mineral Water
lkalized Water 500 ml
Half Ultra Pasteurized

4.5.1

4.6

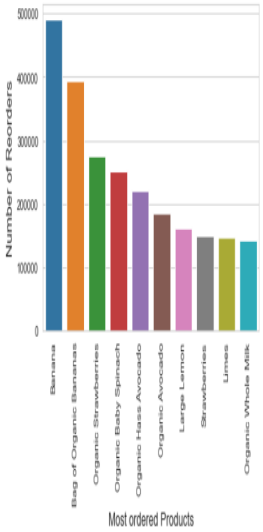
```
product_name
Banana                  491291
Bag of Organic Bananas  394930
Organic Strawberries    275577
Organic Baby Spinach    251705
Organic Hass Avocado    220877
Organic Avocado         184224
Large Lemon             160792
Strawberries            149445
Limes                   146660
Organic Whole Milk      142813
Name: Total_reorders, dtype: int64
```



4.6.1

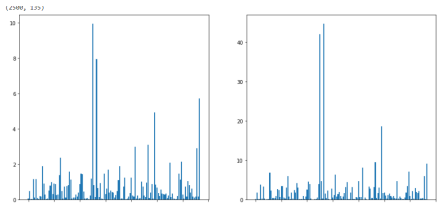
4.7

```
product_name
Banana                  491291
Bag of Organic Bananas  394930
Organic Strawberries    275577
Organic Baby Spinach    251705
Organic Hass Avocado    220877
Organic Avocado         184224
Large Lemon             160792
Strawberries            149445
Limes                   146660
Organic Whole Milk      142813
Name: Total_reorders, dtype: int64
```



4.7.1

4.8

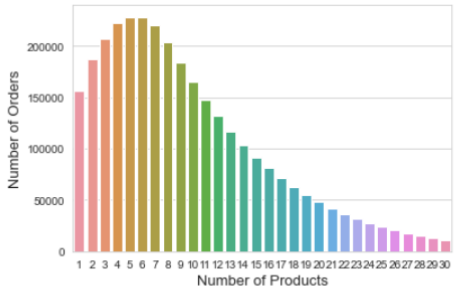


4.8.1

4.9

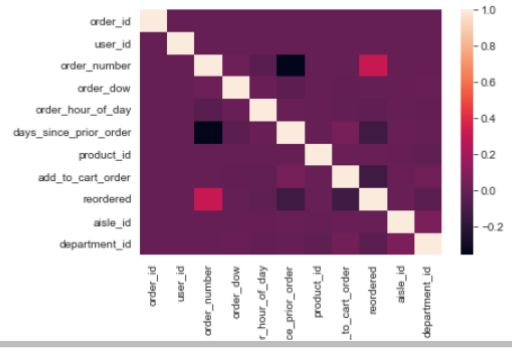
	item_A	item_B	freqAB	supportAB	freqA	supportA	freqB	supportB	confidenceAtoB	confidenceBtoA	lift
0	1337	7922	5	0.012986	5	0.012986	6	0.015583	1.000000	0.833333	64.173333
2	16668	19064	4	0.010389	5	0.012986	5	0.012986	0.800000	0.800000	61.606400
213	22864	3567	6	0.015583	7	0.018180	6	0.015583	0.857143	1.000000	55.005714
232	21640	20739	6	0.015583	7	0.018180	6	0.015583	0.857143	1.000000	55.005714
241	16096	42305	5	0.012986	7	0.018180	5	0.012986	0.714286	1.000000	55.005714
...	...	...	...	...	...	...	...	...	...	...	...
17886	47626	5876	4	0.010389	1899	4.931955	1074	2.789321	0.002106	0.003724	0.000755
9263	9076	13176	7	0.018180	801	2.080303	4745	12.323395	0.008739	0.001475	0.000709
23971	16797	21137	6	0.015583	1624	4.217744	3402	8.835446	0.003695	0.001764	0.000418
6405	13176	24852	7	0.018180	4745	12.323395	5912	15.354249	0.001475	0.001184	0.000096
9459	24852	13176	5	0.012986	5912	15.354249	4745	12.323395	0.000846	0.001054	0.000069

4.10



[64]:
```
corr = Train_data.corr()
sns.heatmap(corr,xticklabels=corr.columns, yticklabels=corr.columns)
```

[64]: <AxesSubplot:>



4.11

	itemA	itemB	freqAB	supportAB	freqA	supportA	freqB	supportB	confidenceAtoB	confidenceBtoA	lift
0	Yogurt, Sheep Milk, Strawberry	Blueberry Sheep Milk Yogurt	5	0.012986	5	0.012986	6	0.015583	1.000000	0.833333	64.173333
2	Bamba Peanut Snack	Bissli Pizza Flavor Snack	4	0.010389	5	0.012986	5	0.012986	0.800000	0.800000	61.606400
213	Iced Bhakti Chai Coffee Blend	Apple Mango Passion Fruit Fruit Snack	6	0.015583	7	0.018180	6	0.015583	0.857143	1.000000	55.005714
232	Tai Pei Chicken Chow Mein	Chicken Egg Rolls	6	0.015583	7	0.018180	6	0.015583	0.857143	1.000000	55.005714
241	Filet Mignon Canine Cuisine Wet Dog Food	Dog Food With Beef in Meaty Juices	5	0.012986	7	0.018180	5	0.012986	0.714286	1.000000	55.005714
...	...	...	...	...	...	...	...	...	...	...	...
17886	Large Lemon	Organic Lemon	4	0.010389	1899	4.931955	1074	2.789321	0.002106	0.003724	0.000755
9263	Blueberries	Bag of Organic Bananas	7	0.018180	801	2.080303	4745	12.323395	0.008739	0.001475	0.000709
23971	Strawberries	Organic Strawberries	6	0.015583	1624	4.217744	3402	8.835446	0.003695	0.001764	0.000418
6405	Bag of Organic Bananas	Banana	7	0.018180	4745	12.323395	5912	15.354249	0.001475	0.001184	0.000096
9459	Banana	Bag of Organic Bananas	5	0.012986	5912	15.354249	4745	12.323395	0.000846	0.001054	0.000069

120658 rows × 11 columns

4.12

[44]:

	user_id	cluster	aisle1	product1	aisle2	product2	aisle3	product3	aisle4	product4	aisle5	product5
0	284	0	fresh fruits	Banana	fresh vegetables	Organic Yellow Onion	yogurt	Total 2% with Strawberry Lowfat Greek Strained...	packaged vegetables fruits	Organic Baby Spinach	packaged cheese	Organic Whole String Cheese
1	293	0	fresh fruits	Banana	fresh vegetables	Organic Yellow Onion	yogurt	Total 2% with Strawberry Lowfat Greek Strained...	packaged vegetables fruits	Organic Baby Spinach	packaged cheese	Organic Whole String Cheese
2	319	0	fresh fruits	Banana	fresh vegetables	Organic Yellow Onion	yogurt	Total 2% with Strawberry Lowfat Greek Strained...	packaged vegetables fruits	Organic Baby Spinach	packaged cheese	Organic Whole String Cheese
3	358	0	fresh fruits	Banana	fresh vegetables	Organic Yellow Onion	yogurt	Total 2% with Strawberry Lowfat Greek Strained...	packaged vegetables fruits	Organic Baby Spinach	packaged cheese	Organic Whole String Cheese
4	432	1	fresh vegetables	Organic Yellow Onion	fresh fruits	Banana	packaged vegetables fruits	Organic Baby Spinach	packaged cheese	Organic Whole String Cheese	yogurt	Total 2% with Strawberry Lowfat Greek Strained...
...	...	...	...	...	...	...	...	...	...	...	...	...
2495	205352	0	fresh fruits	Banana	fresh vegetables	Organic Yellow Onion	yogurt	Total 2% with Strawberry Lowfat Greek Strained...	packaged vegetables fruits	Organic Baby Spinach	packaged cheese	Organic Whole String Cheese
2496	205381	0	fresh fruits	Banana	fresh vegetables	Organic Yellow Onion	yogurt	Total 2% with Strawberry Lowfat Greek Strained...	packaged vegetables fruits	Organic Baby Spinach	packaged cheese	Organic Whole String Cheese
2497	205551	0	fresh fruits	Banana	fresh vegetables	Organic Yellow Onion	yogurt	Total 2% with Strawberry Lowfat Greek Strained...	packaged vegetables fruits	Organic Baby Spinach	packaged cheese	Organic Whole String Cheese
2498	205679	0	fresh fruits	Banana	fresh vegetables	Organic Yellow Onion	yogurt	Total 2% with Strawberry Lowfat Greek Strained...	packaged vegetables fruits	Organic Baby Spinach	packaged cheese	Organic Whole String Cheese
2499	205952	0	fresh fruits	Banana	fresh vegetables	Organic Yellow Onion	yogurt	Total 2% with Strawberry Lowfat Greek Strained...	packaged vegetables fruits	Organic Baby Spinach	packaged cheese	Organic Whole String Cheese

2500 rows × 12 columns

4.13

30]:

```
def merge_item_name(rules, item_name):
    columns = ['item_A','item_B','freqAB','supportAB','freqA','supportA','freqB','supportB',
               'confidenceAtoB','confidenceBtoA','lift']
    rules = (rules
                .merge(item_name, left_on='item_A', right_on='product_id')
                .merge(item_name, left_on='item_B', right_on='product_id'))
   # print(rules)
    return rules[columns]

rules_final = merge_item_name(rules, item_name).sort_values('lift', ascending=False)
display(rules_final)
```

4.14

```
def association_rules(order_item, min_support):

    print("Starting order_item: {:22d}".format(len(order_item)))


    # Calculate item frequency and support
    item_stats             = freq(order_item).to_frame("freq")
    item_stats['support']  = item_stats['freq'] / order_count(order_item) * 100

    # Filter from order_item items below min support
    qualifying_items       = item_stats[item_stats['support'] >= min_support].index
    order_item             = order_item[order_item.isin(qualifying_items)]

    print("Items with support >= {}: {:15d}".format(min_support, len(qualifying_items)))
    print("Remaining order_item: {:21d}".format(len(order_item)))


    # Filter from order_item orders with less than 2 items
    order_size             = freq(order_item.index)
    qualifying_orders      = order_size[order_size >= 2].index
    order_item             = order_item[order_item.index.isin(qualifying_orders)]

    print("Remaining orders with 2+ items: {:11d}".format(len(qualifying_orders)))
    print("Remaining order_item: {:21d}".format(len(order_item)))


    # Recalculate item frequency and support
    item_stats             = freq(order_item).to_frame("freq")
    item_stats['support']  = item_stats['freq'] / order_count(order_item) * 100

    # Get item pairs generator
    item_pair_gen          = get_item_pairs(order_item)

     # Calculate item pair frequency and support
    item_pairs             = freq(item_pair_gen).to_frame("freqAB")
```

4.15

```
prod_f  = prod_f.groupby(['product_name'])['Total_reorders'].sum().sort_values(ascending=False)
print(prod_f)
sns.set_style('whitegrid')
plt.xticks(rotation='vertical')
sns.barplot(x=prod_f.index,y=prod_f.values)
plt.ylabel('Number of Reorders', fontsize=13)
plt.xlabel('Most ordered Products', fontsize=13)
plt.show()
```

5 Test Cases

Table 6.1: Test Case - Login and Register

Test Case ID	Test Case Name	Test Data	Expected Output	Actual Output	Result
1	User enter user id and password	Enters the correct user id and password	Log in Successful	Home Page	Pass
2	User enter user id and password	Enters the user id and password	Prompt error	Prompt error	Pass
3	User enter user id and password	Valid user id and password which doesn't exist in Database	Registered Successfully	Login Page	Pass
4	User enter user id and password	Invalid user id and password which contains in Database	Prompt error	Prompt error	Pass

Table 6.2: Test Case - Others

Test Case ID	Test Case Name	Test Data	Expected Output	Actual Output	Result
1	User enters invalid information in marks section	Invalid marks Details	Prompt Message showing error	Successful Updated	Fail
2	User enters valid information in Profile	Valid Profile Details	Profile Updated successful	Profile Updated successful	Pass

6 Limitations

- It needs internet to be accessed.
- It does not have a feature to connect with shopkeeper directly for queries and support.
- Information regarding groceries can not be uploaded
- It does not have a feature to buy a items directly from the shop.
- Graphical representation of prices are not shown.

7 Future Enhancements

- Automatic price graph generation for groceries.
- Feature of buying items directly from the shop.
- The benefits of the groceries to the health.

8 User Manual

Part 1 – Register
Upon opening the application, user will be greeted with the registration screen. If the user has no account, user can click on register and register self.
After verification of user id, User's account will be saved in our database. The user can now proceed to login.

Part 2 – Login
User needs to enter email first and then password. If there is an active internet connection, user can proceed to login.

Part 3 – Profile
User can access/add/update personal details (eg. Name, Contact No.). User can update this details anytime.

Part 4 – Grocery Section
User can access and add grocery items to the list details. User can update this details any time.

Part 5 – Grocery Prize updates
User can view Grocery Prize updates received from the shopkeeper on basis user can choose the items.system generates a fairly accurate suggestion.

9 Bibliography

9.1 Web References

[1.] http://developer.android.com/docs/
[2.] https://firebase.google.com/docs
[3.] https://www.youtube.com/user/Firebase
[4.] https://stackoverflow.com/
[5.] https://www.draw.io/
[6.] https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/
[7.] https://www.geeksforgeeks.org/