

Increment 3 Report

Plan Your Shopping

Project Group 18

Vikas Kondapalli (39)

Swatvik Gunamaneni (27)

Gopi Krishna Bodapati (8)

Progress from Increment 2

1. User Interface

In order to accommodate the suggestions offered during the evaluation of the previous increment, we have made changes to the User Interface (UI). We have made improvements to the aesthetics of the registration, login pages and the dashboard page. We have tried to make the fields involved in the registration page similar to the ones we come across in any real time website.

2. List of Cities

The dashboard page in the previous increment displayed only two cities belonging to the state of Missouri. But now we have made it possible to include all the states along with their cities. This gives the user a much better option with all the cities in the U S being displayed in drop down box. This makes the application more generalized and suited to everyone across the U S than to only a user from the state of Missouri.

3. Google Maps

The map we displayed in the previous increment showed the route from the user's current location to only the city where the store is located but not the precise location of the store. This time the user has the complete route leading to the particular store accurately. This improves the usability of our application as now the user will not have to worry about maps giving him only partial information of the intended store's location.

4. Speech Recognition

We have planned to include speech recognition as a way to accept input from the user but this could materialize. We have begun working in this area but could not complete. We hope to include the completed speech recognition module in the final phase.

Features

In the previous phase our website enabled the user to login and register. It also gave him a list of stores for two cities in Missouri. The map displayed earlier pointed only to the city the user wished to go to but not the route to his desired store. We discuss the improvements made from the previous increment and the new features included in the current increment as follows.

List of Stores

The current list of stores is not just limited to the state of Missouri and the few cities in it. We now have a very exhaustive list of cities from all the states of US. The earlier work supported only users from Missouri but now the application can be useful to a user from any state in the United States.

Enhanced User Interface

The user interface earlier was mediocre with few fields being used for registration. But now the number of fields used for registration have been increased. This makes the look and feel of our app similar to any other website the users come across in their day to day lives. The dashboard now looks more user friendly.

Map

The map displayed in the current increment not only points to the city of the desired store but gives the complete route from the user's current location.

Existing APIs

During the course of the current increment we have used Supermarket API, Google Maps, and the MongoDB API.

Supermarket API

We have used this API to get the details of the stores like store Id, location, address e.t.c. Supermarket API is Open Source and is the first API within the Grocery space . Apart from Target, Whole Foods, Trader Joes, Safeway,Costco and Wal-Mart we can have information from over 9000 Supermarkets across the US. Product pictures, details and Information on over 1,000,000 grocery products can be had from this API.

Google Maps

Google offers a desktop web mapping service in the form of Google Maps. Satellite imagery, street maps, route planning for traveling by foot, car, bicycle, or public transportation and 360° panoramic views of streets, real-time traffic conditions through Google Traffic, and are the features offered by Google Maps.

MongoDB API

Our website has to enable its users to register their details and validate those credentials when they want to log in to the website. So we need to store the details the user gives when he registers in a database and fetch them to validate the user. We make use of the MongoDB API for this purpose. The details of the user have been stored in the Cloud using the MongoDB database. The API allows to fetch these details whenever they are needed.

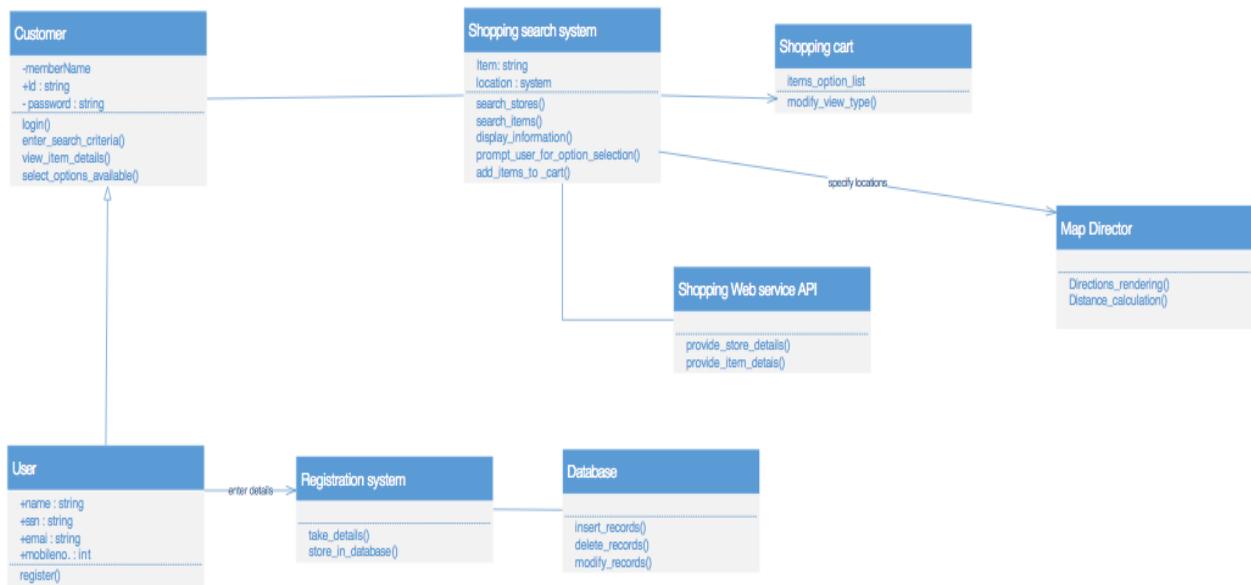
Mashup

In our application we have made a mash up involving two APIs. We have used the location we have from the Supermarket API in the Google Maps API to display the map to the store the wishes to shop at.

Detailed Design of Services

UML Diagrams

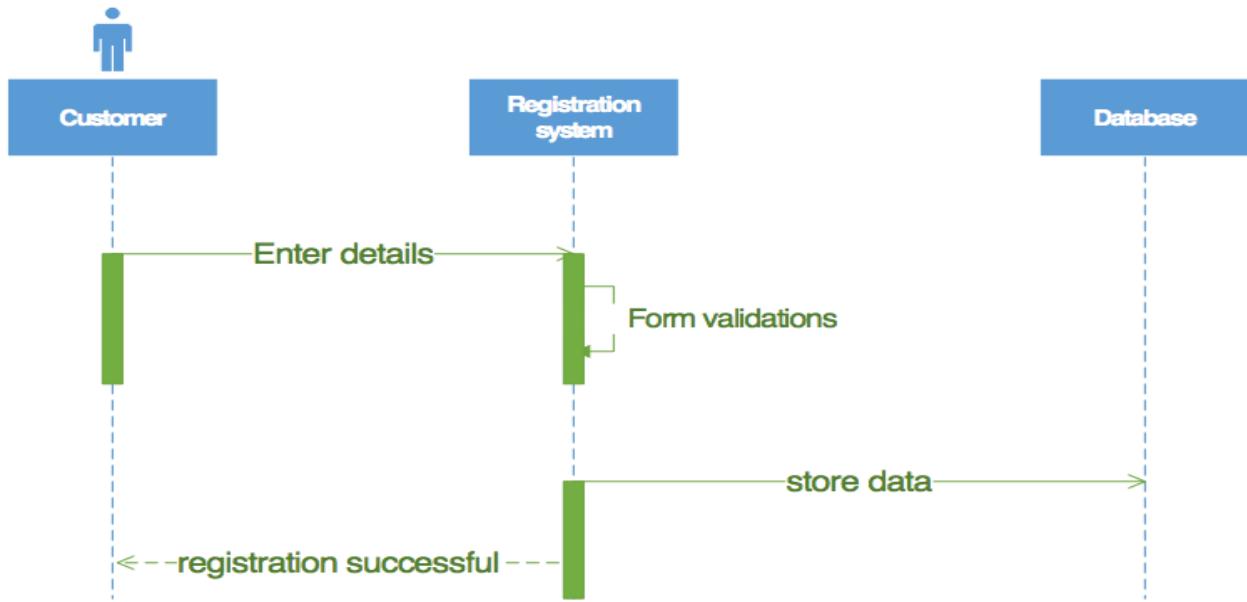
Class Diagram



Customer, User, Shopping system, Cart, Registration are classes used in this class diagram. For a normal user to become the customer he has to get registered and have his details stored in the database. Every user has to interact with at least one registration class. The Customer now interacts with the Shopping System class to search for his choice. Shopping system class has WebService API which is a class based on information provided by the user.

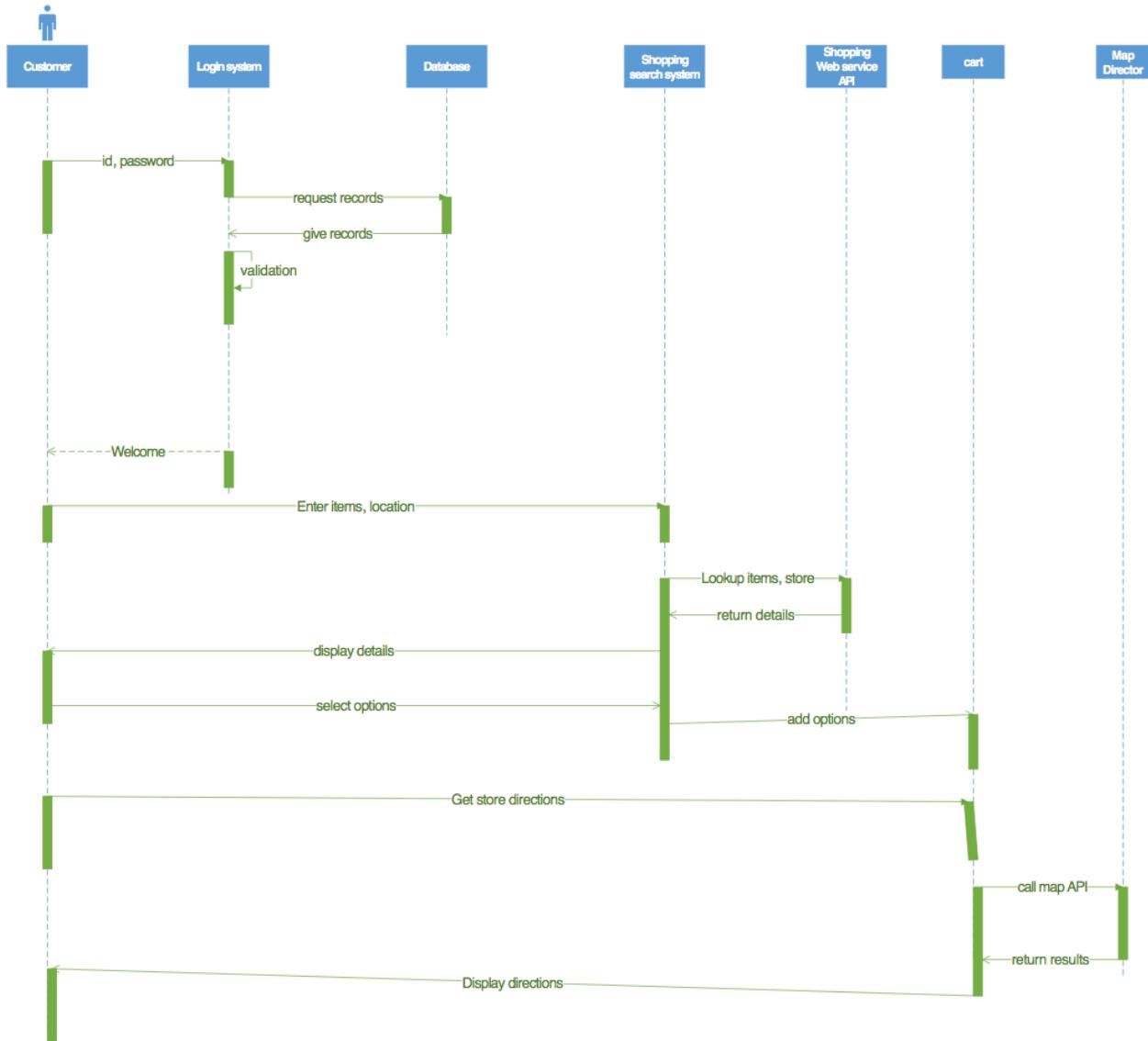
Sequence Diagram

For registration



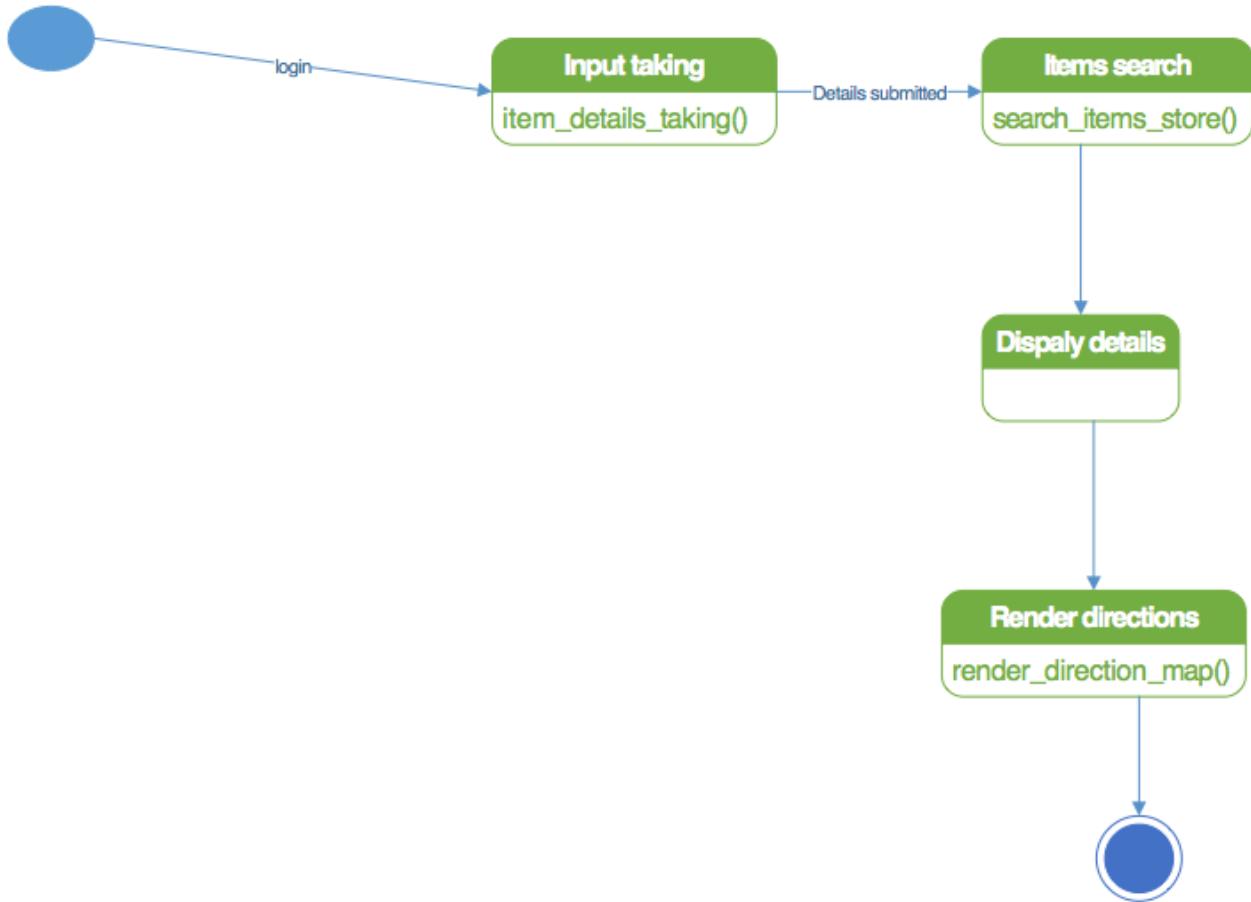
The above sequence diagram illustrates how the customer registers himself and how his details get stored in the database. When he tries to log into this system his details are verified with those already in the database.

For Search and display service



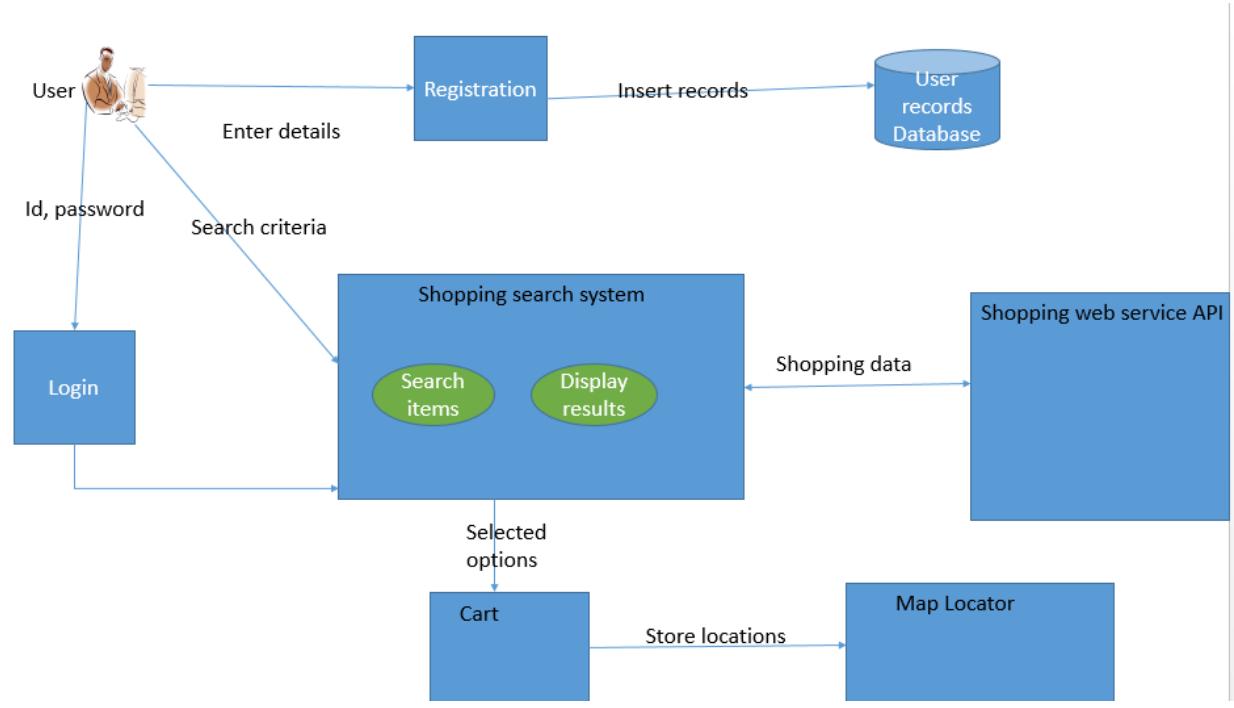
The above diagram shows the scenario where the user searches for a product and has his results displayed on the dashboard.

State Diagram



In the above state diagram we see transition from the start to end and all the various stages involved.

System Architecture Diagram

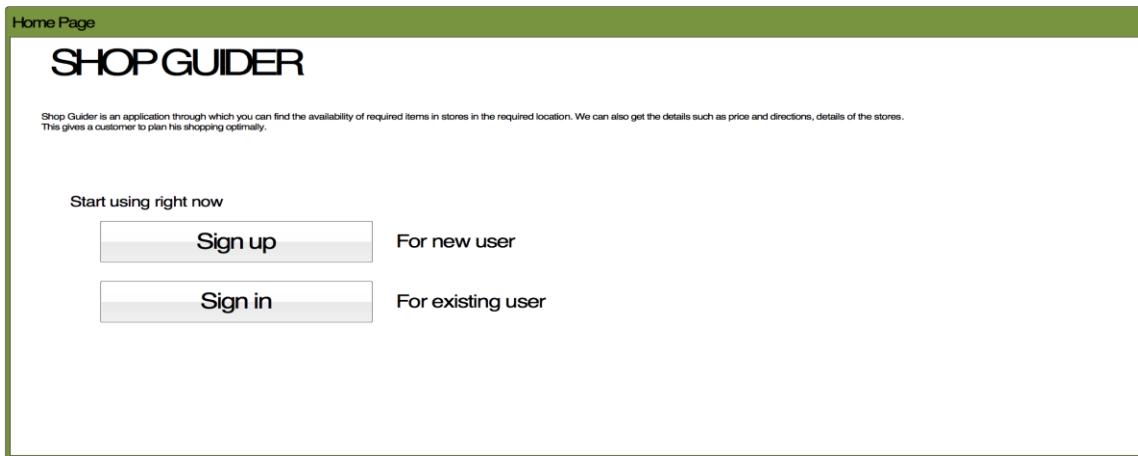


As shown above the system is interacting with various API services to present information to user. A New user has access to the system by registering mandatory information which gets stored in database. The directions to a particular store are had from the map locator API

WireFrames and Mockups

Wireframes

Home page



This is the home page where the user has two options. If he is new he gets registered. If he is not a new user he simply logs into his account. If he chooses to get registered he gets directed to the registration page.

Registration Page

Registration Page

Enter details for Sign up and Registration

First Name	<input type="text"/>	Last Name	<input type="text"/>
Email or Mobile no.	<input type="text"/>		
Re-enter Email or Mobile no.	<input type="text"/>		
Password	<input type="text"/>		
Re-enter Password	<input type="text"/>		

This is the page to which the user is directed when he chooses to sign up. He gives all the details he is asked for which in turn are stored in a database.

Login Page

The screenshot shows a window titled "Login Page" with a green header bar. The main content area has a white background and contains the following elements:

- A title "Sign in for Login" centered at the top.
- Two input fields: "Email ID" and "Password", each with a small placeholder icon to its right.
- A link "Forgot my password?" located below the "Email ID" field.
- A "Login" button located to the right of the "Forgot my password?" link.

This is the login page where an already existing user gives his login details and logs into the system.

Dashboard

The screenshot shows a user interface titled "Dashboard". At the top, there is a "Welcome User" message. Below it are two input fields: "Enter Item:" and "Enter Location:", each with a corresponding "Get details" button. A large grid table follows, with columns labeled "Item", "Store name", "cost", and "location". The first column has 10 rows, and the last three columns have 10 rows each, all containing small checkboxes. Below the grid is a "Check to select" label. At the bottom of the grid is an "Add to cart" button.

This is how the dashboard looks like. The user can choose an item. He can choose his state and city. Now the details of the stores with the product are displayed to him. He picks the item from a particular store and adds his to his cart which is shown next.

Cart Page

This page has all the items the user chooses from whatever store he likes. The user can have all the items in the cart sorted either by cost or by the store he has buy them from.

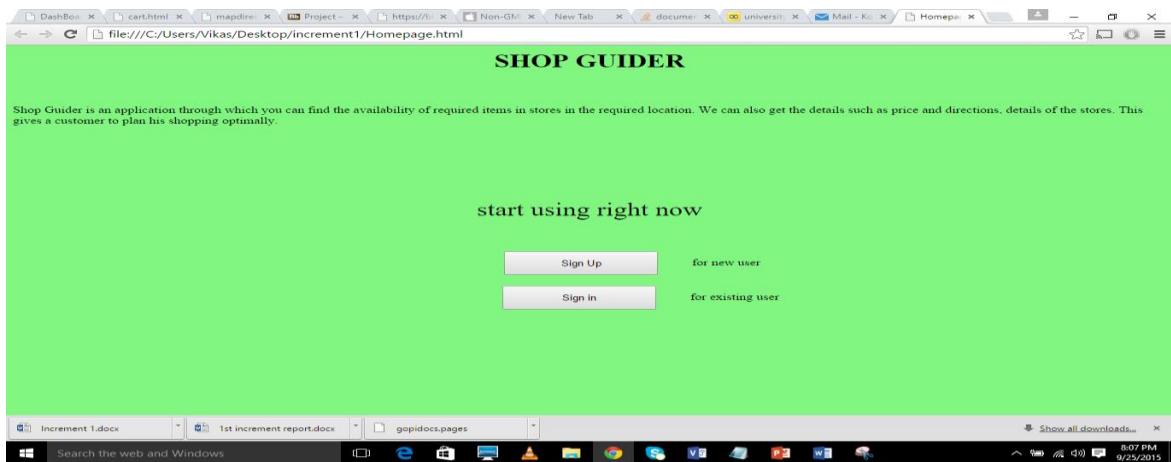
Map Locator Page



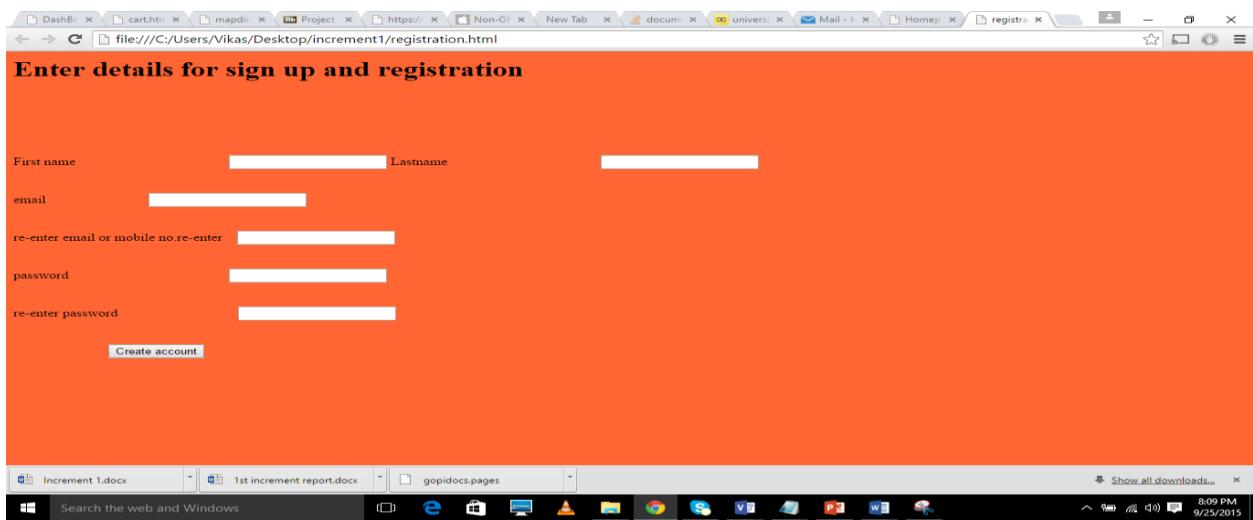
The above page shows the route to the desired store.

Mockups

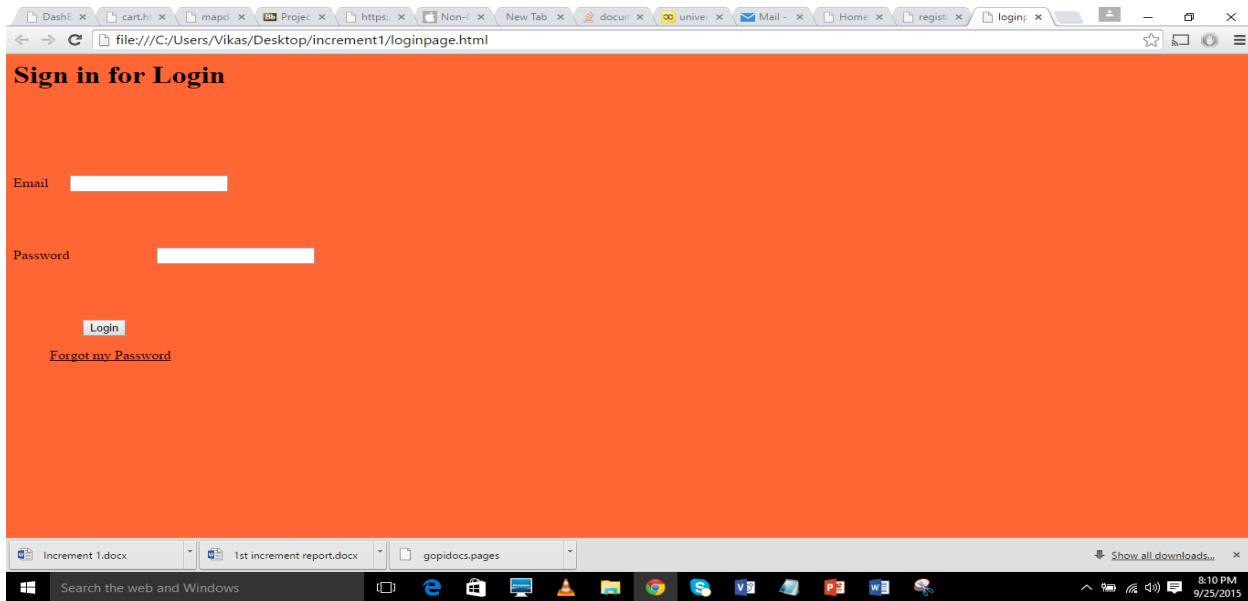
Home page



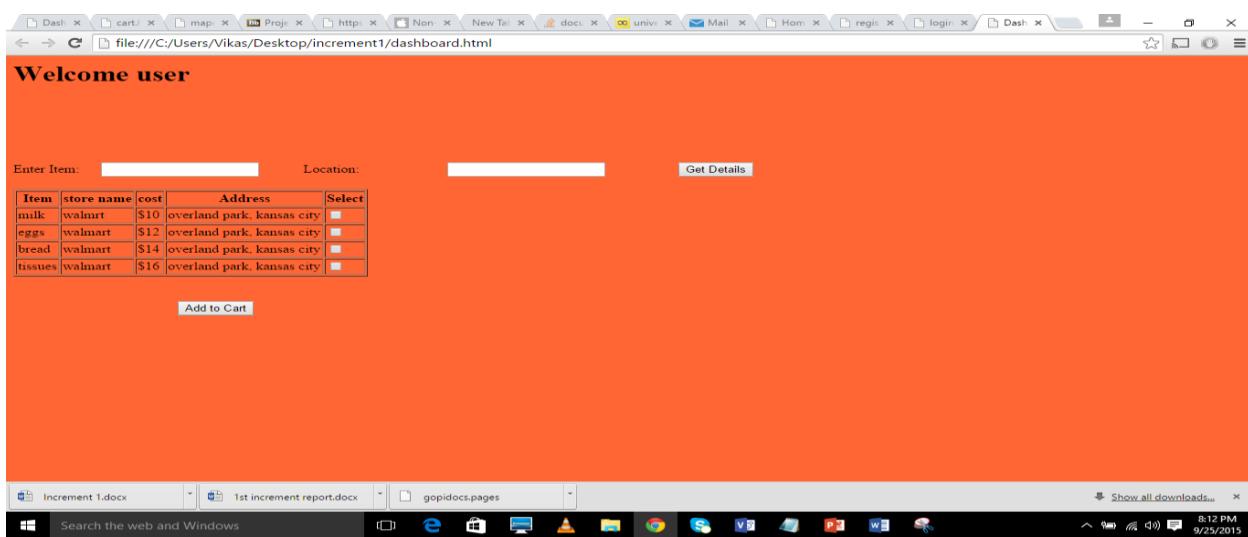
Registration Page



Login Page



Dashboard page



Cart Page

The screenshot shows a web browser window with multiple tabs open at the top. The active tab displays a table titled "Your Selection" containing four rows of grocery items. Below the table are three buttons: "sort by Location", "sort by Cost", and "Go to map locator". The browser's address bar shows the URL as file:///C:/Users/Vikas/Desktop/increment1/cart.html. The taskbar at the bottom shows several open documents and system icons.

Item	store name	cost	Address
milk	walmart	\$10	overland park, kansas city
eggs	walmart	\$12	overland park, kansas city
bread	walmart	\$14	overland park, kansas city
tissues	walmart	\$16	overland park, kansas city

sort by Location
sort by Cost | Go to map locator

Map Locator Page

The screenshot shows a web browser window with multiple tabs open at the top. The active tab displays a form titled "Map Locator Get Directions for" with fields for "Location1", "Location2", "Location3", and "Map locator". Below the form is a large empty area, likely a placeholder for a map or directions. The browser's address bar shows the URL as file:///C:/Users/Vikas/Desktop/increment1/mapdirector.html. The taskbar at the bottom shows several open documents and system icons.

STORYBOARD

In this increment

As a system, it should allow the user to search for the items in all the states in USA.

As a system, it should display the correct state and city combination.

As a system, it should display the store details in the specified location only.

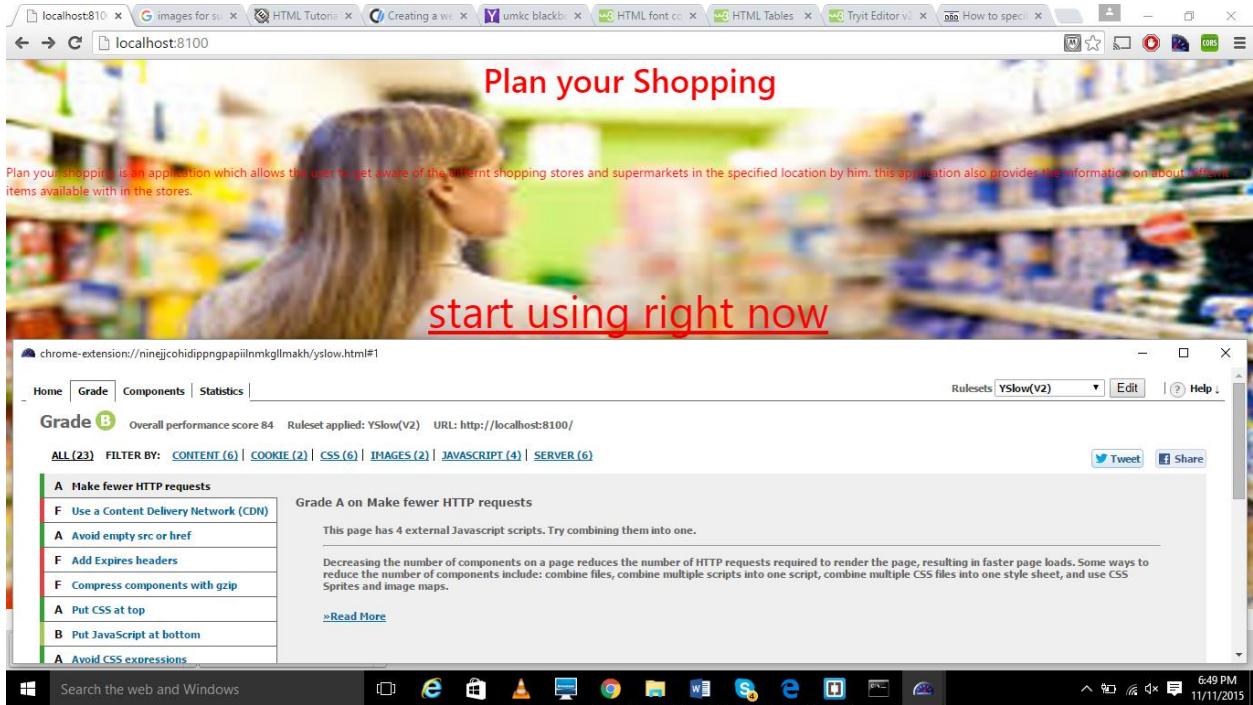
As a system, it should display the items specified in the specified store.

Design Pattern:

In our project, **Singleton design pattern** is used for displaying the welcome message by specifying the name of the user. In this application only one object for displaying welcome message is created and the same object is used in all the objects. This design pattern is designed but not yet implemented, but will be implemented in the further increment.

Testing

Performance Testing



Plan your Shopping

Plan your shopping is an application which allows the user to get aware of the different shopping stores and supermarkets in the specified location by him. this application also provides the information on about different items available with in the stores.

start using right now

chrome-extension://ninejcohidippngpapiiinmkglmakh/yslow.html#1

Home Grade Components Statistics Rulesets YSlow(V2) Edit Help

Statistics The page has a total of 10 HTTP requests and a total weight of 2473.5K bytes with empty cache

WEIGHT GRAPHS

HTTP Requests - 10	
Total Weight - 2473.5K	
1 HTML/Text	3.1K
4 JavaScript File	2111.0K
2 Stylesheet File	223.7K
1 CSS Image	14.8K
1 Favicon	0.02K
1 undefined	120.7K

HTTP Requests - 10	
Total Weight - 60.8K	
1 HTML/Text	3.1K
4 JavaScript File	42.7K
2 Stylesheet File	0.0K
1 CSS Image	14.8K
1 Favicon	0.02K
1 undefined	0.0K

Plan your Shopping

Plan your shopping is an application which allows the user to get aware of the different shopping stores and supermarkets in the specified location by him. this application also provides the information on about different items available with in the stores.

start using right now

chrome-extension://ninejcohidippngpapiiinmkglmakh/yslow.html#1

Home Grade Components Statistics Rulesets YSlow(V2) Edit Help

Components The page has a total of 10 components and a total weight of 2473.5K bytes

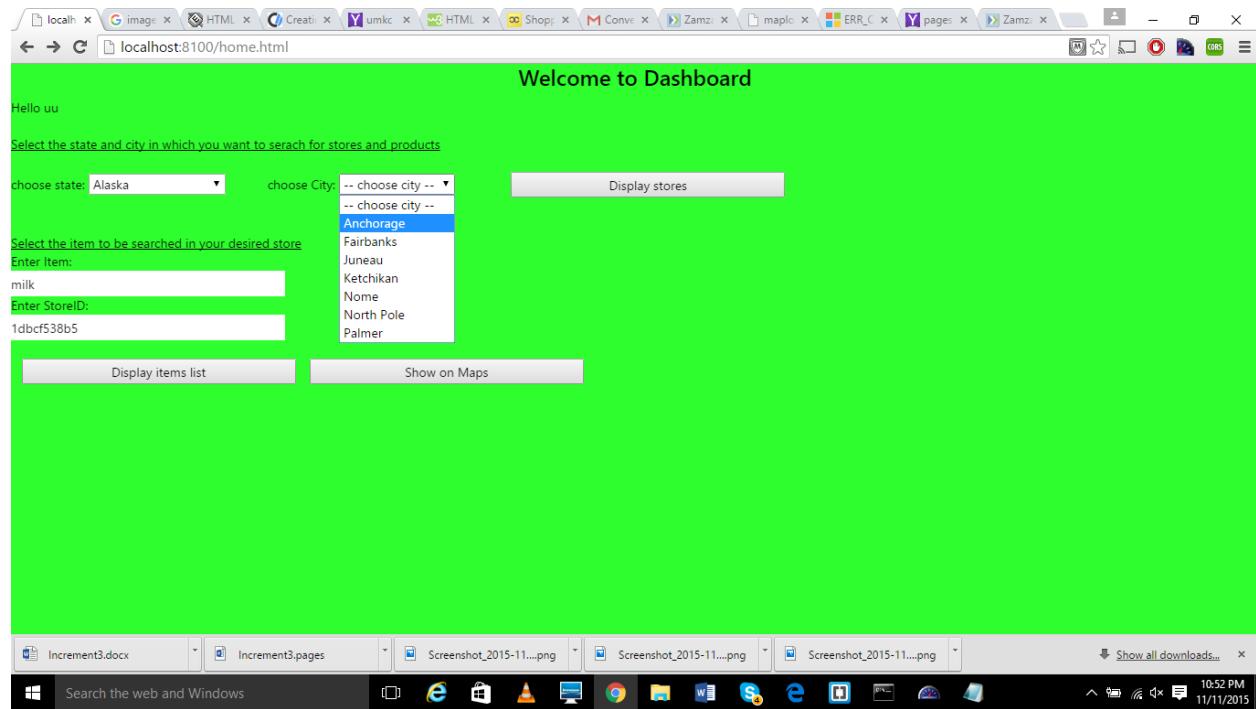
TYPE	SIZE (KB)	GZIP (KB)	COOKIE RECEIVED (bytes)	COOKIE SENT (bytes)	HEADERS	URL	EXPIRES (Y/M/D)	RESPONSE TIME (ms)	ETAG	ACTION
[+ doc (1)]	3.1K									
[+ js (4)]	2111.0K									
[+ css (2)]	223.7K									
[+ cssimage (1)]	14.8K									
[+ favicon (1)]	0.02K									
[+ font (1)]	120.7K									

The above screenshots of the application show the performance testing of the application using Yslow extension to the chrome browser. The overall grade given to this Application in terms of performance is B.

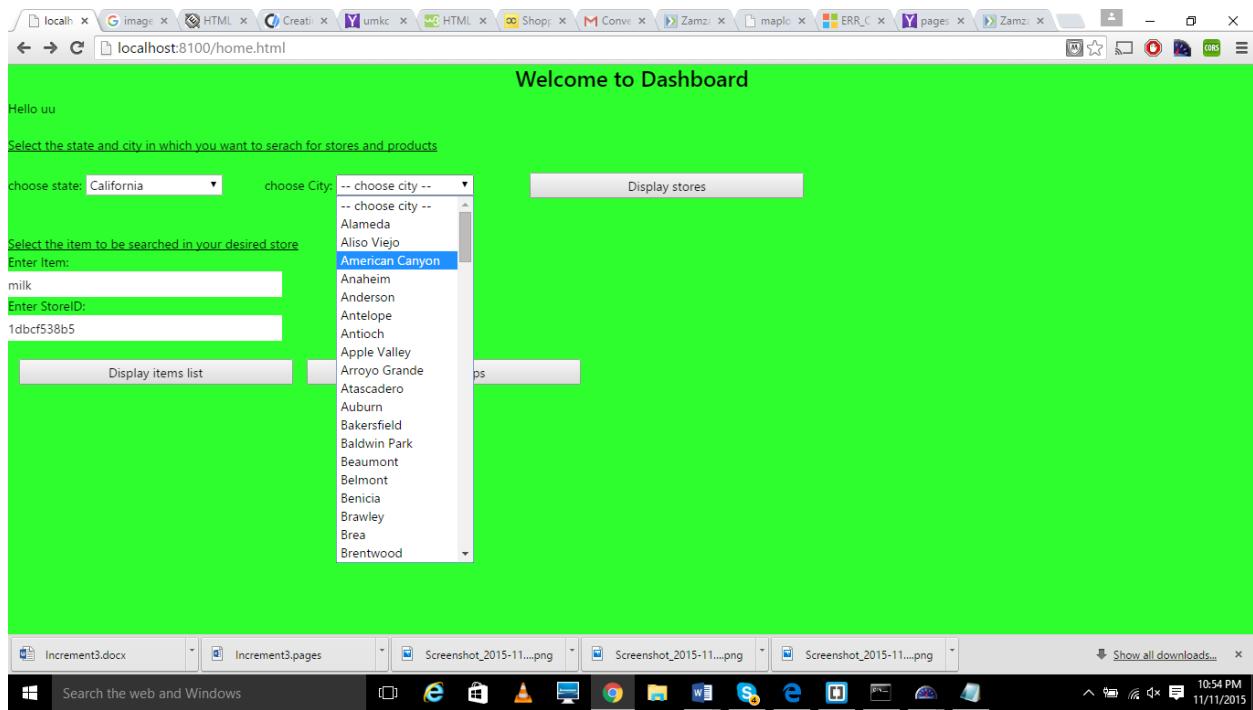
Unit Testing: (Manual)

DashBoard page:

Testing the dynamic refresh of the Choose city select box.



When state “Alaska” is selected, the Choose city select box is dynamically refreshed with the cities in Alaska state only.



When state “California” is selected, the Choose city select box is dynamically refreshed with the cities in California state only.

Testing the DashBoard for displaying of the stores in the correct location as specified by the user.

Welcome to Dashboard

Hello uu

Select the state and city in which you want to search for stores and products

choose state: Florida choose City: Jacksonville Display stores

Select the item to be searched in your desired store

Enter Item:

Enter StoreID:

Storename	Address	City	State	Zip	StoreID
Publix	6001 Argyle Forest Blvd Ste 45	Jacksonville	FL	32244-6127	16eafc0001
Publix	2771 Monument Rd	Jacksonville	FL	32225-5549	1dbcfc38b5
Publix	4320 Deerwood Lake Pkwy	Jacksonville	FL	32216-1177	4f8864d47f
Publix	9964 Old Baymeadows Rd	Jacksonville	FL	32256-8103	06026b5caf
Publix	731 Duval Station Rd Ste 4	Jacksonville	FL	32218-0804	95a53efc2e
Publix	5210 Norwood Ave	Jacksonville	FL	32208-5036	790a4465a6
Publix	13170 Atlantic Blvd Ste 29	Jacksonville	FL	32225-4150	20a049a428
Publix	1100 Dunn Ave Bldg 100	Jacksonville	FL	32218-4832	842191ccc2
Publix	12620 Beach Blvd Ste 12	Jacksonville	FL	32246-7130	b8878edbd0
Publix	11406 San Jose Blvd	Jacksonville	FL	32223-7963	05dc93b380
Publix	14444 Beach Blvd Ste 6	Jacksonville	FL	32250-2010	f8fb87638d
Publix	7749 Normandy Blvd	Jacksonville	FL	32221-7657	723b7177bd
Publix	7639 102nd St. Ste 2A	Jacksonville	FL	32210-8710	98e2310468

When “Jacksonville” city of the Florida state is selected for displaying of the stores is selected, only the stores with in that city are displayed.

Welcome to Dashboard

Hello uu

Select the state and city in which you want to serach for stores and products

choose state: California choose City: Aliso Viejo Display stores

Select the item to be searched in your desired store

Enter Item:
milk

Enter StoreID:
1dbcfc538b5

Display items list Show on Maps

Storename	Address	City	State	Zip	StoreID
Trader Joe's	26541 Aliso Creek Road	Aliso Viejo	CA	92656	05c7e08fb3

When “Alieso Viejo” city of the California state is selected for displaying of the stores is selected, only the stores with in that city are displayed.

Implementation

Server Side Implementation:

The server side implementation is nothing but the calling of the Rest services (APIs). The APIs existing in the server are called by the client and they are fetched on to the client and executed there.

The APIs existing on the server side “Supermarketapi”, “GoogleMaps” API, “MongoDB” API are used in this project.

MongoDB API is used for connecting to the Mongo Database existing in the remote location (cloud). We use MongoDB API for in the login and Registration pages. The MongoDB API should allow the client to perform CURD operations on the corresponding MongoDB server.

Link for MongoDB Data API used in this project:

https://api.mongolab.com/api/1/databases/vikas2/collections/users?apiKey=HxPILPvmcIj3SyZBMrPV38t7_BiBKUJ

Supermarket API is used for getting the details of all the stores in all the states of the USA and also the list of all the items in a particular store. This API contains all these details in XML format and these are stored on the server side. Whenever Supermarket API is called the server responds with data in XML upon request by the client.

Link for Supermarketapi used in this project:

<http://supermarketapi.com/Default.aspx>

Google Maps API in this project is used to render Google maps in the application. The Google Maps API is used for locating the store selected on the Google Maps.

Link for Google Maps API used in this project:

<https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=true>

Client Side Implementation:

In client side there are various functionalities to be implemented, these functionalities are mostly fetching the data from server side and processing, acting on that data by performing some client side implementations. The Client side implementations in this project are

Registration:

This functionality implementation allow the user to get himself registered in to the application and become a valid customer of the application. Here the user enters all his details and then submits them, these details are stored in the Mongo DB using MongoDB API. Then the user can start using the services of this application by logging in to the system.

Login:

Login functionality implementation allow the user to enter his username and password which he has mentioned at the time of the registration. The username and password combination should exactly match with the username and password at the time of registration. This functionality involves fetching the corresponding the details from the MongoDB using MongoDB API and validating the details entered by the user for login. If these details (credentials: username, password) are found to be correct then the login system allows the user to start using the system by redirecting then user to Home page.

Stores and Items search (on Home page):

This functionality implementation allows the user to search for the stores in a specified location. The user can specify the location to be searched by entering the state and the city in that state in which search has to be carried out. The details of all the stores and the items available in them are fetched using the Supermarket API. In this increment the search has been extended to entire USA i.e details of stores in all the states of the USA are available instead of search for stores only in the Missouri state in the second increment. These user specify this search criteria by using two select boxes, in the first select box the user can select a state in USA, the second select box dynamically refreshes itself with the list all of the cities in that selected state in first select box. Then search is performed on the data retrieved from the Supermarket API by using the values from the two select boxes mentioned above and results are displayed.

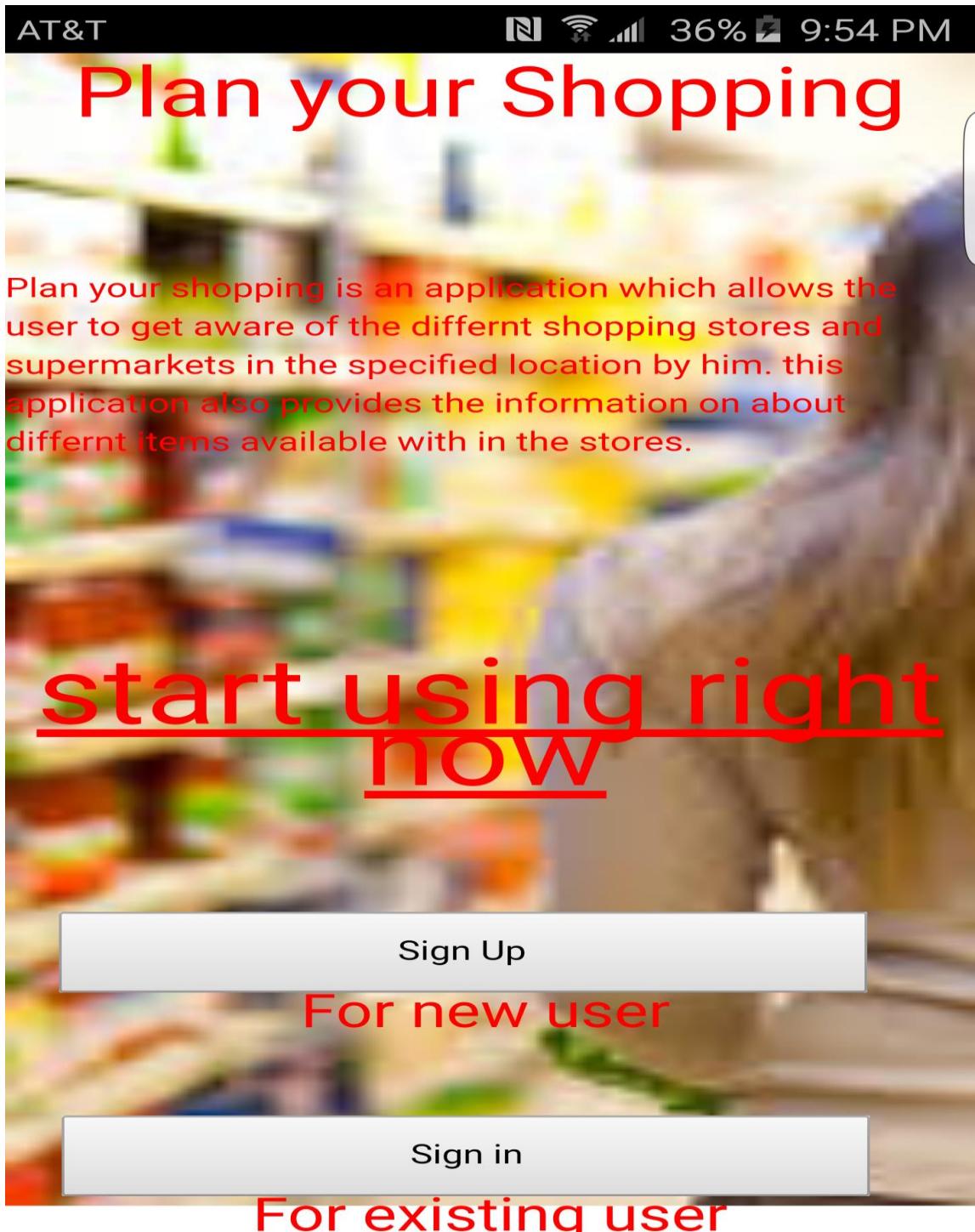
Store location on Google Maps:

The store selected by the user are located on the Google maps using Google maps API. The address, city, state and zip of the store are retrieved from the Supermarket API and these details are Google Maps API, the Google Maps API locates the store exactly on the Google Maps using

the details. There is also implementation for the detection of the GeoLocation of the user and locating him on the Google Maps. The directions are rendered from his location to the location of the store and also distance to be travelled to reach the store is displayed.

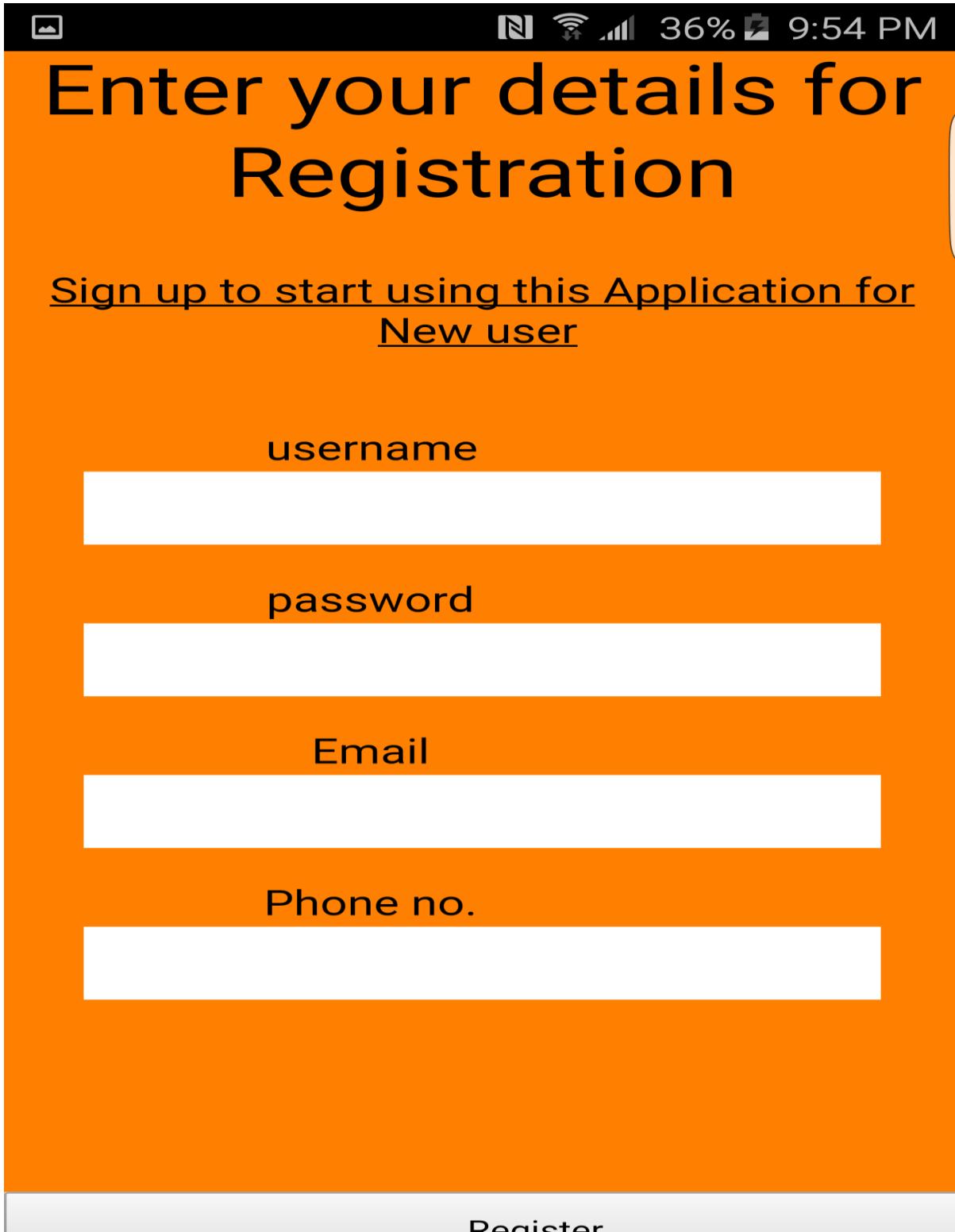
Screenshots of the Application:

Index Page:



Index page introduces the title of the Application and a brief explanation of the Application. Index page provides navigations to the Registration page and the Login pages. For this navigation two buttons “Sign Up” for registration of new user by clicking on which it navigates to Registration page and “Sign in” for Login for existing user by clicking on which it navigates to the Login page are used.

Registration Page:





36%



9:54 PM

Registration

Sign up to start using this Application for
New user

username

Gopi

password

.....

Email

g.bodapati@yahoo.com

Phone no.

8168598182



36%



9:55 PM

Registration

Sign up to start using this Application for
New user

username

Gopi

password

Alert

....

successfully registered

Email

OK

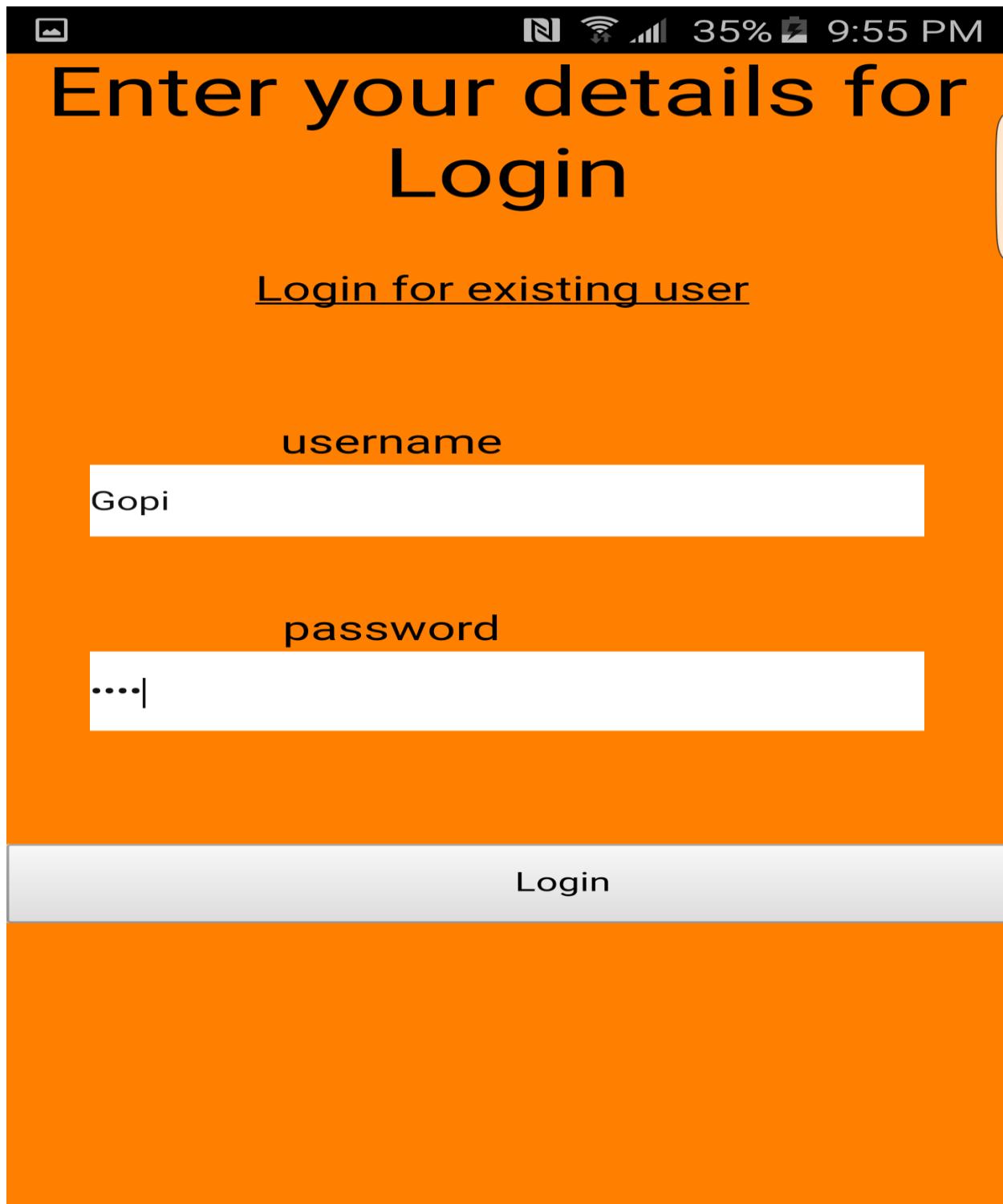
Phone no.

8168598182

Register

Registration Page allows the user to enter his details such as username, password, Email-ID and Phone no. The Registration page contains textboxes for reading these details and after entering all the details the user can click on the Register button to store his details in the Database and get himself registered. The Registration page in turn communicates with the MongoDB API to get Connected with the MongoDB. Upon successful registration the user will be shown an alert box indicating that he has completed registration successfully.

Login page:





35%



9:55 PM

Enter your details for Login

Login for existing user

username

Gopi

Alert

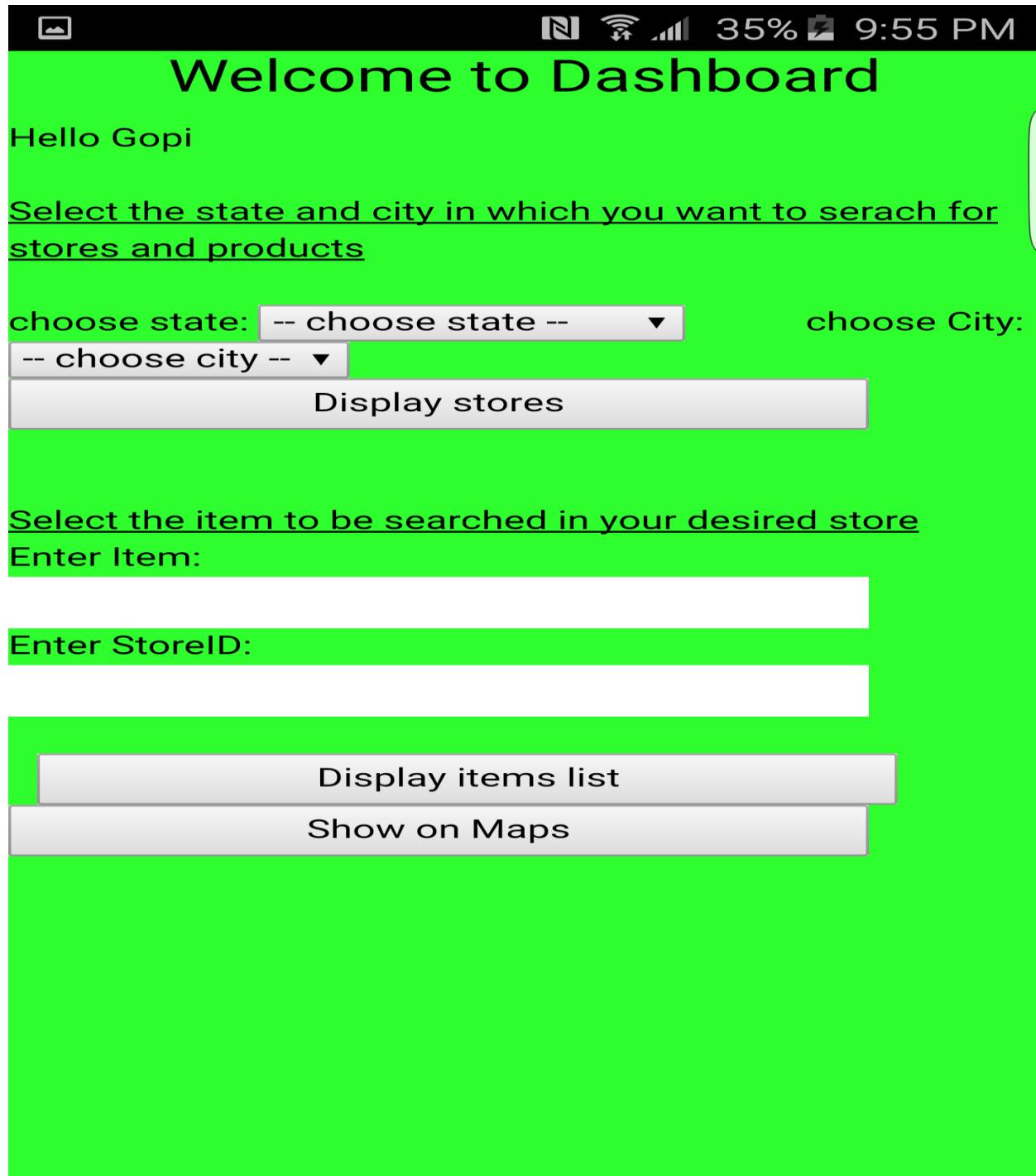
login successful

OK

Login

The Login page allows the user to login into the system by specifying his correct credentials i.e username and password. The Login page contains textboxes for entering the username and password of the user. The login page reads these details and validates them with the details fetched from the MongoDB i.e it compares the username and password combination entered by the user with all username and password combinations in the documents of the MongoDB. If the details entered by the user are found to be correct then the user will be directed to home page where he will be able to access all the services provided by the Application.

Home Page:





32%



10:19 PM

Select the state and city in which you want to serach for stores and products

choose state: Florida

choose City:

Jacksonville

Display stores

Select the item to be searched in your desired store

Enter Item:

Enter StoreID:

Display items list

Show on Maps

Store name	Address	City	State	Zip	Stock
Publix	6001 Argyle Forest Blvd Ste 45	Jacksonville	FL	32244-16e	6127
Publix	2771 Monument Rd	Jacksonville	FL	32225-1db	5549

Home page of the application provides the user with all of the core services of the Application. Home page allows the user to specify the location (state and a particular city in that state) in which the search for the stores has to be carried out. For specifying this location, Home page consists of two select boxes. In the first select box the user can select a state in USA, then the second select box dynamically refreshes itself with the list all of the cities in that selected state in first select box. Then the user clicks on the “Display Stores” button to see the list of all stores available in his specified location. For fetching all these details Home page contacts the Supermarket API where all these data is stored in XML format, this data is fetched and parsed and displayed to the user.



31%



10:21 PM

Enter Item:

Milk

Enter StoreID:

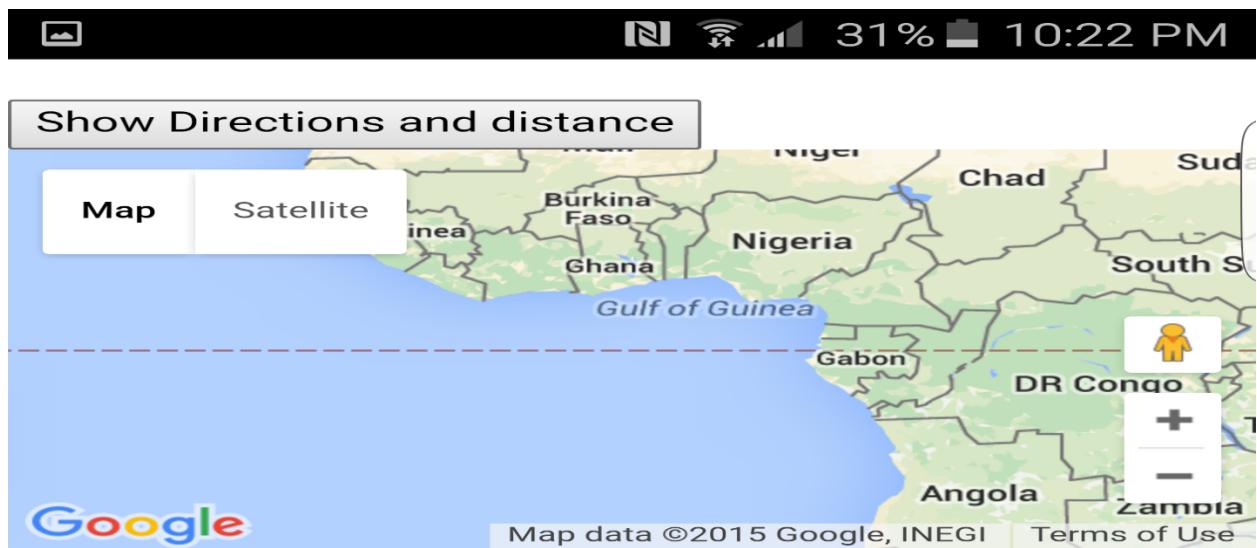
1dbcf538b5

[Display items list](#)[Show on Maps](#)

Itemname	ItemDescription	ItemCategory	ItemID
Enfamil	Enfamil Enfacare	Baby	75360
Enfacare	Lipil Milk-Based		
Lipil Milk- Based	Powder - 12.8 Oz Canister		
Powder - 12.8 Oz Canister			
Enfamil A.R.	Enfamil A.R. Lipil	Baby	75361
Lipil Milk- Based	Milk-Based Powder w/Iron - 12.9 Oz		
Powder w/Iron - 12.9 Oz Canister	Canister		
Enfamil Lipil w/Iron Milk- Based	Enfamil Lipil w/Iron	Baby	75370
Powder - 25.7 Oz Canister	Milk-Based Powder - 25.7 Oz Canister		
Enfagrow Premium	Enfagrow Premium	Baby	75371
	Next Step Lipil Milk		

The Home page also allows the user to search for the items available in a specific store. For this, Home page consists of two textboxes one for entering any keyword related to required item and the other for entering the Store ID of the store in which the search has to be carried out for the required item. For this information also Supermarket API is contacted and the data is fetched, parsed and displayed.

MapLocator Page:



Distance:

The MapLocator page renders the direction from the current location of the user to the location of the store which he has selected on Google Maps. For this purpose it uses Google Maps API to get the Google Maps and perform operations on it. It also shows the distance to be travelled to reach the store.

Project Management

Kanban tool has been used to manage this project. The progress up to second increment was mentioned in the second increment report. There are two more increments remaining to complete this project. In this increment(third) the following tasks were planned, designed, implemented and completed.

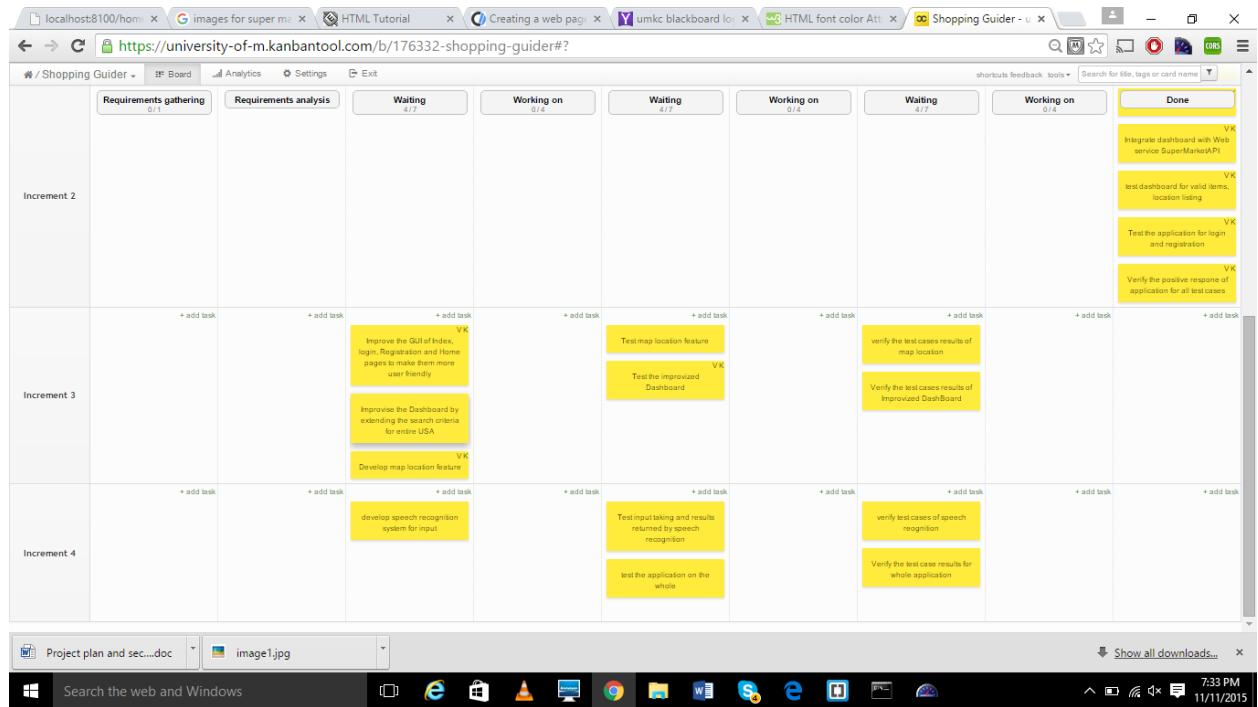
- **Development:**

1. Improvise the Dashboard page for extending the search criteria.
2. Improvise the MapLocation page to detect the user current location and render the directions and distance from his location to the store location.
3. Improve the GUI of the Application to make it more user friendly.

- **Testing:**

1. Testing the functionality of improvised DashBoard page.
2. Testing the functionality of MapLocation page.

Initially all the tasks to be completed were in the Waiting phase.



The task which was first carried out was improving the GUI of the important pages such as Index page, Login page, Registration page and Home page and making them more user friendly.

	Requirements gathering 0/1	Requirements analysis 3/7	Waiting 3/7	Working on 1/4	Waiting 4/7	Working on 0/4	Waiting 4/7	Working on 0/4	Done VK
Increment 2									Develop dashboard page VK
Increment 3	+ add task	+ add task	+ add task	+ add task V.K. Improve the Dashboard by extending the search criteria for entire USA Develop map location feature	+ add task V.K. Improve the GUI of Index, login, Registration and Home pages to make them more user friendly	+ add task Test map location feature Test the improved Dashboard	+ add task Verify the test cases results of map location Verify the test cases results of improved Dashboard	+ add task	Integrate dashboard with Web service SuperMarketAPI test dashboard for valid items, location listing Test the application for login and registration Verify the positive response of application for all test cases
Increment 4	+ add task	+ add task	+ add task	+ add task develop speech recognition system for input	+ add task Test input taking and results returned by speech recognition Test the application on the whole	+ add task Verify test cases of speech recognition Verify the test case results for whole application	+ add task	+ add task	+ add task

This task was completed to our strength and moved to done phase.

	Requirements gathering 0/1	Requirements analysis 3/7	Waiting 3/7	Working on 0/4	Waiting 4/7	Working on 0/4	Waiting 4/7	Working on 0/4	Done VK
Increment 2									Develop dashboard page VK
Increment 3	+ add task	+ add task	+ add task	+ add task V.K. Improve the Dashboard by extending the search criteria for entire USA Develop map location feature	+ add task V.K. Test map location feature Test the improved Dashboard	+ add task Verify the test cases results of map location Verify the test cases results of improved Dashboard	+ add task	+ add task V.K. Integrate dashboard with Web service SuperMarketAPI test dashboard for valid items, location listing Test the application for login and registration Verify the positive response of application for all test cases	
Increment 4	+ add task	+ add task	+ add task	+ add task develop speech recognition system for input	+ add task Test input taking and results returned by speech recognition Test the application on the whole	+ add task Verify test cases of speech recognition Verify the test case results for whole application	+ add task	+ add task	+ add task V.K. Improve the GUI of Index, login, Registration and Home pages to make them more user friendly

The next task was to improvise the Dashboard page by extending the search criteria to the entire USA i.e all the states in the USA.

	Requirements gathering	Requirements analysis	Waiting	Working on	Waiting	Working on	Waiting	Working on	Done
Increment 2									
Increment 3	+ add task	+ add task	+ add task Develop map location feature	+ add task Improve the Dashboard by extending the search criteria for entire USA	+ add task Test map location feature	+ add task Test the improved Dashboard	+ add task verify the test cases results of map location	+ add task Verify the test cases results of Improved Dashboard	+ add task Improve the GUI of index, login, Registration and Home pages to make them more user friendly
Increment 4	+ add task	+ add task	+ add task develop speech recognition system for input	+ add task	+ add task Test input taking and results returned by speech recognition	+ add task test the application on the whole	+ add task verify test cases of speech recognition	+ add task Verify the test case results for whole application	+ add task

This task of improvising the dashboard for extending the search criteria was completed successfully and moved to done phase.

	Requirements gathering	Requirements analysis	Waiting	Working on	Waiting	Working on	Waiting	Working on	Done
Increment 2									
Increment 3	+ add task	+ add task	+ add task Develop map location feature	+ add task Test map location feature	+ add task Test the improved Dashboard	+ add task verify the test cases results of map location	+ add task Verify the test cases results of Improved Dashboard	+ add task Improve the GUI of index, login, Registration and Home pages to make them more user friendly	+ add task Improve the Dashboard by extending the search criteria for entire USA
Increment 4	+ add task	+ add task	+ add task develop speech recognition system for input	+ add task	+ add task Test input taking and results returned by speech recognition	+ add task test the application on the whole	+ add task verify test cases of speech recognition	+ add task Verify the test case results for whole application	+ add task

The next task was developing the Map Location page for displaying the directions from the current location of the user to the location of the store selected by the user.

	Requirements gathering	Requirements analysis	Waiting	Working on	Waiting	Working on	Waiting	Working on	Done	
Increment 2									✓K Develop dashboard page	
Increment 3	+ add task	+ add task	+ add task	+ add task	+ add task	+ add task	+ add task	+ add task	✓K Integrate dashboard with Web service SuperMarketAPI ✓K Test dashboard for valid items, location listing ✓K Test the application for login and registration ✓K Verify the positive response of application for all test cases	
Increment 4	+ add task	+ add task	+ add task	+ add task	+ add task	+ add task	+ add task	+ add task	✓K Improve the GUI of Index, login, Registration and Home pages to make them more user friendly ✓K Improve the Dashboard by extending the search criteria for entire USA	
										+ add task

This task was successfully completed and moved to done phase.

	Requirements gathering	Requirements analysis	Waiting	Working on	Waiting	Working on	Waiting	Working on	Done	
Increment 2									✓K Integrate dashboard with Web service SuperMarketAPI ✓K Test dashboard for valid items, location listing ✓K Test the application for login and registration ✓K Verify the positive response of application for all test cases	
Increment 3	+ add task	+ add task	+ add task	+ add task	+ add task	+ add task	+ add task	+ add task	✓K Improve the GUI of Index, login, Registration and Home pages to make them more user friendly ✓K Improve the Dashboard by extending the search criteria for entire USA ✓K Develop map location feature	
Increment 4	+ add task	+ add task	+ add task	+ add task	+ add task	+ add task	+ add task	+ add task	✓K Develop speech recognition system for input ✓K Test input taking and results returned by speech recognition ✓K Test the application on the whole	
										+ add task

The next task was to test the correct functioning of the improvised Dashboard.

The screenshot shows a Kanban board interface for a project titled "Shopping Guider". The board is organized into three vertical columns: Requirements gathering, Requirements analysis, and Waiting. Each column is further divided into sub-columns: Requirements gathering, Requirements analysis, Waiting, Working on, and Done. The board is divided into three horizontal sections representing increments: Increment 2, Increment 3, and Increment 4. Each increment has a row of cards corresponding to the sub-columns. A yellow box highlights the "Test the improved Dashboard" card in the "Working on" column of Increment 3. The card contains the text "Test the improved Dashboard" and a "VK" icon. Other cards in the increment include "Test the Map location feature", "verify the test cases results of map location", and "Verify the test cases results of Improved Dashboard". The "Done" column contains several completed tasks, each with a "VK" icon. The bottom of the screen shows a Windows taskbar with various icons and a system tray indicating the date and time as 11/11/2015 at 8:14 PM.

This task was successfully completed and moved to done phase.

The screenshot shows a Kanban board with three increments of tasks:

- Increment 2:** Contains 4 tasks in the "Done" column. The tasks are:
 - Integrate dashboard with Web service SuperMarketAPI
 - test dashboard for valid items, location finding
 - Test the application for login and registration
 - Verify the positive response of application for all test cases
- Increment 3:** Contains 8 tasks across four columns. The tasks are:
 - + add task
 - + add task

The tasks are:
 - Test the Map location feature
 - verify the test cases results of map location
 - Verify the test cases results of Improved Dashboard
 - Improve the Search feature, login, Register and Home pages to make them more user-friendly
 - Improve the Dashboard by extending the search criteria for online USA
 - Develop map location feature
 - Test the improved Dashboard
- Increment 4:** Contains 4 tasks across four columns. The tasks are:
 - + add task
 - + add task

The tasks are:
 - develop speech recognition system for input
 - Test input being and results returned by speech recognition
 - Test the application on the
 - Verify and outcome of speech recognition
 - Verify the test case results for whole application

The next task was testing the Map Location page for its correct functionality.

A screenshot of a Kanban board titled "Shopping Guider". The board is organized into columns: Requirements gathering, Requirements analysis, Waiting, Working on, Waiting, Working on, Waiting, Working on, and Done. Rows represent different increments: Increment 2, Increment 3, and Increment 4. Tasks are represented by yellow cards with descriptions and checkmarks. In Increment 2, there are tasks for integrating the dashboard with SuperMarket API, testing the application for login and registration, and verifying positive responses. In Increment 3, tasks include testing the map location feature, verifying test cases for improved dashboard, improving search criteria, developing map location features, and testing the improved dashboard. In Increment 4, tasks involve developing speech recognition systems, testing input taking, and verifying test cases for speech recognition.

This task was successfully completed and moved to done phase.

A screenshot of the same Kanban board after some tasks have been completed. The "Done" column now contains several tasks from previous increments, indicating they have been successfully finished. The board structure remains the same, with columns for Requirements gathering, Requirements analysis, Waiting, Working on, Waiting, Working on, Waiting, Working on, and Done. Rows for Increments 2, 3, and 4 are visible, showing the progression and completion of various project tasks.

The next task was verifying the results of test cases of DashBoard page and the MapLocation page.

	Requirements gathering 0/1	Requirements analysis	Waiting 1/7	Working on 0/4	Waiting 2/7	Working on 0/4	Waiting 2/7	Working on 2/4	Done
Increment 2									<ul style="list-style-type: none"> Integrate dashboard with SuperMarket API Test dashboard for valid item location listing Test the application for login and registration Verify the positive response of application for all test cases
Increment 3	+ add task	+ add task	+ add task	+ add task	+ add task	+ add task	+ add task	+ add task	<ul style="list-style-type: none"> verify the test cases results of map location Verify the test cases results of improved DashBoard Improve the GUI of DashBoard, login, Registration and Home pages to make them more user friendly Improve the Dashboard by extending the search criteria for entire USA Develop map location feature Test the improved Dashboard Test the Map location feature
	+ add task	+ add task	+ add task	+ add task	+ add task	+ add task	+ add task	+ add task	<ul style="list-style-type: none"> develop speech recognition feature for input Test input box and results returned by speech recognition verify test cases of speech recognition

This task was successfully completed and moved to done phase.

The screenshot shows a Kanban board interface with the following details:

- Columns:** Requirements gathering, Requirements analysis, Waiting, Working on, Waiting, Working on, Waiting, Working on, Done.
- Rows (Increments):** Increment 2 and Increment 3.
- Tasks in Increment 2:** One task in the Done column: "Integrate dashboard with Web service SuperMarketAPI".
- Tasks in Increment 3:** Ten tasks in the Done column:
 - test dashboard for valid items, location listing
 - Test the application for login and registration
 - Verify the positive response of application for all test cases
 - Improve the GUI of Index, login, Registration and Home pages to make them more user friendly
 - Improve the Dashboard by extending the functionality for entire USA
 - Develop map location feature
 - Test the improved Dashboard
 - Test the Map location feature
 - verify the test cases results of map location
 - Migrate from local storage results of

Work Completed:

Description:

1. Improvised the DashBoard page by extending the search criteria of the Stores using the Supermarket API to all the states in the USA.
2. Developed the MapLocation page to show the directions and distance from the current location of the user to the location of the store selected by the user.
3. Enhanced the GUI of the Application to make it more user friendly.

Responsibilities:

Vikas Kondapalli: Improvising the DashBoard page to extend the search criteria using the SuperMarket API, Enhancing the GUI of the Application, Testing, documentation.

Gopi Krishna Bodapati: Developing the Map location page to detect user location and to render distance, directions from user location to the store location.

Swatvik Gunamaneni:

Enhancing the GUI of the Application to make more user friendly using CSS, Testing, documentation.

Time taken:

- Improving the DashBoard page: 110 hours
- Developing the Maplocation page : 80 hours
- Enhancing the GUI : 10 hours
- Testing : 20 hours
- Documentation: 20 hours

Contributions:

- Vikas kondapalli : 33.33%
- Gopi Krishna Bodapati: 33.33%
- Swatvik Gunamaneni: 33.33%

Work to be completed:

Description:

- Using of the Walmart API to get the prices of the items from Walmart store.
- Developing a cart page and integrating it with Dashboard page.
- Testing the cart page.
- Developing a speech recognition feature for letting the user enter his input through his voice.

Responsibilities:

- Vikas kondapalli :
Using the Walmart API and integrating it with DashBoard, Speech Recognition feature, testing, documentation
- Gopi Krishna Bodapati:
Developing a cart page and integrating it with Dashboard page, Speech Recognition feature, testing, documentation.
- Swatvik Gunamaneni:
Speech Recognition feature, Testing, documentation.

Time to be taken:

- Using the Walmart API and integrating it with DashBoard: 80 hours
- Developing cart page: 10 hours
- Speech recognition feature: 60 hours
- Testing: 20 hours
- Documentation: 20 hours

Link for Kanban tool (project management tool):

<https://university-of-m.kanbantool.com/b/176332-shopping-guidr#?>

Username and password are required for accessing it.

Bibliography

Link for Supermarket API used in this project:

<http://supermarketapi.com/Default.aspx>