



# Calorie 2 Grocery

## ASE Final Project Document

Project Group #12

Sravani Punyamurthula

Vaishnavi Aienampudi

Leela Naga Devi Gajula

Vinaya Podduturi

## Introduction

Diet plays a major role in leading a healthy life style. Due to busy schedules, it has become difficult to plan for a healthy and nutritious diet. Diet is nothing but a pattern of eating food. It is important to concentrate on diet to stay healthy and happy. Proper diet reduces the risk of many diseases.

Obesity and weight gain are the most common problems today. Irregular and improper diet leads to obesity. So it is important to plan our meal to include necessary ingredients and avoid over consumption of food. Calorie information gives the energy content of food. Hence, Calorie intake is a good measure to check on weight gain.

Another common problem in day to day life is grocery management. Now-a-days, it is common to forget the groceries available at home. Not keeping track of the expiry dates leads to wastage of food. There are many web apps/Android apps which give the calorie information for different varieties of food items. Also, there are apps which aid in grocery management. But there is no single application that handles both the functionalities. An integrated app is much more useful to the user because it helps in proper planning of the meal based on calories per serving and groceries available.

## Overall Goal & Objectives

Taste wins over health for most of the people. Also, after a day of work it is natural to prepare meal based on the ingredients that are available at home. Sometimes if vegetables are not available at home, people tend to eat junk food to fulfill their hunger. Hence it is important to plan for proper meal to maintain healthy life style.

Mobile apps are fun to use. Our endeavor is to develop an application that can be very useful in maintaining proper diet along with managing groceries efficiently. Our project aims to fulfill two main objectives:

- Firstly to provide calorie information about the ingredients used in preparing a recipe. This helps the user to take healthy choices about their meal.
- Secondly, to automatically generate a shopping list so that the user doesn't miss something he needs to buy.

Users can create a recipe/search for a recipe and get the calorie count of the ingredients in the recipe. Based on the calorie information, user can choose to add or remove some of the ingredients in the recipe. Thus the app helps in planning for a healthy meal.

Our app provides the flexibility to log the groceries available at home as well as prepare a shopping list. Our app will generate alerts & notifications about the expiry date of the groceries, thus it helps in reduction of wastage of food. Also, based on the recipe chosen, our app prepares the list of ingredients to buy in order to prepare the recipe. Thus the app functions as a unified meal planner which takes care of grocery management as well as provide calorie information to plan for a healthy meal.

## Significance

Our project focuses on providing all the required information for the user to plan for a healthy meal. A user can have a clear view of the groceries available at home from anywhere. This helps the user to plan his/her meal, search for the recipe and buy the required ingredients before reaching home. The app also gives the calorie information for each ingredient in the recipe thus providing nutritional information to the user. This information could drive the user to go for healthier meal options. Also, the app provides flexibility to add/remove ingredients in the recipe or create his/her own recipes. Our app works as follows:

Suppose a user would like to have vegetable sandwich for dinner. Say, he has tomatoes, onions and bread at home. Using our app he can do the following:

- 1) User can add tomatoes and onions to his currently available groceries list. User can update the list every time he purchases groceries.
- 2) User can search for a recipe of veg sandwich
- 3) The app displays the ingredients for the sandwich. Eg : Bread, Potatoes, Onions, Cucumber, Tomatoes, Lettuce
- 4) Based on the calorie information, user decides to remove potatoes from the sandwich and add olives. The app then updates the overall calorie count of the sandwich.
- 5) Once the user is satisfied with the recipe he can click ok, the app then displays the groceries he/she needs to buy to prepare the sandwich. Also, the quantity of tomatoes and onions will be decreased in the backend to reflect the usage. In this example, the app shows that the user needs to buy Cucumber, lettuce and olives.

## Project Background & Related Work

There are few applications that gives calorie information of the particular item that could help the user to maintain proper diet, but this does not include the grocery information. Some of the other apps that are developed provides user with only grocery management and recipe information. Our idea is to facilitate the user with both the calorie information and grocery management. In this project, we are going to implement this idea which combines both the functionalities.

The Grocery management application titled "Fresh Box" addresses the mundane task by allowing users to simply snap a photo and upload it into their own virtual ice box. They call this their "what you see is what you get" feature, and the beautifully illustrated interface makes stocking the fridge a fun process similar to updating your Instagram. Once photos are uploaded, users can set an expiration date so the milk doesn't go sour, produce doesn't go bad. The following are the features of this application:

- Take snapshots of the items purchased
- Upload the photos in virtual ice box
- Set expiry date for the items

The Calorie intake application named "Fooducate" is helpful in making choices at the grocery store. You can scan an item's barcode and the app will share nutritional highlights, compare it to similar

items, and provide alternatives. But this app doesn't provide the option to log the groceries available at home. Using this app, we can only compare the calorie count of similar products. Main features:

- The user gets the calorie details of the product
- Similar items are retrieved as alternatives which are having same count of calories

One of the Grocery shopping list application “Anylist” is specifically designed for grocery lists and recipes. When we start typing to add a new item, AnyList displays an auto-complete list. Tap an item to add it, and AnyList groups items in your list by store sections: Bakery, dairy, meat, frozen food, and so on. We can create multiple lists—one for each store and we can share them with others who use the app. You can also add recipes, so each item needed gets added to your grocery list. But this app doesn't give the calorie information about the ingredients. The features of this application:

- Search for the particular recipe and get the ingredients
- Categorize the groceries into different groups
- Share the list with others

<b>Fresh Box</b>	<b>Fooducate</b>	<b>Anylist</b>	<b>Proposed System</b>
User can add the items along with expiration date and get notifications whenever the product expired.	User can add the items based on barcodes but cannot add products which doesn't have barcodes.	User can just add the groceries list depending upon the category.	User can add the items along with expiration date and get notifications whenever the product expired. Expiration date will be automatically calculated if not entered.
Does not provide calorie information	Provides calorie information for a particular product but not for a recipe.	Calorie information is not evaluated for a recipe.	User can get the calorie count for each ingredient and can edit the quantity of ingredients depending on his/her choice and get the overall calorie count per serving.
No option to create and save recipes	No option to create and save recipes	User can search for a recipe but cannot save it for future reference	User can search for recipes/create his own recipe and save it.
Does not provide information regarding nearest grocery stores.	Does not provide information regarding nearest grocery stores.	Does not provide the information regarding nearest grocery stores.	Whenever ingredient is not available for the recipe the user can search for the nearest grocery store based on current location.

## **Proposed System**

### **Requirement Specification**

#### **Functional:**

The goal of the project is to develop an application that allows the user to

- 1) Register or login securely through Facebook
- 2) Maintain a list of groceries available in their home
- 3) Search for recipes and save them for quick reference
- 4) Get the detailed information of calories per each ingredient required for the recipe
- 5) Add/remove the ingredients and adjust the quantity of the ingredients
- 6) Get the final count of calories per serving of the recipe
- 7) Based on the ingredients, get a list of groceries to buy in order to prepare the recipe.
- 8) Get the nearest grocery stores based on the current location
- 9) Get alerts/notifications about the expiry date of the groceries in home.

#### **Non-Functional:**

- ❖ Security: The app doesn't require access to any sensitive information of the users. The app requests access for only the current location of the user. Also, users can login securely using Facebook.
- ❖ Stability: The app is designed to function as expected on any device with android version 4.0 and above.
- ❖ Visual Quality: The app displays all the text blocks and forms in acceptable formats.
- ❖ Performance: The app needs to load quickly and respond within 3-5 seconds.

#### **Technological and architectural requirements:**

<b>Requirement</b>	<b>Tools</b>
Operating System	Android 4.0 or above
Development Operating System	Windows
Platform	Android Studio/Eclipse
UML Diagrams	Microsoft Visio
Languages	Java, C# , ASP.Net
Database	SQL Lite
Planning	ScrumDo
Version Control System	GIT
Rest Services	Facebook API
	Google Places API
	NDB API

### Existing Services:

#### **1) Recipe Search & Diet API :**

This API provides services for the following:

- Nutritional Analysis
- Recipe Search

We are using Recipe Search and Diet API to search for a recipe. This API provides information about the ingredients, the preparation time and procedure for a recipe. This API provides detailed information about the quantity of ingredients, total calorie count of the recipe etc. Also this API provides additional search capabilities like we can include restrictions on allergic items etc. Hence we are using this API to get the list of recipes based on a keyword and import the ingredient list.

API: [https://api.edamam.com/search?q=chicken&app\\_id=51aba909&app\\_key=9fc3aa5746d2a423a350cee3ea4d57d](https://api.edamam.com/search?q=chicken&app_id=51aba909&app_key=9fc3aa5746d2a423a350cee3ea4d57d)

Reference: <https://developer.edamam.com/>

#### **2) USDA Nutrition API :**

NDB is the nutrition database provided by the United States Department of Agriculture. Each food item is assigned a NDB number. We can fetch the calorie information of any ingredient only by knowing the NDB No.

The following API fetches the NDB No. of a food item. Ingredient name is passed as parameter to this url:

[http://api.nal.usda.gov/usda/ndb/search/?format=json&q=spinach&sort=n&max=1&offset=0&api\\_key=DgxqZYmXG8YsyCIEfFLkJaAkIM4ybQtEHORaKedx](http://api.nal.usda.gov/usda/ndb/search/?format=json&q=spinach&sort=n&max=1&offset=0&api_key=DgxqZYmXG8YsyCIEfFLkJaAkIM4ybQtEHORaKedx)

The following API accepts NDB No. as an input and fetches the calorie information:

[http://api.nal.usda.gov/usda/ndb/reports/?ndbno=03127&type=b&format=json&api\\_key=DgxqZYmXG8YsyCIEfFLkJaAkIM4ybQtEHORaKedx](http://api.nal.usda.gov/usda/ndb/reports/?ndbno=03127&type=b&format=json&api_key=DgxqZYmXG8YsyCIEfFLkJaAkIM4ybQtEHORaKedx)

By using this rest service, we are displaying the calorie information for each ingredient in the recipe.

### New Services:

We are storing the grocery and recipe information in the database. Our functionality mainly deals with managing this information. Hence most of the features in the application require interaction with the database. To achieve that, we have created the following rest services for implementing the server side logic for this application.

- 1) **Grocery List:** User can add the grocery items available at their home using our application. The grocery items added by the user will be stored in the database. This API is used to maintain the grocery lists. We have implemented 4 methods in this API.

- a) **Add grocery Item:** This method is used to insert the data entered by the user in the database. User name, grocery name, quantity and units are given as input to this method.  
 API: [http://kc-sce-cs551-  
2.kc.umkc.edu/aspnet\\_client MPG12/Sravani/AddGrocery/Service1.svc/addgrocery/Sravani/butter/30/gm](http://kc-sce-cs551-2.kc.umkc.edu/aspnet_client MPG12/Sravani/AddGrocery/Service1.svc/addgrocery/Sravani/butter/30/gm)
  
- b) **View grocery list:** This method is used to retrieve the grocery list from the database. Username is passed as an input to this method.  
 API: [http://kc-sce-cs551-  
2.kc.umkc.edu/aspnet\\_client MPG12/Sravani/viewGrocery/Service1.svc/JsonGrocery/Sravani](http://kc-sce-cs551-2.kc.umkc.edu/aspnet_client MPG12/Sravani/viewGrocery/Service1.svc/JsonGrocery/Sravani)
  
- c) **Delete Grocery Item:** This method is used to delete grocery items from the grocery list. We need to provide username and grocery item name as an input to the API URL.  
 API: [http://kc-sce-cs551-  
2.kc.umkc.edu/aspnet\\_client MPG12/Sravani/viewGrocery/Service1.svc/deleteGrocery/Sravani/butter](http://kc-sce-cs551-2.kc.umkc.edu/aspnet_client MPG12/Sravani/viewGrocery/Service1.svc/deleteGrocery/Sravani/butter)
  
- d) **Update Grocery Item:** This method is used to update the quantity & units of a grocery item. We need to pass the grocery name, quantity & units as input to the URL.  
 API: [http://kc-sce-cs551-  
2.kc.umkc.edu/aspnet\\_client MPG12/Sravani/viewGrocery/Service1.svc/updateGrocery/Sravani/butter/30/gm](http://kc-sce-cs551-2.kc.umkc.edu/aspnet_client MPG12/Sravani/viewGrocery/Service1.svc/updateGrocery/Sravani/butter/30/gm)

- 2) **Recipes:** One of the key functionalities of our application is to search for recipes or create their own recipes. So we need rest services to import the recipe information and store it in the database and manage the recipes. Following are the services created to provide different functionalities to the user:

- a) **Import Recipe:** This API is created to import the recipe information from the recipe search API and store the recipe name and ingredient list in the database. We need to import this information to the database so that we can enable the users to customize the recipe.  
 API: [http://kc-sce-cs551-  
2.kc.umkc.edu/aspnet\\_client MPG12/Sravani/viewGrocery/Service1.svc/importRecipe/](http://kc-sce-cs551-2.kc.umkc.edu/aspnet_client MPG12/Sravani/viewGrocery/Service1.svc/importRecipe/)
  
- b) **Add Ingredient to recipe:** This rest service is used to add ingredients to the recipe based on user's taste preference.  
 API: [http://kc-sce-cs551-  
2.kc.umkc.edu/aspnet\\_client MPG12/vinaya2/RecipeIngredient/Service1.svc/data/](http://kc-sce-cs551-2.kc.umkc.edu/aspnet_client MPG12/vinaya2/RecipeIngredient/Service1.svc/data/)

c) **Remove ingredient to recipe:** This rest service is used to remove any of the ingredients from the recipe information in the database.

d) **Update the ingredients in recipe:** This service is used to update the quantity and units of ingredients in the recipe.

API: <http://kc-sce-cs551->

[2.kc.umkc.edu/aspnet\\_client MPG12/Leela/UpdateRecipe/Service1.svc/UpdateRecipe/](2.kc.umkc.edu/aspnet_client MPG12/Leela/UpdateRecipe/Service1.svc/UpdateRecipe/)

3) **Shopping List:** This rest service is used to generate the shopping list based on the ingredients in the recipe table and the items in grocery list. This service has the following methods:

a) **Add to shopping list:** Whenever a user clicks on generate shopping list after importing recipe, the difference between the recipe ingredients and the grocery list are stored in a shopping list table in the database.

API: <http://kc-sce-cs551->

[2.kc.umkc.edu/aspnet\\_client MPG12/Sravani/viewGrocery/Service1.svc/importRecipe](2.kc.umkc.edu/aspnet_client MPG12/Sravani/viewGrocery/Service1.svc/importRecipe)

b) **View Shopping List:** This rest service is used to fetch the data from the shopping list table in database.

API: <http://kc-sce-cs551->

[2.kc.umkc.edu/aspnet\\_client MPG12/Sravani/viewGrocery/Service1.svc/userShoppingList](2.kc.umkc.edu/aspnet_client MPG12/Sravani/viewGrocery/Service1.svc/userShoppingList)

c) **Delete from shopping list:** This rest service is used to delete the grocery items from the shopping list table. We need to provide the name of the grocery item which needs to be deleted as an input to this service.

API: <http://kc-sce-cs551->

[2.kc.umkc.edu/aspnet\\_client MPG12/Sravani/viewGrocery/Service1.svc/deleteShoppingList/](2.kc.umkc.edu/aspnet_client MPG12/Sravani/viewGrocery/Service1.svc/deleteShoppingList/)

### **System Architecture Diagram:**

The overall system architecture can be represented as below: Users interact with the application through GUI. Based on the user input, a request will be sent to the database using rest services. The response is then redirected to the user through the GUI.

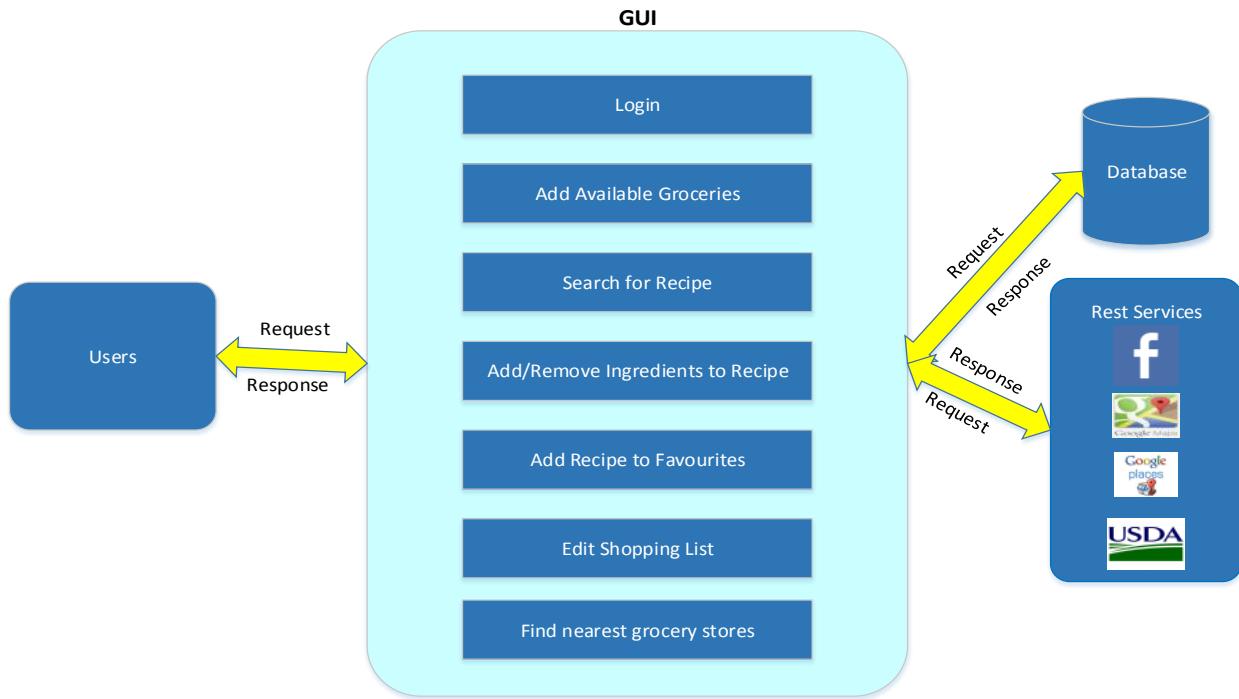
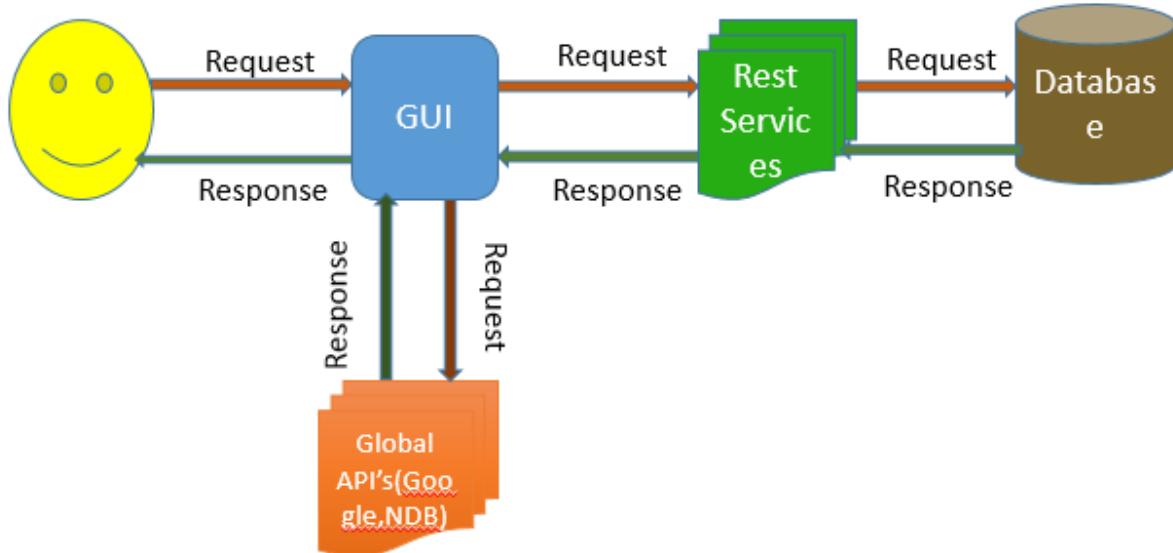


Figure 1.2 System Model

#### System Architecture Diagram:



## Class Diagram:

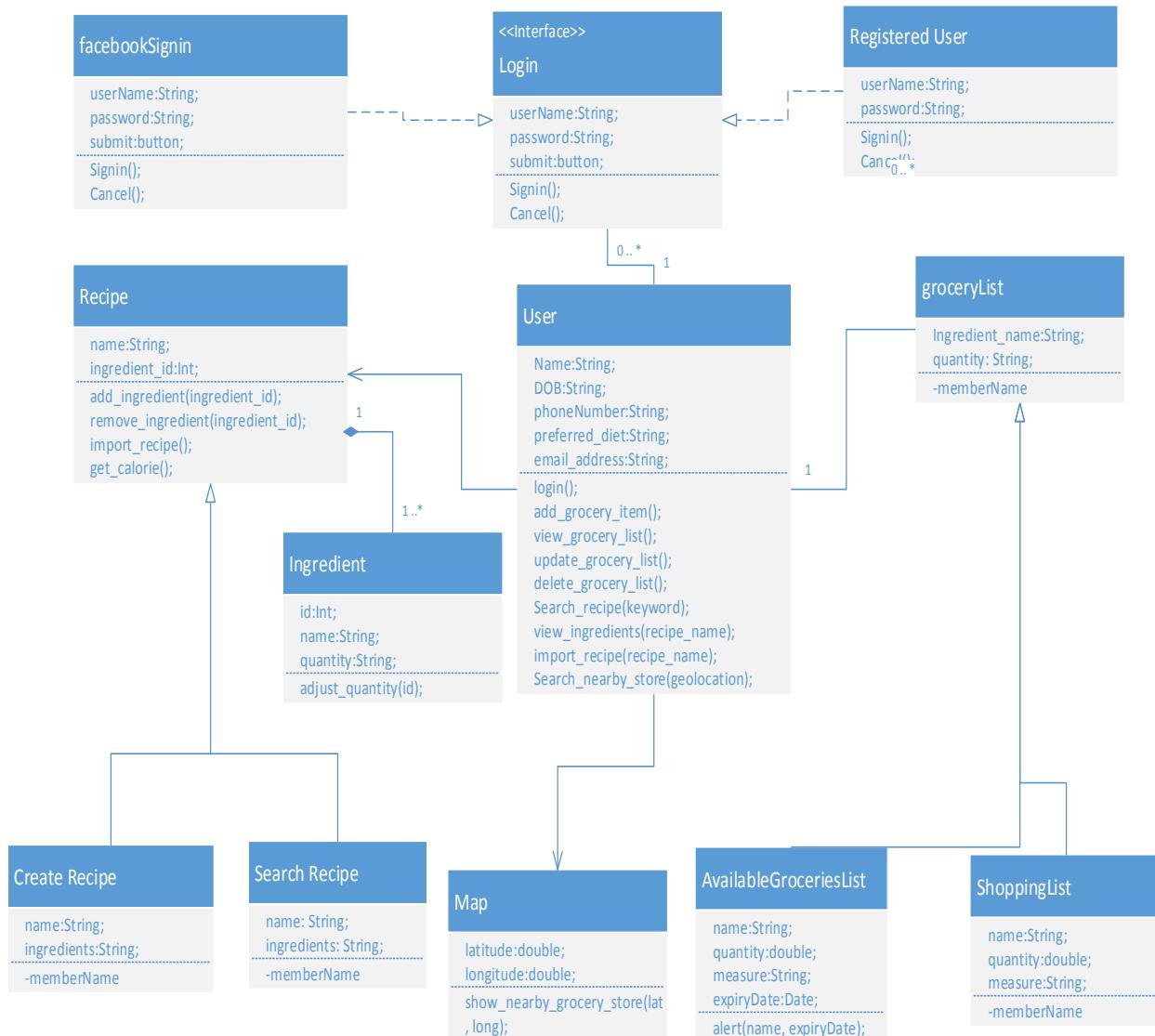


Figure 1.1 Class Diagram

## Activity Diagram:

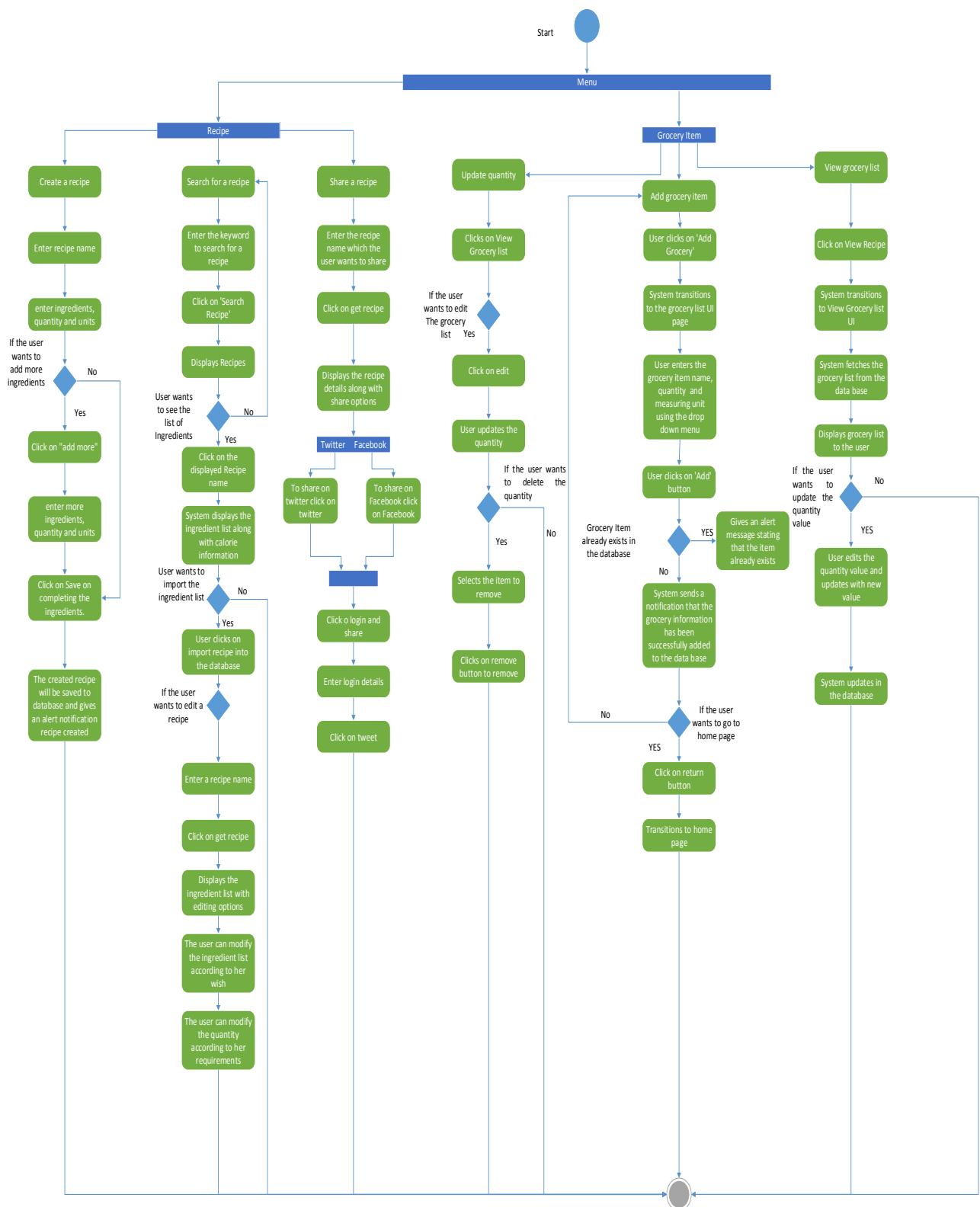
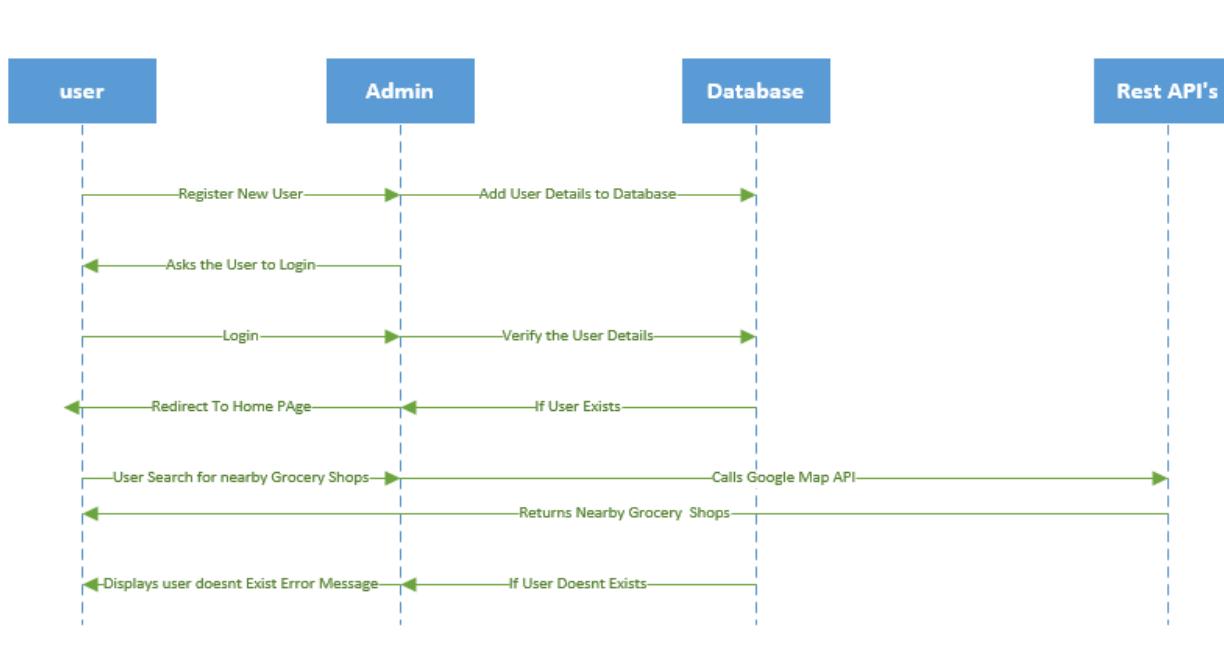


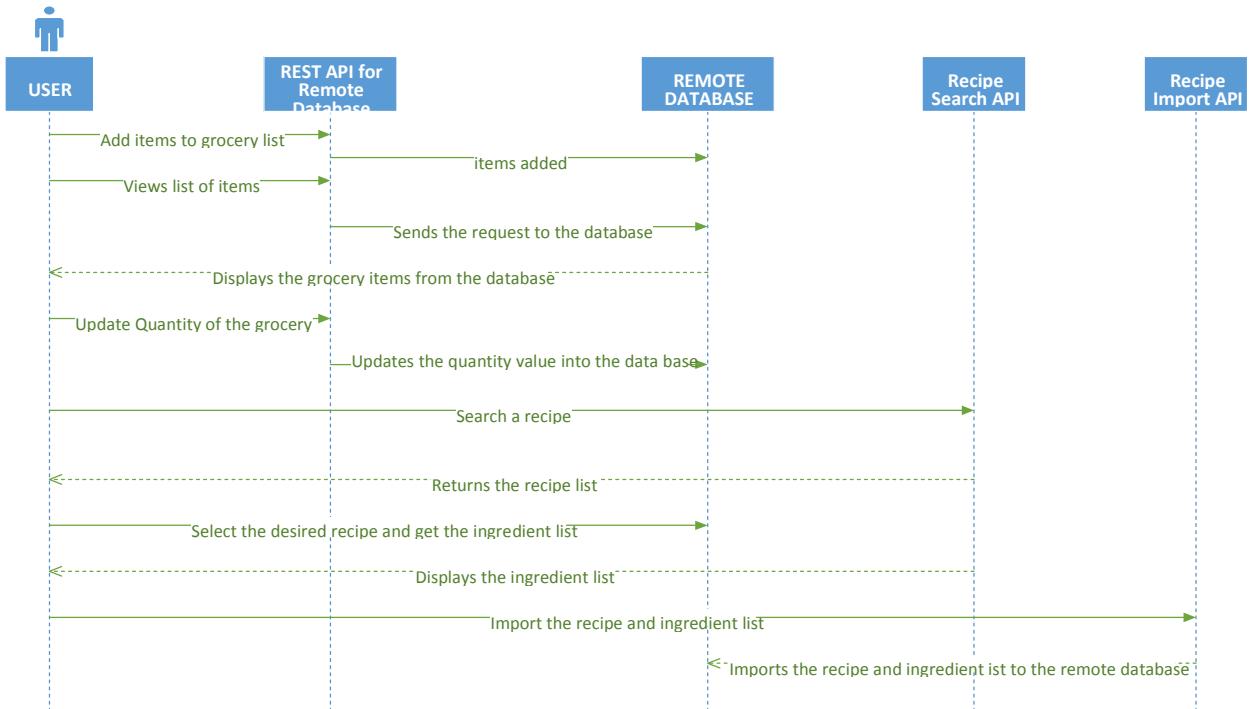
Figure 1.4 Activity Diagram

## Sequence diagrams

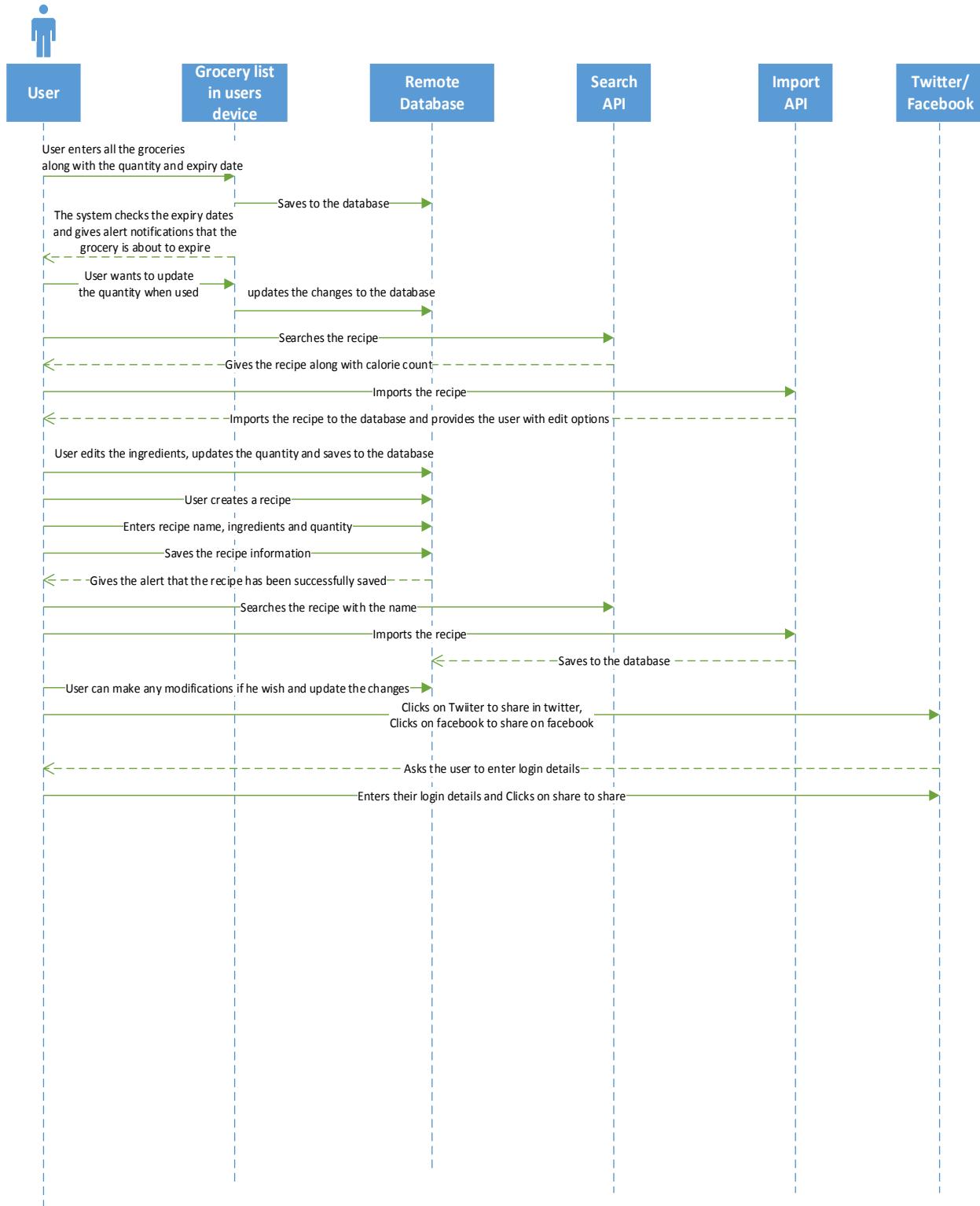
### 1. For Login Page



### 2. Adding Grocery Items



### 3. Importing recipes along with calorie count



## Use Case Templates

### Use Case Template for Adding grocery items:

<b>Use Case Name</b>	Add grocery items
<b>Primary Actor</b>	User
<b>Pre-Conditions</b>	<ol style="list-style-type: none"><li>1. User has the app installed on his/her mobile.</li><li>2. User should be logged into the app.</li></ol>
<b>Guarantee(Post Conditions)</b>	Grocery items are stored in the database
<b>Main Success Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on “Add Grocery” button in the home page.</li><li>2. The system transitions the UI to the “Add Grocery” layout.</li><li>3. User enters the grocery item name, quantity, selects the measuring unit from the drop down menu and clicks on “Add”.</li><li>4. System adds the grocery information to the database.</li><li>5. System sends a notification that the grocery item is successfully added to the database.</li><li>6. User can go back to the home page by clicking on return button.</li></ol>
<b>Extensions</b>	<p>3a. User enters incorrect value for quantity. Systems throws an error asking the user to add proper value.</p> <p>4a. If the grocery item is already present in the list, systems sends an alert saying the given name already exists in the list. The user can update the quantity of the item in the list if desired.</p> <p>5a. User can add as many groceries items as he needed to by following steps 3 &amp; 4 and then go back to the home page.</p>

### Use Case Template to View the grocery list:

<b>Use Case Name</b>	View grocery list
<b>Primary Actor</b>	User
<b>Pre-Conditions</b>	<ol style="list-style-type: none"><li>1. User has the app installed on his/her mobile.</li><li>2. User should be logged into the app.</li></ol>
<b>Guarantee(Post Conditions)</b>	User can view the list of groceries.

	User can update the quantity of grocery items.
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. User clicks on “View Grocery List” button in the home page.</li> <li>2. The system transitions the UI to the “View Grocery” layout.</li> <li>3. System fetches the grocery list of the user from the database.</li> <li>4. System displays the grocery list to the user.</li> <li>5. User can update the quantity of any of the grocery items.</li> <li>6. System updates the database to reflect the changes made by the user.</li> <li>7. System sends a notification saying the update is successful.</li> </ol>
<b>Extensions</b>	None

**Use Case Template to update grocery item:**

<b>Use Case Name</b>	Update grocery items
<b>Primary Actor</b>	User
<b>Pre-Conditions</b>	8. User has the app installed on his/her mobile. 9. User should be logged into the app.
<b>Guarantee(Post Conditions)</b>	Quantity of grocery items is updated in database.
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. User clicks on “View/Edit grocery list” button in the home page.</li> <li>2. The system transitions the UI to the “View Grocery” layout.</li> <li>3. System fetches the grocery list of the user from the database.</li> <li>4. User clicks on the quantity of the grocery item he/she needs to update.</li> <li>5. System changes the quantity cell to editable mode.</li> <li>6. User updates the quantity and hits enter.</li> <li>7. System updates the data in the remote database.</li> </ol>
<b>Extensions</b>	3a. User clicks the units cell of the grocery item. 4a. System changes the units cell to editable mode. 5a. User updates the unit for the grocery item. 6a. System updates the measure of the grocery item in database.

**Use Case Template to delete grocery item:**

<b>Use Case Name</b>	Delete grocery item
----------------------	---------------------

<b>Primary Actor</b>	User
<b>Pre-Conditions</b>	<ol style="list-style-type: none"> <li>1. User has the app installed on his/her mobile.</li> <li>2. User should be logged into the app.</li> </ol>
<b>Guarantee(Post Conditions)</b>	Grocery item is deleted from the database.
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>3. User clicks on “View/Edit Grocery List” button in the home page.</li> <li>4. The system transitions the UI to the “View Grocery” layout.</li> <li>3. System fetches the grocery list of the user from the database.</li> <li>4. System displays the grocery list to the user.</li> <li>5. User can click on “Edit” button</li> <li>6. System display minus sign in front of each grocery item in the list.</li> <li>7. User clicks on the minus sign in front of the grocery item he/she wants to delete.</li> <li>8. System opens an alert dialog asking the user to confirm deletion.</li> <li>9. User hits “OK”.</li> <li>10. System deletes the grocery item from the database.</li> <li>11. System removes the row in the display</li> </ol>
<b>Extensions</b>	<ol style="list-style-type: none"> <li>9a. User hits “Cancel”.</li> <li>9b. System doesn't delete the grocery item.</li> </ol>

#### Use Case Templates for searching and importing a recipe:

<b>Use Case Name</b>	Search & import a recipe
<b>Primary Actor</b>	API
<b>Pre-Conditions</b>	The user must be a valid user to use the app
<b>Post-Conditions</b>	Retrieves the results for the search
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. If the user wants to search for a recipe, then, the user enters the keyword to search.</li> <li>2. User clicks on the search recipes.</li> <li>3. The system displays ten recipe names.</li> <li>4. If the user wants to see the ingredient list, then the user can click on the recipe name.</li> <li>5. System displays ingredient list along with quantity.</li> <li>6. If the user wants to import the ingredient list then click on import.</li> </ol>

	7. The system stores the ingredient list in the database.
<b>Extensions</b>	None
<b>Special Requirements</b>	None

**Use Case Templates to Add Ingredient to a recipe:**

<b>Use Case Name</b>	Search & import a recipe
<b>Primary Actor</b>	User
<b>Pre-Conditions</b>	The user must be a valid user to use the app
<b>Post-Conditions</b>	Retrieves the results for the search
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. If the user wants to search for a recipe, then, the user enters the keyword to search.</li> <li>2. User clicks on the search recipes.</li> <li>3. The system displays ten recipe names.</li> <li>4. If the user wants to see the ingredient list, then the user can click on the recipe name.</li> <li>5. System displays ingredient list along with quantity.</li> <li>6. If the user wants to import the ingredient list then click on import.</li> <li>7. The system stores the ingredient list in the database.</li> </ol>
<b>Extensions</b>	User can view and update the recipe

**Use Case Templates to View Favorite Recipe:**

<b>Use Case Name</b>	View recipe
<b>Primary Actor</b>	User
<b>Pre-Conditions</b>	The user must be a valid user to use the app
<b>Post-Conditions</b>	Retrieves the results for the search
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. If the user wants to get the favorite recipe then user clicks the view favorite recipe button</li> <li>2. User clicks on the favorite recipes.</li> <li>3. User gets all the favorite recipe list from the database.</li> </ol>
<b>Extensions</b>	User can update the recipe

**Use Case Templates to Update Recipe:**

<b>Use Case Name</b>	View recipe
<b>Primary Actor</b>	User
<b>Pre-Conditions</b>	The user must be a valid user to use the app
<b>Post-Conditions</b>	Retrieves the results for the search
<b>Main Success Scenario</b>	<ol style="list-style-type: none"><li>1. If the user wants to get the favorite recipe then user clicks the recipe name.</li><li>2. User clicks on the get recipe button</li><li>3. User gets all the recipe name, ingredient list , quantity and label from the database.</li><li>4. User can add, remove, update the ingredient , quantity and label</li><li>5. User can click the update button.</li><li>6. System uploads the changed the recipe list to the database.</li></ol>

**Use Case Template to Create Recipe:**

<b>Use Case Name</b>	View recipe
<b>Primary Actor</b>	User
<b>Pre-Conditions</b>	The user must be a valid user to use the app
<b>Post-Conditions</b>	Retrieves the results for the search
<b>Main Success Scenario</b>	<ol style="list-style-type: none"><li>1. If the user wants to create the favorite recipe then user enters the recipe name , ingredients list and their quantity and labels.</li><li>2. User clicks on the add more button for more ingredients to the recipe.</li><li>3. User adds all the ingredients and quantity and their labels and clicks the save recipe button.</li><li>4. System uploads the recipe tot the database.</li></ol>
<b>Extensions</b>	User can Share the recipe in twitter and facebook.

**Use Case Templates to Share Favorite Recipe:**

<b>Use Case Name</b>	View recipe
<b>Primary Actor</b>	User

<b>Pre-Conditions</b>	The user must be a valid user to use the app
<b>Post-Conditions</b>	Users should be able to share the recipe using twitter or facebook.
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. User can search the recipe he would like to search from his list of recipes in database.</li> <li>2. System displays the ingredients list, quantity and label for the recipe.</li> <li>3. User can share the recipe in the twitter by providing twitter login</li> <li>4. Recipe information will be shared in the users twitter account as a tweet</li> <li>5. Similarly, User can share the recipe in the facebook using the facebook link.</li> </ol>

**Use Case Template to View & Edit shopping list:**

<b>Use Case Name</b>	View shopping list
<b>Primary Actor</b>	User
<b>Pre-Conditions</b>	<ol style="list-style-type: none"> <li>1. User has the app installed on his/her mobile.</li> <li>2. User should be logged into the app.</li> </ol>
<b>Guarantee(Post Conditions)</b>	User can view the shopping list User can edit the shopping list
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. User clicks on "Shopping List" button in the home page.</li> <li>2. The system transitions the UI to the "Shopping List" layout.</li> <li>3. System fetches the list of grocery items that need to be purchased from the database.</li> <li>4. System displays the shopping list to the user.</li> <li>5. User can delete any of the items from the list by clicking on edit button.</li> <li>6. System displays minus sign against each item in the list.</li> <li>7. User can click on the minus sign in the same row as the shopping item to be deleted.</li> <li>8. System deletes the item from the database.</li> </ol>
<b>Extensions</b>	None

**Use Case Template to Generate Notifications:**

<b>Use Case Name</b>	Generate Notifications
----------------------	------------------------

Primary Actor	System
<b>Pre-Conditions</b>	<ol style="list-style-type: none"> <li>1. User has the app installed on his/her mobile.</li> <li>2. User should be logged into the app.</li> <li>3. User should enter the grocery items along with expiry dates.</li> </ol>
<b>Guarantee(Post Conditions)</b>	User can check the expiry dates of the grocery items
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. User clicks on show notifications</li> <li>2. The system checks the expiry dates of all the items and generates notifications which are about to expire within 1 day.</li> <li>3. The system generates notification “grocery expiry dates” in the notification bar of the android device.</li> <li>4. User clicks on the notification to get the complete details.</li> <li>5. The system transitions to other page which lists the grocery items which are about to expire.</li> </ol>
<b>Extensions</b>	None

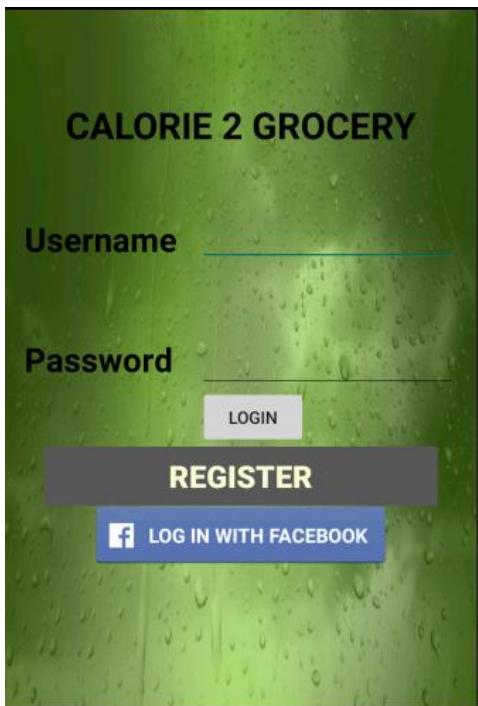
## Implementation

### 1) Implementation of User Interface:

We have grouped the functionalities and developed the following layouts as part of UI implementation.

- 1) Registration and Login Page: Allows the user to register to our application and login by providing username and password or by using facebook login
- 2) Home Page: Provides the necessary buttons to navigate to different functions
- 3) My Groceries : Once we go to My groceries page, there are 2 options:
  - a) Add Grocery
  - b) View/Edit Grocery
- 4) Recipes: We have created the following buttons in this page:
  - a) Search Recipe
  - b) Create Recipe
  - c) Share Recipe
  - d) Update Recipe
- 5) Show Notifications: Notifies user about the expiry dates of the grocery items.
- 6) Shopping List
- 7) Grocery Stores

### Screenshots:

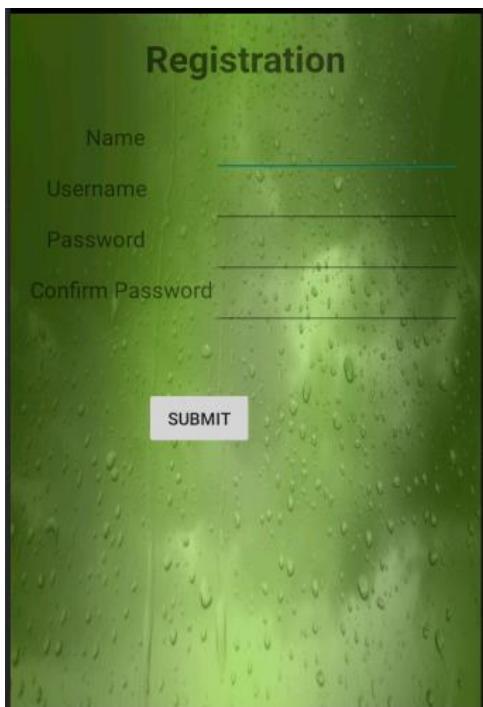


#### Login Page:

- User can login with username and password
- User can login with Facebook
- New user can register for the application

#### Technical Details:

- We have created a web service for retrieving the data from the registration table and comparing the data entered by the user and generating a message response
- Client side logic is implemented in android studio as a native app.

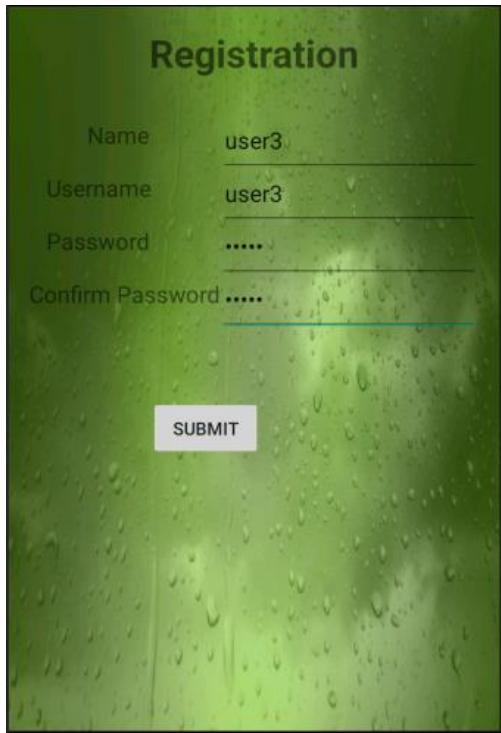


#### Registration Page:

- New user can register by filling form with name, username and password details and by clicking submit button

#### Technical Details:

- We have created a web service for inserting the data entered by the user in the registration table in the remote database.

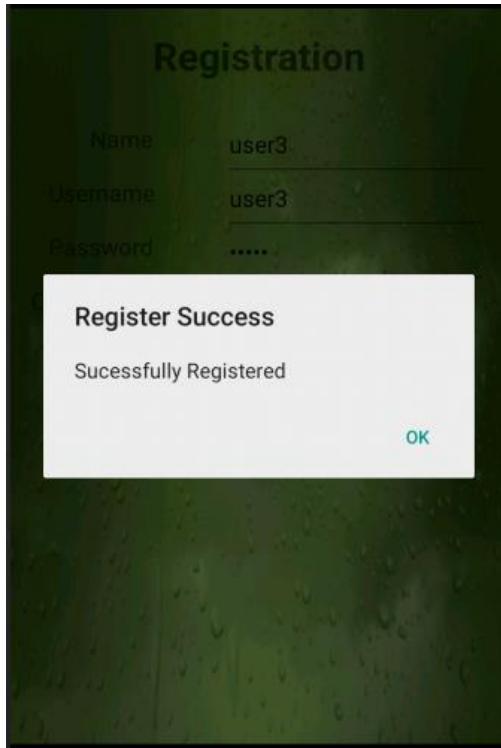


#### Registration page:

- The form filled with the details by the new user

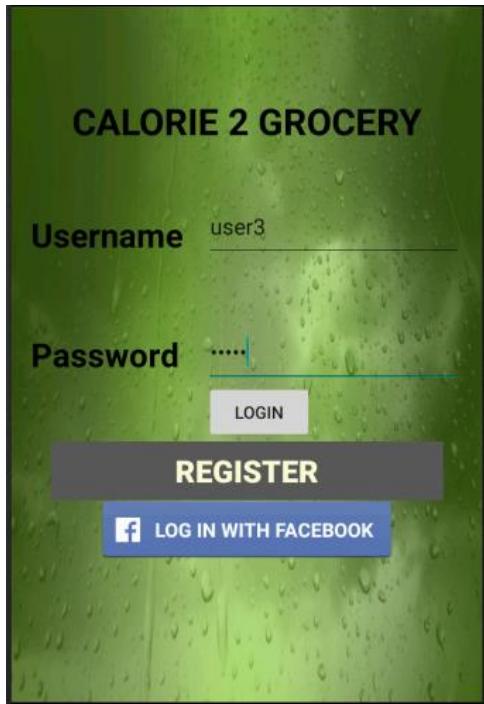
#### Technical Details:

- Client side logic is implemented in android studio as a native app.



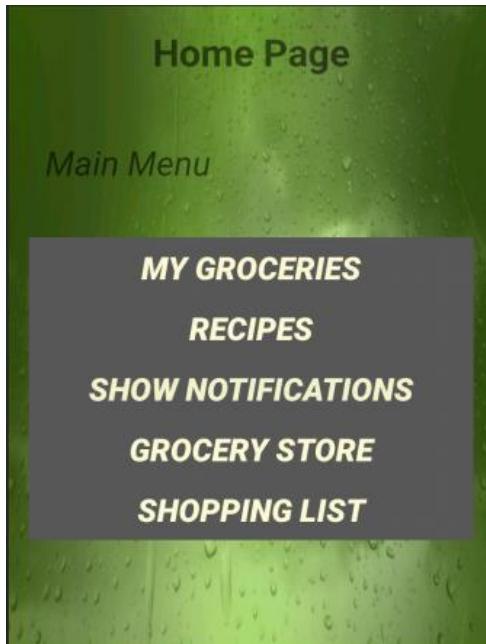
#### Registration page:

- The registration is successful and the new user is registered
- The message response generated showing "Successfully Registered"



#### Login page:

- The registered user can now login by giving his/her username and password
- The registered user is thus logged in to the application

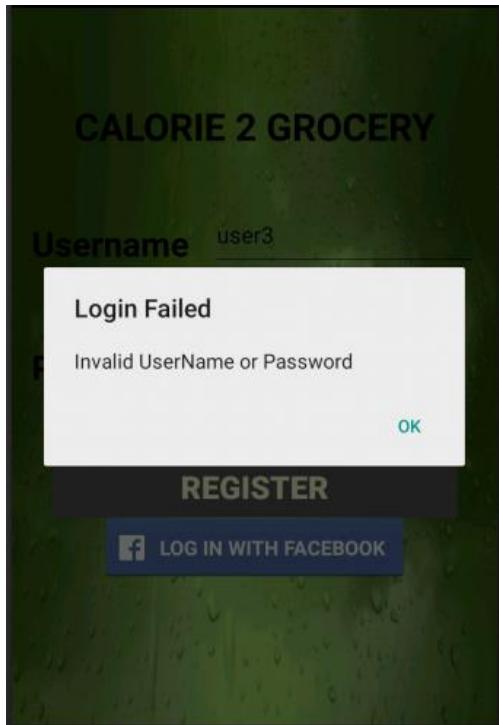


#### Home page:

- The user is logged in successfully in to the application
- The Home page will be displayed if login successful

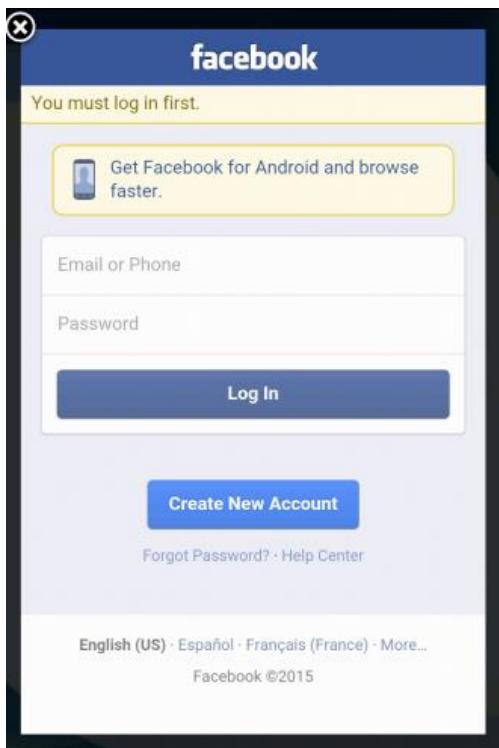
#### Technical Details:

- Client side logic is implemented in android studio as a native app.



#### Login page:

- If the user has entered incorrect username or password the login fails
- The dialog box pops up saying login failed
- The message response generated showing "Invalid Username or Password"

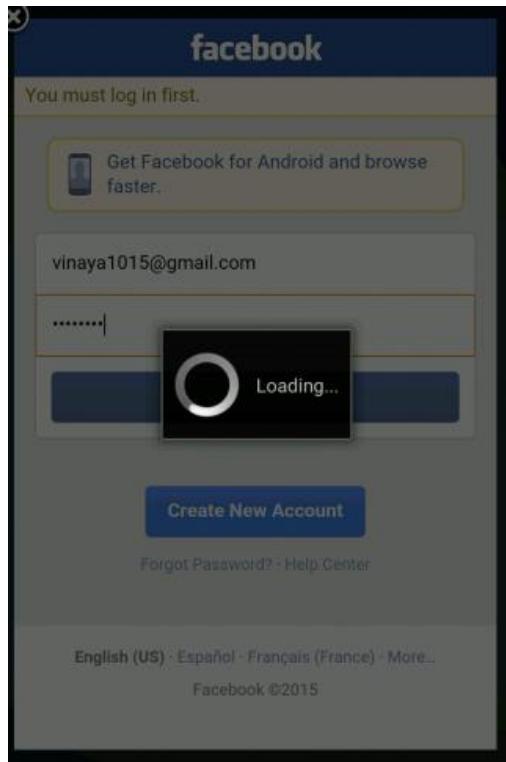


#### Facebook Login page:

- If the user has Facebook account, he can login with his account
- The user can login with Facebook by entering his/her Facebook username and password details

#### Technical Details:

- The Facebook SDK is imported into the application. Facebook App ID is configured and linked to the application. The android key hash is generated and linked to the Facebook API configuration



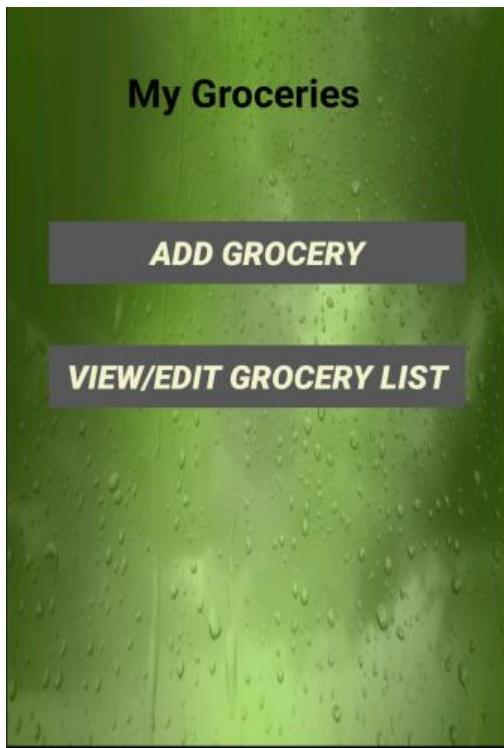
#### Facebook Login page:

- The user has entered his/her Facebook credentials and is logging in with Facebook account



#### Home page:

- After successful login with application or with Facebook, the user is redirected to the home page
- It contains the Menu in which there are buttons like My Groceries, Recipes, Notifications, Find Grocery store and Shopping list



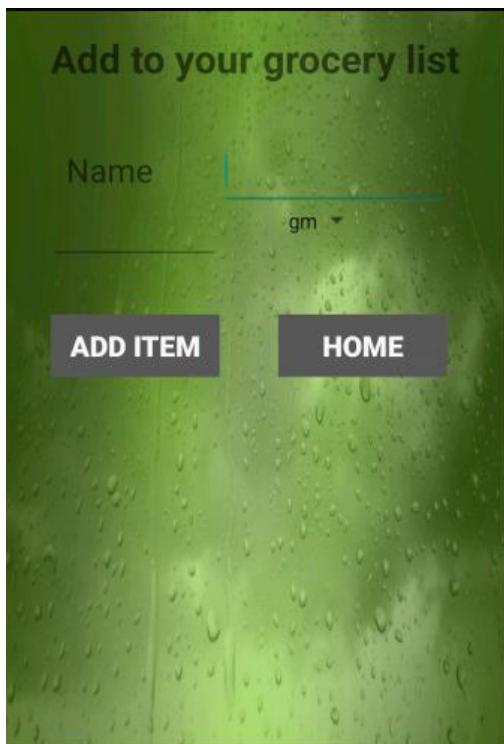
## My Groceries

**ADD GROCERY**

**VIEW/EDIT GROCERY LIST**

### My Groceries page:

- User is provided with two buttons:
  - Add Grocery
  - View/Edit Grocery list
- *Add Grocery*: User can click this button if he wants to add grocery items to his list
- *View/Edit Grocery list*: If user wants to view the grocery list, he/she could click on this button



## Add to your grocery list

Name

gm ▾

**ADD ITEM**

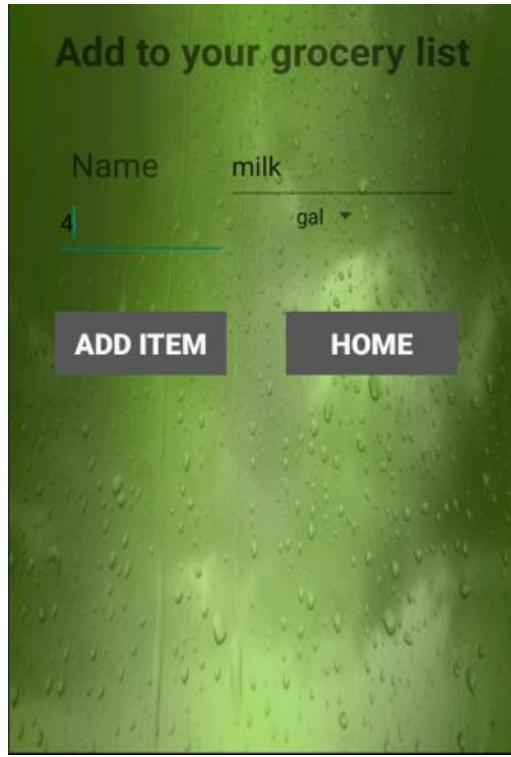
**HOME**

### Add Grocery page:

- User enters the grocery item name, quantity.
- User can select the measuring unit from the drop down menu

### Technical Details:

- Client side logic is implemented in android studio as a native app.

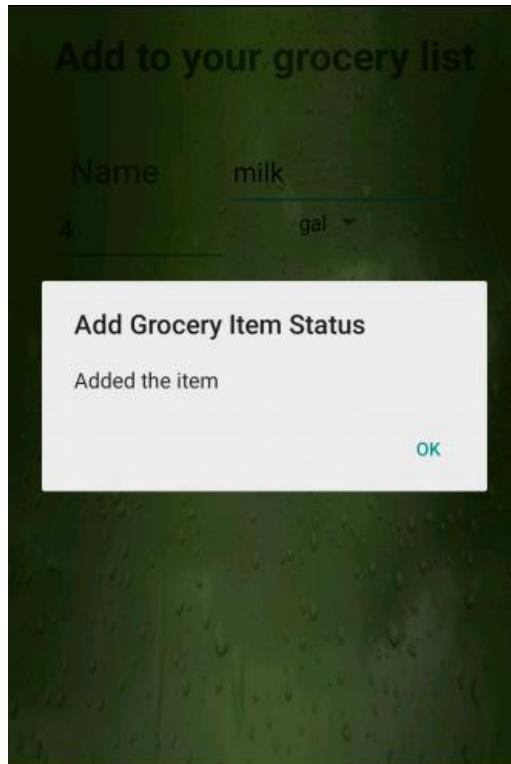


### Add Grocery

- After entering the details about the grocery item, user clicks on add.
- System adds the information to the SQL server database.

### Technical Details:

- We have created a web service for inserting the data entered by the user in the grocery\_list table in the remote database.

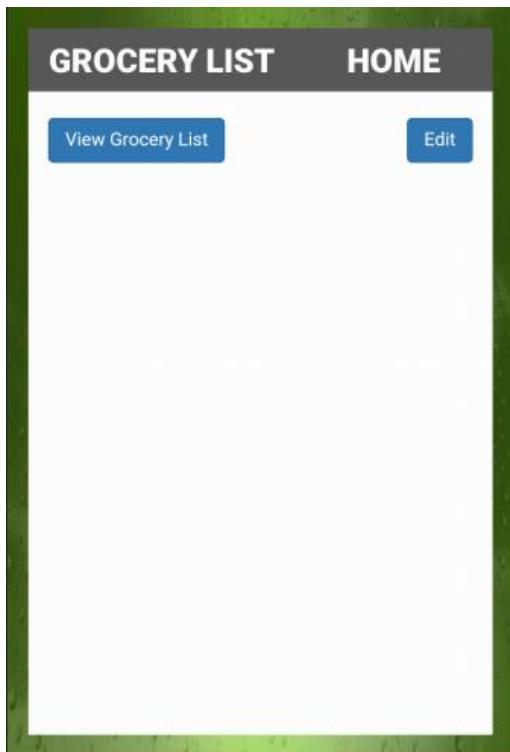


### Add Grocery

- System sends a notification to the user saying the grocery item is added successfully.
- After user clicks ok the add grocery UI is refreshed, the user can add more grocery items or use the home button to go to the home page.

### Technical Details:

- Web service to add the grocery items to database sends the result of the insert operation in the form of JSON.
- Created a class for parsing the JSON string.
- Used alert dialog to display the output message from the web service as a notification.



### View Grocery List

- If the user clicks on “View grocery list” button in home page then the UI transitions to this page.
- Once the user clicks on “Grocery List” button, the web page is loaded into the web view.

### Technical Details:

- Implemented the functionality for displaying the grocery list using web technologies.
- Used Web view to display the page.

GROCERY LIST			HOME
View Grocery List		Edit	
Name	Quantity	Units	
cheese	8	gm	
milk	4	gal	
beetroot	1	kg	
cheese pizza	40	gm	
beans	50	gm	
peas	1	kg	

### View Grocery List

- Once the user clicks on View Grocery List button within the page, the list of groceries is displayed in a table.

### Technical Details:

- We have created a web service to retrieve data from the remote database
- We have used Ajax function to call the API and display the data
- Used twitter Bootstrap styling for table.

GROCERY LIST			HOME
View Grocery List		Edit	
	Name	Quantity	Units
–	cheese	8	gm
–	milk	4	gal
–	beetroot	1	kg
–	cheese pizza	40	gm
–	beans	50	gm
–	peas	1	kg

### View Grocery List

- When user clicks on Edit button within the page, the list of groceries is displayed in a table.

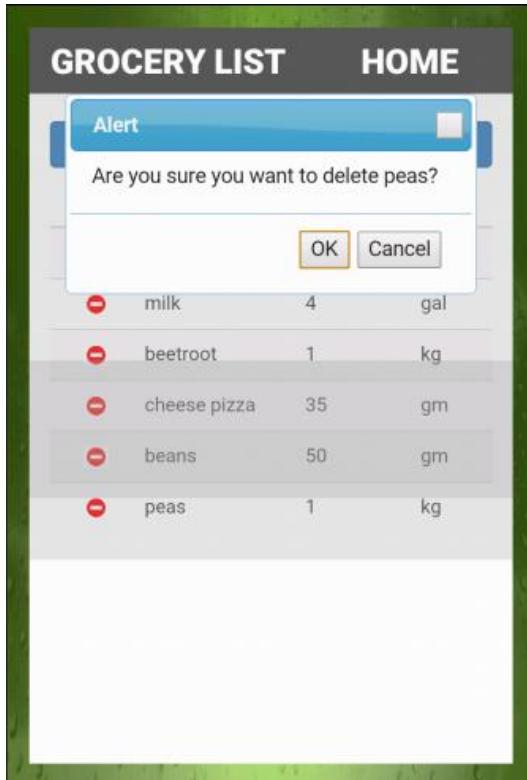
### Technical Details:

- We have created a web service to modify table and update table in the database.

GROCERY LIST			HOME
View Grocery List		Edit	
	Name	Quantity	Un
–	cheese	8	gm
–	milk	4	ga
–	beetroot	1	kg
–	cheese pizza	35	gm
–	beans	50	gm
–	peas	1	kg

### View Grocery List

- When user clicks on Edit button within the page, the list of groceries is displayed in a table.
- The user can edit particular item by clicking on it.



#### View Grocery List

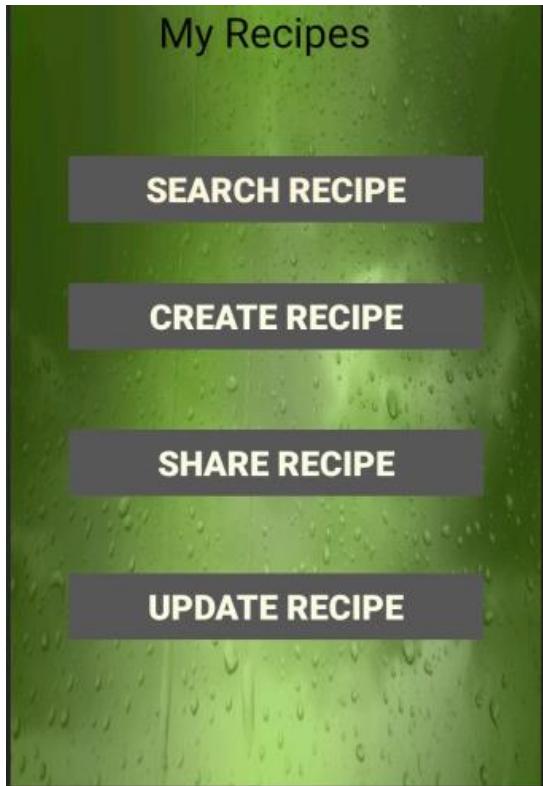
- When user clicks on Edit button within the page, the list of groceries is displayed in a table.
- By click on the red color button on the left side, the particular item is deleted from the grocery list and the table gets updated.
- The alert dialog box pops up to confirm the deletion.

The screenshot shows a mobile application interface titled "GROCERY LIST" and "HOME". At the top are two buttons: "View Grocery List" and "Edit". Below is a table of grocery items:

	Name	Quantity	Units
-	cheese	8	gm
-	milk	4	gal
-	beetroot	1	kg
-	cheese pizza	35	gm
-	beans	50	gm

#### View Grocery List

- When user clicks on View Grocery List button within the page, the updated list of groceries is displayed in a table.



#### My Recipes page:

- User is provided with four buttons:
  - Search Recipe
  - Create Recipe
  - Share Recipe
  - Update Recipe
- *Search Recipe*: User can search recipes by clicking this button.
- *Create recipe*: If user wants to create new recipe he/she could click on this button.
- *Share recipe*: User can also share recipe to twitter and Facebook by clicking this button.
- *Update Recipe*: If user wants to update, view or delete the recipes he/she could click this button.

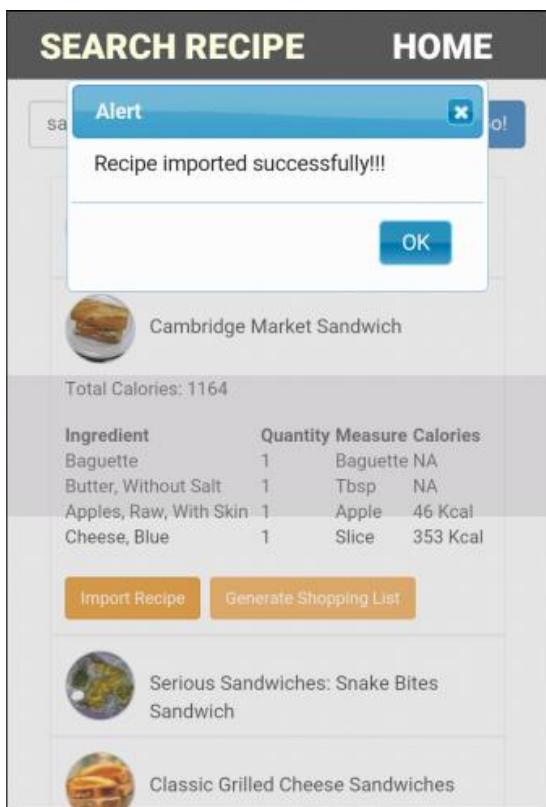
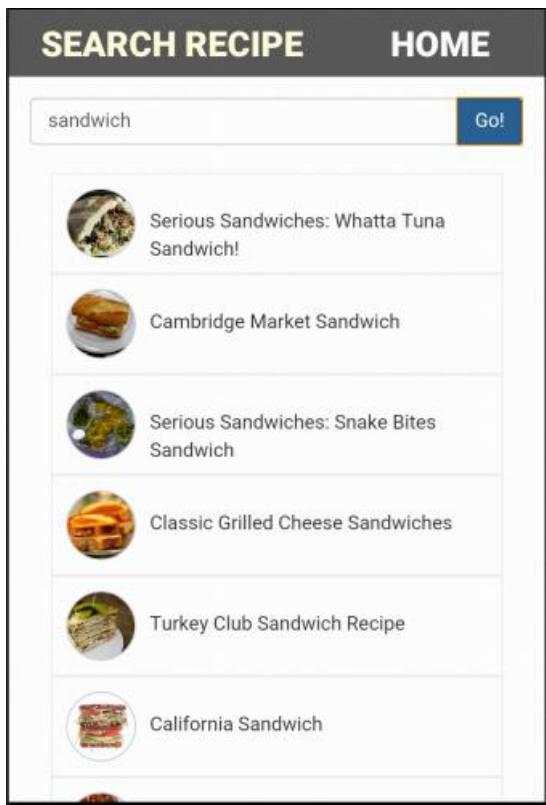


#### Search Recipe page:

- The user gets the search recipe page when he clicks the search recipe button in My Recipes page.
- The user can search any recipe by giving the keywords in the search bar.

#### Technical Details:

- The client side logic is implemented using web technologies JavaScript, JQuery & Ajax
- Used bootstrap style sheets.



### Search Recipe

- User enters sandwich as a search key to get various recipe on sandwich
- When user clicks the Go button the list of recipes of sandwich are displayed.
- When user selects a particular recipe, the two buttons are displayed.
- Import recipe button:* The user can store the particular recipe for future reference by clicking this button.
- Generate Shopping List button:* The user can see whether he/she are having all the ingredients needed for a particular recipe.

### Technical Details:

- We are using the recipe & diet api to fetch the recipe information based on keyword search.

### Search Recipe

- When user clicks on import recipe button, alert box is displayed showing Recipe imported successfully.
- The recipe is stored in the ingredients table in the remote database when import recipe button is clicked.

### Technical Details:

- We have created a web service to add these imported recipes into ingredients table in the database.

The screenshot shows a search interface for recipes. At the top, there are two buttons: "SEARCH RECIPE" and "HOME". Below the search bar, the word "sandwich" is entered, and a blue "Go!" button is visible. The search results list two items:

- 1. Serious Sandwiches: Whatta Tuna Sandwich! (with a thumbnail image)
- 2. Cambridge Market Sandwich (with a thumbnail image)

Below the results, it says "Total Calories: 1164". A table displays the ingredients with their quantities, measures, and calories:

Ingredient	Quantity	Measure	Calories
Baguette	1	Baguette	NA
Butter, Without Salt	1	Tbsp	NA
Apples, Raw, With Skin	1	Apple	46 Kcal
Cheese, Blue	1	Slice	353 Kcal

At the bottom, there are two buttons: "Import Recipe" and "Generate Shopping List".

**Shopping List**

- Apple
- Baguette
- Blue Cheese
- Unsalted Butter

### Search Recipe

- If user wants to check whether the groceries sufficient for the recipe ingredients, he/she will get this information by clicking Generate Shopping List button.

### Technical Details:

- We have created a web service to compare the grocery\_list table with the ingredients table and retrieve the items accordingly.

The screenshot shows a shopping list page. At the top, there are two buttons: "View Shopping List" and "Edit". Below the buttons, the word "Name" is displayed above a list of grocery items:

- black olives
- blue cheese
- butter
- charred
- cumin seeds
- eggplant
- garlic
- ginger
- lemon juice
- oil
- olive oil

### Search Recipe

- When user clicks the Generate Shopping List button, the grocery items which he need to buy are generated and displayed to the user.
- The user can also delete any grocery items from the shopping list by clicking the edit button.

**CREATE RECIPE**      **HOME**

Create New Recipe

Recipe Name  
ingredient  
quantity  
units

Add More

Save

**CREATE RECIPE**      **HOME**

Create New Recipe

orange juice  
oranges  
2  
pieces

Add More

Save

### Create Recipe

- User can create his/her own recipes by clicking the create recipe button on My Recipes page.
- Create new recipe page will be displayed where user can enter the recipe information.
- The Add More button is used to add more ingredients.
- The recipe will be created by clicking the Save button.

### Create Recipe

- The user has entered the recipe details in the fields.

### Technical Details:

- We have created a web service to insert the data into the ingredients table in the remote database.

**CREATE RECIPE**      **HOME**

Create New Recipe

orange juice
oranges
2
pieces

**Add More**

ingredient	quantity
units	
ingredient	quantity
units	
ingredient	quantity
units	

**Save**

**CREATE RECIPE**      **HOME**

Create New Recipe

orange juice
oranges
2
pieces

**Add More**

milk	100
ml	
sugar	50
gm	
water	20
ml	

**Save**

### Create Recipe

- When user clicks on Add more button, more fields are displayed to add more ingredients.

### Create Recipe

- When user clicks on save button the recipe is created and inserted into the ingredients table in the remote database.



### Share Recipe

- User can share recipes by clicking on Share Recipe button in My Recipes page.
- In share recipe page, user can get the particular recipe by giving the recipe name and can share with others.
- User can share imported recipes or his own created recipes to twitter or Facebook.

A screenshot of the "SHARE RECIPE" application. The header shows "SHARE RECIPE" and "HOME".

mango juice

[Get Recipe](#)

[Tweet](#)

Ingredient	Quantity	Label
mangoes	2	cups
milk	50	ml
sugar	100	grams
water	100	ml

[Facebook](#)

### Share Recipe

- By clicking Get recipe button the list of ingredients for the particular recipe name given are retrieved.
- It also displays two options i.e. twitter or Facebook through which user can share the recipes to others.

The screenshot shows the Twitter login interface. At the top, the URL is https://twitter.com/intent/tw. Below it, there's a Twitter logo and a 'Sign up' button. The main area has a heading 'What's happening?' followed by a box containing a mango juice recipe: 'Try this mango juice Recipe-- mangoes 2 cups, milk 50 ml, sugar 100 grams, water 100 ml.' Below this are input fields for 'Phone, email or username' (with a character count of 40) and 'Password'. There's also a 'Remember me' checkbox and a 'Forgot password?' link. A blue 'Log in and Tweet' button is at the bottom. A 'New to Twitter?' section is visible at the bottom, with a 'Sign up' button.

### Share Recipe

- If user wants to share the recipe to twitter the popup is displayed with recipe name and ingredients list.
- The user should login to twitter if he/she wants to share.

The screenshot shows a tweet from 'vinaya reddy' (@vinayareddy15) posted at 2:05am - 1 May 15. The tweet content is the same mango juice recipe: 'Try this mango juice Recipe-- mangoes 2 cups, milk 50 ml, sugar 100 grams, water 100 ml,'. Below the tweet are standard mobile Twitter interaction icons: back, forward, star, and more. A reply box is visible at the bottom, starting with 'Reply to @vinayareddy15'.

### Share Recipe

- After sharing the recipe with others, the recipe is posted on to the wall.



### Share Recipe

- User can also share the recipe link to the Facebook.
- User must sign in through Facebook account to share the link.



### Share Recipe

- After sharing through Facebook, it is posted to Facebook wall.

**UPDATE RECIPE**      **HOME**

RecipeName

**Get Recipe** **Add** **Delete**

**UPDATE RECIPE**      **HOME**

mango juice

**Get Recipe** **Add** **Delete**

Ingredient	Quantity	Units
mangoes	2	cups
milk	50	ml
sugar	100	grams
water	100	ml

### Update Recipe

- The recipes imported or created can be updated by clicking update recipe button on My Recipes page.
- The Get Recipe button is used to retrieve the recipe details.
- The Add button is used to add the ingredients for a particular recipe.
- The delete button deletes the ingredient from the recipe.

### Update Recipe

- The ingredients list for a particular recipe are retrieved by clicking Get Recipe button.

### Technical Details:

- We have created a web service to retrieve the data from the ingredients table in the remote database.

Ingredient	Quantity	Units
mangoes	2	cups
milk	50	ml
sugar	100	grams
water	100	ml

### Update Recipe

- By clicking the delete button, the red colored icons are displayed on left side for each ingredient.

### Technical Details:

- We have created a web service to update the modified recipes in the table.

milk	50	ml
sugar	100	grams
water	100	ml

### Update Recipe

- When the red colored icon for a particular ingredient is clicked, alert box pops up to confirm deletion.
- By clicking OK, the specific ingredient will be deleted.

**UPDATE RECIPE** **HOME**

mango juice

Get Recipe	Add	Delete
Ingredient	Quantity	Units
- mangoes	2	cups
- milk	50	ml
- sugar	100	grams

### Update Recipe

- The recipe ingredient list which was previously modified.
- The specific ingredient is deleted and the recipe is updated.

**UPDATE RECIPE** **HOME**

mango juice

Get Recipe	Add	Delete
Ingredient	Quantity	Units
mangoes	<input type="text" value="4"/>	cups
milk	50	ml
sugar	100	grams

### Update Recipe

- The ingredients list can also modified by clicking on the particular field.

**UPDATE RECIPE**      **HOME**

mango juice

<a href="#">Get Recipe</a>	<a href="#">Add</a>	<a href="#">Delete</a>
Ingredient	Quantity	Units
mangoes	4	cups
milk	50	ml
sugar	100	grams

### Update Recipe

- The ingredients list for a particular recipe are retrieved by clicking Get Recipe button which shows the updated recipe ingredients list.

**UPDATE RECIPE**      **HOME**

mango juice

<a href="#">Get Recipe</a>	<a href="#">Add</a>	<a href="#">Delete</a>
Ingredient	Quantity	
mangoes	2	
milk	50	
sugar	100	
<input type="text" value="ingredient"/>	<input type="text" value="quantity"/>	
<a href="#">Update</a>		

### Update Recipe

- By clicking Add button, the field is displayed where we can add the ingredient for a recipe.

**UPDATE RECIPE**      **HOME**

mango juice

Get Recipe	Add	Delete
Ingredient	Quantity	
mangoes	2	
milk	50	
sugar	100	
water	10	
<b>Update</b>		

### Update Recipe

- The ingredients are entered in the fields displayed.

**UPDATE RECIPE**      **HOME**

mango juice

Alert
X

Recipe updated successfully!!!

OK

Get Recipe	Add	Delete
Ing	Quantity	
mangoes	2	
milk	50	
sugar	100	
water	10	
<b>Update</b>		

### Update Recipe

- The ingredients are added by clicking update button.
- Alert dialog pops up showing recipe updates successfully.

The screenshot shows a mobile application interface titled "UPDATE RECIPE" at the top left and "HOME" at the top right. Below the title, the text "mango juice" is displayed. At the bottom of the screen are three buttons: "Get Recipe" (white background), "Add" (blue background), and "Delete" (blue background). A table lists the ingredients:

Ingredient	Quantity	Units
mangoes	4	cups
milk	50	ml
sugar	100	grams
water	10	ml

### Update Recipe

- The updated ingredients list for the recipe are displayed.

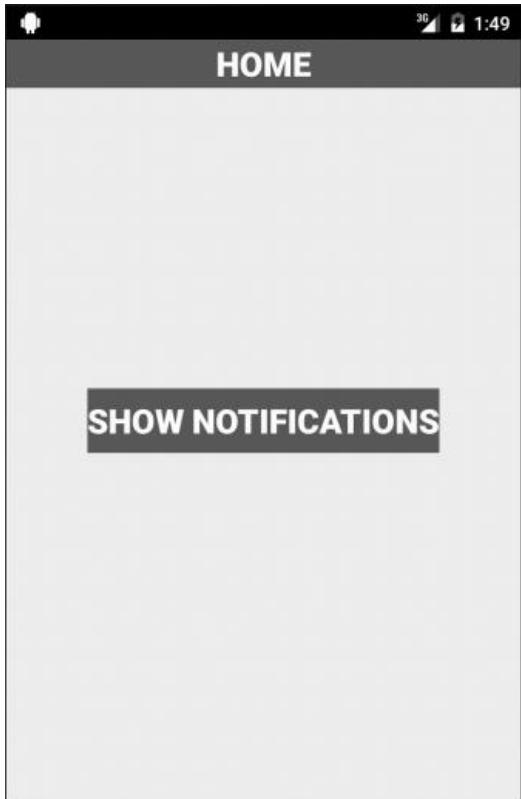
The screenshot shows a mobile application interface titled "HOME" at the top. At the bottom of the screen is a large button with the text "SHOW NOTIFICATIONS" in white capital letters.

### Show Notifications Page

- The notifications if any item is being expired are displayed by clicking show notifications button in home page.

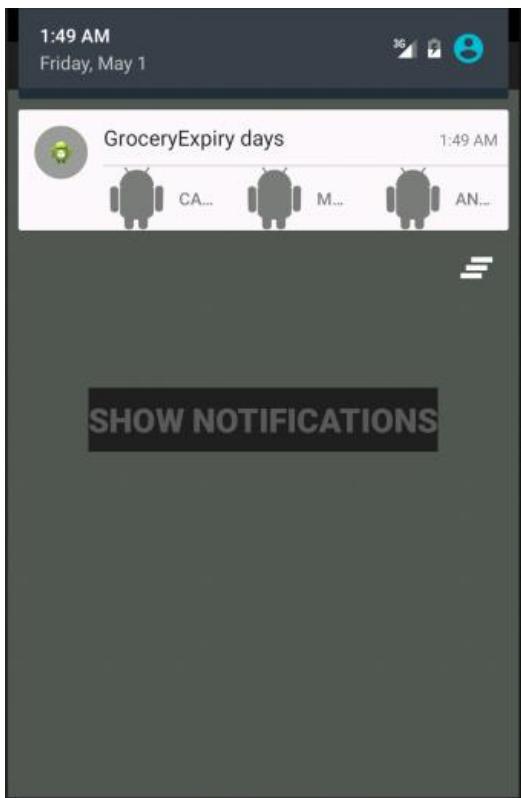
### Technical Details:

- Client side logic is implemented in android studio as a native app.



#### Show Notifications Page

- The android icon pops up when show notifications button is clicked.



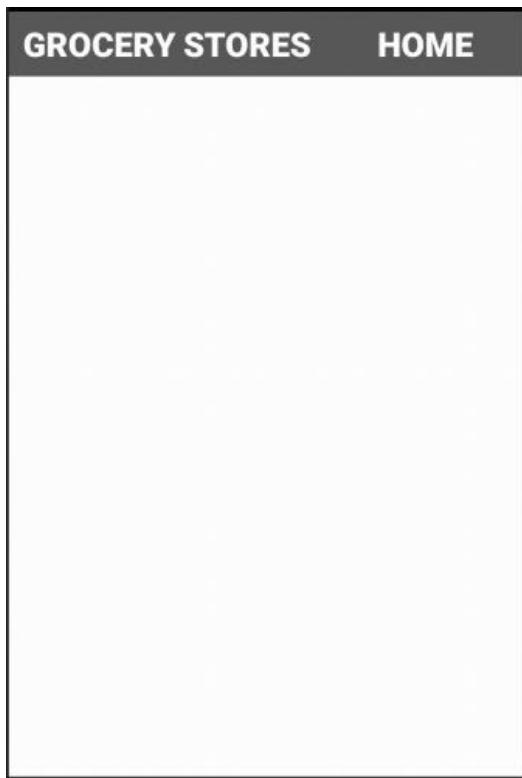
#### Show Notifications Page

- The notifications are displayed saying Grocery Expiry days when the popup icon is scrolled down.



#### Show Notifications Page

- The notifications are displayed for the grocery items which are about to expire.

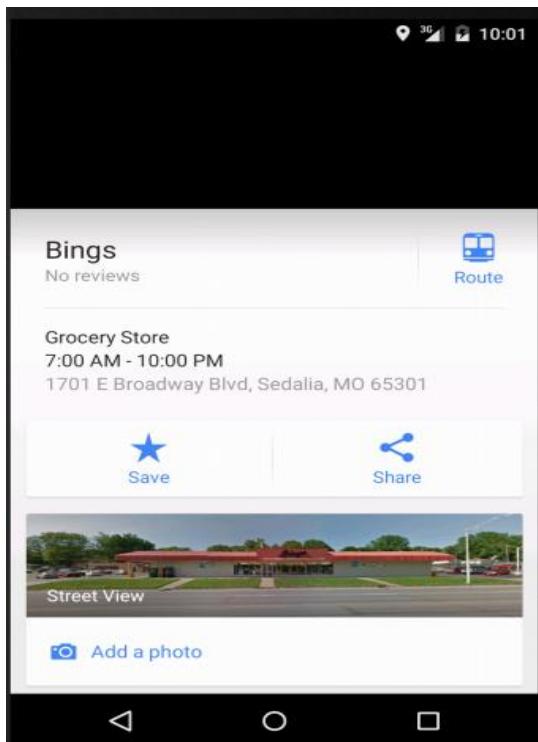


#### Grocery Stores Page

- The nearby grocery stores could be found by clicking the Grocery Stores button in Home page.

#### Technical Details:

- Client side logic is implemented in android studio as a native app.



By Clicking on grocery stores, the application displays the nearby grocery stores as shown in the screenshot.



#### Shopping List Page

- The shopping list can be known to user by clicking on shopping list button on Home page.

**SHOPPING LIST**      **HOME**

**View Shopping List**      **Edit**

Name
apple
baguette
butter
cumin seeds
eggplant
garlic
ginger
lemon juice
oil
olive oil
onions

### Shopping List Page

- The list is displayed when user clicks on view shopping list button.
- The list can be modified by clicking on edit button.

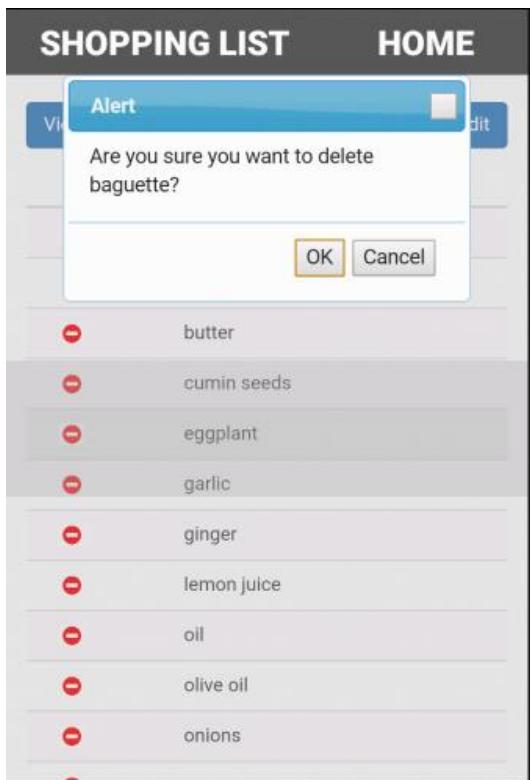
**SHOPPING LIST**      **HOME**

**View Shopping List**      **Edit**

Name
apple
baguette
butter
cumin seeds
eggplant
garlic
ginger
lemon juice
oil
olive oil
onions

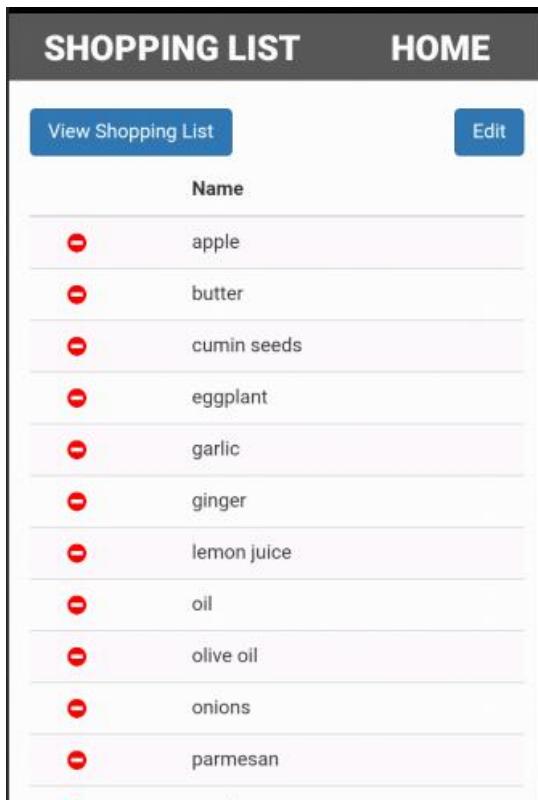
### Shopping List Page

- When user clicks edit button the red colored icons are displayed on left side of each item.
- If user clicks this icon the particular item will be deleted.



### Shopping List Page

- The alert dialog box pops up to confirm the deletion of the item.
- On clicking OK button, the item will be deleted from the shopping list.



### Shopping List Page

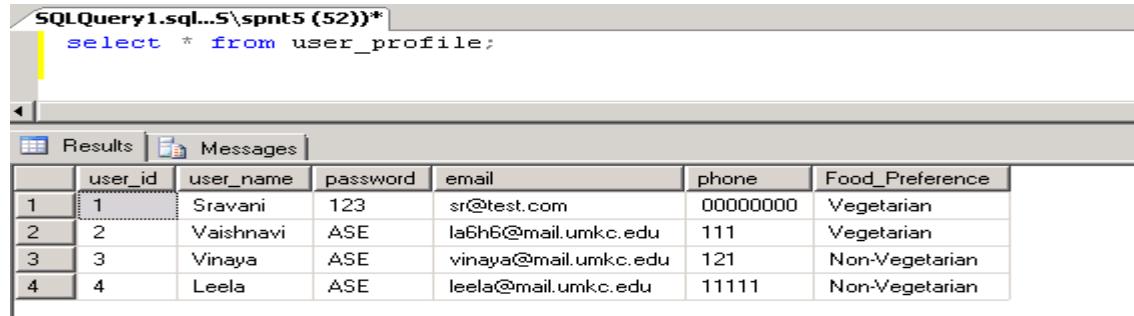
- The updated list is displayed after deleting the "baguette" from the shopping list.

## 2) Database Implementation

We have created Calorie2Grocery database in SQL Server and developed the following tables for our application:

### User\_profile

This table is used to store the user information like user\_id, username, password, email, phone number and food preference.



The screenshot shows a SQL query window with the following content:

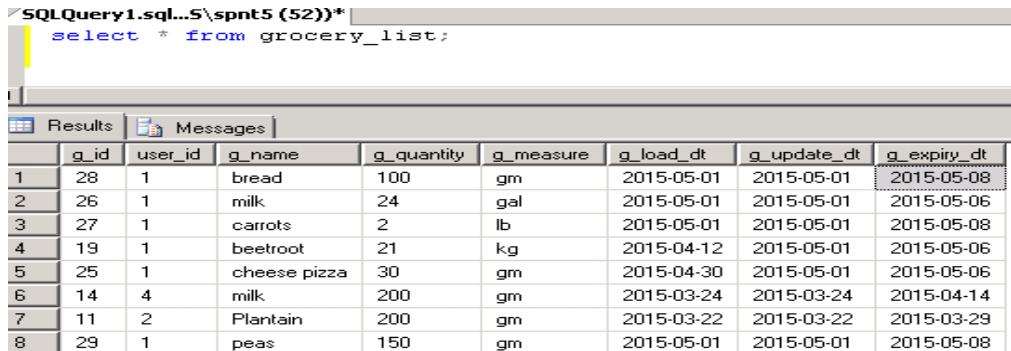
```
SQLQuery1.sql...5\spnt5 (52)*
select * from user_profile;
```

The results pane displays the following data:

	user_id	user_name	password	email	phone	Food_Preference
1	1	Sravani	123	sr@test.com	00000000	Vegetarian
2	2	Vaishnavi	ASE	la6h6@mail.umkc.edu	111	Vegetarian
3	3	Vinaya	ASE	vinaya@mail.umkc.edu	121	Non-Vegetarian
4	4	Leela	ASE	leela@mail.umkc.edu	11111	Non-Vegetarian

- **Grocery\_list**

This table is used to store the list of grocery items that are available at users home. In this table user can save the grocery item along with the quantity.



The screenshot shows a SQL query window with the following content:

```
SQLQuery1.sql...5\spnt5 (52)*
select * from grocery_list;
```

The results pane displays the following data:

	g_id	user_id	g_name	g_quantity	g_measure	g_load_dt	g_update_dt	g_expiry_dt
1	28	1	bread	100	gm	2015-05-01	2015-05-01	2015-05-08
2	26	1	milk	24	gal	2015-05-01	2015-05-01	2015-05-06
3	27	1	carrots	2	lb	2015-05-01	2015-05-01	2015-05-08
4	19	1	beetroot	21	kg	2015-04-12	2015-05-01	2015-05-06
5	25	1	cheese pizza	30	gm	2015-04-30	2015-05-01	2015-05-06
6	14	4	milk	200	gm	2015-03-24	2015-03-24	2015-04-14
7	11	2	Plantain	200	gm	2015-03-22	2015-03-22	2015-03-29
8	29	1	peas	150	gm	2015-05-01	2015-05-01	2015-05-08

- **Recipes**

This table stores the information regarding the recipe and the ingredient list when the user searches for a recipe and import the ingredients of the recipe.

SQLQuery1.sql...S\spnt5 (52)\*

```
select * from recipes;
```

Results | Messages

	recipe_name	ingredient_name	quantity	units	load_dt
1	Chorizo And Roasted Pepper Pizza	cheese pizza	14	oz	2015-05-01 20:31:00.000
2	Cheeseless Pizza	roasted red bell peppers	3	pepper	2015-04-30 18:56:00.000
3	Cheeseless Pizza	roasted red bell peppers	1	pepper	2015-04-30 18:56:00.000
4	Grilled eggplant and olive pizza	provolone	5	oz	2015-04-30 19:17:00.000
5	Grilled eggplant and olive pizza	pizza	1	lb	2015-04-30 19:17:00.000
6	Grilled eggplant and olive pizza	green olives	0.33333334	cup	2015-04-30 19:17:00.000
7	Grilled eggplant and olive pizza	provolone	5	oz	2015-04-30 19:17:00.000
8	Grilled eggplant and olive pizza	parsley	0.25	cup	2015-04-30 19:17:00.000
9	Grilled eggplant and olive pizza	garlic	1	clove	2015-04-30 19:17:00.000

- Shopping List:** This table stores the aggregated result for all the shopping lists generated by the user during different recipe searches.

SQLQuery1.sql...S\spnt5 (52)\*

```
select * from shopping_list;
```

Results | Messages

	grocery_name	load_dt
1	roasted red peppers	2015-05-01 20:00:00.000
2	unsalted butter	2015-04-30 21:47:00.000
3	sandwich bread	2015-04-30 21:48:00.000
4	ginger	2015-04-30 22:03:00.000
5	lemon juice	2015-04-30 20:59:00.000
6	onions	2015-04-30 20:59:00.000
7	parmesan	2015-04-30 20:59:00.000
8	parsley	2015-04-30 20:59:00.000
9	pepper	2015-04-30 20:59:00.000
10	pizza	2015-04-30 20:59:00.000
11	pizza dough	2015-04-30 20:59:00.000
12	prosciutto	2015-04-30 20:59:00.000
13	provolone	2015-04-30 20:59:00.000
14	potatoes	2015-04-30 22:03:00.000
15	tomato sauce	2015-04-30 20:59:00.000
16	red chile pepper	2015-04-30 22:03:00.000
17	red onion	2015-04-30 22:03:00.000
18	sea salt	2015-04-30 22:03:00.000
19	spinach	2015-04-30 22:03:00.000
20	frozen pizzas	2015-05-01 19:32:00.000
21	pine nuts	2015-05-01 16:00:00.000

Query executed successfully.

## Testing

### 1) UI Testing:

The following test cases has been written for UI testing and the actual output is validated with expected output. All of the below test cases which have been written obtained positive results.

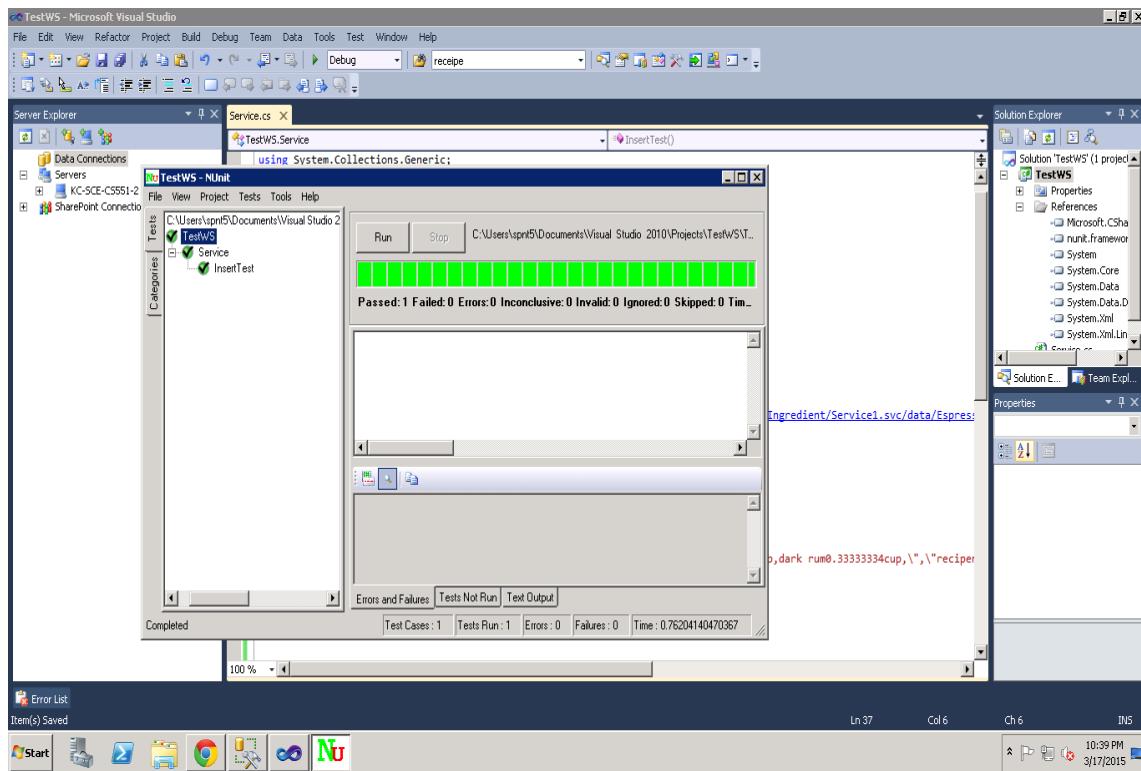
Page	Button	Expected	Actual
View Grocery list	view	Should display grocery list	Displaying grocery list
	edit	Should modify the ingredients, quantity and label	Able to modify the ingredients, quantity and label
	delete	Should delete the grocery items	Deleted the grocery items
Calorie count	Browse for a recipe	Should display list of the different recipes	Displays the list of different recipes
	Select a particular recipe	Should display the ingredients list for the selected recipe	Displays the ingredients list for the selected recipe
	Calorie count	Should display individual and total calorie count of ingredients	Displaying individual and total calorie count of ingredients
Get recipe and modify	Import recipe	The list of recipes should be added to the database	The list of recipes are added to the database
	View recipe	Should display the recipe list	Displays the recipe list
	Update recipe	Should allow us to update the ingredient, quantity , label of recipe	Allowing us to update the ingredient, quantity , label of recipe
Create new recipe	Create recipe	Add new recipe ingredients, quantity, label to database	Adding new recipe ingredients, quantity, label to database
Get recipe and share	Get recipe	should get imported and created recipes	Able to get imported and created recipes
	Share recipe	Should share recipes to twitter and facebook	Able to share recipes to twitter and facebook
Notification	Send Notifications	Send Notifications to expiry dates	Sending Notifications to expiry dates

Generate Shopping List	Shopping List	Generates an aggregated shopping list of all the recipes	Generating all the shopping list items from all the imported recipes
------------------------	---------------	--	--

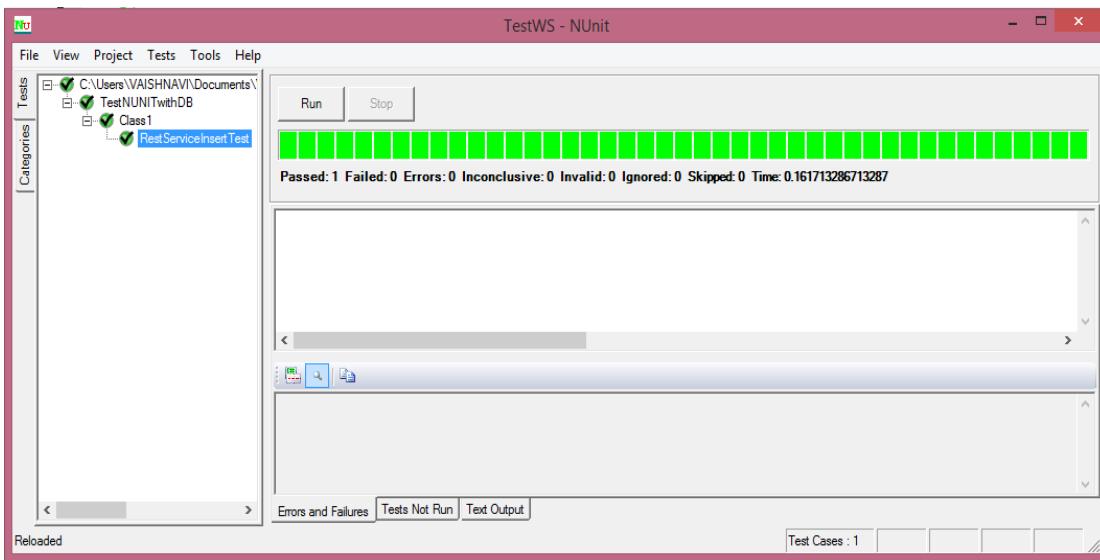
## 2) Data Validation:

We have implemented N-Unit test cases to validate the data for all the newly created web services. We have used Assert methods to compare the actual output with the expected output.

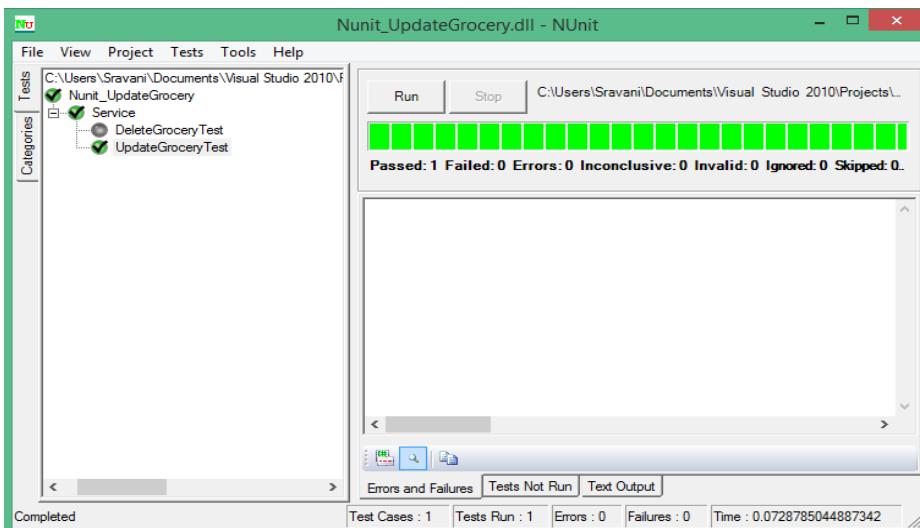
**Screenshot of N-Unit validation: This is used to validate the data for Import recipe API:**



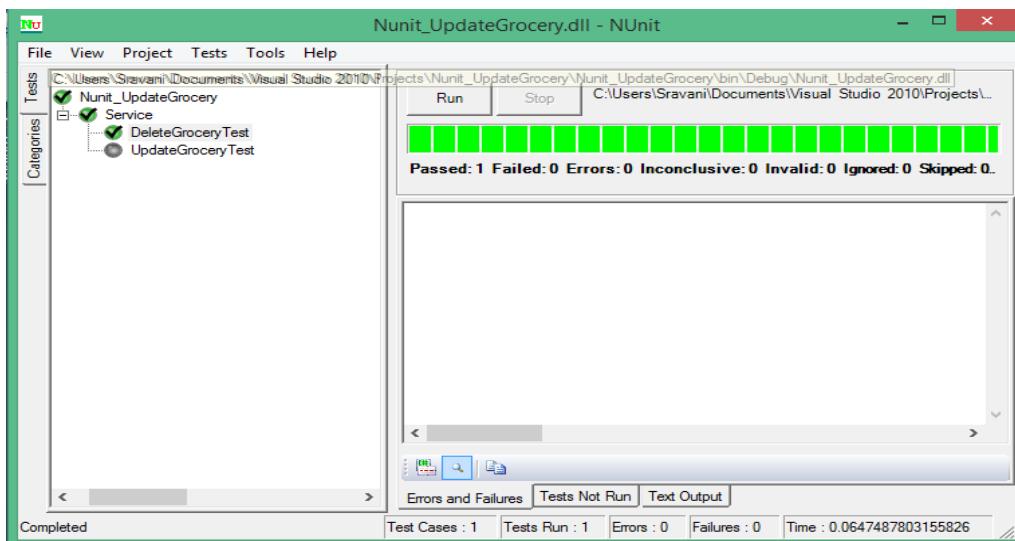
### Screenshot for N-Unit validation of Add groceries API:



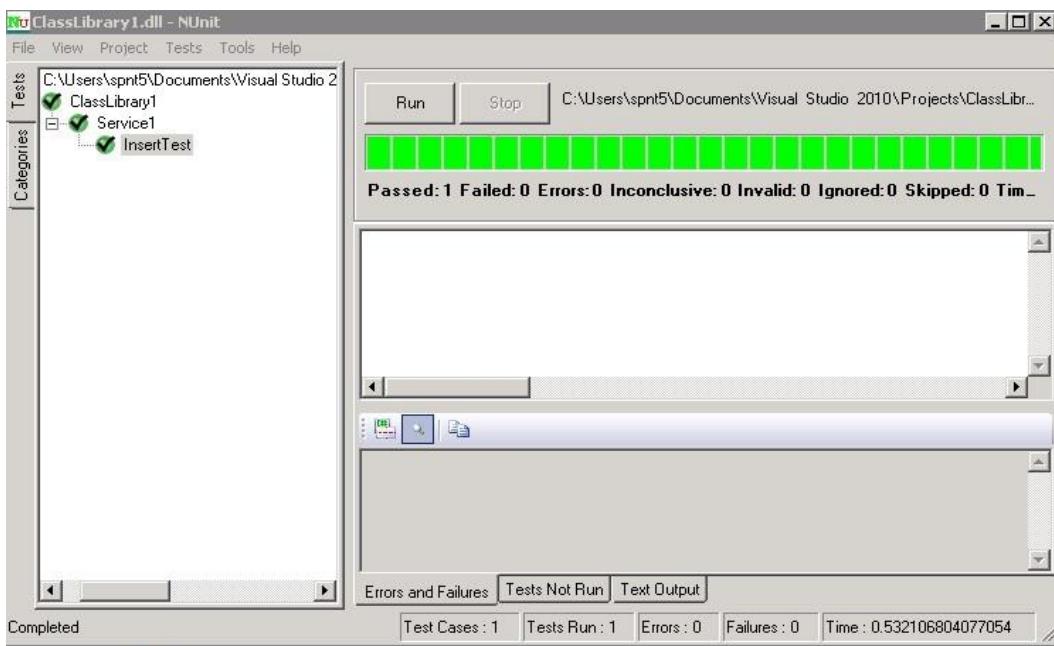
**Screenshot of N-Unit validation:** This is used to validate the data for update grocery item method in the rest service.



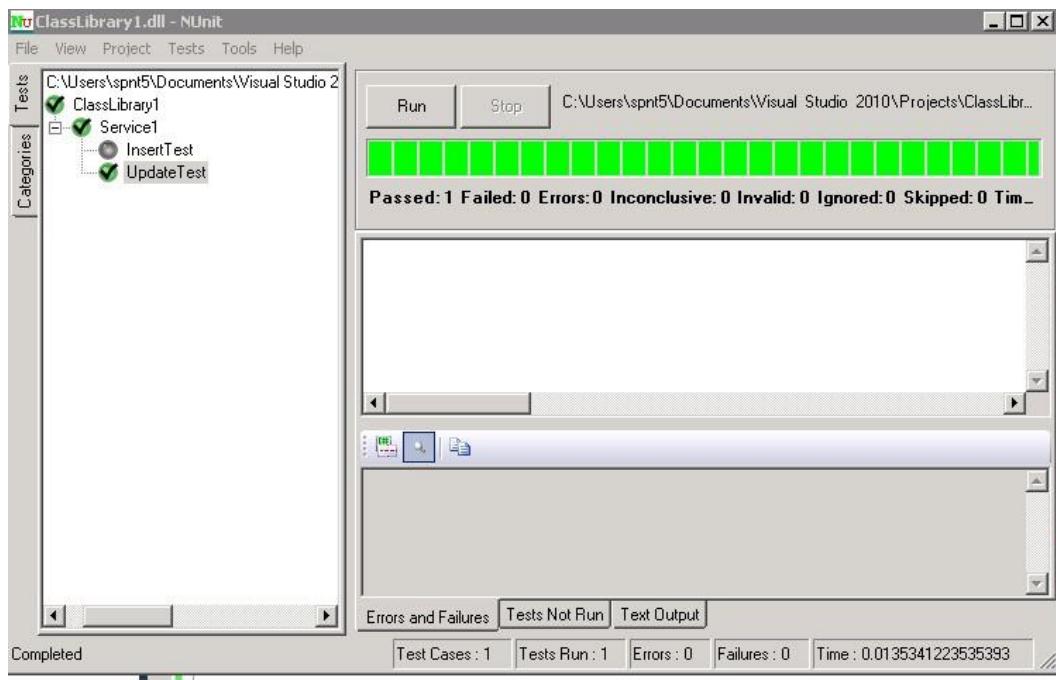
### Screenshot for N-Unit validation of delete groceries API:



Screenshot for N-Unit validation of Import groceries API:



Screenshot for N-Unit validation of Update groceries API:



### **Performance Testing:**

The following are the tools we used for performance testing:

- YSlow analyzer
- Firebug

1. **YSlow Analyzer:** This is a free, open source testing tool that analyzes web pages and suggests ways to improve their performance. The below screenshots shows the performance analysis of the web pages that are created in our app. This analysis is done using YSlow analyzer(Google chrome extension).

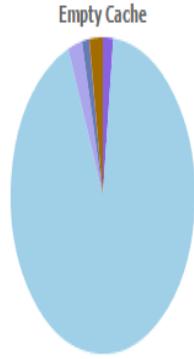
**Testing for a web page that generates notification about expiry dates**

[Home](#) [Grade](#) [Components](#) [Statistics](#)

## Statistics

The page has a total of 36 HTTP requests and a total weight of 1802.8K bytes with empty cache

### WEIGHT GRAPHS

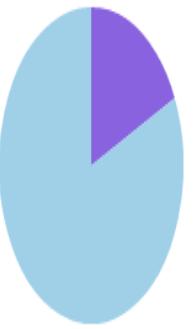


#### HTTP Requests - 36

##### Total Weight - 1802.8K

1 HTML/Text	32.7K
23 JavaScript File	1659.7K
6 Stylesheet File	44.9K
4 CSS Image	22.3K
1 Favicon	1.0K
1 undefined	41.9K

#### Primed Cache



#### HTTP Requests - 7

##### Total Weight - 182.0K

1 HTML/Text	32.7K
4 JavaScript File	149.2K
1 Favicon	0.0K
1 undefined	0.0K

Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 1

chrome-extension://ninejjcohidippngpapiilmkgllmakh/yslow.html#655

Home Grade Components Statistics | Rulesets YSlow(V2) Edit Help

**Grade C** Overall performance score 74 Ruleset applied: YSlow(V2) URL: http://jsfiddle.net/la6h6/8ksfm4b0/6/

ALL (23) FILTER BY: CONTENT (6) COOKIE (2) CSS (6) IMAGES (2) JAVASCRIPT (4) SERVER (6)

**F Make fewer HTTP requests**

**F Use a Content Delivery Network (CDN)**

**A Avoid empty src or href**

**F Add Expires headers**

**B Compress components with gzip**

**A Put CSS at top**

**F Put JavaScript at bottom**

**A Avoid CSS expressions**

**n/a Make JavaScript and CSS external**

**A Reduce DNS lookups**

**F Minify JavaScript and CSS**

**A Avoid URL redirects**

**A Remove duplicate JavaScript and CSS**

**A Configure entity tags (ETags)**

**A Make AJAX cacheable**

**A Use GET for AJAX requests**

**A Reduce the number of DOH elements**

**A Avoid HTTP 404 (Not Found) error**

**A Reduce cookie size**

**F Use cookie-free domains**

**A Avoid AlphaImageLoader filter**

**A Do not scale images in HTML**

**A Make favicon small and cacheable**

This page has 23 external Javascript scripts. Try combining them into one.  
This page has 6 external stylesheets. Try combining them into one.

Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.

[»Read More](#)

Copyright © 2015 Yahoo Inc. All rights reserved.

Screen Shot: 2

chrome-extension://ninejjcohidippngpapiilmkgllmakh/yslow.html#655

Home Grade Components Statistics | Rulesets YSlow(V2) Edit Help

**Components** The page has a total of 36 components and a total weight of 1802.8K bytes [»Expand All](#)

†TYPE	SIZE (KB)	GZIP (KB)	COOKIE RECEIVED (bytes)	COOKIE SENT (bytes)	HEADERS	URL	EXPIRES (Y/M/D)	RESPONSE TIME (ms)	ETAG	ACTION
doc (1)	32.7K									
js (23)	1684.5K									
css (6)	44.9K									
cssimage (4)	22.3K									
favicon (1)	1.0K									
font (1)	41.9K									

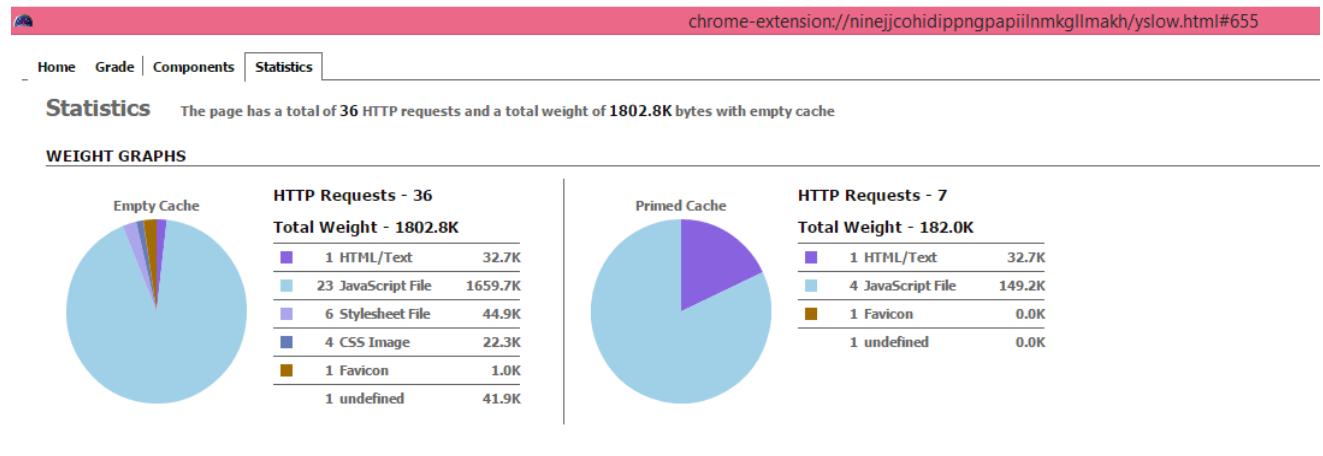
\* type column indicates the component is loaded after window.onload event

† denotes 1x1 pixels image that may be image beacon

Copyright © 2015 Yahoo Inc. All rights reserved.

Screen Shot: 3

## Testing for a web page that searches a recipe from the database and share it on facebook and twitter



Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 1

The screenshot shows the YSlow Components report with a table of network requests. The table includes columns for TYPE, SIZE (KB), GZIP (KB), COOKIE RECEIVED (bytes), COOKIE SENT (bytes), HEADERS, URL, EXPIRES (Y/M/D), RESPONSE TIME (ms), ETag, and ACTION.

TYPE	SIZE (KB)	GZIP (KB)	COOKIE RECEIVED (bytes)	COOKIE SENT (bytes)	HEADERS	URL	EXPIRES (Y/M/D)	RESPONSE TIME (ms)	ETAG	ACTION
doc(1)	35.9K									
js(23)	1684.5K									
css(6)	44.9K									
cssimage(5)	22.6K									
favicon(1)	1.0K									
font(1)	41.9K									

\* type column indicates the component is loaded after window.onload event  
† denotes 1x1 pixels image that may be image beacon

Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 2

chrome-extension://ninejjcohidippngpapiilmkgllmakh/yslow.html#655

Home Grade Components Statistics Rulesets YSlow(V2) Edit Help

**Grade C** Overall performance score 74 Ruleset applied: YSlow(V2) URL: http://jsfiddle.net/la6h6/yfsbp3mq/1/

All (23) FILTER BY: CONTENT (6) | COOKIE (2) | CSS (6) | IMAGES (2) | JAVASCRIPT (4) | SERVER (6)

**F Make fewer HTTP requests**

This page has 23 external Javascript scripts. Try combining them into one. This page has 6 external stylesheets. Try combining them into one.

Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.

[Read More](#)

Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 3

## Testing for a web page for Creating a recipe

chrome-extension://ninejjcohidippngpapiilmkgllmakh/yslow.html#655

Home Grade Components Statistics Rulesets YSlow(V2) Edit Help

**Statistics** The page has a total of 36 HTTP requests and a total weight of 1802.8K bytes with empty cache

**WEIGHT GRAPHS**

HTTP Requests - 36		HTTP Requests - 7	
Total Weight - 1802.8K		Total Weight - 182.0K	
1 HTML/Text	32.7K	1 HTML/Text	32.7K
23 JavaScript File	1659.7K	4 JavaScript File	149.2K
6 Stylesheet File	44.9K	1 Favicon	0.0K
4 CSS Image	22.3K	1 undefined	0.0K
1 Favicon	1.0K		
1 undefined	41.9K		

Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 1

chrome-extension://ninejjcohidipngpapiiinmkglmak/h/yslow.html#655

Home Grade Components Statistics Rulesets YSlow(V2) Edit Help »Expand All

**Components** The page has a total of 36 components and a total weight of 1802.8K 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)	32.5K									
js (23)	1684.5K									
css (6)	44.9K									
cssimage (4)	22.3K									
favicon (1)	1.0K									
font (1)	41.9K									

\* type column indicates the component is loaded after window onload event  
† denotes 1x1 pixels image that may be image beacon

Copyright © 2015 Yahoo Inc. All rights reserved.

Screen Shot: 2

chrome-extension://ninejjcohidipngpapiiinmkglmak/h/yslow.html#655

Home Grade Components Statistics Rulesets YSlow(V2) Edit Help »

**Grade C** Overall performance score 74 Ruleset applied: YSlow(V2) URL: http://jsfiddle.net/vp8t3/z0ef8wf0/

ALL (23) FILTER BY: CONTENT (6) | COOKIE (2) | CSS (6) | IMAGES (2) | JAVASCRIPT (4) | SERVER (6) Tweet Share

**F Make fewer HTTP requests**

Grade F on Make fewer HTTP requests

This page has 23 external Javascript scripts. Try combining them into one.  
The page has 6 external stylesheets. Try combining them into one.

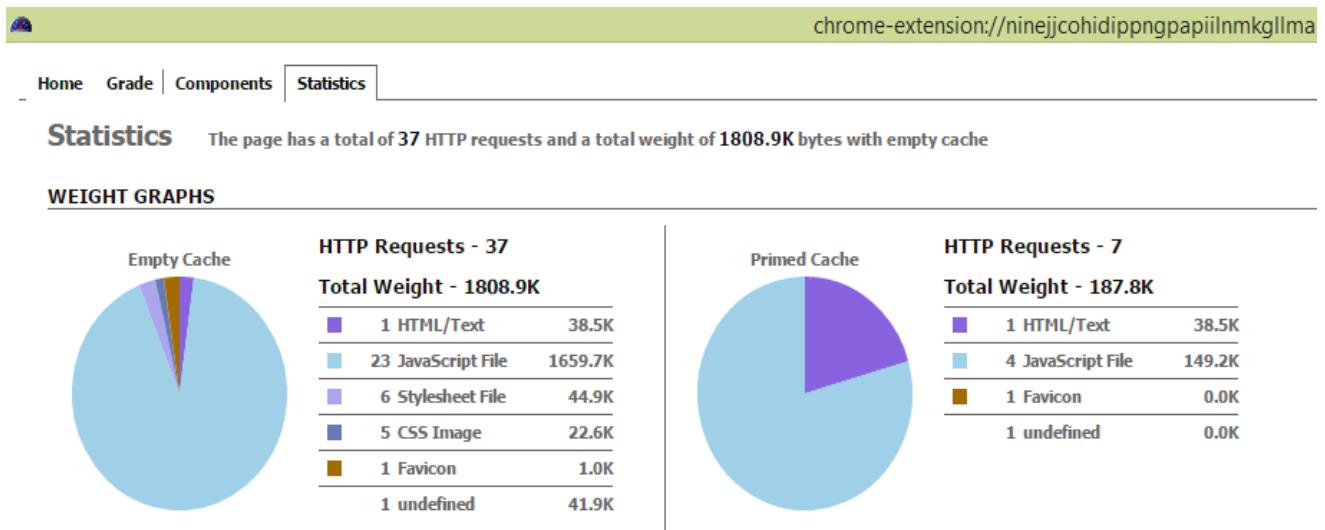
Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.

»Read More

Copyright © 2015 Yahoo Inc. All rights reserved.

Screen Shot: 3

## Web Page for displaying a recipe long with calorie count



Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 1

The screenshot shows a detailed table of components with the following columns: TYPE, SIZE (KB), GZIP (KB), COOKIE RECEIVED (bytes), COOKIE SENT (bytes), HEADERS, URL, EXPIRES (Y/M/D), RESPONSE TIME (ms), ETag, and ACTION.

TYPE	SIZE (KB)	GZIP (KB)	COOKIE RECEIVED (bytes)	COOKIE SENT (bytes)	HEADERS	URL	EXPIRES (Y/M/D)	RESPONSE TIME (ms)	ETAG	ACTION
doc (1)	38.5K					<a href="http://jsfiddle.net/la6h6/qf6y2smq/">http://jsfiddle.net/la6h6/qf6y2smq/</a>				
doc	38.5K	38.5K		665		<a href="http://jsfiddle.net/la6h6/qf6y2smq/8/">http://jsfiddle.net/la6h6/qf6y2smq/8/</a>	no expires	0		
is (23)	1684.5K									
css (6)	44.9K									
css	0.6K	0.6K		665		<a href="http://jsfiddle.net/css/normalize.css?...">http://jsfiddle.net/css/normalize.css?...</a>	2015/5/3	0		
css	0.2K	0.2K		665		<a href="http://jsfiddle.net/css/tricks.css?...">http://jsfiddle.net/css/tricks.css?...</a>	2015/5/3	0		
css	18.7K	18.7K		665		<a href="http://jsfiddle.net/css/elusive-webfont.css?...">http://jsfiddle.net/css/elusive-webfont.css?...</a>	2015/5/3	0		
css	8.0K	8.0K		665		<a href="http://jsfiddle.net/is/codemirror/lib/codemirror.css?...">http://jsfiddle.net/is/codemirror/lib/codemirror.css?...</a>	2015/5/3	0		
css	17.2K	17.2K		665		<a href="http://jsfiddle.net/css/screen.css?...">http://jsfiddle.net/css/screen.css?...</a>	2015/5/3	0		
css	0.0K			665		<a href="http://jsfiddle.net/css/light.css?...">http://jsfiddle.net/css/light.css?...</a>	2015/5/3	0	"5120150e-0"	
cssimage (5)	22.6K									
cssimage	20.8K			665		<a href="http://jsfiddle.net/img/initializing.png">http://jsfiddle.net/img/initializing.png</a>	2015/5/3	0	"5128cfbb-515f"	
cssimage	1.1K			665		<a href="http://jsfiddle.net/img/logo.png">http://jsfiddle.net/img/logo.png</a>	2015/5/3	0	"5120150e-488"	
cssimage	0.2K			665		<a href="http://jsfiddle.net/img/remove-resources.png">http://jsfiddle.net/img/remove-resources.png</a>	2015/5/3	0	"5120150e-120"	
cssimage	0.2K			665		<a href="http://jsfiddle.net/img/handle-h.png">http://jsfiddle.net/img/handle-h.png</a>	2015/5/3	0	"5120150e-ca"	
cssimage	0.1K			665		<a href="http://jsfiddle.net/img/handle-v.png">http://jsfiddle.net/img/handle-v.png</a>	2015/5/3	0	"5120150e-c4"	
favicon (1)	1.0K									
font (1)	41.9K									

\* type column indicates the component is loaded after window onload event  
† denotes 1x1 pixels image that may be image beacon

Copyright © 2015 Yahoo! Inc. All rights reserved.

chrome-extension://ninejjcohidippngpapiilmkgillma/vslow.html#757

Screen Shot: 2

Home | Grade | Components | Statistics | Rulesets: YSlow(V2) | Edit | Help

**Grade C** Overall performance score 74 Ruleset applied: YSlow(V2) URL: http://jsfiddle.net/la6h6/gf6y2smg/8/

**All (23)** FILTER BY: CONTENT (6) | COOKIE (2) | CSS (6) | IMAGES (2) | JAVASCRIPT (4) | SERVER (6)

**F Make fewer HTTP requests**

**F Use a Content Delivery Network (CDN)**

**A Avoid empty src or href**

**F Add Expires headers**

**B Compress components with gzip**

**A Put CSS at top**

**F Put JavaScript at bottom**

**A Avoid CSS expressions**

**n/a Make JavaScript and CSS external**

**A Reduce DNS lookups**

**F Minify JavaScript and CSS**

**A Avoid URL redirects**

**A Remove duplicate JavaScript and CSS**

**A Configure entity tags (ETags)**

**A Make AJAX cacheable**

**A Use GET for AJAX requests**

**B Reduce the number of DOM elements**

**A Avoid HTTP 404 (Not Found) error**

**A Reduce cookie size**

**F Use cookie-free domains**

**A Avoid AlphaImageLoader filter**

**A Do not scale images in HTML**

**A Make favicon small and cacheable**

**Grade F on Make fewer HTTP requests**

This page has 23 external Javascript scripts. Try combining them into one. This page has 6 external stylesheets. Try combining them into one.

Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.

[Read More](#)

Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 3

### Testing a web page that edits/removes the grocery item

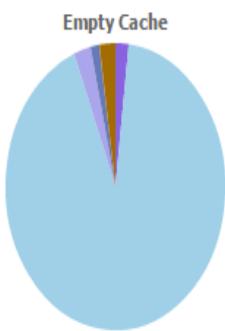
chrome-extension://ninejjcohidippngpapiilnmkgllm/

Home | Grade | Components | **Statistics**

### Statistics

The page has a total of 37 HTTP requests and a total weight of 1805.1K bytes with empty cache

### WEIGHT GRAPHS



#### HTTP Requests - 37

#### Total Weight - 1805.1K

1 HTML/Text	34.7K
23 JavaScript File	1659.7K
6 Stylesheet File	44.9K
5 CSS Image	22.6K
1 Favicon	1.0K
1 undefined	41.9K

#### Primed Cache

#### HTTP Requests - 7

#### Total Weight - 184.0K

1 HTML/Text	34.7K
4 JavaScript File	149.2K
1 Favicon	0.0K
1 undefined	0.0K

Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 1

<u>TYPE</u>	<u>SIZE (KB)</u>	<u>GZIP (KB)</u>	<u>COOKIE RECEIVED (bytes)</u>	<u>COOKIE SENT (bytes)</u>	<u>HEADERS</u>	<u>URL</u>	<u>EXPIRES (Y/M/D)</u>	<u>RESPONSE TIME (ms)</u>	<u>ETAG</u>	<u>ACTION</u>
doc (1)	34.7K									
doc	34.7K	34.7K		665	⌚	<a href="http://jsfiddle.net/spmt5/n6rl5r3s/2/">http://jsfiddle.net/spmt5/n6rl5r3s/2/</a>	no expires	0		
is (23)	1684.5K									
css (6)	44.9K									
css	0.6K	0.6K		665	⌚	<a href="http://jsfiddle.net/css/normalize.css?_">http://jsfiddle.net/css/normalize.css?_</a>	2015/5/3	0		
css	0.2K	0.2K		665	⌚	<a href="http://jsfiddle.net/css/tricks.css?_">http://jsfiddle.net/css/tricks.css?_</a>	2015/5/3	0		
css	18.7K	18.7K		665	⌚	<a href="http://jsfiddle.net/css/elusive-webfont.css?_">http://jsfiddle.net/css/elusive-webfont.css?_</a>	2015/5/3	0		
css	8.0K	8.0K		665	⌚	<a href="http://jsfiddle.net/is/codemirror/lib/codemirror.css?_">http://jsfiddle.net/is/codemirror/lib/codemirror.css?_</a>	2015/5/3	0		
css	17.2K	17.2K		665	⌚	<a href="http://jsfiddle.net/css/screen.css?_">http://jsfiddle.net/css/screen.css?_</a>	2015/5/3	0		
css	0.0K			665	⌚	<a href="http://jsfiddle.net/css/light.css?_">http://jsfiddle.net/css/light.css?_</a>	2015/5/3	0	"5120150e-0"	
cssimage (5)	22.6K									
cssimage	20.8K			665	⌚	<a href="http://jsfiddle.net/img/initializing.png">http://jsfiddle.net/img/initializing.png</a>	2015/5/3	0	"5128cfbb-515f"	
cssimage	1.1K			665	⌚	<a href="http://jsfiddle.net/img/logo.png">http://jsfiddle.net/img/logo.png</a>	2015/5/3	0	"5120150e-488"	
cssimage	0.2K			665	⌚	<a href="http://jsfiddle.net/img/remove-resources.png">http://jsfiddle.net/img/remove-resources.png</a>	2015/5/3	0	"5120150e-120"	
cssimage	0.2K			665	⌚	<a href="http://jsfiddle.net/img/handle-h.png">http://jsfiddle.net/img/handle-h.png</a>	2015/5/3	0	"5120150e-ca"	
cssimage	0.1K			665	⌚	<a href="http://jsfiddle.net/img/handle-v.png">http://jsfiddle.net/img/handle-v.png</a>	2015/5/3	0	"5120150e-c4"	
favicon (1)	1.0K									
favicon	1.0K			665	⌚	<a href="http://jsfiddle.net/favicon.png">http://jsfiddle.net/favicon.png</a>	no expires	0	"5120150e-444"	
font (1)	41.9K									
font	41.9K			665	⌚	<a href="http://jsfiddle.net/font/Elusive-Icons.eot?_">http://jsfiddle.net/font/Elusive-Icons.eot?_</a>	no expires	0	"51af0907-a404"	

\* type column indicates the component is loaded after window.onload event  
 † denotes 1x1 pixels image that may be image beacon

Copyright © 2015 Yahoo! Inc. All rights reserved.

## Screen Shot: 14

chrome-extension://ninejcohidippngpapiilmkgilmakh/yslow.html#757

Home | Grade | Components | Statistics | Rulesets YSlow(V2) | Edit | Help |

**Grade C** Overall performance score 74 Ruleset applied: YSlow(V2) URL: <http://jsfiddle.net/spmt5/n6rl5r3s/2/>

**All (23) FILTER BY: CONTENT (6) COOKIE (2) CSS (6) IMAGES (2) JAVASCRIPT (4) SERVER (6)**

**F Make fewer HTTP requests**

**Grade F on Make fewer HTTP requests**

This page has 23 external Javascript scripts. Try combining them into one.  
 This page has 6 external stylesheets. Try combining them into one.

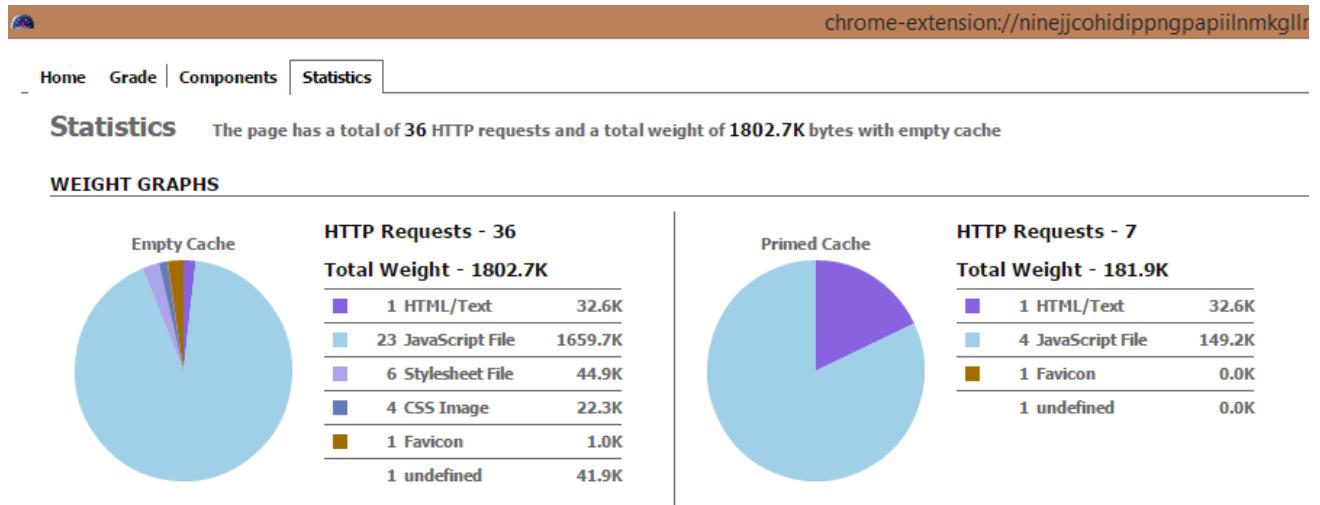
Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.

[»Read More](#)

Copyright © 2015 Yahoo! Inc. All rights reserved.

## Screen Shot: 15

## Testing a web page for updating a recipe



Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 1

The screenshot displays a table of 36 components, each with its type, size, GZIP size, cookie received, cookie sent, headers, URL, expiration date, response time, etag, and action.

TYPE	SIZE (KB)	GZIP (KB)	COOKIE RECEIVED (bytes)	COOKIE SENT (bytes)	HEADERS	URL	EXPIRES (Y/M/D)	RESPONSE TIME (ms)	ETAG	ACTION
doc (1)	32.6K									
js (23)	1684.5K									
css (6)	44.9K									
css	0.6K	0.6K		665	🔗	http://jsfiddle.net/css/normalize.css?...	2015/5/3	0		
css	0.2K	0.2K		665	🔗	http://jsfiddle.net/css/tricks.css?...	2015/5/3	0		
css	18.7K	18.7K		665	🔗	http://jsfiddle.net/css/elusive-webfont.css?...	2015/5/3	0		
css	8.0K	8.0K		665	🔗	http://jsfiddle.net/js/codemirror/lib/codemirror.css?...	2015/5/3	0		
css	17.2K	17.2K		665	🔗	http://jsfiddle.net/css/screen.css?...	2015/5/3	0		
css	0.0K			665	🔗	http://jsfiddle.net/css/light.css?...	2015/5/3	0	"5120150e-0"	
cssimage (4)	22.3K									
cssimage	20.8K			665	🔗	http://jsfiddle.net/img/initializing.png	2015/5/3	0	"5128cfbb-515f"	
cssimage	1.1K			665	🔗	http://jsfiddle.net/img/logo.png	2015/5/3	0	"5120150e-488"	
cssimage	0.2K			665	🔗	http://jsfiddle.net/img/handle-h.png	2015/5/3	0	"5120150e-ca"	
cssimage	0.1K			665	🔗	http://jsfiddle.net/img/handle-v.png	2015/5/3	0	"5120150e-c4"	
favicon (1)	1.0K									
favicon	1.0K			665	🔗	http://jsfiddle.net/favicon.png	no expires	0	"5120150e-444"	
font (1)	41.9K									
font	41.9K			665	🔗	http://jsfiddle.net/font/Elusive-Icons.eot?...	no expires	0	"51af0907-a404"	

\* type column indicates the component is loaded after window onload event  
† denotes 1x1 pixels image that may be image beacon

Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 2

chrome-extension://ninejcohidippngpapiilmkgillm/yslow.html#757

Home Grade Components Statistics

**Grade C** Overall performance score 74 Ruleset applied: YSlow(V2) URL: http://jsfiddle.net/la6h6/9ywq2u2n/1/

ALL (23) FILTER BY: [CONTENT \(6\)](#) | [COOKIE \(2\)](#) | [CSS \(6\)](#) | [IMAGES \(2\)](#) | [JAVASCRIPT \(4\)](#) | [SERVER \(6\)](#)

**F Make fewer HTTP requests**

This page has 23 external Javascript scripts. Try combining them into one.

This page has 6 external stylesheets. Try combining them into one.

Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.

[»Read More](#)

Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 3

## Testing a web page for getting the favorite recipe

chrome-extension://ninejcohidippngpapiilmkgillm

Home Grade Components **Statistics**

**Statistics** The page has a total of 36 HTTP requests and a total weight of 1801.3K bytes with empty cache

**WEIGHT GRAPHS**

**Empty Cache**

Type	Weight
1 HTML/Text	31.2K
23 JavaScript File	1659.7K
6 Stylesheet File	44.9K
4 CSS Image	22.3K
1 Favicon	1.0K
1 undefined	41.9K

**HTTP Requests - 36**

**Total Weight - 1801.3K**

Type	Weight
1 HTML/Text	31.2K
23 JavaScript File	1659.7K
6 Stylesheet File	44.9K
4 CSS Image	22.3K
1 Favicon	1.0K
1 undefined	41.9K

**Primed Cache**

Type	Weight
1 HTML/Text	31.2K
4 JavaScript File	149.2K
1 Favicon	0.0K
1 undefined	41.9K

**HTTP Requests - 7**

**Total Weight - 180.4K**

Type	Weight
1 HTML/Text	31.2K
4 JavaScript File	149.2K
1 Favicon	0.0K
1 undefined	0.0K

Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 1

chrome-extension://ninejjcohidippngpapiilmkgllmakh/yslow.html#1008

Home Grade Components Statistics Rulesets YSlow(V2) Edit Help »Expand All

**Components** The page has a total of 36 components and a total weight of 1801.3K 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)	31.2K									
js (23)	1684.5K									
css (6)	44.9K									
css	0.6K	0.6K		667	♂	<a href="http://jsfiddle.net/css/normalize.css?_m">http://jsfiddle.net/css/normalize.css?_m</a>	2015/5/4	0		
css	0.2K	0.2K		667	♂	<a href="http://jsfiddle.net/css/tricks.css?_m">http://jsfiddle.net/css/tricks.css?_m</a>	2015/5/4	0		
css	18.7K	18.7K		667	♂	<a href="http://jsfiddle.net/css/elusive-webfont.css?_m">http://jsfiddle.net/css/elusive-webfont.css?_m</a>	2015/5/4	0		
css	8.0K	8.0K		667	♂	<a href="http://jsfiddle.net/js/codemirror/lib/codemirror.css?_m">http://jsfiddle.net/js/codemirror/lib/codemirror.css?_m</a>	2015/5/4	0		
css	17.2K	17.2K		667	♂	<a href="http://jsfiddle.net/css/screen.css?_m">http://jsfiddle.net/css/screen.css?_m</a>	2015/5/4	0		
css	0.0K			667	♂	<a href="http://jsfiddle.net/css/light.css?_m">http://jsfiddle.net/css/light.css?_m</a>	2015/5/4	0	"5120150e-0"	
cssimage (4)	22.3K									
cssimage	20.8K			667	♂	<a href="http://jsfiddle.net/img/initializing.png">http://jsfiddle.net/img/initializing.png</a>	2015/5/4	0	"5128cfbb-515f"	
cssimage	1.1K			667	♂	<a href="http://jsfiddle.net/img/logo.png">http://jsfiddle.net/img/logo.png</a>	2015/5/4	0	"5120150e-488"	
cssimage	0.2K			667	♂	<a href="http://jsfiddle.net/img/handle-h.png">http://jsfiddle.net/img/handle-h.png</a>	2015/5/4	0	"5120150e-ca"	
cssimage	0.1K			667	♂	<a href="http://jsfiddle.net/img/handle-v.png">http://jsfiddle.net/img/handle-v.png</a>	2015/5/4	0	"5120150e-c4"	
favicon (1)	1.0K									
favicon	1.0K			667	♂	<a href="http://jsfiddle.net/favicon.png">http://jsfiddle.net/favicon.png</a>	no expires	0	"5120150e-444"	
font (1)	41.9K			667	♂	<a href="http://jsfiddle.net/font/Elusive-Icons.eot?_m">http://jsfiddle.net/font/Elusive-Icons.eot?_m</a>	no expires	0	"51af0907-a404"	

\* type column indicates the component is loaded after window onload event  
† denotes 1x1 pixels image that may be image beacon

Copyright © 2015 Yahoo! Inc. All rights reserved.

### Screen Shot: 2

chrome-extension://ninejjcohidippngpapiilmkgllmakh/yslow.html#1008

Home Grade Components Statistics Rulesets YSlow(V2) Edit Help

**Grade C** Overall performance score 74 Ruleset applied: YSlow(V2) URL: <http://jsfiddle.net/la6sh6/qlyn9naw/11/>

ALL (23) FILTER BY: [CONTENT \(6\)](#) | [COOKIE \(2\)](#) | [CSS \(6\)](#) | [IMAGES \(2\)](#) | [JAVASCRIPT \(4\)](#) | [SERVER \(6\)](#)

[Tweet](#) [Share](#)

**F Make fewer HTTP requests**

**F Use a Content Delivery Network (CDN)**

**A Avoid empty src or href**

**F Add Expires headers**

**B Compress components with gzip**

**A Put CSS at top**

**F Put JavaScript at bottom**

**A Avoid CSS expressions**

**N/A Make JavaScript and CSS external**

**A Reduce DNS lookups**

**F Minify JavaScript and CSS**

**A Avoid URL redirects**

**A Remove duplicate JavaScript and CSS**

**A Configure entity tags (ETags)**

**A Make AJAX cacheable**

**A Use GET for AJAX requests**

**A Reduce the number of DOM elements**

**A Avoid HTTP 404 (Not Found) error**

**A Reduce cookie size**

**F Use cookie-free domains**

**A Avoid AlphaImageLoader filter**

**A Do not scale images in HTML**

**A Make favicon small and cacheable**

Grade F on Make fewer HTTP requests

This page has 23 external Javascript scripts. Try combining them into one.  
This page has 6 external stylesheets. Try combining them into one.

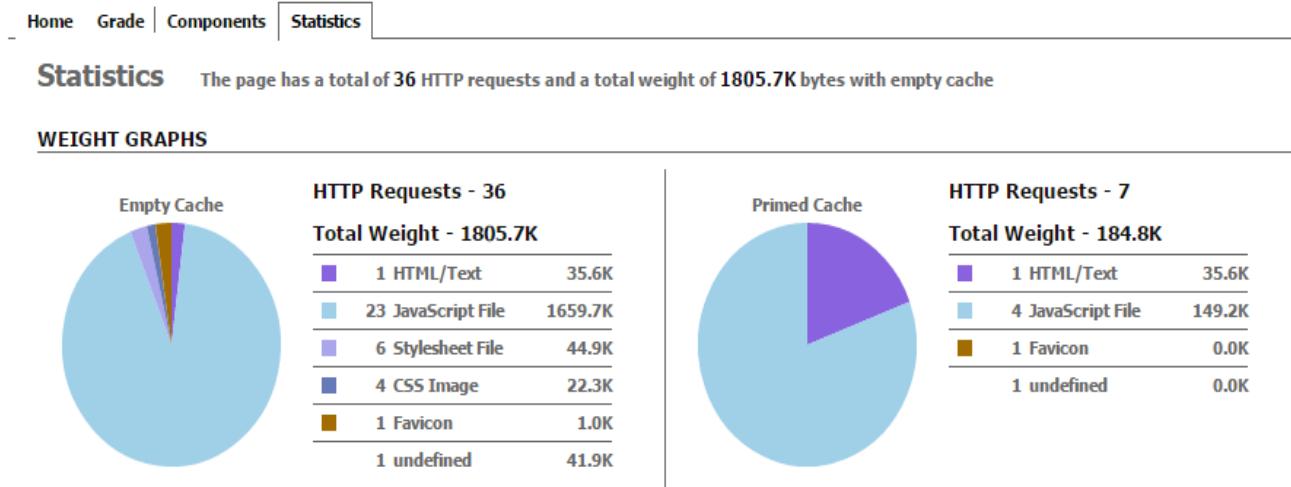
Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.

[»Read More](#)

Copyright © 2015 Yahoo! Inc. All rights reserved.

### Screen Shot: 3

## Testing a web page for import and updating a recipe



Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 1

chrome-extension://ninejjcohidipppngpapiinmkglmakh/yslow.html#1008

Rulesets YSlow(V2) Edit Help

**Components** The page has a total of 36 components and a total weight of 1805.7K bytes [»Expand All](#)

TYPE	SIZE (KB)	GZIP (KB)	COOKIE RECEIVED (bytes)	COOKIE SENT (bytes)	HEADERS	URL	EXPIRES (Y/H/D)	RESPONSE TIME (ms)	ETAG	ACTION
[+] doc (1)	35.6K									
[+] js (23)	1684.5K									
[+] css (6)	44.9K									
css	0.6K	0.6K		670	⌚	<a href="http://jsfiddle.net/css/normalize.css?...">http://jsfiddle.net/css/normalize.css?...</a>	2015/5/4	0		
css	0.2K	0.2K		670	⌚	<a href="http://jsfiddle.net/css/tricks.css?...">http://jsfiddle.net/css/tricks.css?...</a>	2015/5/4	0		
css	18.7K	18.7K		670	⌚	<a href="http://jsfiddle.net/css/elusive-webfont.css?...">http://jsfiddle.net/css/elusive-webfont.css?...</a>	2015/5/4	0		
css	8.0K	8.0K		670	⌚	<a href="http://jsfiddle.net/js/codemirror/lib/codemirror.css?...">http://jsfiddle.net/js/codemirror/lib/codemirror.css?...</a>	2015/5/4	0		
css	0.0K			670	⌚	<a href="http://jsfiddle.net/css/light.css?...">http://jsfiddle.net/css/light.css?...</a>	2015/5/4	0 "5120150e-0"		
css	17.2K	17.2K		670	⌚	<a href="http://jsfiddle.net/css/screen.css?...">http://jsfiddle.net/css/screen.css?...</a>	2015/5/4	0		
[+] cssimage (4)	22.3K									
cssimage	20.8K			670	⌚	<a href="http://jsfiddle.net/img/initializing.png">http://jsfiddle.net/img/initializing.png</a>	2015/5/4	0 "5128cfbb-515f"		
cssimage	1.1K			670	⌚	<a href="http://jsfiddle.net/img/logo.png">http://jsfiddle.net/img/logo.png</a>	2015/5/4	0 "5120150e-488"		
cssimage	0.2K			670	⌚	<a href="http://jsfiddle.net/img/handle-h.png">http://jsfiddle.net/img/handle-h.png</a>	2015/5/4	0 "5120150e-ca"		
cssimage	0.1K			670	⌚	<a href="http://jsfiddle.net/img/handle-v.png">http://jsfiddle.net/img/handle-v.png</a>	2015/5/4	0 "5120150e-c4"		
[+] favicon (1)	1.0K									
favicon	1.0K			670	⌚	<a href="http://jsfiddle.net/favicon.png">http://jsfiddle.net/favicon.png</a>	no expires	0 "5120150e-444"		
[+] font (1)	41.9K									

\* type column indicates the component is loaded after window.onload event  
† denotes 1x1 pixels image that may be image beacon

Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 2

The screenshot shows the YSlow extension interface. At the top, it displays the URL `chrome-extension://ninejjcohidippngpapiilmkgilmakh/yslow.html#1008`. Below this, there are tabs for Home, Grade, Components, and Statistics. The Grade section shows a score of 74 with a 'Grade C'. It includes a 'FILTER BY' dropdown set to 'CONTENT (6)' and a link to 'Edit' rulesets. The main content area is titled 'Grade F on Make fewer HTTP requests' and lists various optimization suggestions. Some items are marked with a red border, indicating they are applicable or recommended. The suggestions include using a Content Delivery Network (CDN), avoiding empty src or href, adding expires headers, compressing components with gzip, putting CSS at the top, putting JavaScript at the bottom, avoiding CSS expressions, and many others related to minimizing HTTP requests and file sizes.

Screen Shot: 3

### Fire Bug tool:

We used the fire bug tool to do the performance testing for html pages we used. The firebug tool will automatically show us if there are any errors in the JavaScript, CSS or HTML code. It also notify us if there are any errors in request URL's or XHR requests. We have done this testing on all of the HTML pages we created in this increment. Mostly there are no errors detected while we performed this testing.

#### 1. Createrecipe.html

This screenshot shows a browser window with multiple tabs open. The active tab is titled 'Create Recipe' and contains a form for creating a new recipe. The form fields include 'Create New Recipe' (with options like 'veg fried rice', 'boiled rice', '1', and 'bowl'), and a table for ingredients ('carrots', '1', 'cup'), ('onions', '1', 'cup'), and ('spices', '1', 'spoon'). A modal dialog box is displayed in the center, stating 'recipe created!' with an 'OK' button. At the bottom of the browser window, a Firebug toolbar is visible, showing tabs for Console, HTML, CSS, Script, DOM, Net, and Cookies. The 'Console' tab is selected, and the status bar at the bottom right shows the date and time as 4/13/2015 10:54 PM.



Share Recipe    Create Recipe    DirectionAndWeat...    DirectionAndWeat...    Connecting...    file:///C...ories.html    Title of the document...

file:///C:/Users/VINAYA/Desktop/createrecipe.html

Create New Recipe

veg fried rice  
boiled rice  
1  
bowl

recipe created!!  
 Prevent this page from creating additional dialogs

OK

Add More

carrots	1	cup
onions	1	cup
spices	1	spoon

Save

Net

Console HTML CSS Script DOM Net Cookies

Search within Net panel

Clear Persist All HTML CSS JavaScript XHR Images Plugins Media Fonts

URL Status Domain Size Remote IP Timeline

Net panel activated. Any requests while the net panel is inactive are not shown.

⊕ GET AuthenticationService.Au..._c...	200 OK	maps.googleapis.com	57 B	216.58.218.202:443	51ms
⊕ GET bowl\$cup\$cup\$spoon?callb...	200 OK	kc-sce-cs551-2.kc.umkc.edu	136 B	134.193.136.128:80	151ms
⊕ GET bowl\$cup\$cup\$spoon?callb...	200 OK	kc-sce-cs551-2.kc.umkc.edu	136 B	134.193.136.128:80	142ms
⊕ GET bowl\$cup\$cup\$spoon?callb...	200 OK	kc-sce-cs551-2.kc.umkc.edu	136 B	134.193.136.128:80	58ms
4 requests					465 B      402ms (onload: 482ms)



## 2. Share Recipe.html

Share Recipe    Create Recipe    DirectionAndWeat...    DirectionAndWeat...    Connecting...    file:///C...ories.html    Title of the document...

file:///C:/Users/VINAYA/Desktop/sharerecipe.html

Share Favorite Recipe

sweet

Get Recipe

recipename : sweet

ingredients : jaggery sugar water

quantity : 0 50 1

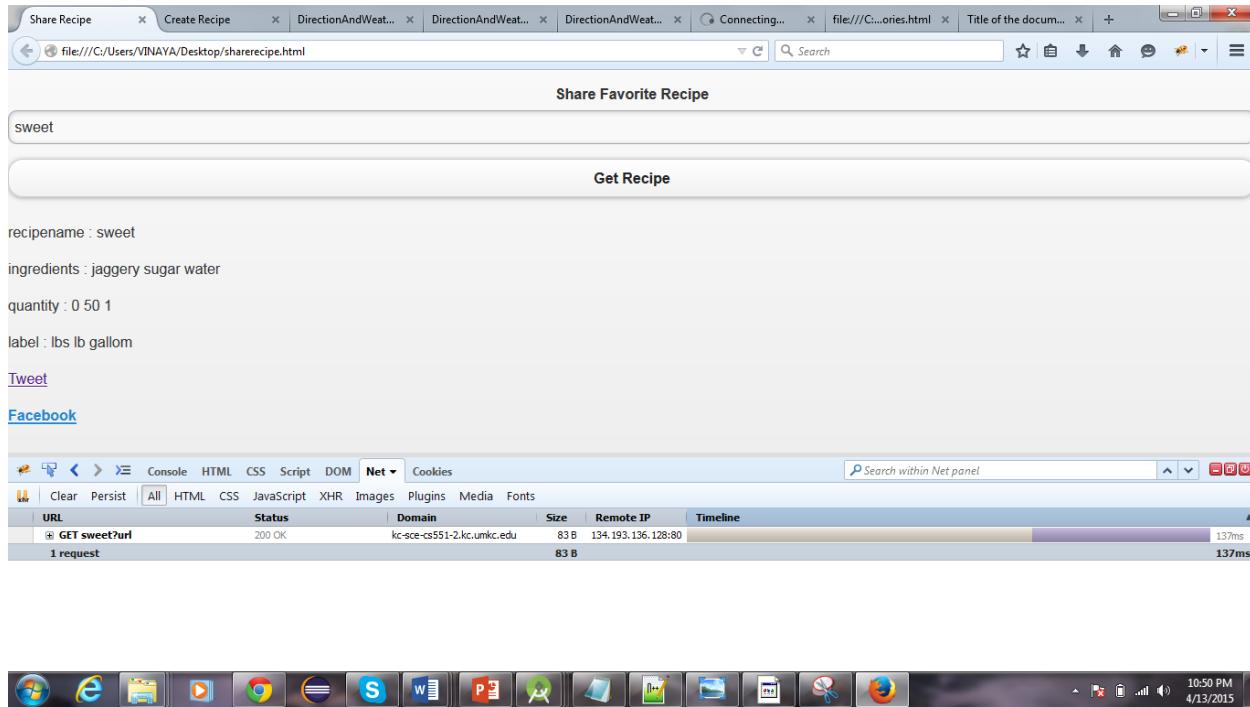
label : lbs lb gallom

Console

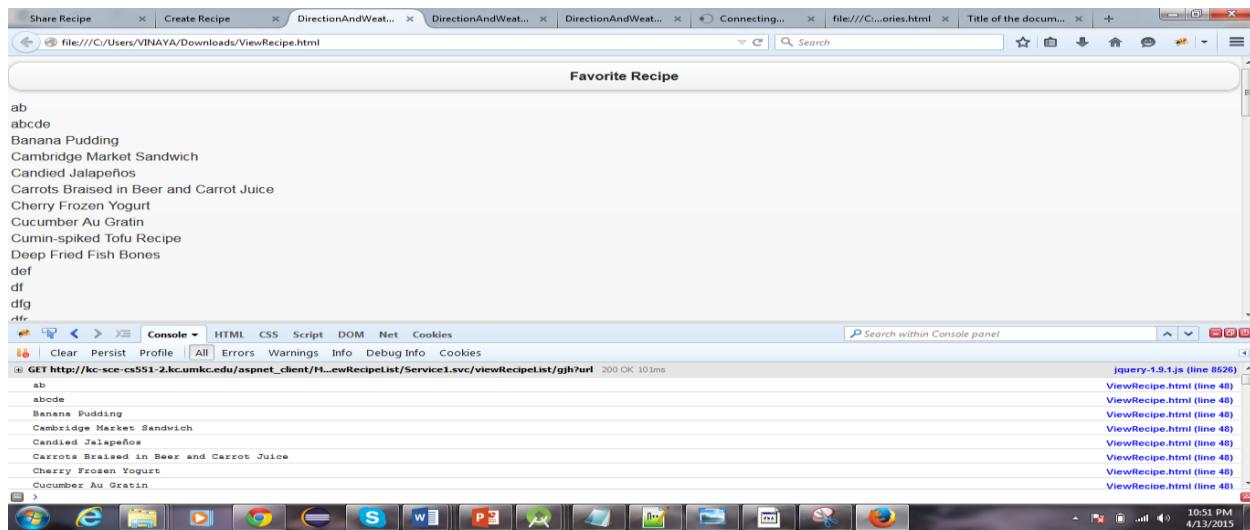
Search within Console panel

```
GET http://kc-sce-cs551-2.kc.umkc.edu/aspnet_client/M...maya2/displayrecipe/Service1.svc/data/sweet?url 200 OK 137ms
Object { ingredients:"jaggery:sugar:water", label:"lbs:lb:gallom", quantity:"0:50:1" }
"jaggery:sugar:water"
" jaggery sugar water"
" 0 50 1"
" lbs lb gallom"
```





### 3. viewRecipe.html



Share Recipe | Create Recipe | DirectionAndWeat... | DirectionAndWeat... | Connecting... | file:///C:/...ries.html | Title of the document | +

file:///C:/Users/VINAYA/Downloads/ViewRecipe.html

Search

### Favorite Recipe

ab  
abcde  
Banana Pudding  
Cambridge Market Sandwich  
Candied Jalapeños  
Carrots Braised in Beer and Carrot Juice  
Cherry Frozen Yogurt  
Cucumber Au Gratin  
Cumin-spiked Tofu Recipe  
Deep Fried Fish Bones  
def  
df  
dfg  
dfr

Net panel activated. Any requests while the net panel is inactive are not shown.

URL	Status	Domain	Size	Remote IP	Timeline
GET gfh7url	200 OK	kc-sce-cs551-2.kc.umkc.edu	2.3 KB	134.193.136.128:80	101ms
GET AuthenticationService.Au...c_	200 OK	maps.googleapis.com	57 B	216.58.218.202:443	49ms

2 requests 2.3 KB 150ms (onload: 442ms)



#### 4. Updaterecipe.html

Share Recipe | Create Recipe | DirectionAndWeat... | DirectionAndWeat... | Connecting... | file:///C:/...ries.html | Title of the document | +

file:///C:/Users/VINAYA/Downloads/UpdateRecipe.html

Search

veg fried rice

recipename : veg  
 ingredients : boiled,rice,carrots,spices  
 quantity : 1,1,1  
 label : bowl,cup,spoon

veg fried rice

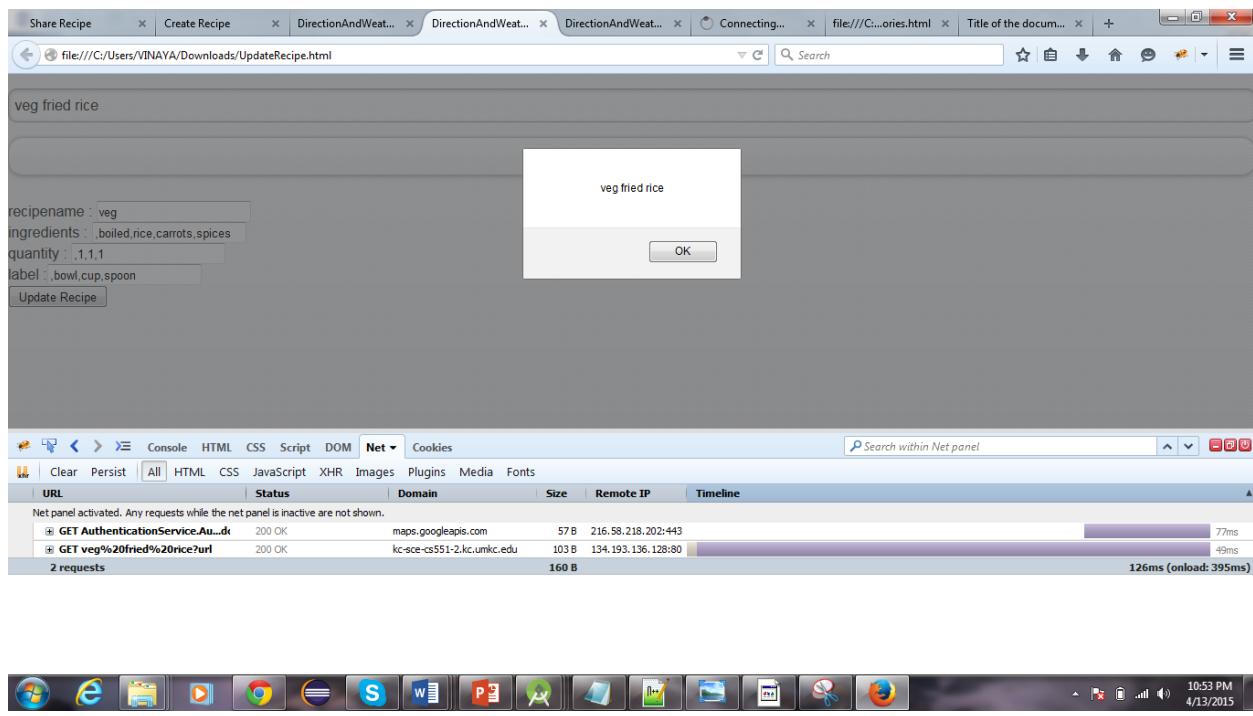
Console | HTML | CSS | Script | DOM | Net | Cookies

Search within Console panel

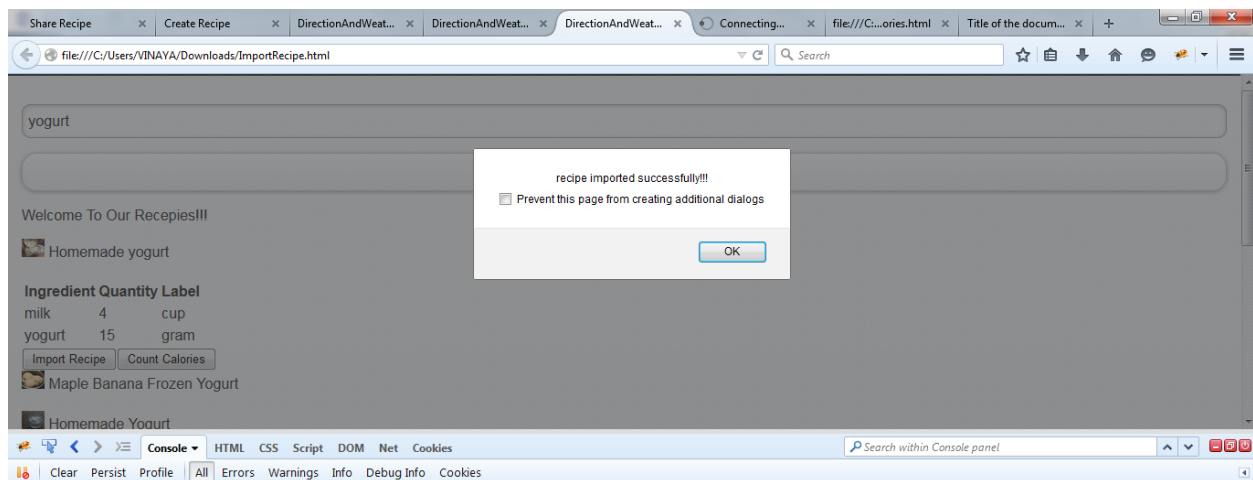
GET http://kc-sce-cs551-2.kc.umkc.edu/aspnet\_client/M...yrecipe/Service1.svc/data/veg%20fried%20rice?url 200 OK 49ms

jquery-1.9.1.js (line 8526)





## 5. Import Recipe.html



Share Recipe | Create Recipe | DirectionAndWeather - jsFiddle... | DirectionAndWeather - jsFiddle... | DirectionAndWeather - jsFiddle... | +

file:///C:/Users/VINAYA/Downloads/ImportRecipe.html

yogurt

**Search Recipe**

Welcome To Our Receiptes!!!

Homemade yogurt

**Ingredient Quantity Label**

milk	4	cup
yogurt	15	gram

**Import Recipe** | **Count Calories**

Net | Console | HTML | CSS | Script | DOM | Net | Cookies

Search within Net panel

Clear Persist All HTML CSS JavaScript XHR Images Plugins Media Fonts

18 requests 214.1 KB 10:42 PM 4/13/2015

- ④ GET /apple&app\_id=... 200 OK api.edamam.com 20.2 KB 107.20.173.119:443 825ms
- ④ GET AuthenticationService.Au...dc 200 OK maps.googleapis.com 57 B 216.58.218.170:443 45ms
- ④ GET 8bd76a8582aa638378f5fc24 200 OK edamam.com 24.0 KB 107.20.173.119:443
- ④ GET e507a640dcf013a573d2d87cd 200 OK edamam.com 10.7 KB 107.20.173.119:443
- ④ GET c75a65078289acf1c74057923 200 OK edamam.com 0 B 107.20.173.119:443
- ④ GET 31d8f5d3977560d2209be6ca 200 OK edamam.com 13.0 KB 107.20.173.119:443
- ④ GET feff415120db72c77290dc062e 200 OK edamam.com 9.5 KB 107.20.173.119:443
- ④ GET 47c75b8e933d0337fd5f84315 200 OK edamam.com 18.2 KB 107.20.173.119:443
- ④ GET 9811c168e13ab9da0e66716 200 OK edamam.com 7.4 KB 107.20.173.119:443
- ④ GET 50319d4eccfd12fa10fb0248 200 OK edamam.com 12.1 KB 107.20.173.119:443
- ④ GET ea7d9d38b512c3de759dff5f 200 OK edamam.com 9.1 KB 107.20.173.119:443
- ④ GET 0775818ccdb07cc522d0b0b 200 OK edamam.com 13.5 KB 107.20.173.119:443
- ④ GET search?q=yogurt&app\_id=... 200 OK api.edamam.com 18.2 KB 107.20.173.119:443 2.22s
- ④ GET 13f3d6891c17b3ea39398f75 200 OK edamam.com 64.0 KB 107.20.173.119:443 1.41s

Windows Taskbar icons: Internet Explorer, Google Chrome, Mozilla Firefox, Microsoft Edge, File Explorer, Paint, Snipping Tool, Task View, Task Manager, Task Scheduler, Task History, Task List, Task View, Task Manager, Task History, Task List.

## 6. Calorie.html

Share Recipe | Create Recipe | DirectionAndWeather - ... | DirectionAndWeather - ... | Connecting... | file:///C:/U...alories.html | +

file:///C:/Users/VINAYA/Downloads/calories.html

Strawberry

Strawberry Shake

Total Calories: 692

Ingredient	Quantity	Calories
Ice Creams, Strawberry	1pint	353 Kcal
Strawberries, Raw	4oz	400 Kcal
Milk, Whole, 3.25% Milkfat, With Added Vitamin D	1cup	84 Kcal

Pink Grapefruit, Strawberry, and Champagne Granita with Sugared Strawberries

Net | Console | HTML | CSS | Script | DOM | Net | Cookies

Search within Net panel

Clear Persist All HTML CSS JavaScript XHR Images Plugins Media Fonts

18 requests 214.1 KB 10:45 PM 4/13/2015

- ④ GET 087c32b6f70b9016e8ce568 200 OK edamam.com 10.4 KB 107.20.173.119:443 431ms
- ④ GET c00d07f5b898cd446ec60e10 200 OK edamam.com 14.3 KB 107.20.173.119:443 528ms
- ④ GET 2495c263291a4d48ec13afc1 200 OK edamam.com 13.0 KB 107.20.173.119:443
- ④ GET 3057774>22a554674e6bb3b 200 OK edamam.com 18.2 KB 107.20.173.119:443
- ④ GET a475e9c073daace44dc51cb2 200 OK edamam.com 14.4 KB 107.20.173.119:443
- ④ GET b7dc59bf463c15e4427fd9c 200 OK edamam.com 19.6 KB 107.20.173.119:443
- ④ GET 79d3947d96eb0493b73aa 200 OK edamam.com 20.7 KB 107.20.173.119:443 738ms
- ④ GET fformat=json&q=strawber 200 OK api.nal.usda.gov 246 B 54.165.187.228:80 840ms
- ④ GET ?format=json&q=strawber 200 OK api.nal.usda.gov 217 B 54.165.187.228:80
- ④ GET ?format=json&q=milk&sort 200 OK api.nal.usda.gov 209 B 54.165.187.228:80 390ms
- ④ GET ?ndbno=18547&type=b&for 200 OK api.nal.usda.gov 1.1 KB 54.165.187.228:80
- ④ GET ?ndbno=03681&type=b&for 200 OK api.nal.usda.gov 1.5 KB 54.165.187.228:80 29.2ms
- ④ GET ?ndbno=35001&type=b&for 200 OK api.nal.usda.gov 955 B 54.165.187.228:80 166ms

Windows Taskbar icons: Internet Explorer, Google Chrome, Mozilla Firefox, Microsoft Edge, File Explorer, Paint, Snipping Tool, Task View, Task Manager, Task History, Task List, Task View, Task Manager, Task History, Task List.

Screenshot of a web browser showing a nutrition facts page for a strawberry shake.

The page displays the following information:

- Strawberry Shake**
- Total Calories: 692**
- Ingredient** (Ice Creams, Strawberry; Strawberries, Raw; Milk, Whole, 3.25% Milkfat, With Added Vitamin D)
- Quantity** (1pint, 4oz, 1cup)
- Calories** (353 Kcal, 400 Kcal, 84 Kcal)

The browser's developer tools (Console tab) show several requests to the USDA API:

```

GET http://api.nal.usda.gov/usda/ndb/search/?format=j...key=DgxqZYmXG8YsyCIEfLkJaAklH4ybQTEHORaKedx&url 200 OK 399ms jquery....min.js (line 4)
GET http://api.nal.usda.gov/usda/ndb/search/?format=j...key=DgxqZYmXG8YsyCIEfLkJaAklH4ybQTEHORaKedx&url 200 OK 391ms jquery....min.js (line 4)
GET http://api.nal.usda.gov/usda/ndb/search/?format=j...key=DgxqZYmXG8YsyCIEfLkJaAklH4ybQTEHORaKedx&url 200 OK 390ms jquery....min.js (line 4)
GET http://api.nal.usda.gov/usda/ndb/reports/?ndbno=1...key=DgxqZYmXG8YsyCIEfLkJaAklH4ybQTEHORaKedx&url 200 OK 167ms jquery....min.js (line 4)
GET http://api.nal.usda.gov/usda/ndb/reports/?ndbno=0...key=DgxqZYmXG8YsyCIEfLkJaAklH4ybQTEHORaKedx&url 200 OK 292ms jquery....min.js (line 4)
GET http://api.nal.usda.gov/usda/ndb/reports/?ndbno=3...key=DgxqZYmXG8YsyCIEfLkJaAklH4ybQTEHORaKedx&url 200 OK 166ms jquery....min.js (line 4)

```



## 7. Notification.html

Screenshot of a web browser showing a notification page.

The page contains a single button labeled "Show notifications".

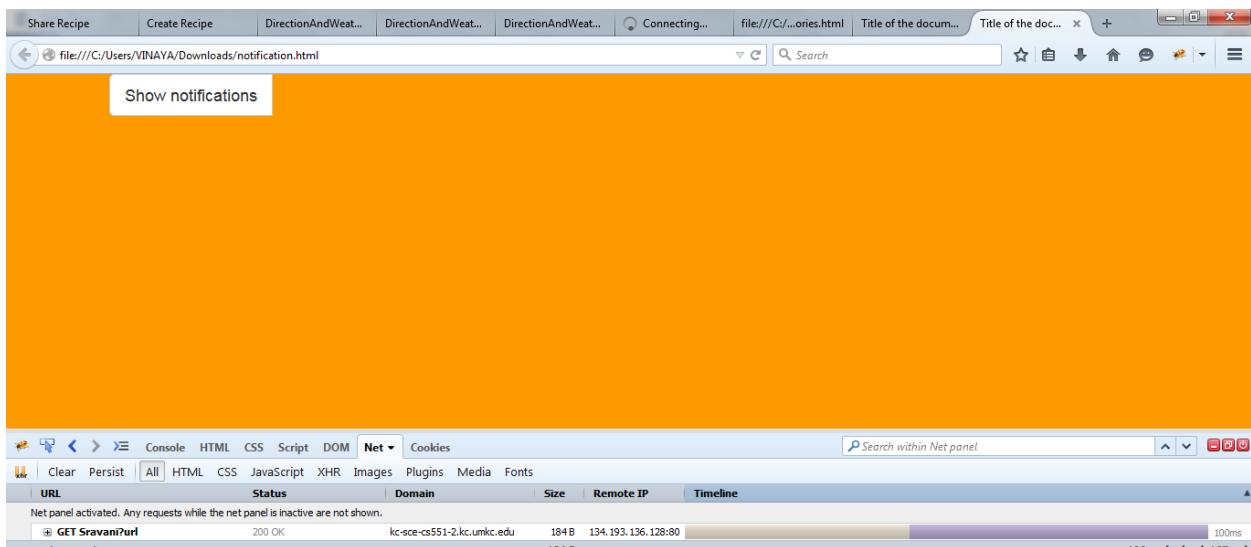
The browser's developer tools (Console tab) show the following logs:

```

GET http://kc-sce-cs551-2.kcumkc.edu/aspnet_client/M...viewGrocery/Service1.svc/JsonGrocery/Sravani?uri 200 OK 100ms jquery....min.js (line 5)
[{"g_name": "cheese", "g_quantity": 50, "g_measure": "gm", "Column1": "2015-04-18"}, {"g_name": "beetroot", "g_quantity": 1, "g_measure": "kg", "Column1": "2015-04-18"}]
2015-04-18
x value is 2015-04-18
2015-04-18
x value is 2015-04-18
notification.html (line 39)
notification.html (line 45)
notification.html (line 47)
notification.html (line 45)
notification.html (line 47)

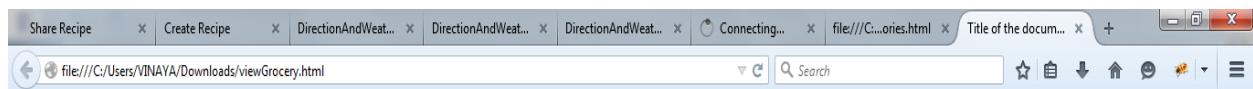
```





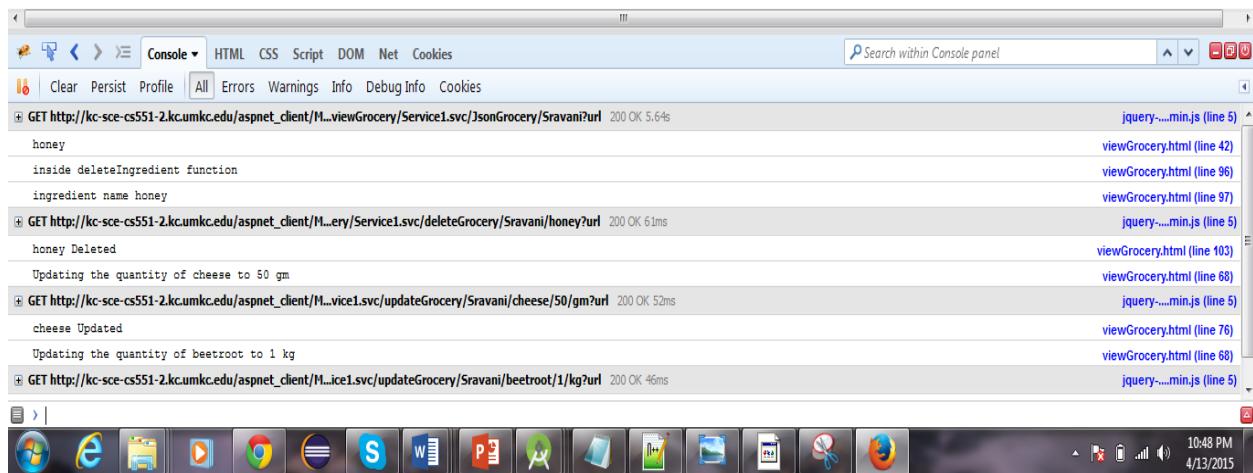
## 8. viewGrocery.html

The screenshot shows a web browser window displaying a grocery list application. The page has a header with "View Grocery List" and "Edit" buttons. Below the header is a table with columns "Name", "Quantity", and "Units". Two items are listed: "cheese" (50 gm) and "beetroot" (1 kg). The table has a light gray background and white text. The Firebug Net panel is open at the bottom, showing network activity. The timeline tab is selected, displaying multiple requests. Some requests are from ajax.googleapis.com (e.g., "GET jquery-uicss", "GET jquery-ui.min.js") and others are from the local host (e.g., "GET Sravani?url", "GET deletebtn.jpg"). The total time for all requests is 6.16s, and the onload time is 1.27s. The system tray on the right shows the date and time as 10:48 PM on 4/13/2015.



[View Grocery List](#)    [Edit](#)

Name	Quantity	Units
cheese	50	gm
beetroot	1	kg



Screen Shots for shopping list generation:

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

Home | Grade | Components | Statistics | Rulesets | YSlow(V2) | Edit |

**Grade C** Overall performance score 79 Ruleset applied: YSlow(V2) URL: http://jsfiddle.net/spt5Lpvchjdo/3/

ALL (23) FILTER BY: [CONTENT \(6\)](#) | [COOKIE \(2\)](#) | [CSS \(6\)](#) | [IMAGES \(2\)](#) | [JAVASCRIPT \(4\)](#) | [SERVER \(6\)](#)

[Twitter](#) [Facebook](#)

**F Make fewer HTTP requests**

This page has 23 external Javascript scripts. Try combining them into one.  
This page has 6 external stylesheets. Try combining them into one.

Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.

[Read More](#)

Copyright © 2015 Yahoo! Inc. All rights reserved.

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

Home | Grade | Components | **Statistics** | Rulesets | YSlow(V2) | Edit |

**Statistics** The page has a total of 36 HTTP requests and a total weight of 1790.5K bytes with empty cache

**WEIGHT GRAPHS**

**Empty Cache**

HTTP Requests - 36		
<b>Total Weight - 1790.5K</b>		
1	HTML/Text	35.2K
23	JavaScript File	1644.9K
6	Stylesheet File	44.9K
4	CSS Image	22.3K
1	Favicon	1.0K
1	undefined	41.9K

**Primed Cache**

HTTP Requests - 7		
<b>Total Weight - 35.2K</b>		
1	HTML/Text	35.2K
4	JavaScript File	0.0K
1	Favicon	0.0K
1	undefined	0.0K

Copyright © 2015 Yahoo! Inc. All rights reserved.

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

Home | Grade | **Components** | Statistics | Rulesets | YSlow(V2) | Edit |

**Components** The page has a total of 36 components and a total weight of 1790.5K bytes

[Expand All](#)

TYPE	SIZE (KB)	GZIP (KB)	COOKIE RECEIVED (bytes)	COOKIE SENT (bytes)	HEADERS	URL	EXPIRES (Y/M/D)	RESPONSE TIME (ms)	ETAG	ACTION
doc (1)	35.2K									
js (23)	2087.9K									
css (6)	44.9K									
cssimage (4)	22.3K									
favicon (1)	1.0K									
font (1)	41.9K									

\* type column indicates the component is loaded after window.onload event  
† denotes 1x1 pixels image that may be image beacon

Screen Shots for Import recipe

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

Home Grade Component Statistics

Rulesets YSlow(V2) Edit

**Grade C** Overall performance score 79 Ruleset applied: YSlow(V2) URL: http://jsfiddle.net/spn5/y12vq7fp/19/

ALL (23) FILTER BY: CONTENT (6) COOKIE (2) CSS (6) IMAGES (2) JAVASCRIPT (4) SERVER (6)

**F Make fewer HTTP requests**

Grade F on Make fewer HTTP requests

A Use a Content Delivery Network (CDN)

A Avoid empty src or href

F Add Expires headers

B Compress components with gzip

A Put CSS at top

F Put JavaScript at bottom

A Avoid CSS expressions

N/A Make JavaScript and CSS external

A Reduce DNS lookups

F Minify JavaScript and CSS

A Avoid URL redirects

A Remove duplicate JavaScript and CSS

A Configure entity tags (ETags)

A Make AJAX cacheable

A Use GET for AJAX requests

B Reduce the number of DOM elements

A Avoid HTTP 404 (Not Found) error

A Reduce cookie size

F Use cookie-free domains

A Avoid AlphaImageLoader filter

A Do not scale images in HTML

A Make favicon small and cacheable

This page has 23 external Javascript scripts. Try combining them into one.  
This page has 6 external stylesheets. Try combining them into one.

Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.

[»Read More](#)

Copyright © 2015 Yahoo! Inc. All rights reserved.

chrome-extension://ninejjcohidippngpapiilnmkgllmakh/

Home Grade Components Statistics

**Statistics** The page has a total of 36 HTTP requests and a total weight of 1790.5K bytes with empty cache

**WEIGHT GRAPHS**

HTTP Requests - 36		HTTP Requests - 7	
Total Weight - 1790.5K		Total Weight - 35.2K	
1 HTML/Text	35.2K	1 HTML/Text	35.2K
23 JavaScript File	1644.9K	4 JavaScript File	0.0K
6 Stylesheet File	44.9K	1 Favicon	0.0K
4 CSS Image	22.3K	1 undefined	0.0K
1 Favicon	1.0K		
1 undefined	41.9K		

**Empty Cache**

**Primed Cache**

Copyright © 2015 Yahoo! Inc. All rights reserved.

## Task Planning

We planned to implement the project by using latest technology & tools. The project is divided into four iterations.

### Increment#1:

We have developed four layouts as part of UI implementation in increment 1.

1. Registration page
2. Login page
3. Welcome/Home page
4. Layout for viewing the google maps

If the user is a new user, then he has to register to use the app. After registering, the app redirects to the login page where the user can login to use the features of this app. When the user logs in successfully, he can perform the following operations:

- He can visualize his current location using google maps.
- He can search the list of nearby grocery stores.

### Increment#2:

As part of Increment#2, We have developed the following features of our application.

- Add groceries items that are available at home to a list.
- View grocery list
- Search for a recipe.

- See the ingredient list for that recipe.
- Import the ingredient list to the local system.

We developed a native android application for adding the list of groceries along with quantity. We have used web technologies for implementing the client side logic rest of the functionalities listed above. We are storing the data in SQL Server remote database. Hence we have created rest services to push/pull the data from the database.

#### **Increment#3:**

In Increment3 we have enhanced the features developed in increment#2 and provided certain new features:

- 1) Update the quantity of grocery items
- 2) Delete the grocery items from the list
- 3) Create a recipe and store the recipe in the database
- 4) Share the recipe through Facebook and twitter
- 5) Send notifications to users about the expiry date of recipes
- 6) Get calorie information for each ingredient of the recipe

#### **Increment#4:**

Following are the new features and enhancements provided in increment#4

- 1) Shopping list generation
- 2) Updating recipes imported/created
- 3) Fixing the issues with notifications
- 4) Integrating the code from all the increments
- 5) Integration testing

In brief each iteration consists of following tasks:

1. Designing graphical user interface.
2. Connecting to the database and implementing web services.
3. Testing the entire application and bug fixing.

#### **ScrumDo**

We have used Scrumdo to track the progress of the project. The whole process of development and tasks division has been completely displayed in the scrum Do tool and below is the link for the Scrum DO action of our project.

Link: <https://www.scrumdo.com/projects/project/calorie2grocery3/iteration/122933#>

## **User Stories for increment#1:**

#4 As a developer i need to develop a welcome/home page for my application. This page will be modified through the project as we add functionalities to the application	   	Done	Tasks   0 Comments	Story4	 sravani	3
#6 As a developer i need to establish a connection to the database so that i can store the user information Story6	   	Done	Tasks   0 Comments	Story6		5
#5 As a user i need to use google maps to get the near by grocery stores Story5	   	Done	Tasks   0 Comments	Story5	 sravani	8
#2 As a developer i need to develop a user interface for login so that users can sign in to my application Story2	   	Done	Tasks   0 Comments	Story2		2
#3 As a developer i need to develop a user interface for register page so that users can register for my application Story3	   	Done	Tasks   0 Comments	Story3		3
#19 As a developer, i need to validate the data entered in the database. TestStory2	   	Done	Tasks   0 Comments	TestStory2	 sravani	3
#18 As a developer i need to test the UI interface to verify the transitions are happening as expected from login page, register page and welcome page. TestStory1	   	Done	Tasks   0 Comments	TestStory1	 sravani	5
#20 As a developer i need to verify that the authentication logic is working as expected. TestStory3	   	Done	Tasks   0 Comments	TestStory3		2
#1 As a developer i need to install android studio so that i can develop a application Story1	   	Done	Tasks   0 Comments	Story1		3
#4 As a developer i need to develop a welcome/home page for my application. This page will be modified through the project as we add functionalities to the application	   	Done	Tasks   0 Comments	Story4	 sravani	3

## **User Stories for increment#2:**

#7 As a user i should be able to view the grocery list so that i can keep track of the groceries available at home	   
Story7 <a href="#">Done</a> <a href="#">Tasks</a>   <a href="#">0 Comments</a> <a href="#">Story7</a>	
#8 As a user i should be able to search for recipes based on some keywords.	   
Story8 <a href="#">Done</a> <a href="#">Tasks</a>   <a href="#">0 Comments</a> <a href="#">Story8</a>	
#22 As a user i should be able to add grocery items available at home to a list in the application	   
<a href="#">Done</a> <a href="#">Tasks</a>   <a href="#">0 Comments</a>  <a href="#">sravani</a>	
#25 As a user, i should be able to view the list of ingredients required for a selected recipe.	   
<a href="#">Done</a> <a href="#">Tasks</a>   <a href="#">0 Comments</a>  <a href="#">Story#25</a>  <a href="#">sravani</a>	
#21 As a system, i should be able to store the grocery information provided by the user in the database	   
<a href="#">Done</a> <a href="#">Tasks</a>   <a href="#">0 Comments</a>  <a href="#">sravani</a>	
#35 As a developer, i need to test the import recipe API to ensure that the recipe information is inserted appropriately.	   
<a href="#">Done</a> <a href="#">Tasks</a>   <a href="#">0 Comments</a>	
#34 As a developer, i need to test the API for adding groceries to the database to validate the data.	   
<a href="#">Done</a> <a href="#">Tasks</a>   <a href="#">0 Comments</a>	
#33 As a system, i should store the details like ingredients,quantity, preparation method etc for the recipes imported by the user in the database.	   
<a href="#">Done</a> <a href="#">Tasks</a>   <a href="#">0 Comments</a>	
#32 As a system, i should display the UI for viewing the grocery list	   
<a href="#">Done</a> <a href="#">Tasks</a>   <a href="#">0 Comments</a>  <a href="#">sravani</a>	
#31 As a system,I should display the UI for searching recipes.	   
<a href="#">Done</a> <a href="#">Tasks</a>   <a href="#">0 Comments</a>	
#30 As a system, i should display the UI for adding groceries to the database.	   
<a href="#">Done</a> <a href="#">Tasks</a>   <a href="#">0 Comments</a>  <a href="#">sravani</a>	
#29 As a system,i should be able to fetch the grocery items data from the database and display it to the user.	   
<a href="#">Done</a> <a href="#">Tasks</a>   <a href="#">0 Comments</a>  <a href="#">sravani</a>	
#26 As a user i should be able to import my favorite recipes for customization.	   
<a href="#">Done</a> <a href="#">Tasks</a>   <a href="#">0 Comments</a>	

### Userstories for increment#3:

#46 As a developer i need to test the APP performance using the tools available.	   
<a href="#">Done</a> <a href="#">Tasks</a>   0 Comments	
#45 As a developer i need to perform Nunit testing on newly built .Net rest services.	   
<a href="#">Done</a> <a href="#">Tasks</a>   0 Comments  srawani	
#44 As a developer i need to test that all UI transitions are happening as expected	   
<a href="#">Done</a> <a href="#">Tasks</a>   0 Comments	
#43 As a system i should store the login data in the remote server - Extension to increment 1 (Used SQLite in Inc1)	   
<a href="#">Done</a> <a href="#">Tasks</a>   0 Comments	
#42 As a user I should be able to login using facebook and transition to the app - Fix on increment 1	   
<a href="#">Done</a> <a href="#">Tasks</a>   0 Comments	
#40 As a user I should be able to get the calorie information of individual ingredients of the recipe	   
<a href="#">Done</a> <a href="#">Tasks</a>   0 Comments  srawani	
#39 As a user I should be able to get the calorie count of the complete recipe	   
<a href="#">Done</a> <a href="#">Tasks</a>   0 Comments  srawani	
#38 As a user I should be able to share my recipes using facebook/twitter	   
<a href="#">Done</a> <a href="#">Tasks</a>   0 Comments	
#37 As a user I should be able to create a recipe	   
<a href="#">Done</a> <a href="#">Tasks</a>   0 Comments	
#47 As a system i need to store the recipes created by the user in the remote database	   
<a href="#">Done</a> <a href="#">Tasks</a>   0 Comments	
#36 As a user I should be able to add ingredients to the recipe	   
<a href="#">Done</a> <a href="#">Tasks</a>   0 Comments	
#28 As a system, I should send notifications to the user about the expiry date of the grocery items.	   
<a href="#">Done</a> <a href="#">Tasks</a>   0 Comments	
#27 As a user I should be able to delete some of the entries in the grocery list	   
<a href="#">Done</a> <a href="#">Tasks</a>   0 Comments  srawani	
#12 As a user I should be able to remove ingredients from the recipe	   
<a href="#">Done</a> <a href="#">Tasks</a>   0 Comments	3
#9 As a user I should be able to update the quantity of the grocery items in my list	   
<a href="#">Done</a> <a href="#">Tasks</a>   0 Comments  srawani	3

### User stories for Increment#4:

#53 As a developer, i need to fix the issues identified during integration testing	
<b>Done</b> Tasks   0 Comments	
#52 As a developer, i need to enhance the functionality of update recipe	
<b>Done</b> Tasks   0 Comments	
#51 As a developer, i need to enhance the UI of the application	
<b>Done</b> Tasks   0 Comments	
#50 As a system, i need to send notifications about the expiry of the grocery items	
<b>Done</b> Tasks   0 Comments	
#49 As a developer, we need to enhance the functionality of import recipe	
<b>Done</b> Tasks   0 Comments	
#16 As a team, we need to integrate the code developed by all the team members.	3
<b>Done</b> Tasks   0 Comments	
#15 As a developer i need to develop the UI for getting the shopping list	3
<b>Done</b> Tasks   0 Comments	
#13 As a user i need a shopping list so that i wont miss something needed for preparing the recipe	3
<b>Done</b> Tasks   0 Comments	
#11 As a developer, we need to perform integration testing on the application.	3
<b>Done</b> Tasks   0 Comments	

#### Work Completed for Increment#4

User Story#	User Story	Hours Worked	Team Member	Status
52	As a developer, I need to enhance the functionality of update recipe	10	Leela	Done
51	As a developer, I need to enhance the UI of the application	5	Vinaya	Done
50	As a system, I need to send notifications about the expiry of the grocery items	15	Vaishnavi	Done
49	As a developer, we need to enhance the functionality of import recipe	10	Sravani, Leela	Done
16	As a team, we need to integrate the code developed by all the team members.	20	Vinaya, Sravani	Done
15	As a developer I need to develop the UI for getting the shopping list	3	Sravani	Done

13	As a user I need a shopping list so that I won't miss something needed for preparing the recipe	10	Sravani	Done
11	As a developer, we need to perform integration testing on the application.	15	Vinaya, Leela	Done
53	As a developer, I need to fix the issues identified during integration testing	15	Leela, Vaishnavi	Done

**GIT Hub URL:**

The Project code and documentation are uploaded to the following GIT hub URL:

[https://github.com/vaishnavi5054/AdvSoftEng/tree/master/ASE%20Increment\\_4%20PG\\_12](https://github.com/vaishnavi5054/AdvSoftEng/tree/master/ASE%20Increment_4%20PG_12)

**Youtube URL:**

<http://youtu.be/1NwWGkUfANo>

**Bibliography**

<https://developer.android.com/training/index.html>

<http://www.apartmenttherapy.com/whats-for-dinner-6-fridge-management-apps-weekly-smartphone-app-roundup-189441>

<http://www.snaptohealth.org/nutrition-hub/nutrition-on-the-go/>

<http://www.techhive.com/article/2455133/six-grocery-shopping-apps-to-replace-your-paper-list.html>

<http://www.groceryiq.com/>

<https://www.anylistapp.com/>