



CS5591 Advanced Software Engineering

Calorie 2 Grocery

Increment#3 Report

Project Group #12

Team Members

Sravani Punyamurthula

Vaishnavi Aienampudi

Leela Naga Devi Gajula

Vinaya Podduturi

Project Goal:

The goal of our project is to develop an android application “Calorie2grocery”, which helps its users to plan for a proper meal to maintain healthy life style. Firstly, it provides the calorie count of a recipe. The app has a browser which allows the user to import a recipe and get a list of ingredients. The app then displays the calorie count for each ingredient in the recipe. Each user has his/her own specifications and preferences for recipes. Hence the app provides its users with an option to edit the ingredient list. Users can add or remove ingredients and can get the total calorie count of that recipe. The calorie count helps its users to take healthy choices about their meal. Secondly, it automatically generates a shopping list so that the user does not miss anything that he needs to buy. It provides us the flexibility to log the groceries available at our home. Our app will generate alert notifications about the expiry dates of groceries which in turn helps in reducing the wastage of food. Based on the recipe chosen it, 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.

Increment #3

In Increment#2, we have developed an application which allows the user to a) Add grocery items available at their home to the database and view the list b) Search for a recipe and import the ingredients in the recipe.

In Increment3 we have added the following features to the application:

- 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) Add/Remove ingredients from recipes
- 6) Send notifications to users about the expiry date of recipes
- 7) Get calorie information for each ingredient of the recipe

Application Specifications:

	Tools
Platform	Android 4.0.3
UML Diagrams	Microsoft Visio
Languages	Java, C# , ASP.Net , JQuery , JavaScript , AJAX
Database	SQL Server
Planning	ScrumDo
Version Control System	GIT
Existing Rest Services	Recipe Search Calorie API
New API's	Grocery list - Added Update, Delete methods

	Import API – Added Add, remove ingredient methods
--	---------------------------------------------------

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

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 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=DgxqZYmXG8YsyCIEfFLkJaAklM4ybQtEHORaKedx

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=DgxqZYmXG8YsyCIEfFLkJaAklM4ybQtEHORaKedx

For increment#3, we are displaying the calorie count per serving of the ingredient. We need to enhance the logic to take the quantity of the ingredient and display the exact calorie count in the final increment.

New Services:

We have created 2 rest services for implementing the server side logic for this application.

1) Grocery List: This API is used to maintain the grocery lists. We added two methods to this API.

a) **Delete Grocery Item:** This method is used to delete grocery items from the grocery list. We need to provide 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](#)

- b) **Update Grocery Item:** This method is used to update the quantity & units of a grocery item. We need to pass the 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

- 2) **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 added two methods to this rest service.
- a) **Add Ingredient:** This method 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/
- b) **Remove ingredient:** This method is used to remove any of the ingredients from the recipe.
- API: http://kc-sce-cs551-2.kc.umkc.edu/aspnet_client MPG12/Leela/ViewRecipeList/Service1.svc/viewRecipeList/gih

Use Case Template to update grocery item:

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

Use Case Name	Delete grocery item
----------------------	---------------------

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

Use Case Templates for Add Ingredient to a recipe:

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

	7. The system stores the ingredient list in the database.
Extensions	User can view and update the recipe

Use Case Templates for View Favorite Recipe:

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

Use Case Templates for Update Favorite Recipe:

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

Use Case Templates for Create Favorite Recipe:

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

Use Case Templates for Share Favorite Recipe:

Use Case Name	View recipe
Primary Actor	User
Pre-Conditions	The user must be a valid user to use the app
Post-Conditions	Retrieves the results for the search
Main Success Scenario	<ol style="list-style-type: none">1. If the user search for the recipe .2. User gets the ingredients list, quantity and label for the recipe.3. User can share the recipe in the twitter.4. User can share the recipe in the facebook.

UML CLASS DIAGRAM:

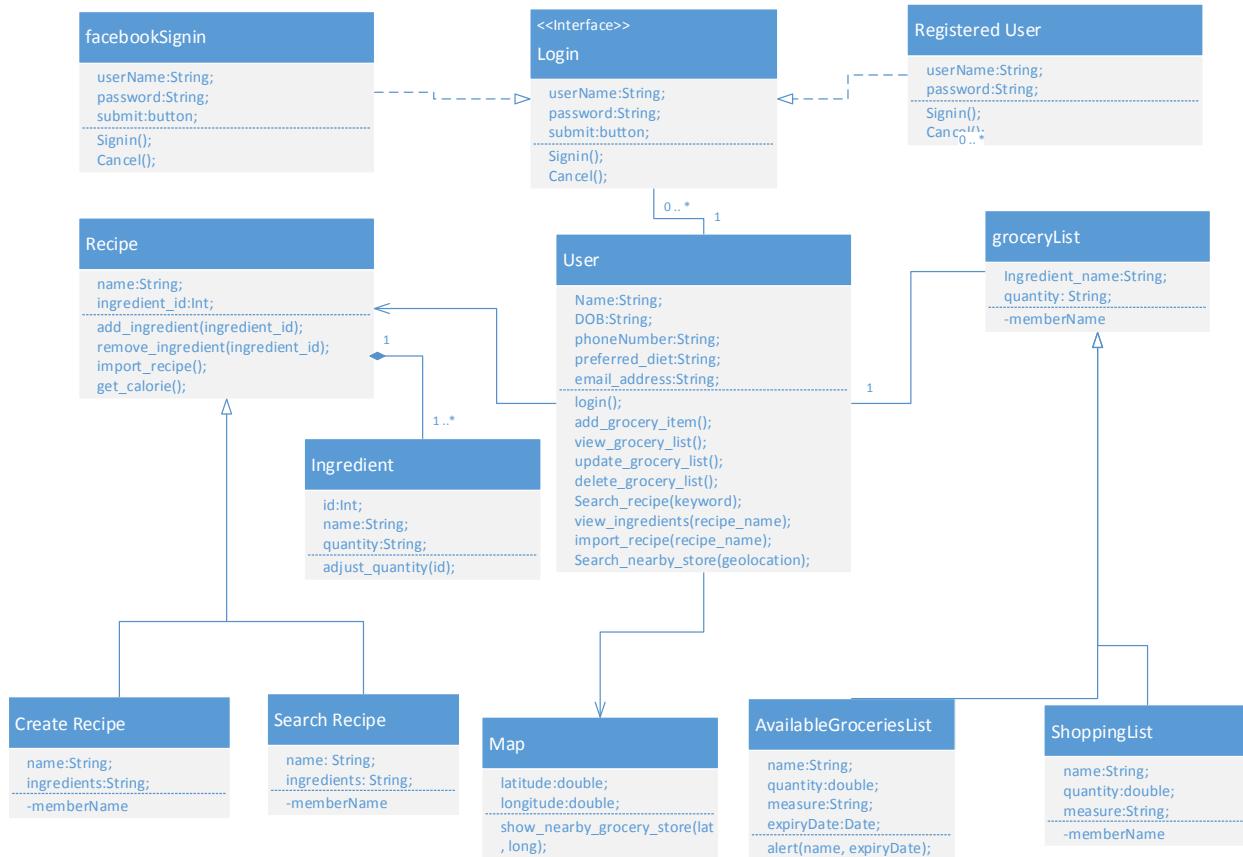
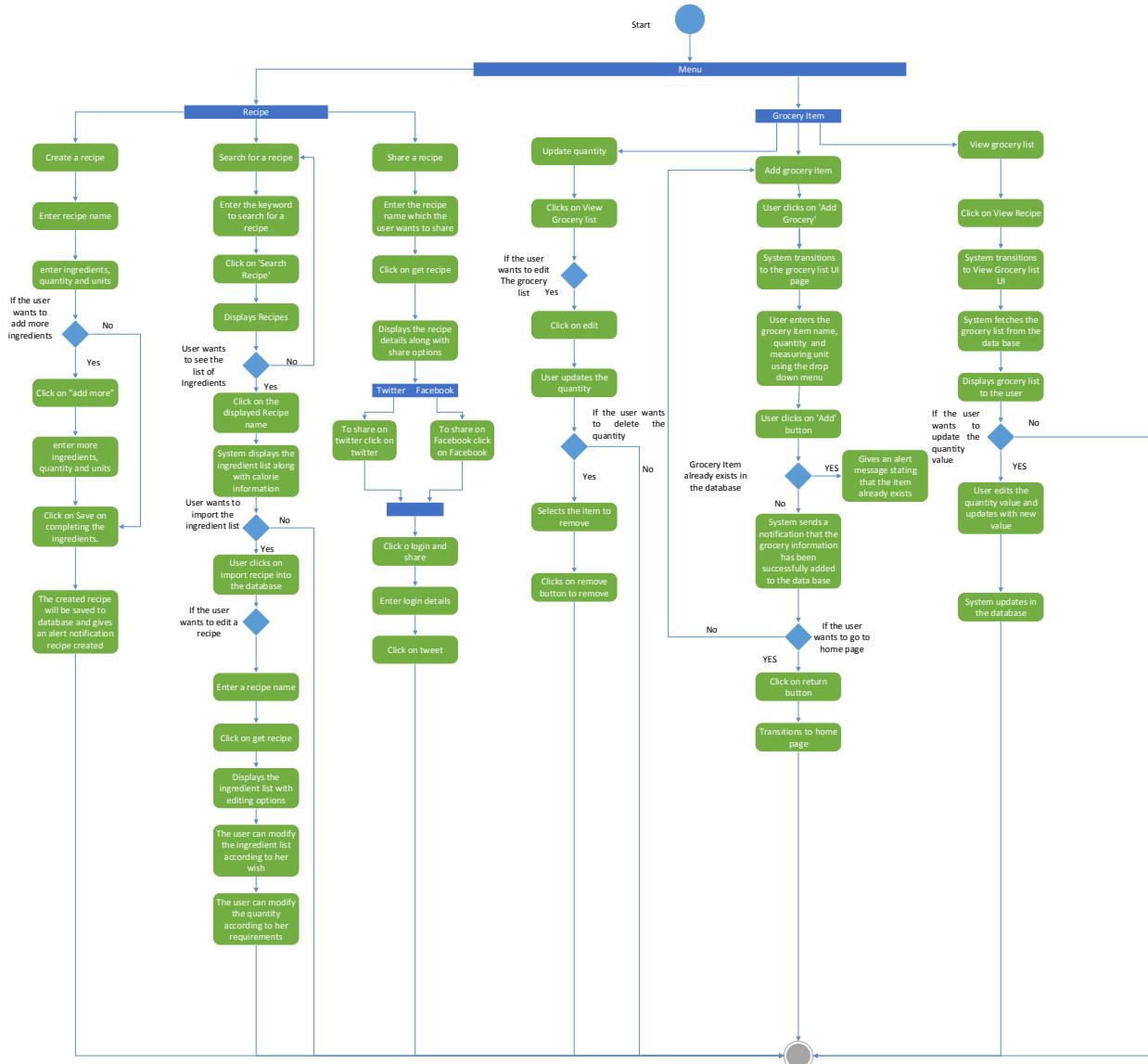


Figure 1: UML Class Diagram of the entire system

Activity Diagram for the functionality implemented in Increment 3:



Sequence Diagram:

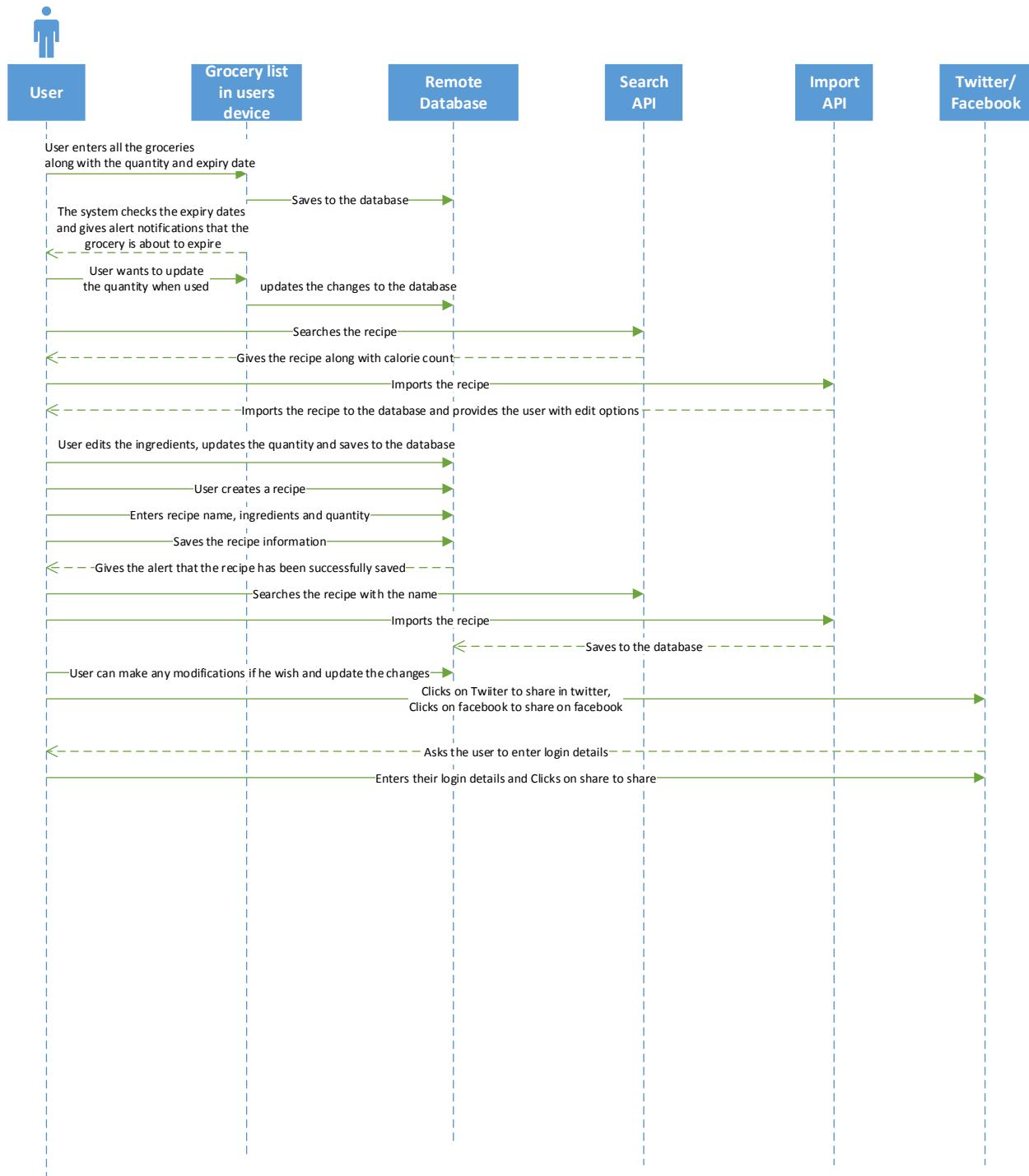


Figure 3: Sequence Diagram

Increment#3 Implementation:

1) Implementation of User Interface:

We have grouped the functionalities and developed 3 new layouts as part of UI implementation.

- 1) Home Page
- 2) My Groceries : Once we go to My groceries page, there are 2 options:
 - a) Add Grocery
 - b) View/Edit Grocery (Enhanced the functionality in this page to update/delete the grocery item)
- 3) Recipes: We have created two buttons in this page:
 - a) Search Recipe (Added functionality to add/remove ingredients and display calorie information of each ingredient)
 - b) Create Recipe (User can create his/her own recipe and save it to the database)
 - c) Share Recipe (User can share recipes through Facebook or twitter.)
- 4) Show Notifications: This page posts notifications to the user about the expiry dates of the grocery items.

Screenshots:

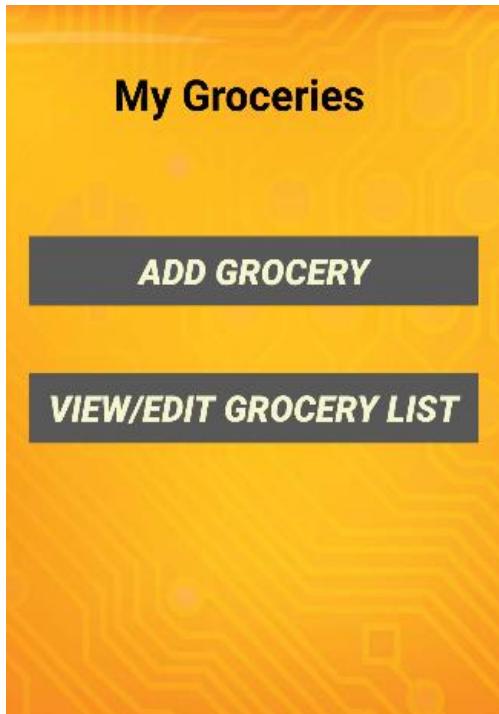


Home Page

User is provided with three buttons:

- My Groceries: User can click this button For managing his grocery list.
- Recipe: User can click this button to search for recipes or create recipes.
- Show Notifications: User can click this button to view the notifications.

Screenshot 1: Home Page



Screenshot 2: My Groceries page

My Groceries

- When the user clicks on My Groceries button in the home page, the UI is transitioned to My Groceries page.
- User can add grocery items by clicking on Add Grocery button.
- User can view the grocery list, edit the quantity of the grocery items, and delete the grocery items by clicking on View/Edit grocery List.

GROCERY LIST		
View Grocery List	Edit	
Name	Quantity	Units
cheese	50	gm
honey	100	gm
beetroot	1	kg
carrots	5	kg
butter	100	kg

Screenshot 3: View/Edit Grocery page

View/Edit Grocery List

- User can click the View grocery list button to view the grocery list.
- System fetch the grocery information from the remote database and displays it to user.

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			
View Grocery List		Edit	
Name	Quantity	Units	
- cheese	50	gm	
- honey	100	gm	
- beetroot	1	kg	
- carrots	5	kg	
- butter	100	kg	

View/Edit Grocery List

- User clicks on “Edit” button to delete grocery items from the list.
- System displays “Minus” sign in front of every grocery item.
- User needs to click on the “Minus” sign to delete a grocery item.

Technical Details:

- We have implemented the client logic using Jquery.

Screen shot 4: Edit Grocery

GROCERY LIST			
Alert			
Are you sure you want to delete carrots?			
	OK	Cancel	
- beetroot	1	kg	
- carrots	5	kg	
- butter	100	kg	

View/Edit Grocery List

- User clicks on the “Minus” sign in the “Carrots” row to delete carrots from grocery list.
- System displays an alert dialog asking the user to confirm.
- User can click on “Ok” to delete or “Cancel” to cancel the deletion.

Technical Details:

- Implemented the alert dialog using JQuery UI.

Screen shot 5: Delete grocery alert

GROCERY LIST			
View Grocery List		Edit	
Name	Quantity	Units	
cheese	50	gm	
honey	100	gm	
beetroot	1	kg	
carrots	5	kg	
butter	100	kg	

View/Edit Grocery List

- If the user clicks on “Cancel” button in the alert dialog then nothing happens.

GROCERY LIST			
View Grocery List		Edit	
Name	Quantity	Units	
cheese	50	gm	
honey	100	gm	
beetroot	1	kg	
carrots	5	kg	
butter	100	kg	

Delete Grocery

- Once the user clicks on “Ok” button in the alert dialog, grocery item “Carrots” is deleted from the list

Technical Details:

- We have created a rest service for deleting the grocery item in the remote database.

Screen shot 7: Delete Grocery

GROCERY LIST		
View Grocery List		Edit
Name	Quantity	Units
cheese	50	gm
honey	100	gm
beetroot	1	kg
butter	100	kg

View Grocery List

- Once the user clicks on View Grocery List button within the page, the page comes out of the editable mode and refreshes the grocery list.

GROCERY LIST		
Name	Quantity	U
cheese	50	gr
honey	100	gr
beetroot	1	kg
butter	100	kg

1 2 3 4 5 6 7 8 9 0
q w e r t y u i o p
a s d f g h j k l
↑ z x c v b n m ←
?123 , . →

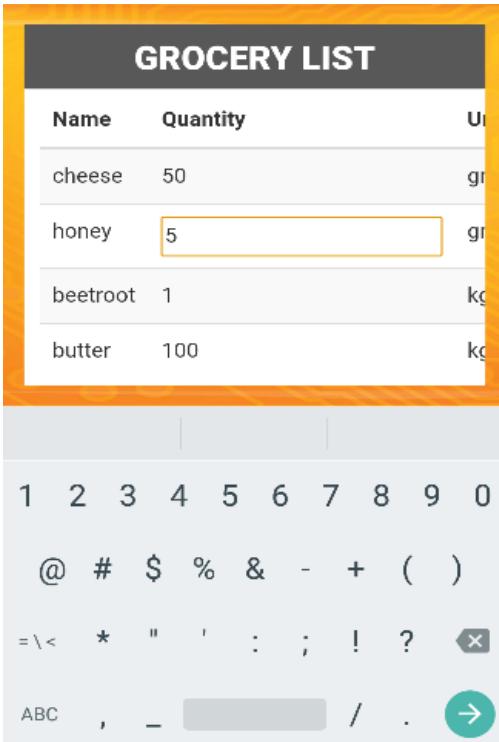
Updating Quantity

- User can click on any quantity/units cell of any grocery item row to update.
Eg: The screen shot shows that after clicking the quantity cell of Honey, the cell is changed to editable mode.

Technical Details:

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

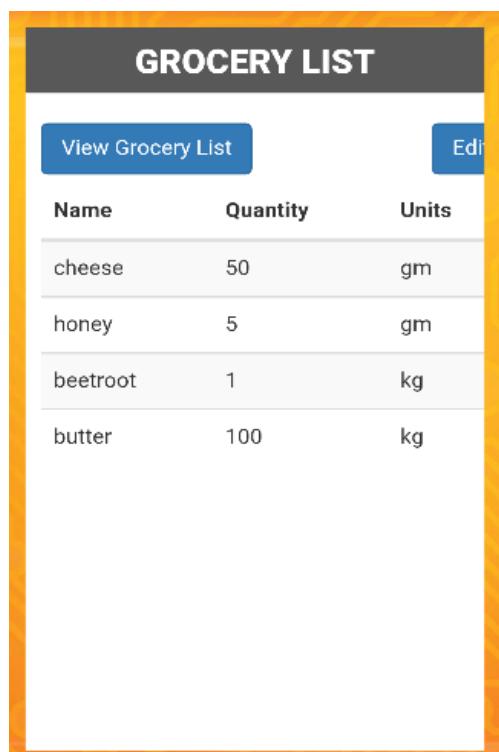
Screenshot 9: Update Grocery



Screenshot 10: Update Grocery

Updating Quantity

- User can change the quantity and hit enter. Eg: changed the quantity of honey from 100 to 5.
- In a similar manner, he can click on the Units cell of any grocery item and update it.



Screenshot 11: Update Grocery

Update Quantity

- The quantity of the grocery item "honey" is changed to 5.

Technical Details:

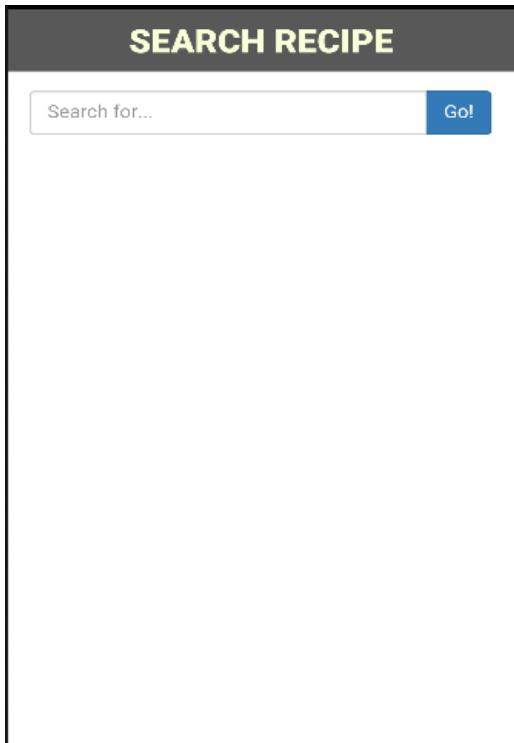
- We have created a update grocery method in the rest service.
- We have used Ajax function to call the API and update the data in the database.



Screenshot 12: Recipes Page

Recipes Page

- If the user clicks on the “Recipes” button in the home page, then the UI is transitioned to My Recipes page.
- There are two buttons in this page, one to search for recipes, the other to create their own recipe.



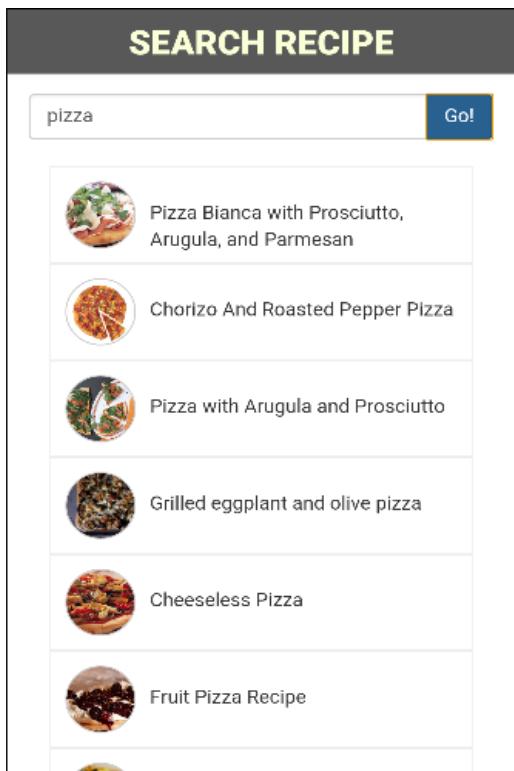
Screenshot 13: Search Recipe

Search Recipes Page

- If the user clicks on the “Search Recipes” button in the Recipes page, then the UI is transitioned to Search Recipe page.
- User can enter in the recipe name in the input box and enter Go!.

Technical Details:

- We have used bootstrap to style the input control with a placeholder for this page.



Screenshot 14: Search Recipe

The screenshot shows the same "SEARCH RECIPE" application interface. The search bar now displays "pizza". Below the search bar, the list of recipes is shown, along with additional information for the first recipe:

Total Calories: 1343

Ingredient	Quantity	Calories
Chorizo, Pork And Beef	2oz	455 Kcal
Peppers, Sweet, Red, Cooked, Boiled, Drained, Without Salt	0.5cup	75 Kcal
Pizza, Cheese Topping, Regular Crust, Frozen, Cooked	14oz	279 Kcal
Parsley, Fresh	2tbsp	23 Kcal

Below this table are two more recipe cards:

- Pizza with Arugula and Prosciutto
- Grilled eggplant and olive pizza

Screenshot 15: Ingredients with calorie info

Search Recipes Page

- In this screenshot, the user entered "Pizza" in the input box and hit Go.
- System displays list of recipes for Pizza.

Technical Details:

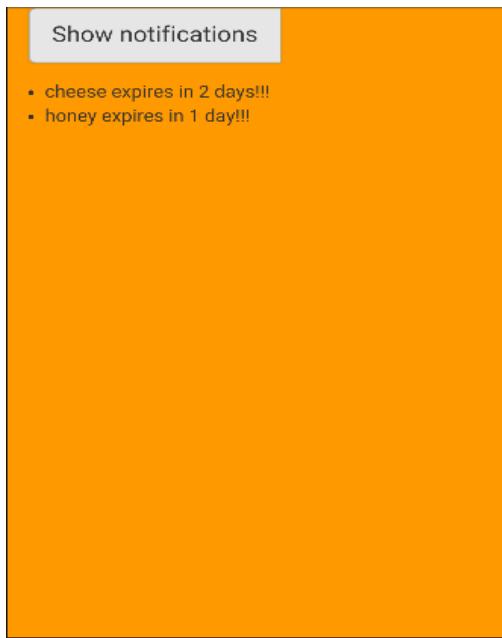
- We have used Ajax to make calls to the rest service in order to fetch the recipes.

Search Recipes Page

- To get the ingredients required for a recipe, user can click on any recipe name in the list.
- System displays the list of ingredients along with their calorie information
- System displays the total count of calories for the recipe.

Technical Details:

- To get the calorie information for each ingredient, we are making two Ajax calls to USDA NDB database.
- The first call gets the `ndb_no` for the ingredient
- The second call gets the ingredient calories based on the `ndb_no`.



Screenshot 16: Notifications

Notifications Page

- Once the user clicks on the “Show Notifications” button on the home page, UI is transitioned to this page.
- System displays the alert messages related to the expiry dates of the grocery items.

Technical Details:

- We have used a rest service to get the expiry date information of the grocery items from the database.



Screenshot 17: Create Recipe

Create Recipe Page

- User can create a new recipe by entering the recipe name, ingredients along with quantity and units.
- Add more: This button lets the user to add more ingredients along with quantity and units.
- Save: Used to save the created recipe.

The screenshot shows the 'CreateRecipe' application interface. At the top, it says 'Create New Recipe'. Below that, there are four input fields: 'strawberry smoothie', 'strawberries', '2', and 'cups'. A large blue button labeled 'Add More' is centered below these. Underneath 'Add More' are three more sets of input fields for ingredients and quantities. At the bottom is a large blue 'Save' button.

Screenshot 17: Adding ingredient

This screenshot shows the same 'CreateRecipe' application interface as the previous one, but with more data entered. The first four fields remain the same: 'strawberry smoothie', 'strawberries', '2', and 'cups'. The 'Add More' button has been pressed, adding three new rows of ingredient and quantity fields. The first row contains 'milk' and '200'. The second row contains 'ml' and an empty field. The third row contains 'sugar' and '100'. The fourth row contains 'grams' and an empty field. The fifth row contains 'water' and '50'. The sixth row contains 'ml' and is highlighted with a yellow border. At the bottom is a large blue 'Save' button.

Screenshot 17: Adding more ingredients

Create Recipe Page

- This screenshot shows creating Strawberry Smoothie recipe using the application.

Create Recipe Page

- Add more: This button lets the user to add more ingredients along with quantity and units.
- The screen shows adding additional ingredients milk, sugar, water to the recipe.

Microsoft SQL Server Management Studio

File Edit View Query Debug Tools Window Community Help

New Query Execute

Object Explorer

SQLQuery5.sql - (local)\...\spn5 SQLQuery4.sql - R3\spn5 (56) SQLQuery3.sql...5\spn5 (57)* SQLQuery2.sql - (local)...1,2 SQLQuery1.sql - (local)...5

SELECT TOP 1000 [item_name]
,[ingredient]
,[quanity]
,[label]
FROM [Calorie2Grocery].[dbo].[Ingredients]

	item_name	ingredient	quanity	label
380	mango juice	sugar	100	grams
381	strawberry smoothie	strawberries	2	cups
382	strawberry smoothie	milk	200	ml
383	strawberry smoothie	sugar	100	grams
384	strawberry smoothie	water	50	ml
385	Strawberry Shake	strawberry ice cream	1	pint
386	Strawberry Shake	strawberry	4	oz

Query executed successfully. (local)\SQLEXPRESS (10.50 SP2) | UMKC-USER5\spn5 (75) | master | 00:00:00 | 387 rows

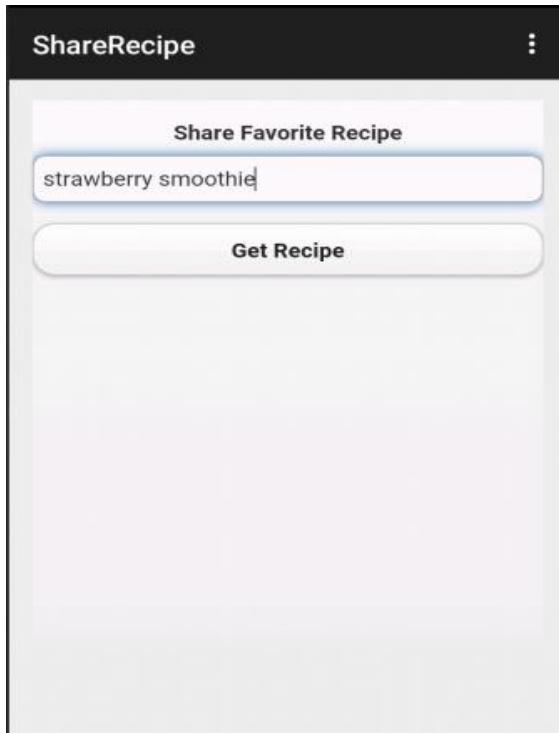
Screen Shot: 18 shows that the above created recipe has been added to the database successfully.



Share Recipe in Facebook and Twitter:

- The user can enter the recipe name and search from the database to get the recipe details.
- Get Recipe:** Gives the list of recipes, that matches the keyword along with share options.

Screen Shot: 19 Share Recipe Page



Screen Shot: 20 Share Recipe

Share Recipe Page

- Screenshot shows searching for strawberry smoothie recipe that was previously created.



Screen Shot: 21 Showing the recipe details

Share Recipe Page

- Screenshot shows displaying the details of the strawberry recipe.

The screenshot shows the Twitter login interface. At the top, the URL is https://twitter.com/intent/tw. Below it, there's a blue header bar with the Twitter logo and a 'Sign up' link. The main area starts with a 'What's happening?' section containing a text input field with the following text: "Try this strawberry smoothie recipe ingredients: milk strawberries sugar water quantity: 200 2 100 50 units: ml cups grams ml". Below this is a form for entering a phone number or email and password, with a red '-1' indicating an error. There are also 'Remember me' and 'Forgot password?' links, and a 'Log in and Tweet' button. A 'New to Twitter?' sidebar on the right includes a 'Sign up' button.

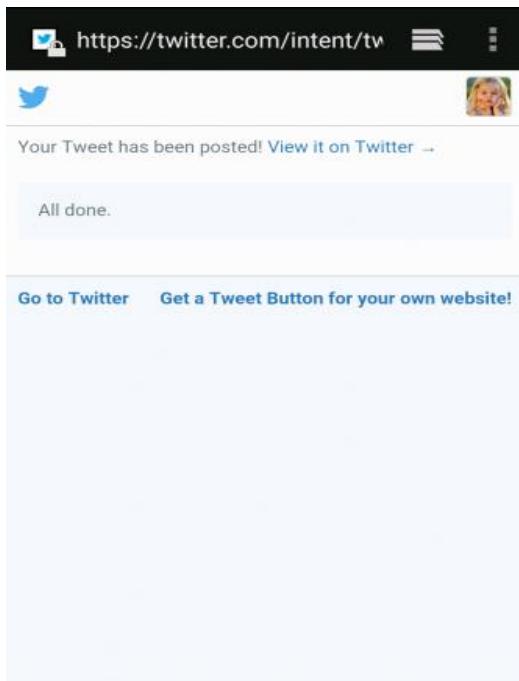
Screenshot: 22

This screenshot is identical to Screenshot 22, but it shows user input in the fields. The phone number input field contains 'vinayareddy15' and the password input field contains '.....'. The rest of the interface, including the recipe text, buttons, and sidebar, remains the same.

Screenshot : 23

Share Recipe using Twitter

- On clicking the Tweet hyperlink, user can login to twitter and share the recipe.



Screenshot : 24

Share Recipe using Twitter

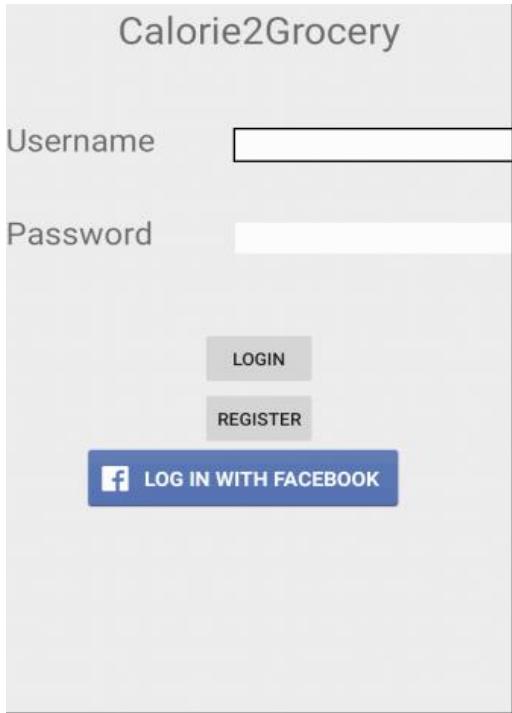
- Screenshot showing success message after posting the tweet.



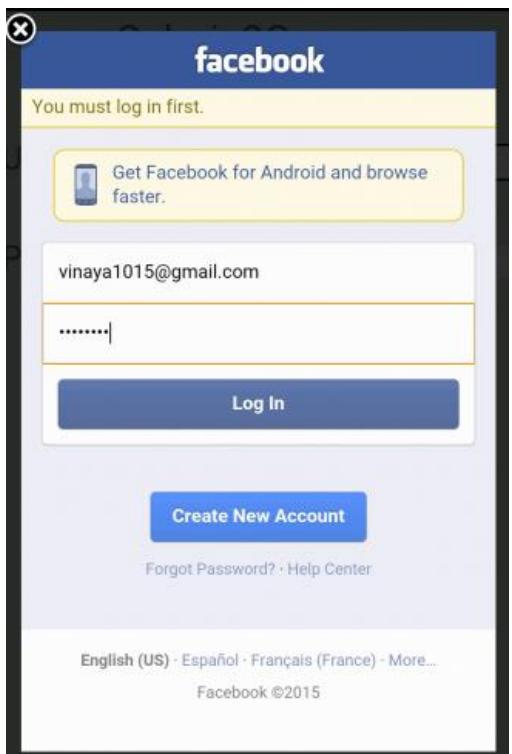
Screenshot :25 Twitter post

Share Recipe using Twitter

- Screen shot showing the post in twitter page.



The screenshot shows the Calorie2Grocery login interface. It features a light gray header with the app name. Below it is a form with 'Username' and 'Password' fields, each with a corresponding input box. Underneath the password field is a 'LOG IN' button. To the right of the password field is a 'REGISTER' button. At the bottom of the form is a blue button labeled 'LOG IN WITH FACEBOOK' with a white 'f' icon.



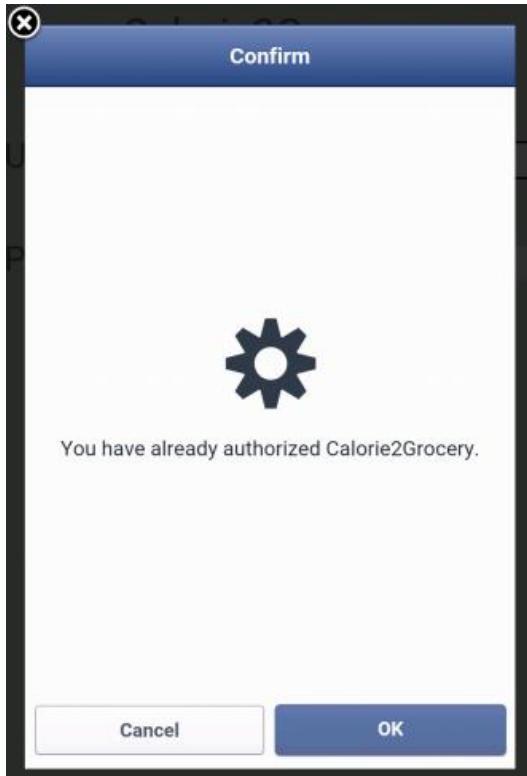
The screenshot shows the Facebook login screen. It has a dark blue header with the word 'facebook'. Below it, a yellow bar displays the message 'You must log in first.' A button for 'Get Facebook for Android and browse faster.' is shown. The main area contains email and password input fields, both containing placeholder text ('vinaya1015@gmail.com' and '.....'). A large blue 'Log In' button is centered below these fields. At the bottom of the screen are links for 'Create New Account', 'Forgot Password? · Help Center', and language options ('English (US) · Español · Français (France) · More...'). The footer also includes the text 'Facebook ©2015'.

Facebook Login

- In Increment#1 , we had an issue with the facebook login. After login, the UI was not transitioning to our application.
- We have fixed that issue in increment#3.

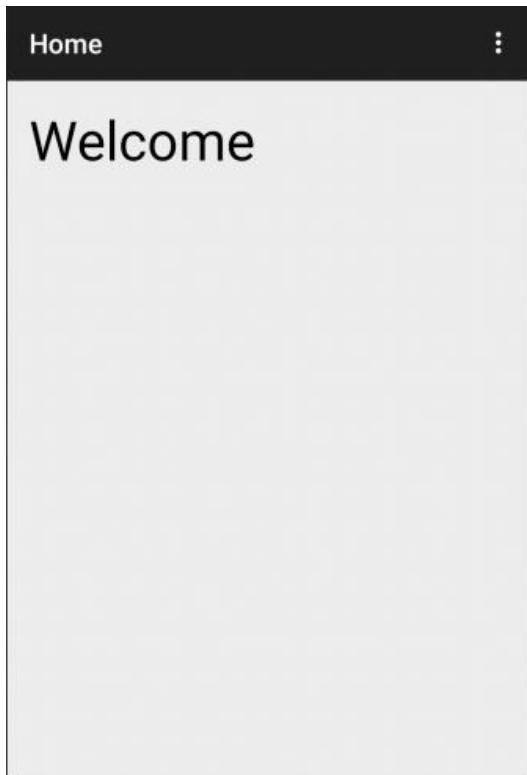
Facebook Login

- This screenshot shows entering the login details for Facebook.



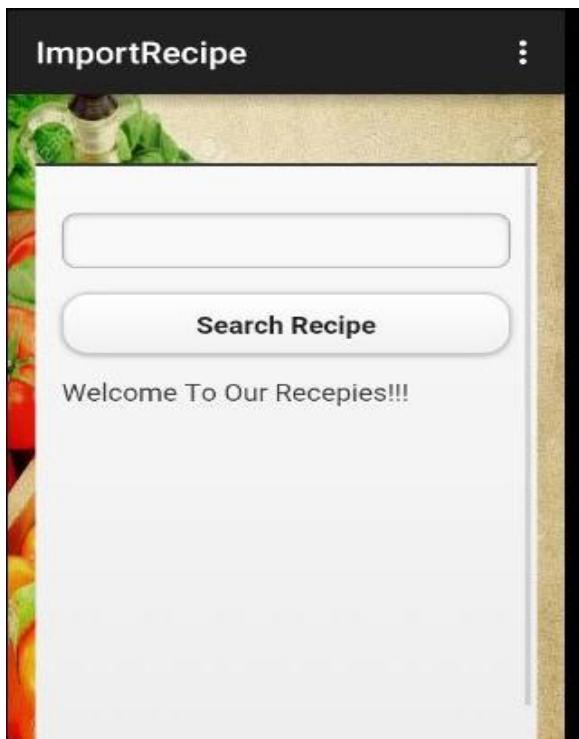
Facebook Login

- This screenshot shows that the user has authorized the Calorie2Grocery application to access basic info from facebook.



Facebook Login

- This screenshot shows the successful transition of UI from Facebook login to our application.

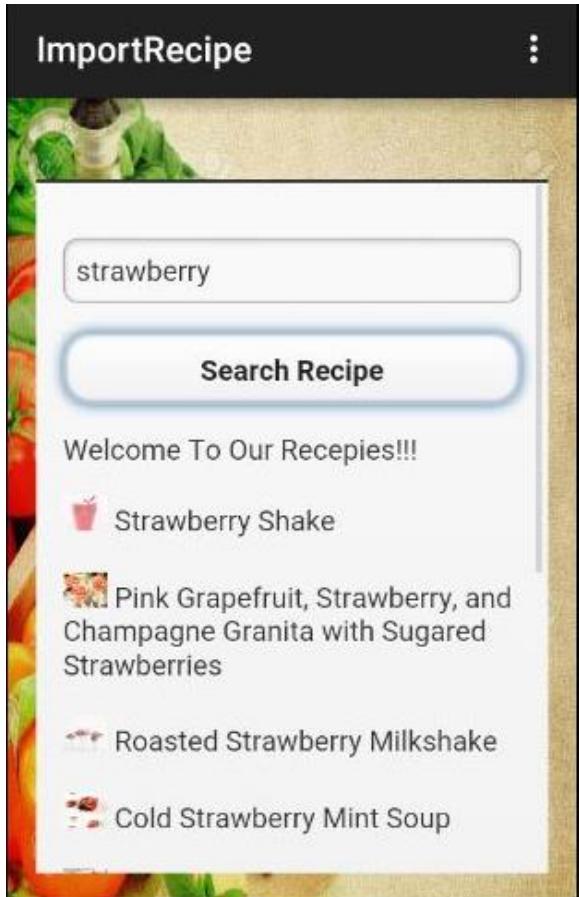


Search Recipe

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

Technical Details:

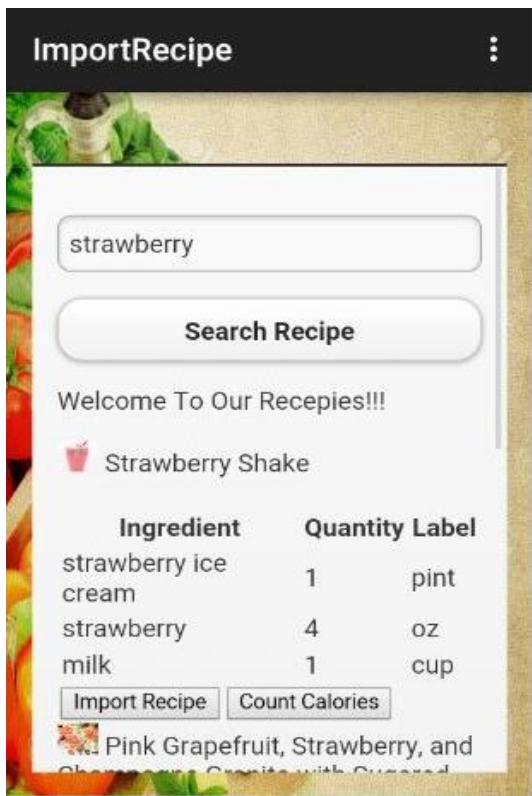
- The client side logic is implemented using web technologies JavaScript, Jquery & Ajax
- Used bootstrap style sheets.



Search Recipe

- If user wants to view the ingredients required for preparing a recipe, user clicks on recipe name
- System displays the ingredient list for the recipe.
- If the user wants to import the recipe, user clicks on "Import Recipe" button

System loads the recipe & ingredients details to the database



Import Recipe:

Once the user clicks import recipe, the recipe is loaded into the database using a rest service.

Microsoft SQL Server Management Studio

File Edit View Query Debug Tools Window Community Help

New Query Execute

Object Explorer

SQLQuery5.sql - (local)\...\(5) SQLQuery4.sql ... RS\spnt5 (S8) SQLQuery3.sql...5\spnt5 (S7)* SQLQuery2.sql - (local)\...\(2) SQLQuery1.sql - (local)\...\(5)

```
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP 1000 [item_name]
    ,[ingredient]
    ,[quantity]
    ,[label]
FROM [Calorie2Grocery].[dbo].[Ingredients]
```

Results Messages

item_name	ingredient	quantity	label
382 strawberry smoothie	milk	200	ml
383 strawberry smoothie	sugar	100	grams
384 strawberry smoothie	water	50	ml
385 Strawberry Shake	strawberry ice cream	1	pint
386 Strawberry Shake	strawberry	4	oz
387 Strawberry Shake	milk	1	cup

Imported the Strawberry Shake into the Ingredients Table

Microsoft SQL Server Management Studio

File Edit View Query Debug Tools Window Community Help

New Query Execute

Object Explorer

Connect to... SQL Server 2008 (local)\...\s...* SQLQuery5.sql - (local)...(5) SQLQuery2.sql - (local)...(2) SQLQuery1.sql - (local)...(5)

SQLQuery6.sql - (local)\...\s...* SQLQuery5.sql - (local)...(5) SQLQuery2.sql - (local)...(2) SQLQuery1.sql - (local)...(5)

select * from [Calorie2Grocery].[dbo].[Ingredients] where item_name='sweet';

Databases System Databases Calorie2Grocery Database Diagrams Tables System Tables dbo.Address dbo.grocery_list dbo.Ingredients dbo.newrecipe dbo.Registration dbo.user_profile Views Synonyms Programmability Stored Procedures System Stored Procedures dbo.Add_Grocery Parameters Functions Table-valued Functions Scalar-valued Functions

Results Messages

	item_name	ingredient	qunality	label
1	sweet	taggey	0	lbs
2	sweet	milk1	6	lite
3	sweet	sugar	50	lb
4	sweet	water	1	gallon

Query executed successfully. (local)\SQLEXPRESS (10.50 SP2) UMKC-USER\\$spnt5 (53) master 00:00:00 4 rows

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'Calorie2Grocery' is selected. In the Results pane, a query is run to select all rows from the 'Ingredients' table where 'item_name' is 'sweet'. The results show four rows of data:

	item_name	ingredient	qunality	label
1	sweet	taggey	0	lbs
2	sweet	milk1	6	lite
3	sweet	sugar	50	lb
4	sweet	water	1	gallon

A status bar at the bottom indicates the query was executed successfully.

Screenshot the Ingredients list before the update the sweet Recipe.

2) Design Of Unit Test Cases

UI Testing:

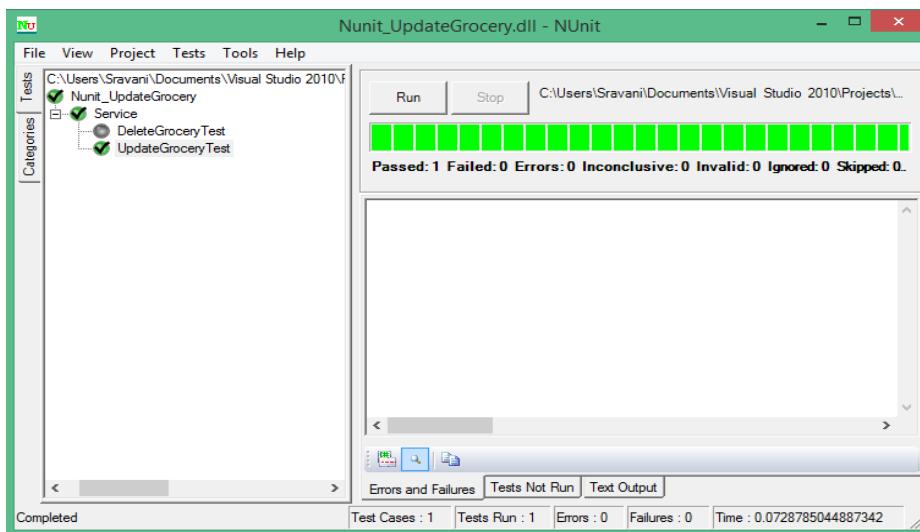
The following test cases has been written for UI testing and the actual output is validated with expected output. All of the below test cases which have been written obtained positive results.

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

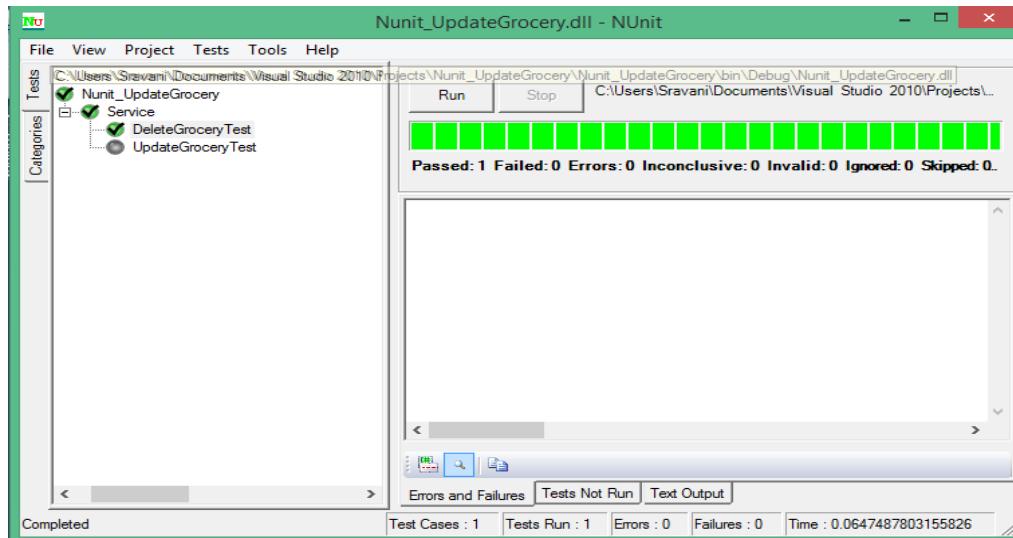
Data Validation:

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.

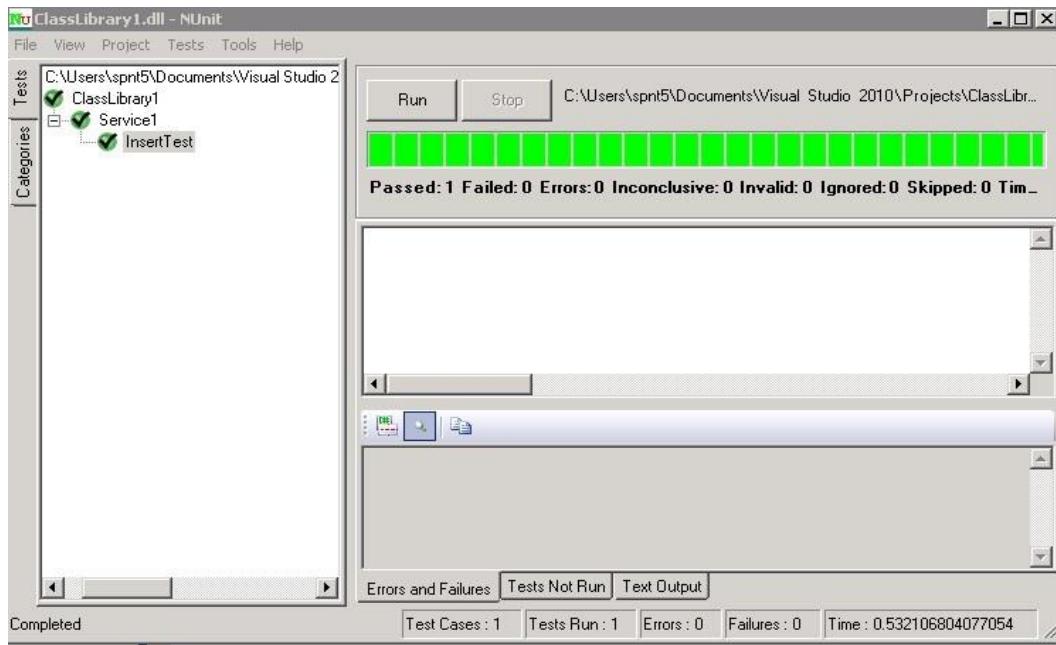
Screenshot of N-Unit validation: This is used to validate the data for update grocery item method in the rest service.



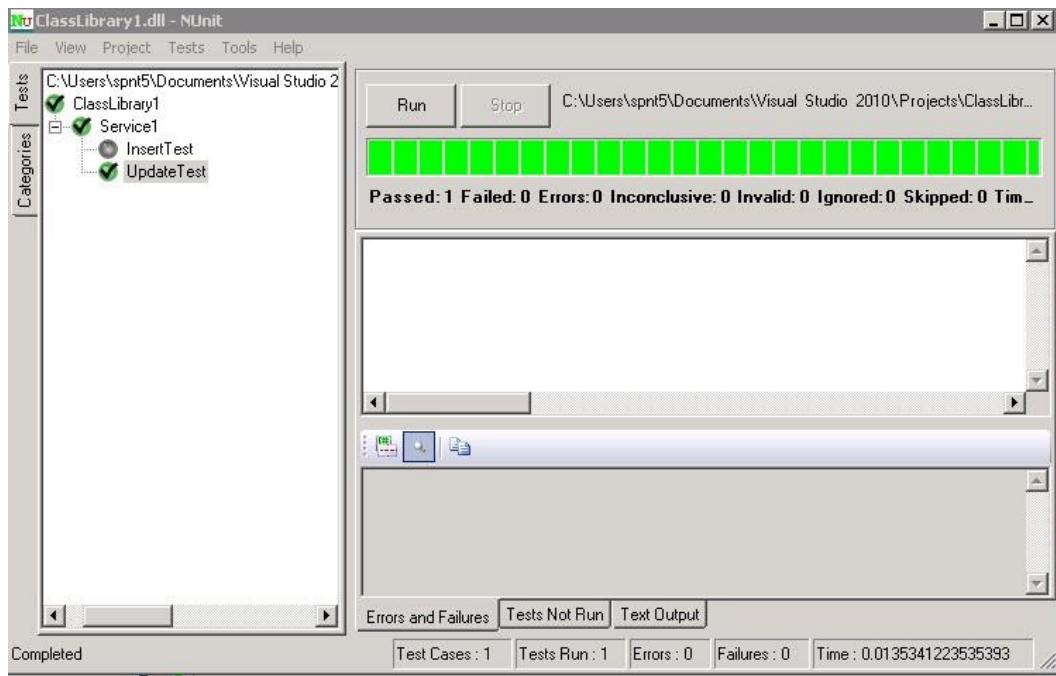
Screenshot for N-Unit validation of delete groceries API:



Screenshot for N-Unit validation of Import groceries API:



Screenshot for N-Unit validation of Update groceries API:



Performance Testing:

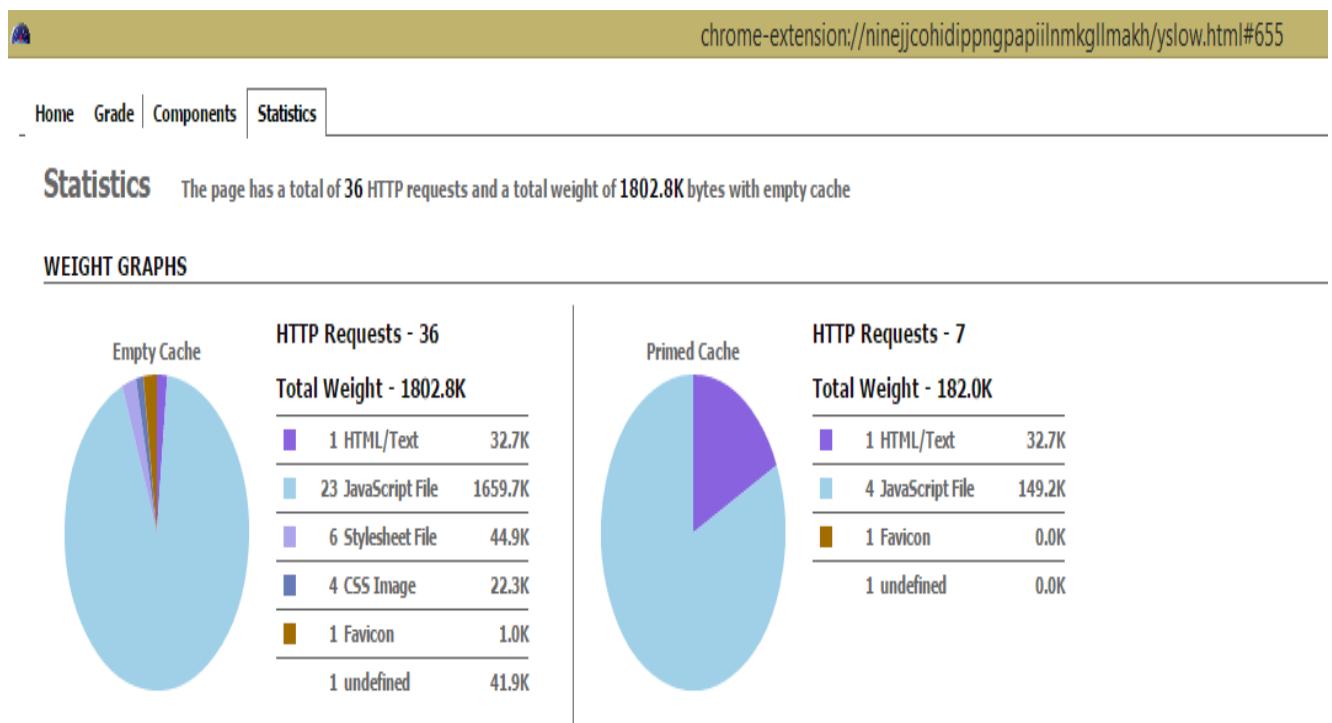
The following are the tools we used for performance testing:

- 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



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									

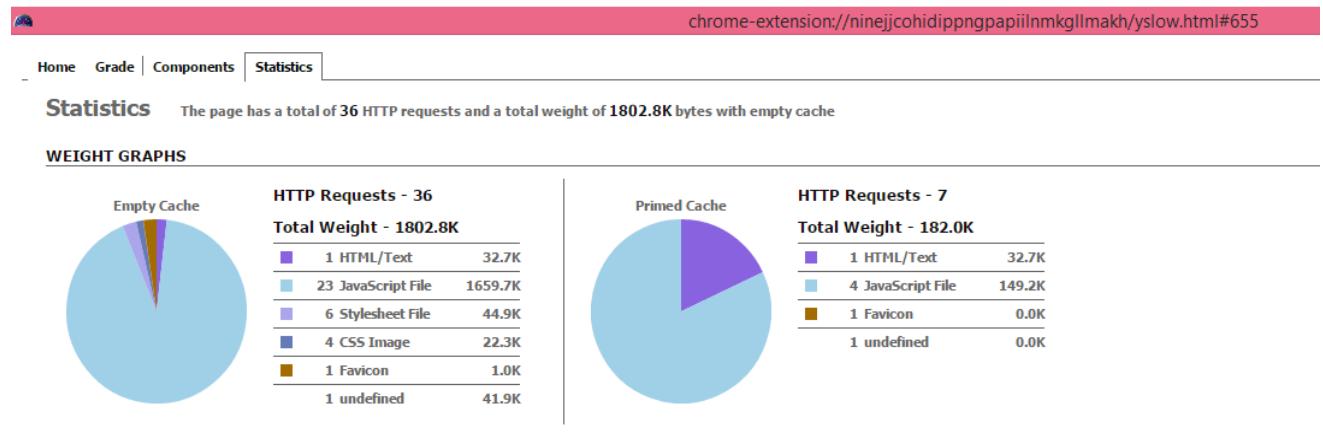
* type column indicates the component is loaded after window.onload event

† 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 a table listing 37 components with their details.

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

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 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

Empty Cache	Primed Cache																				
HTTP Requests - 36	HTTP Requests - 7																				
Total Weight - 1802.8K	Total Weight - 182.0K																				
<table border="1"> <tr> <td>1 HTML/Text</td> <td>32.7K</td> </tr> <tr> <td>23 JavaScript File</td> <td>1659.7K</td> </tr> <tr> <td>6 Stylesheet File</td> <td>44.9K</td> </tr> <tr> <td>4 CSS Image</td> <td>22.3K</td> </tr> <tr> <td>1 Favicon</td> <td>1.0K</td> </tr> <tr> <td>1 undefined</td> <td>41.9K</td> </tr> </table>	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	<table border="1"> <tr> <td>1 HTML/Text</td> <td>32.7K</td> </tr> <tr> <td>4 JavaScript File</td> <td>149.2K</td> </tr> <tr> <td>1 Favicon</td> <td>0.0K</td> </tr> <tr> <td>1 undefined</td> <td>0.0K</td> </tr> </table>	1 HTML/Text	32.7K	4 JavaScript File	149.2K	1 Favicon	0.0K	1 undefined	0.0K
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																				
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://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.

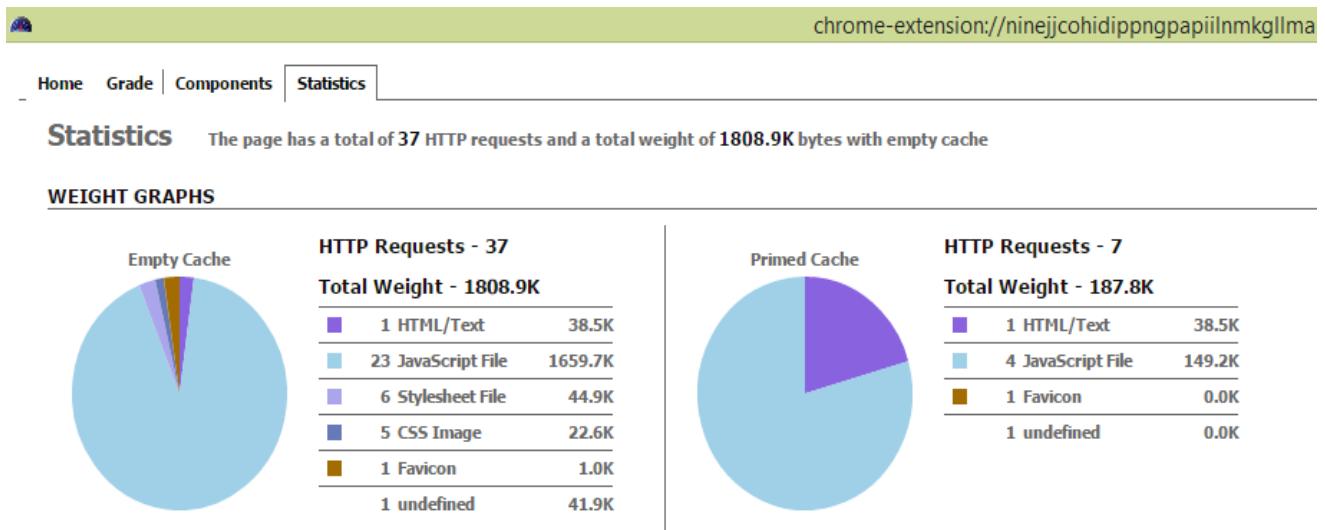
Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, 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					http://jsfiddle.net/la6h6/qf6y2smq/				
doc	38.5K	38.5K		665		http://jsfiddle.net/la6h6/qf6y2smq/8/	no expires	0		
is (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/is/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 (5)	22.6K									
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/remove-resources.png	2015/5/3	0	"5120150e-120"	
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									
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

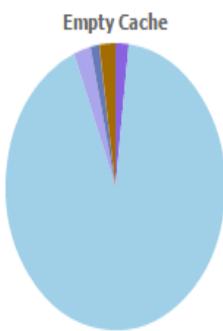
chrome-extension://ninejjcohidippngpapiilnmkgllm/

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

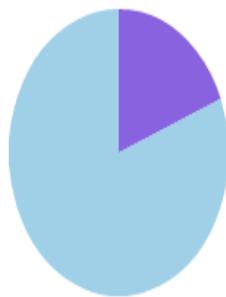


HTTP Requests - 37

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	⌚	http://jsfiddle.net/spmt5/n6rl5r3s/2/	no expires	0		
is (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/is/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 (5)	22.6K									
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/remove-resources.png	2015/5/3	0	"5120150e-120"	
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: 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.

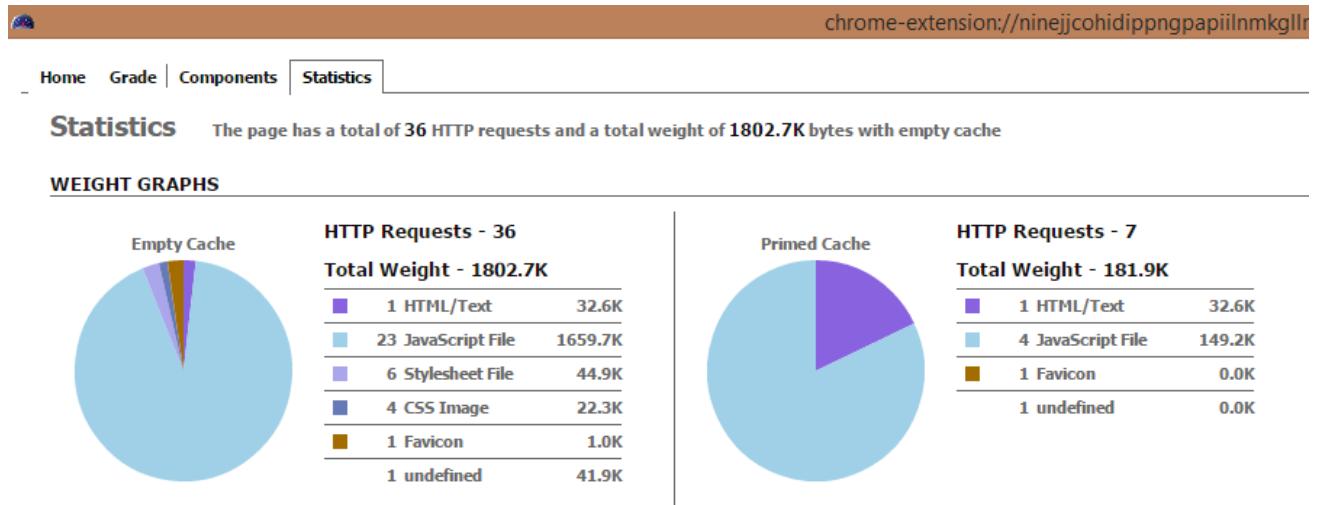
Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, 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	Count	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	Count	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	Count	Weight
1 HTML/Text	31.2K	
4 JavaScript File	149.2K	
1 Favicon	0.0K	
1 undefined	0.0K	

HTTP Requests - 7

Total Weight - 180.4K

Type	Count	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	♂	http://jsfiddle.net/css/normalize.css?_m	2015/5/4	0		
css	0.2K	0.2K		667	♂	http://jsfiddle.net/css/tricks.css?_m	2015/5/4	0		
css	18.7K	18.7K		667	♂	http://jsfiddle.net/css/elusive-webfont.css?_m	2015/5/4	0		
css	8.0K	8.0K		667	♂	http://jsfiddle.net/js/codemirror/lib/codemirror.css?_m	2015/5/4	0		
css	17.2K	17.2K		667	♂	http://jsfiddle.net/css/screen.css?_m	2015/5/4	0		
css	0.0K			667	♂	http://jsfiddle.net/css/light.css?_m	2015/5/4	0	"5120150e-0"	
cssimage (4)	22.3K									
cssimage	20.8K			667	♂	http://jsfiddle.net/img/initializing.png	2015/5/4	0	"5128cfbb-515f"	
cssimage	1.1K			667	♂	http://jsfiddle.net/img/logo.png	2015/5/4	0	"5120150e-488"	
cssimage	0.2K			667	♂	http://jsfiddle.net/img/handle-h.png	2015/5/4	0	"5120150e-ca"	
cssimage	0.1K			667	♂	http://jsfiddle.net/img/handle-v.png	2015/5/4	0	"5120150e-c4"	
favicon (1)	1.0K									
favicon	1.0K			667	♂	http://jsfiddle.net/favicon.png	no expires	0	"5120150e-444"	
font (1)	41.9K			667	♂	http://jsfiddle.net/font/Elusive-Icons.eot?_m	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/qlvn9naw/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.

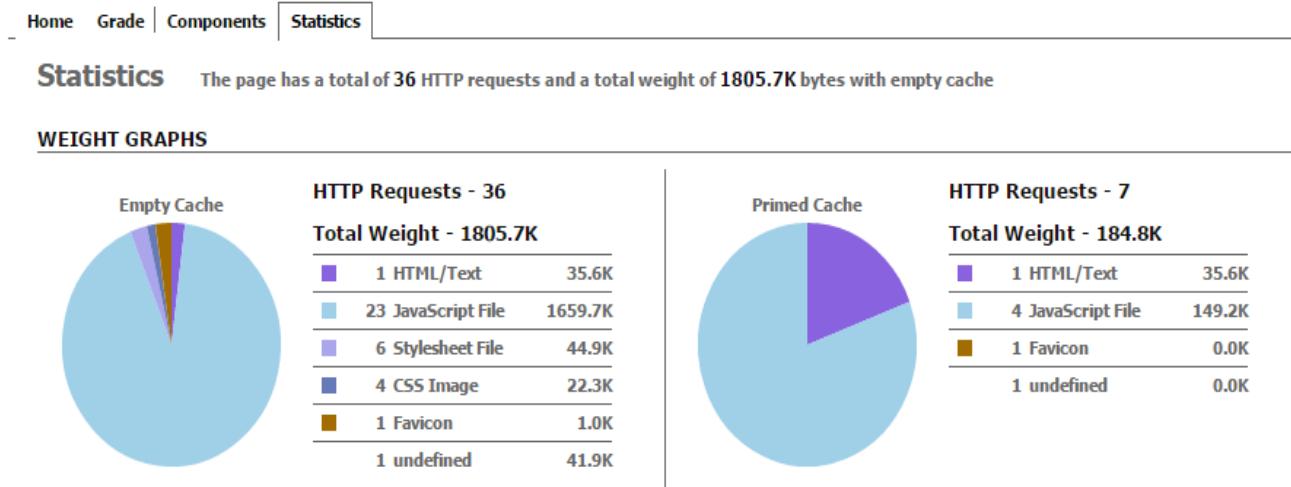
Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, 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	⌚	http://jsfiddle.net/css/normalize.css?...	2015/5/4	0		
css	0.2K	0.2K		670	⌚	http://jsfiddle.net/css/tricks.css?...	2015/5/4	0		
css	18.7K	18.7K		670	⌚	http://jsfiddle.net/css/elusive-webfont.css?...	2015/5/4	0		
css	8.0K	8.0K		670	⌚	http://jsfiddle.net/js/codemirror/lib/codemirror.css?...	2015/5/4	0		
css	0.0K			670	⌚	http://jsfiddle.net/css/light.css?...	2015/5/4	0 "5120150e-0"		
css	17.2K	17.2K		670	⌚	http://jsfiddle.net/css/screen.css?...	2015/5/4	0		
[+] cssimage (4)	22.3K									
cssimage	20.8K			670	⌚	http://jsfiddle.net/img/initializing.png	2015/5/4	0 "5128cfbb-515f"		
cssimage	1.1K			670	⌚	http://jsfiddle.net/img/logo.png	2015/5/4	0 "5120150e-488"		
cssimage	0.2K			670	⌚	http://jsfiddle.net/img/handle-h.png	2015/5/4	0 "5120150e-ca"		
cssimage	0.1K			670	⌚	http://jsfiddle.net/img/handle-v.png	2015/5/4	0 "5120150e-c4"		
[+] favicon (1)	1.0K									
favicon	1.0K			670	⌚	http://jsfiddle.net/favicon.png	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 (Grade C), Components, and Statistics. The Grade section shows an overall performance score of 74 and notes that the ruleset applied is YSlow(V2). The URL is listed as `http://jsfiddle.net/leeladevi/uq3c6k96/3/`. A filter bar at the top says "All (23) FILTER BY: CONTENT (6) COOKIE (2) CSS (6) IMAGES (2) JAVASCRIPT (4) SERVER (6)". On the right, there are links to "Tweet" and "Share". The main content area is titled "Grade C" and "Overall performance score 74 Ruleset applied: YSlow(V2) URL: http://jsfiddle.net/leeladevi/uq3c6k96/3/". It lists various items under the heading "F Make fewer HTTP requests", such as "Use a Content Delivery Network (CDN)", "Avoid empty src or href", "Add Expires headers", "Compress components with gzip", "Put CSS at top", "Put JavaScript at bottom", "Avoid CSS expressions", and "Make JavaScript and CSS external". There are also sections for "Reduce DNS lookups", "Minify JavaScript and CSS", "Avoid URL redirects", "Remove duplicate JavaScript and CSS", "Configure entity tags (ETags)", "Make AJAX cacheable", "Use GET for AJAX requests", "Reduce the number of DOM elements", "Avoid HTTP 404 (Not Found) error", "Reduce cookie size", "Use cookie-free domains", "Avoid AlphaImageLoader filter", "Do not scale images in HTML", and "Make favicon small and cacheable". A note states: "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." A "Read More" link is provided.

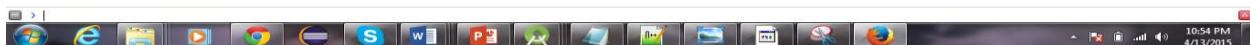
Screen Shot: 3

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

The screenshot shows the Firebug extension interface. The browser tab bar includes "Create Recipe", "DirectionAndWeat...", "DirectionAndWeat...", "Connecting...", "file:///C:/...ries.html", and "Title of the document...". The main content area shows a "Create New Recipe" form with fields for "veg fried rice", "boiled rice", "1", "bowl", and a table for "carrots", "onions", and "spices" with values "1", "cup", "cup", "1", "cup", "spoon". A modal dialog box is centered, displaying "recipe created!!" with a checkbox "Prevent this page from creating additional dialogs" and an "OK" button. At the bottom, there are "Save" and "Console" tabs, and a status bar with "Clear", "Persist", "Profile", and "All" buttons.



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

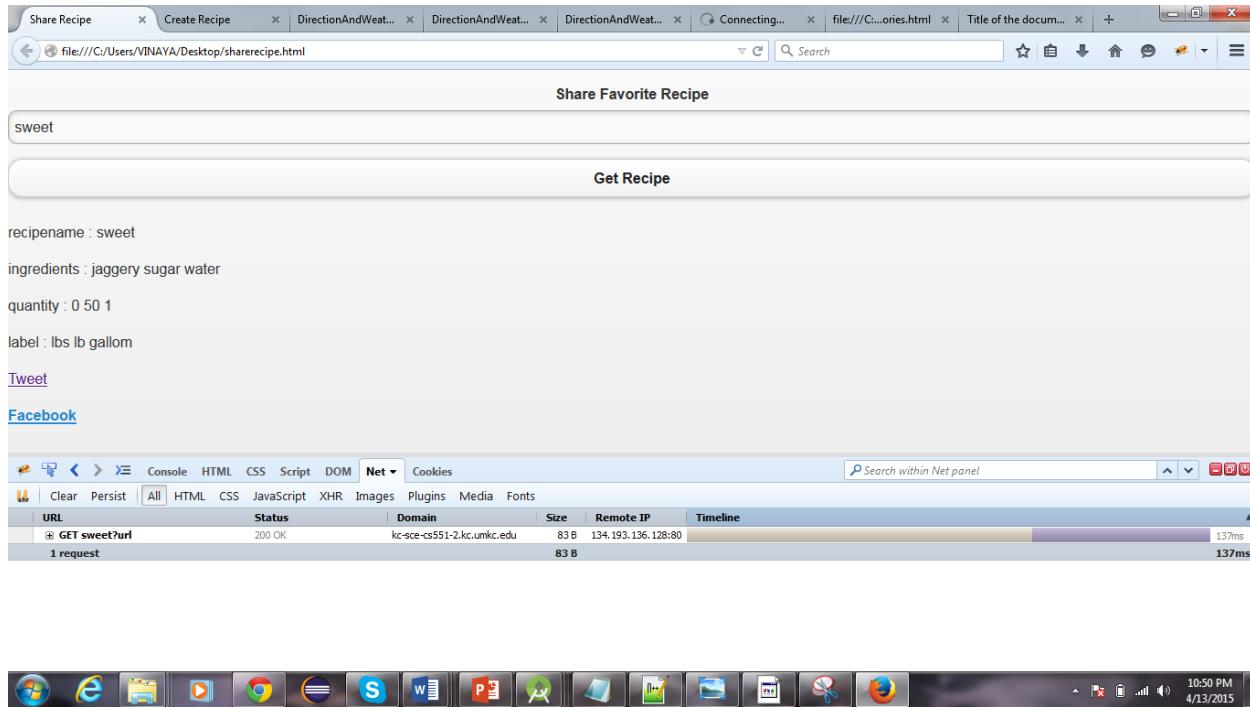
label : lbs lb gallom

Console

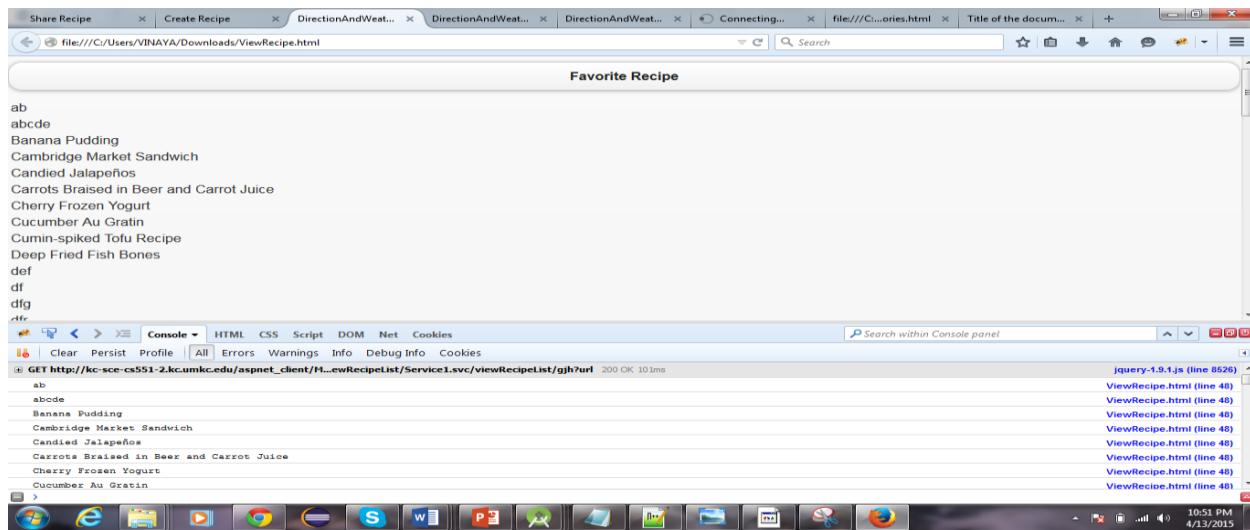
Search within Console panel

```
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



The screenshot shows a web browser window with multiple tabs open. The active tab displays a list of recipes under the heading "Favorite Recipe". The list includes:

- 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

Below the list is a "Net" panel showing network activity. It lists two requests:

URL	Status	Domain	Size	Remote IP	Timeline
GET gih7url	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

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



4. Updaterecipe.html

The screenshot shows a web browser window displaying a form to update a recipe. The form fields are:

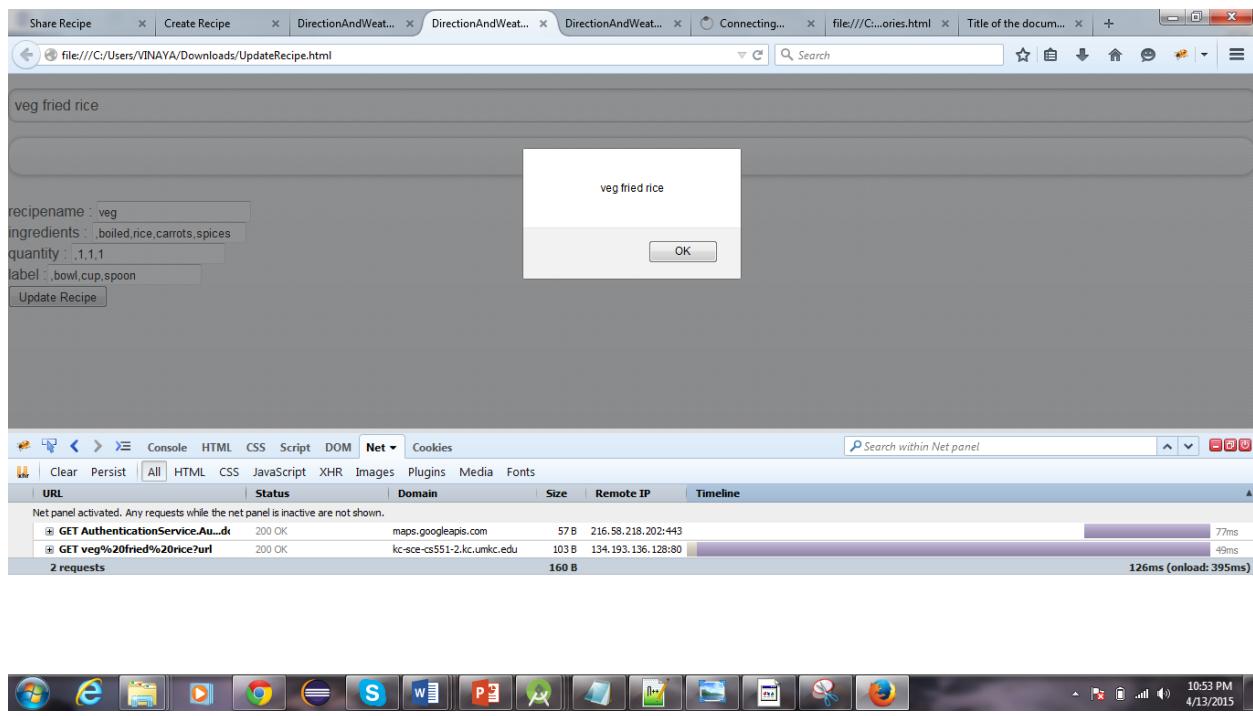
- recipename: veg
- ingredients: boiled,rice,carrots,spices
- quantity: 1,1,1
- label: bowl,cup,spoon

Below the form is a button labeled "Update Recipe". A modal dialog box is displayed with the text "veg fried rice" and an "OK" button.

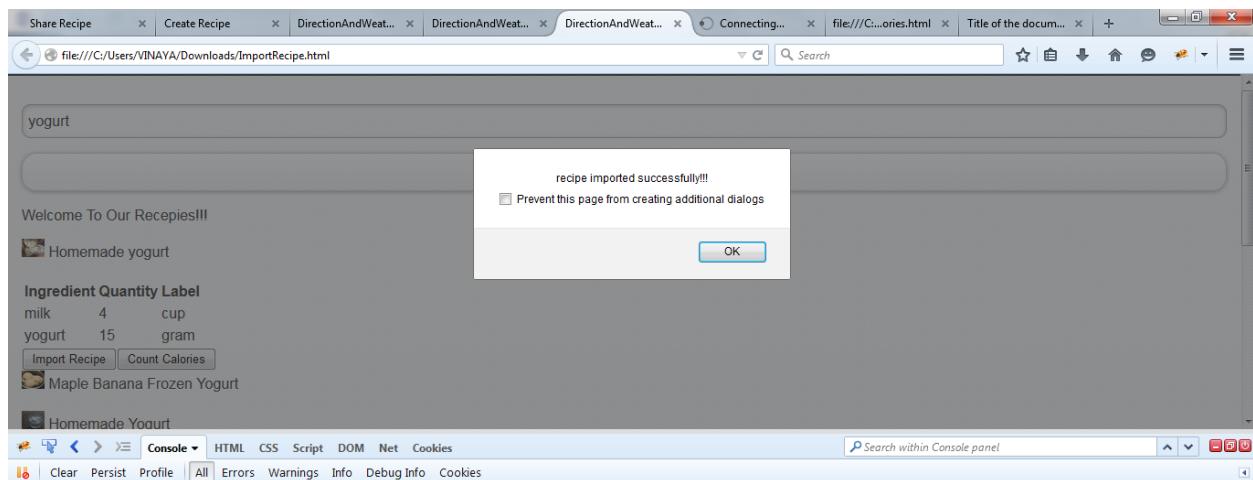
At the bottom of the browser window is a "Console" panel showing the following log entry:

```
jQuery 1.9.1.js (line 8526)
GET http://kc-sce-cs551-2.kc.umkc.edu/aspnet_client/M_yrecipe/Service1.svc/data/veg%20fried%20rice?url 200 OK 49ms
```





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 8b876a85852aa638378f5f4c24 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 c75a65078289ac1f74057923 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 Internet Explorer Firefox Google Chrome Microsoft Edge S Opera Mozilla Firefox

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 2495c263291a4d48ec13afcd 200 OK edamam.com 13.0 KB 107.20.173.119:443 415ms
- ④ GET 3057774>22a554674e6bb3b 200 OK edamam.com 18.2 KB 107.20.173.119:443 529ms
- ④ GET a475e9c073aae44dc51cb2 200 OK edamam.com 14.4 KB 107.20.173.119:443 644ms
- ④ GET b7dc59bf463c15e4427fd9c 200 OK edamam.com 19.6 KB 107.20.173.119:443 738ms
- ④ GET 79d3947d96eb0493b73aa 200 OK edamam.com 20.7 KB 107.20.173.119:443 840ms
- ④ GET ?format=json&q=strawber 200 OK api.nal.usda.gov 246 B 54.165.187.228:80 399ms
- ④ GET ?format=json&q=strawber 200 OK api.nal.usda.gov 217 B 54.165.187.228:80 391ms
- 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 167ms
- 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 Internet Explorer Firefox Google Chrome Microsoft Edge S Opera Mozilla Firefox

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 network 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".

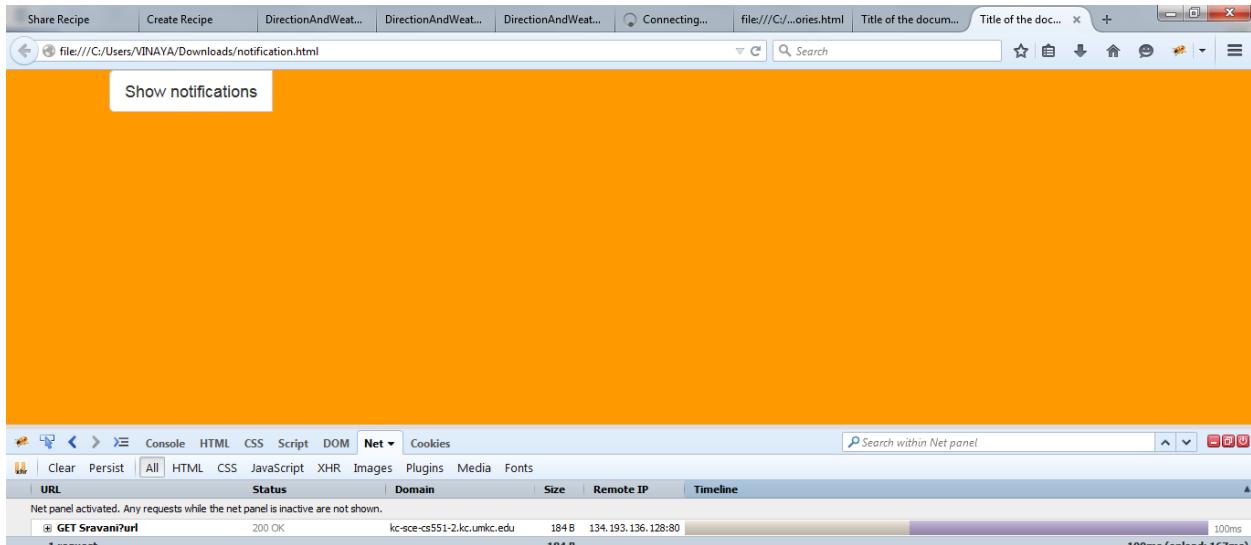
The browser's developer tools (Console tab) show the following logs:

```

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)

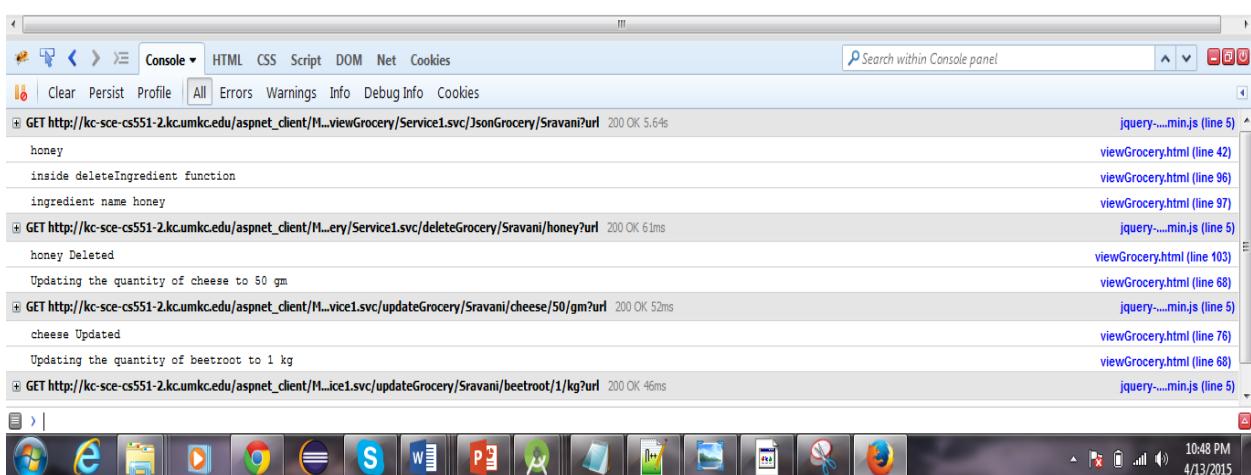
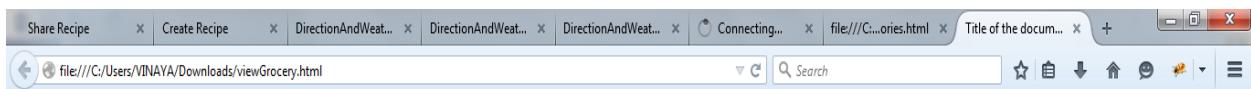
```





8. viewGrocery.html

The screenshot shows a web browser window with a white background. At the top, there is a toolbar with various icons and a search bar. Below the toolbar, a message box displays the text "View Grocery List". The main content area shows a table with a header row "Name", "Quantity", and "Units". Two rows of data are listed: one for "cheese" (50 gm) and one for "beetroot" (1 kg). Below the table, there is a blue "Edit" button. The bottom part of the screenshot shows the Firebug Net panel. The Net panel has a header with tabs like Console, HTML, CSS, Script, DOM, Net (selected), and Cookies. It shows a table of network requests with columns for URL, Status, Domain, Size, and Remote IP. Multiple requests are listed, including JavaScript files from ajax.googleapis.com and various CSS and image files from kc-sce-cs551-2.kc.umkc.edu. The timeline below the table shows the total load time for 10 requests was 6.16s with an onload time of 1.27s.



5) ScrumDo

We have used Scrumdo to track the progress of the project. The project development is divided into four increments. All the functions and development tasks are written as user stories.

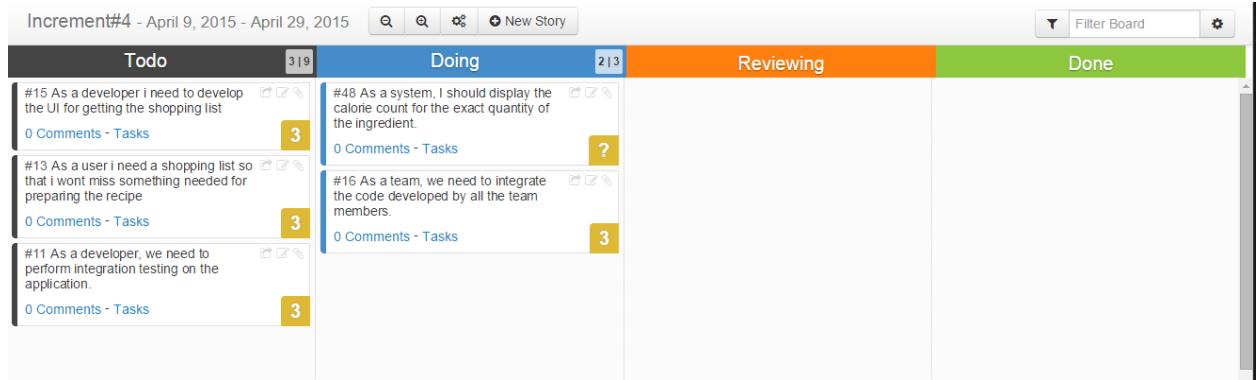
Link: <https://www.scrumdo.com/projects/project/calorie2grocery3/iteration/122932#>

Screenshots for Increment#3 user stories:

All the user stories planned for increment #3 have been successfully implemented.

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

Increment #4 user stories



Work Completed:

User Story#	User Story	Hours Worked	Team Member	Status
9	As a user I should be able to update the quantity of the grocery items in my list	10	Sravani	Done
27	As a user I should be able to delete some of the entries in the grocery list	8	Sravani	Done
28	As a system, I should send notifications to the user about the expiry date of the grocery items.	20	Vaishnavi	Done
12	As a user I should be able to remove ingredients from the recipe	15	Leela	Done
36	As a user I should be able to add ingredients to the recipe	15	Leela	Done
37	As a user I should be able to create a recipe	10	Vinaya	Done
38	As a user I should be able to share my recipes using Facebook/twitter	10	Vinaya	Done
39	As a user I should be able to get the calorie count of the complete recipe	2	Sravani	Done
40	As a user I should be able to get the calorie information of individual ingredients of the recipe	15	Sravani	Done
42	As a user I should be able to login using Facebook and	5	Vinaya	Done

	transition to the app - Fix on increment 1			
47	As a system I need to store the recipes created by the user in the remote database	5	Vinaya	Done
46	As a developer I need to test the APP performance using the tools available.	20	Vaishnavi , Vinaya	Done
45	As a developer I need to perform Nunit testing on newly built .Net rest services.	8	Sravani, Leela	Done
44	As a developer I need to test that all UI transitions are happening as expected	2	Sravani, Leela	Done

GIT Hub URL:

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

<https://github.com/vaishnavi5054/AdvSoftEng>

Bibliography:

1. Reference: <http://api.bigoven.com/>

API: <http://api.bigoven.com/recipes/47725> api_key=dvx30p6vcIMHZWh3G1mghS88YvV6140D

This API provides two functions:

- Get Recipe
- Search Recipe

This is a REST based API. This API helps to search the recipes, display the list of ingredients. It redirects to another webpage where the user can see the recipe reviews etc... It also allows to update the grocery lists in the cloud. This supports either XML or JSON formats.

2. Reference: <https://www.recipal.com>

API: <https://recipal.com/api/v1/recipes/522>

This API helps to search and retrieve the recipes according to the scale. We can update an ingredient into the ingredient list. It also allows the user to create the customer ingredient object. The user can create a new recipe. Update an existing recipe etc...

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

API: https://api.edamam.com/search?q=chicken&app_id=51aba909&app_key=9fcfd3aa5746d2a423a350cee3ea4d57d

This API provides services for the following:

- Nutritional Analysis API
- Recipe Search and Diet API

Here we are using Recipe Search and Diet API to search for a recipe. This API gives retrieves more information about the ingredients, the preparation time and procedure for that recipe. So, this API is more useful for our project as it gives the detailed information which helps us to import the ingredients. With the help of this API we can filter more while searching for a recipe like we can give allergic restrictions etc.