

Calorie2Grocery

PROJECT REPORT

Submitted By

SRAVANI PUNYAMURTHULA, LAKSHMI VAISHNAVI AIENAMPUDI, VINAYA PODDUTURI, LEELA
NAGA DEVI GAJULA

FOR THE PARTIAL FULFILMENT OF THE COURSE

CS551-ADVANCE SOFTWARE ENGINEERING

SPRING – 2015

Instructor

Dr. Yugyung Lee

TA's

Malathy Krishnan

Mayanka Chandrashekhar

Bharath Viswanadham



Table of Contents

<u>S.no</u>	<u>Description</u>	<u>Page No</u>
1.	Project Deployment	1
2.	User Manual	2
3.	Project Management	4
4.	Project Proposal	5
5.	Project Plan	7
6.	First Increment	18
7.	Second Increment	33
8.	Third Increment	55
9.	Fourth Increment	111
10.	Presentation Materials and Project Evaluation	

PROJECT DEPLOYMENT

The entire code of implementation has been completed and deployed in GitHub. The GitHub link is provided below:

https://github.com/vaishnavi5054/AdvSoftEng/tree/master/ASE%20Increment_4%20PG_12

The project description and methodologies are described briefly and are captured as a video. The video has been uploaded to the YouTube which can be viewed using the following link:

<https://youtu.be/tT-zoL18Qd8>

Our project has been planned using the Scrum do which is an agile tool and the link is given below as follows:

<https://www.scrumdo.com/projects/project/calorie2grocery3/summary>

USER MANUAL

GitHub Project URL:

https://github.com/vaishnavi5054/AdvSoftEng/blob/master/ASE%20Increment_4%20PG_12/A SE_Increment4_PG12.pdf

1. 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.

2. How to use the system:

This is an android application. Any user can access our system by registering and logging in for free.

2.1 How to access the system:

2.2 How to access the system:

Upon opening our application, it will display the “Login Page”. It also provides a registration button, which allows the new users to register. We are also providing login with facebook option through which users can login into this application with their facebook credentials. After getting successfully logged into the system, the application redirects to home page. In the home page different options will be provided to the users related to grocery and recipe management.

User can search for a recipe using any keyword. The application retrieves the results based on the search keyword. If the user wants to see the recipe, he can click on the recipe name. It then gives a list of ingredients along with calorie information. If the user wishes, he can import the recipe and generate a shopping list.

The user can also create a recipe and share it using twitter and facebook. The user can also edit the imported recipes, he can delete the ingredients, add more ingredients and can also update the quantity.

The application also provides a feature for grocery management, where user can maintain a grocery list. He can update the quantity whenever used along with units. Our application also provides another feature for notifications. If the user wants, the application generates a notification of the expiry dates of all the groceries.

PROJECT MANAGEMENT

As part of our Project Management, We used Scrum Do for planning our tasks. We divided the project into 4 increments. Created user stories for each independent task in the increments and assigned the tasks to our team.

1. Sravani Punyamurthula
2. Lakshmi Vaishnavi Aienampudi
3. Vinaya Podduturi
4. Leela Naga Devi Gajula

<https://www.scrumdo.com/projects/project/calorie2grocery3/summary>

PROJECT PROPOSAL

PROJECT GOAL AND OBJECTIVES

Motivation:

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.

Since mobile devices are available for every human in the world where we had experienced a lot of innovation in the areas of Education we decided to develop an android application that manages both calorie and grocery information.

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.

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.

Overall Goal and 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 chose 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.

PROJECT PLAN

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 and 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 chose 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 and 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

Fresh Box	Fooducate	Anylist	Proposed System
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:

Requirement	Tools
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

Web Services:

- Google Places API: <https://developers.google.com/places/>
- Google Maps API: <https://developers.google.com/maps/documentation/android/>
- Facebook API: <https://developers.facebook.com/docs/facebook-login/v2.2>
- Google Plus API: <https://developers.google.com/+/web/signin/>
- USDA NDB API: <http://ndb.nal.usda.gov/ndb/api/doc>

Class Diagram:

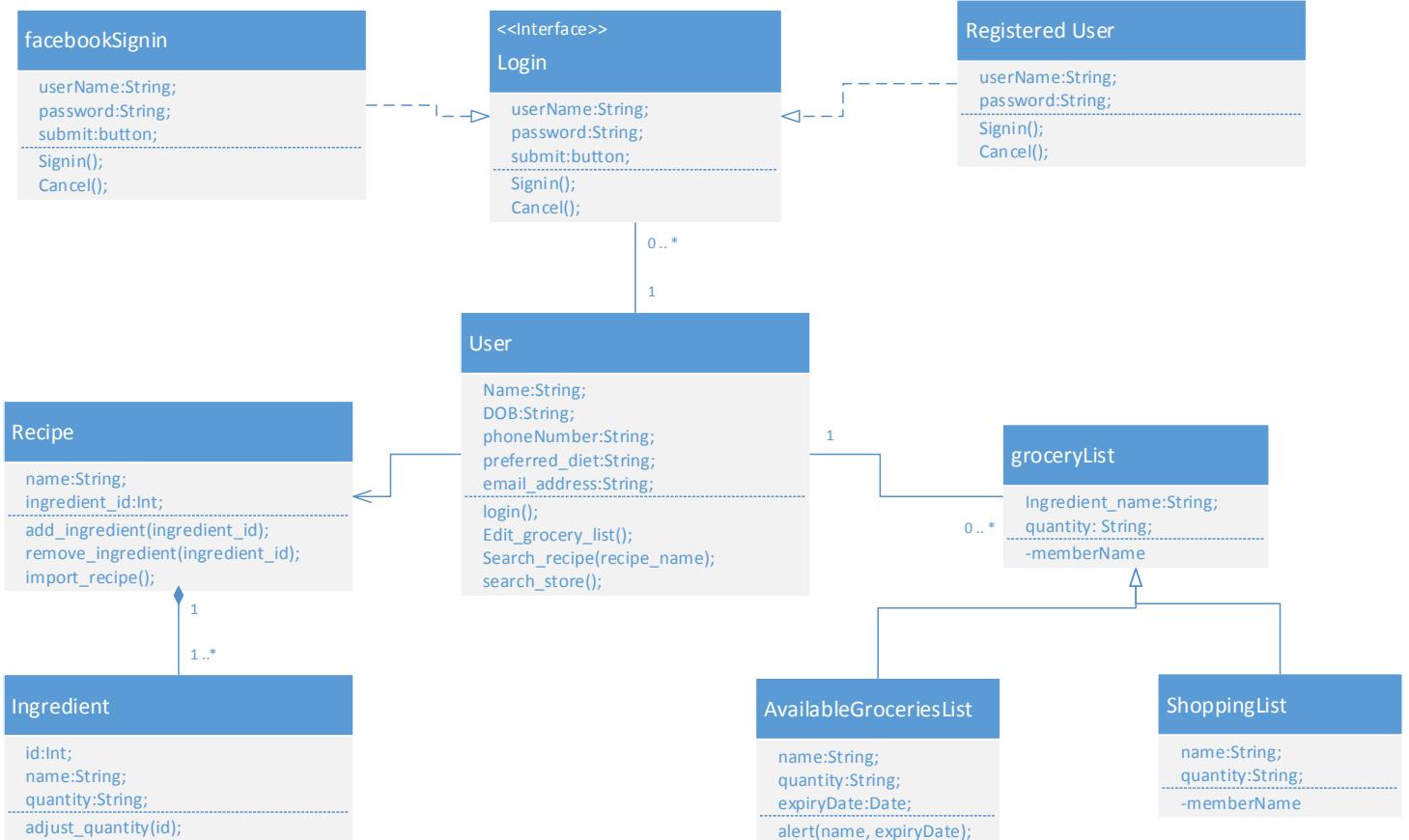


Figure 1.1 Class Diagram

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/rest services. The response is then redirected to the user through the GUI.

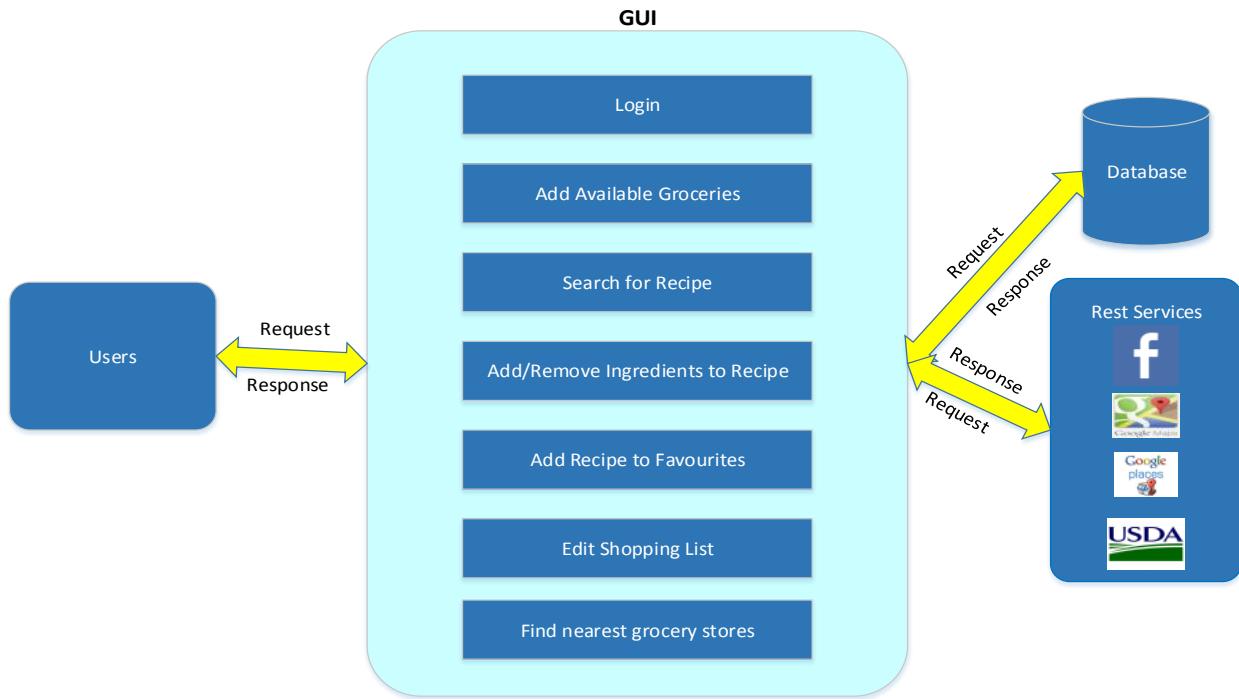
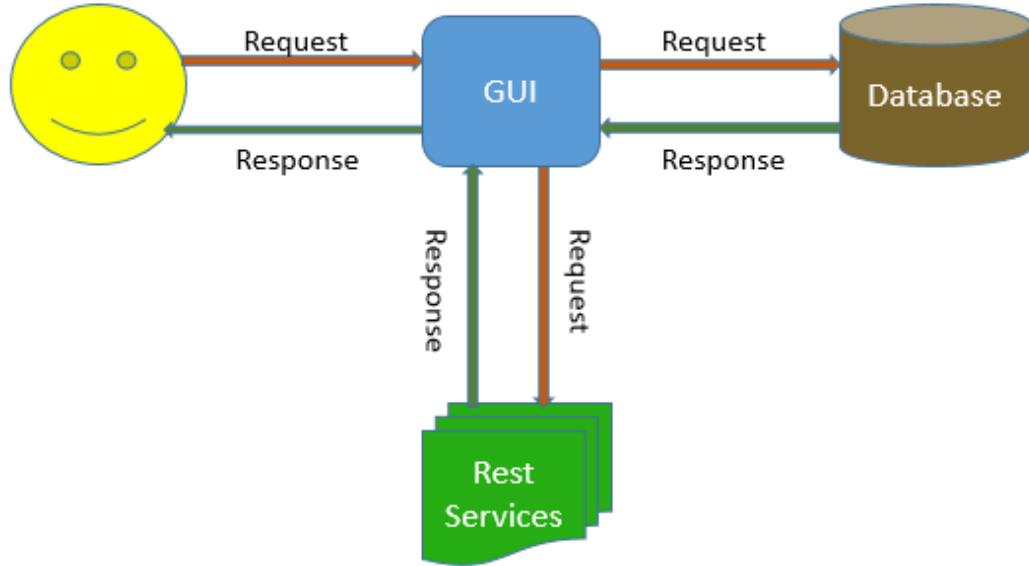


Figure 1.2 System Model

System Architecture Diagram:



Activity Diagram:

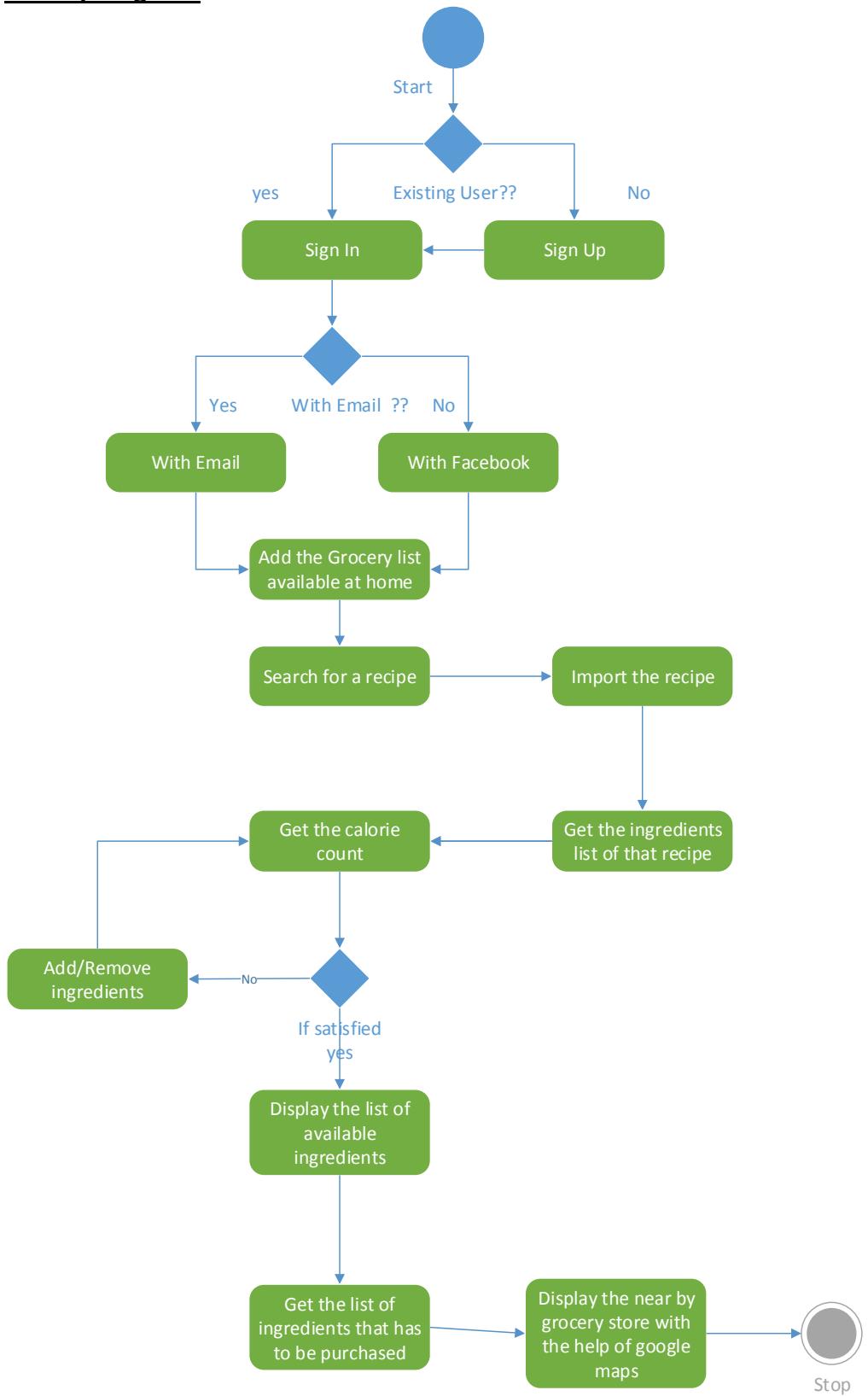


Figure 1.4 Activity Diagram

Sequence diagram:

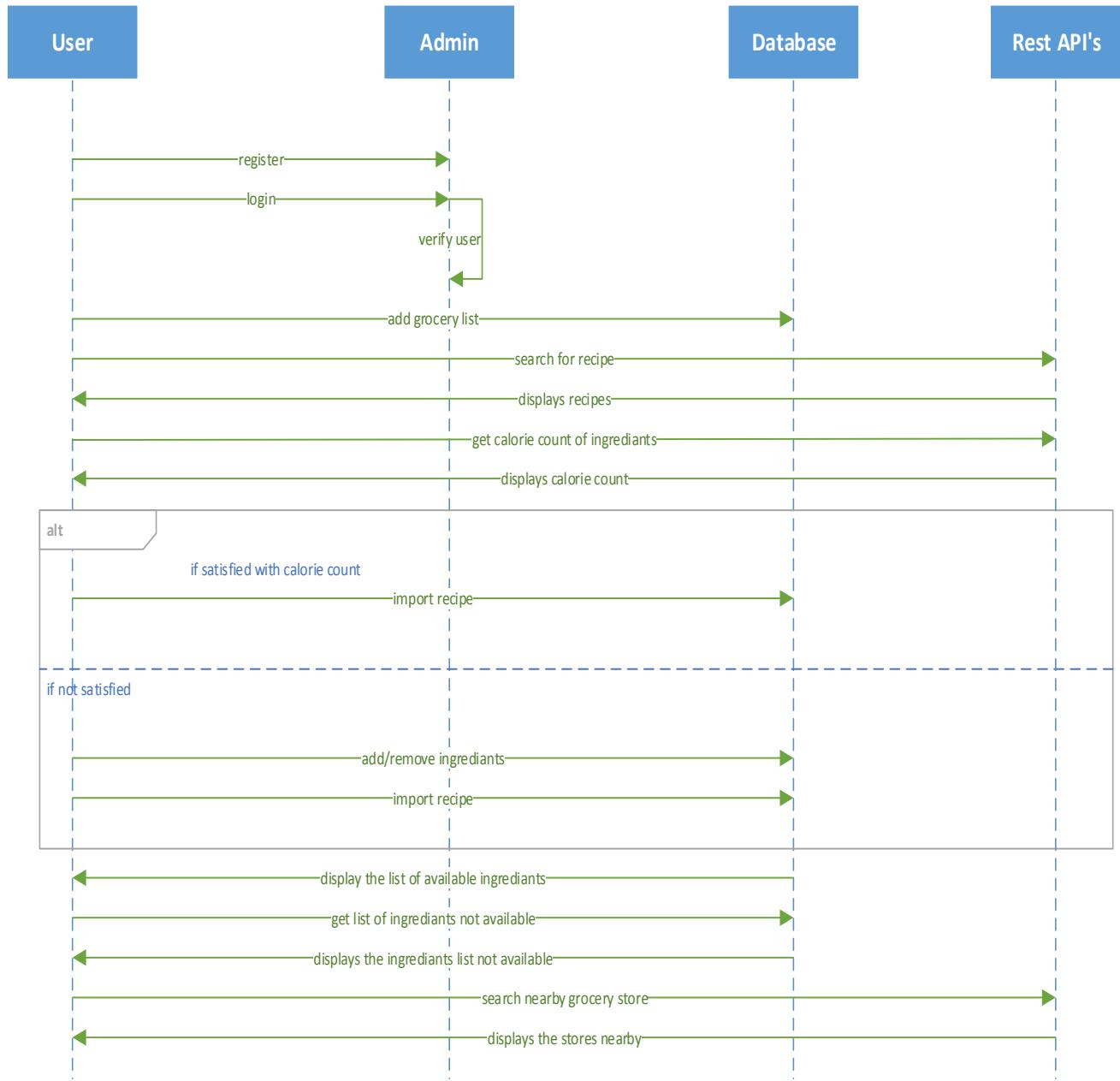


Figure 1.4 Sequence Diagram

Task Planning

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

Increment#1:

- Create a home page for the application.
- Create a function to login with facebook/google plus by using rest services.
- Create a register page.
- Create a feature to search nearby grocery stores using google places api.

Increment#2:

- Add functionality to add groceries to the database
- Add a browser to the app to be able to search recipes.

Increment#3:

- Add functionality to import ingredients from the recipe
- Add functionality to get calorie information using NDB api for the imported ingredients

Increment#4:

- Add functionality to enable the user to add/remove ingredients from the recipe
- Automatically generate shopping list

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.

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.

<https://www.scrumdo.com/projects/project/calorie2grocery2/summary>

Risk Management

Technological & Architectural Requirements:

- ❖ User should be having a smart phone with Android version 4.0 and above.
- ❖ GPS should be turned on to get the current location.

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/>

PROJECT INCREMENT 1

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.

Objective

“Calorie2grocery” is an android application that 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. Our app also displays a list of nearby grocery stores, so that the user can easily go and purchase products.

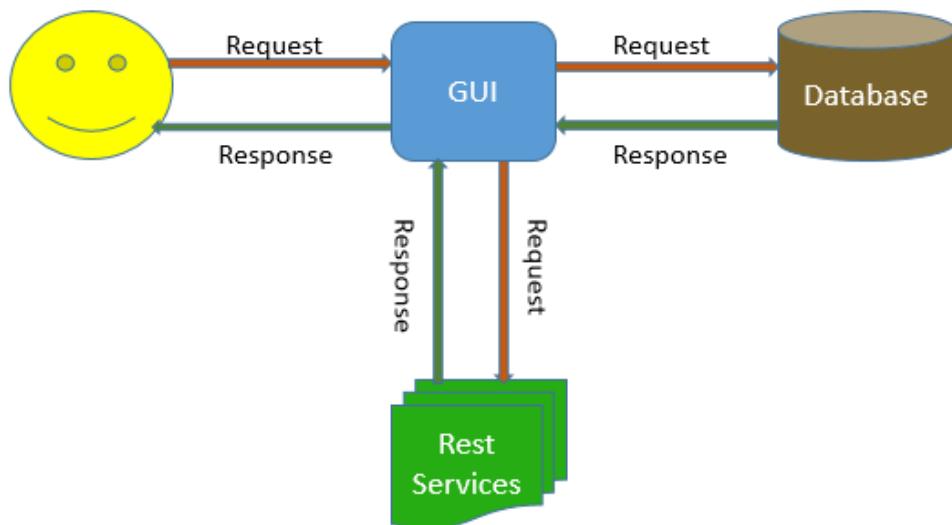
Framework Specification:

Our framework consists of three stages namely GUI, API, Database and a parser. Database handles the storage of data like the user profile information, grocery list, saved recipes etc. GUI editor is developed as mobile application. The GUI connects to the database/API and gets the required data. The data fetched from the API's is parsed by the parser and displayed in the user interface.

Increment#1:

We have built the GUI for login, registration and home pages as part of Increment1. The home page will be modified throughout the project as we add functions to our application. We have established the database connection from the GUI. As part of increment1, we have added functionality to store the user profile information in the database. API is used to display the current location of the user and to display the nearby grocery stores. We have added authentication logic which enables the GUI to communicate with the database and verify the user.

System Architecture:



Application Specifications:

	Tools
Platform	Android 4.0.3
UML Diagrams	Microsoft Visio
Languages	Java, C# , ASP.Net
Database	SQL Lite
Planning	ScrumDo
Version Control System	GIT
Rest Services	Google Places API

Implementation of User Interface:

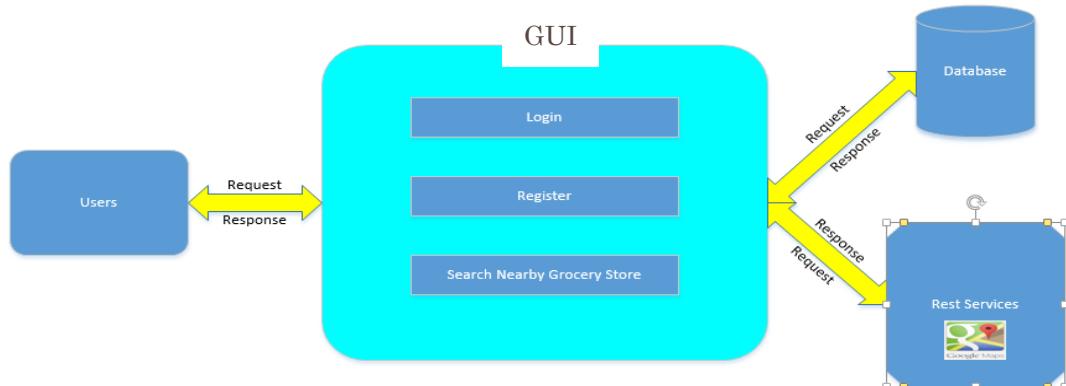
We have developed four layouts as part of UI implementation.

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.

Design of Mobile Client Interface:

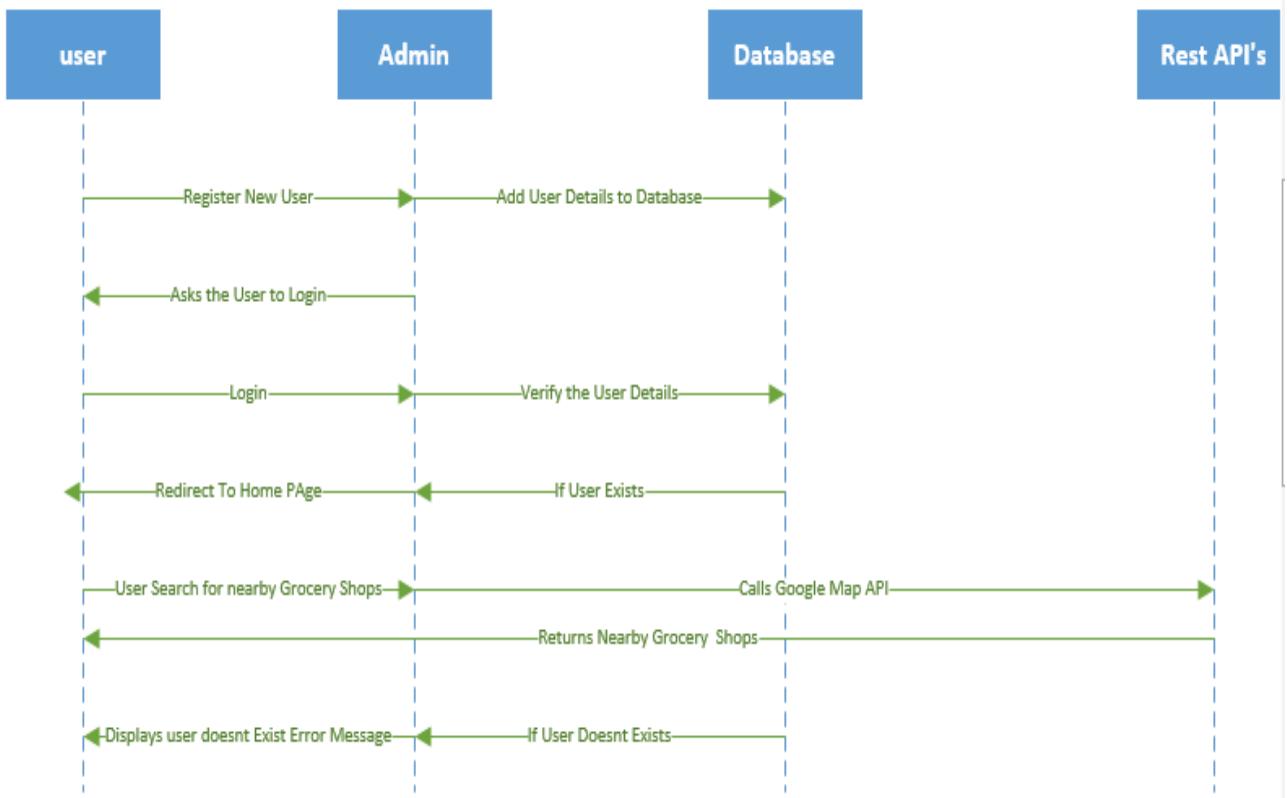


Class Diagram:



Sequence Diagram:

In our sequence diagram, there are five components. When the user is an unregistered or a new user, the GUI asks the user to register with his details. The details of all the users are stored in the data base. If the user is already a registered user, the GUI asks the user to login. When the registered user enters his login credentials, the GUI checks the details from the database, and if he is a valid user, it redirects to the home page. Once, he goes to home page, the user has the options to view the current location and get the list of nearby grocery stores. When the user wishes to search for a nearby grocery store, the GUI communicates with the API and returns the results to the GUI where the user can visualize. The GUI component is displayed as Admin in the following diagram.



Here users interact with the GUI to login or register. When the user logs in, the GUI checks the data from the database and displays Welcome Screen. When the user clicks to Search a nearby grocery store, the GUI interacts with the Rest service API to get the data and displays it on the google map.

Screen Shots of the UI:

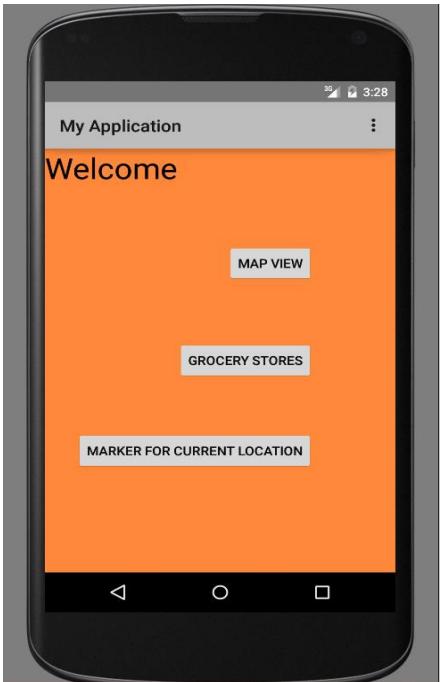
1. Login Page:



- The user interface provides the users with two options:
 - Login for already registered users.
 - Register button for new or unregistered users.

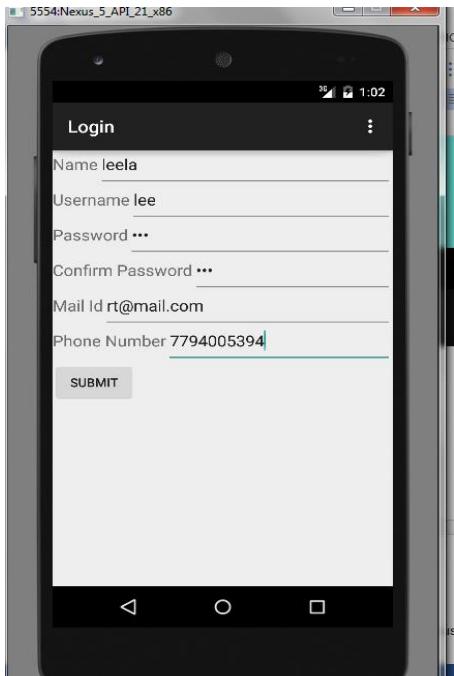


2. Home Page:



- This is the home screen
- It provides user three options:
 - Map View
 - Grocery Stores
 - Marker for current location

3. Registration page for new or unregistered users.



- This is the Registration Page
- User has to provide the following details:
 - Name
 - Username
 - Password
 - Mail Id
 - Phone number

Implementation of existing API:

Our welcome page has three buttons related to location services. The grocery store button displays the nearby grocery stores using the google places API. We have started implementing the API by first fetching the information to display the current location. The button “Marker for current location” implements this logic. Then we have implemented the logic for opening the google maps. The “Map View” button is used for executing this logic.

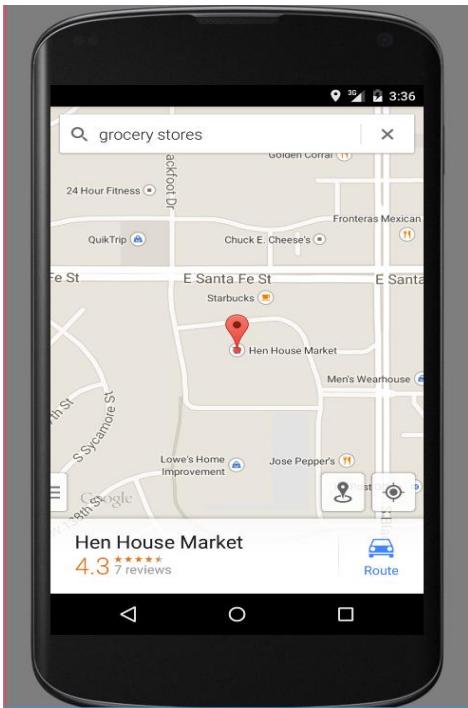
For using google API's, we have installed google play services from android SDK manager. We have created an application in google console to get the API key and API secret key which need to be configured in the manifest file of the android application.

Screen Shots for implementation of API:

Grocery Stores



- This page is displayed when the user clicks on grocery stores. This page has a button which redirects to google maps on clicking it. When the user clicks the button, it displays a google maps page with markers at nearby grocery store.

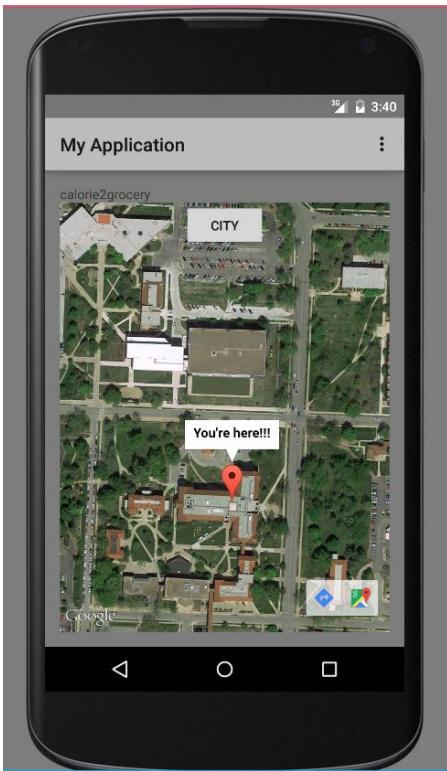


- When the user clicks on grocery stores, it transitions to another page. The screen shot displays the screen how it looks like when we click grocery store.

Marker Button:



- When the user clicks on marker button, it redirects to a google maps page which has a button named city. The below screenshot shows the page that loads on clicking the marker button.
- When the user clicks on CITY button, the user interface displays a marker at the users' current location with a tag "You're here!!!".



- When the user clicks on CITY button, the user interface displays a marker at the users' current location with a tag "You're here!!!".

Database Implementation:

We have established the connection to the SQLite database from the android application. To establish the connection, we have installed the SQLite plugin. We have created the register table in the database. The information that the user inputs from the registration page will be loaded into the database. During login, the username and password entered by the user will be authenticated using the values present in this table.

The below screen shot shows the image of how the values are stored in the database when the user registers. We have created sample users as part of testing.

DB Browser for SQLite - C:/Users/VINAYA/Desktop/user_info

File Edit View Help

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL

Table: Table_Info New Record Delete Record

	name	username	user_pass	cnf_pass	mailid	phoneno
1	rt	t	yv	4256315890	h@d.com	4256315890
2	vinaya	vinayap	user	user	vinaya@gmail.com	9876349089
3	leela	leeladevi	lee	lee	leela@gmail.com	7654897650
4	vyshu	vaishnavi	va56	va56	vaishnavi@gmail...	6578965467
5	sravani	sra	7shrav	7shrav	shravani@gmail...	5237829736
6	ganesh	vignesh	shiva	shiva	parvathi@mice...	9999999999

1 - 6 of 6 Go to: 1

DB Schema

Name

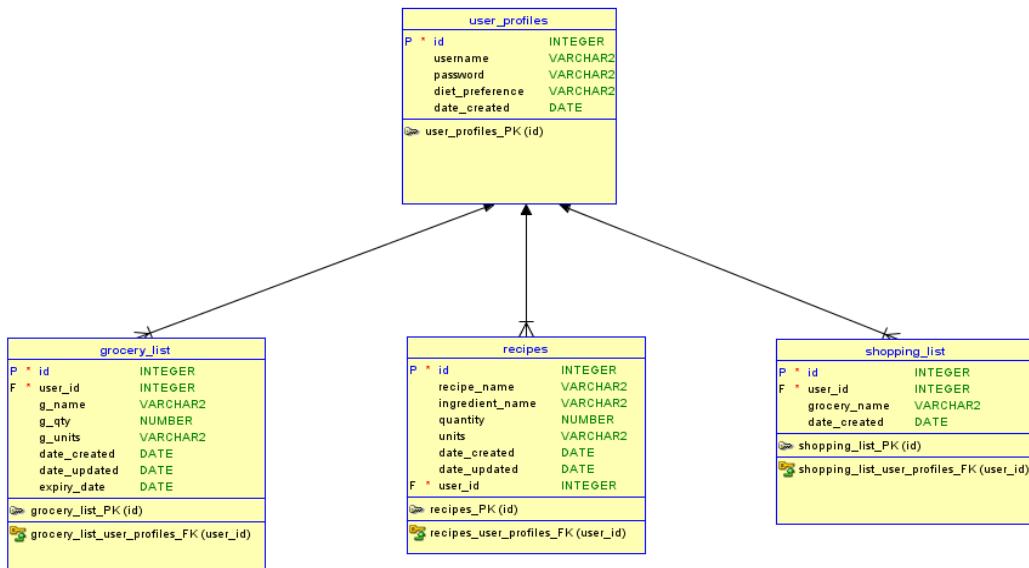
- Tables (2)
 - Table_Info
 - android_metadata
- Indices (0)
- Views (0)
- Triggers (0)

SQL Log Plot DB Schema UTF-8

Data Model

This is the data model for our application. It consists of three database tables.

- a) User_Profiles: Maintains user related data.
- b) Grocery_list: Maintains all grocery related data.
- c) Recipes: Maintains all recipe information.
- d) Shopping_list: Consists of all the items that are generated from all the recipes, when user clicks on generate shopping list.



Design of Test Cases:

We have divided the test cases into two parts:

1. UI Testing
2. Data validation

UI Testing:

As part of UI testing, we have verified that the transitions from page to page happen as expected. Following are the manual test cases for transition of the UI.

Page	Button	Expected	Actual
Login	Register	Registration page	Registration page
Login	Login (with correct username & password)	Welcome Page	Welcome Page
Login	Login (with incorrect credentials)	Error Message, ask to re-enter credentials	Error Message, ask to re-enter credentials
Registration	Submit	Thank you page to show successful registration	Thank you page to show successful registration
Registration Success Page	Click to Login	Login page	Login page
Welcome Page	Grocery Stores	Grocery layout	Grocery layout
Grocery	Click to view nearby grocery stores	Google maps	Google maps
Welcome Page	Marker for current location	Google maps	Google maps

Data Validation Tests:

For Increment#1, we have two test cases for data validation.

- 1) We have verified that the data entered in the register table matches with the user input.
- 2) **User authentication:**

Positive test case: Upon entering the correct user credentials, the authentication should succeed and the GUI need to transition to the Welcome page.

Negative test case: If we enter incorrect username and password, the app should throw an error message.

Scrum Do :

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/organization/umkc240/dashboard>

Following are the screen shots of the scrum board for all the increments:

Increment#1

The ScrumDo board for Increment#1 displays the following tasks:

- Todo:** #4 As a developer i need to develop a welcome/home page for my application. (Story4, 3 comments)
- Doing:** #5 As a user i need to use google maps to get the near by grocery stores. (Story5, 0 comments)
- Reviewing:** #2 As a developer i need to develop a user interface for login so that users can sign in to my application. (Story2, 0 comments)
- Reviewing:** #3 As a developer i need to develop a user interface for register page so that users can register for my application. (Story3, 0 comments)
- Reviewing:** #19 As a developer, i need to validate the data entered in the database. (TestStory2, 0 comments)
- Reviewing:** #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, 0 comments)
- Reviewing:** #20 As a developer i need to verify that the authentication logic is working as expected. (TestStory3, 0 comments)
- Done:** #1 As a developer i need to install android studio so that i can develop a application. (Story1, 0 comments)
- Done:** #6 As a develop i need to establish a connection to the database so that i can store the user information. (Story6, 3 comments)

Increment#2:

The ScrumDo board for Increment#2 displays the following tasks:

- Todo:** #7 As a user i need a grocery list so that i can keep track of the groceries available at home. (Story7, 3 comments)
- Todo:** #8 As a user i need a browser in the app so that i can search for recipes. (Story8, 3 comments)
- Doing:** #17 As a developer i need to develop the UI for grocery list. (0 comments)
- Reviewing:** #14 As a developer i need to develop the UI for searching the recipes. (0 comments)

Increment#3:

Increment#3 - Feb. 19, 2015 - April 8, 2015			Filter Board	
Todo		Doing	Reviewing	Done
#12 As a user i should be able to add/remove ingredients from the recipe				
0 Comments - Tasks	3			
#10 As a user i should be able to import the ingredients required for preparing a recipe				
0 Comments - Tasks	3			
#9 As a user i should be able to mark recipes for future reference				
0 Comments - Tasks	3			

Increment#4:

Increment#4 - April 9, 2015 - April 29, 2015			UMKC CALORIE2GROC Filter Board	
Todo		Doing	Reviewing	Done
#16 As a developer i need to develop the UI for showing the calorie count of the ingredients in the recipe				
0 Comments - Tasks	3			
#15 As a developer i need to develop the UI for getting the shopping list				
0 Comments - Tasks	3			
#13 As a user i need a shopping list so that i wont miss something needed for preparing the recipe				
0 Comments - Tasks	3			
#11 As a user i should get the calorie count for each ingredient of the recipe				
0 Comments - Tasks	3			

Implementation Status Report:

Work Completed:

1) GUI

Description:

Implementation of login & Registration pages along with database connection. UI implementation of these pages include page transition.

Responsibility: Vinaya , Leela

Time taken: 8 Hours

Contribution:

Database Connectivity: Vinaya (40%) , Leela (30%)

Division of tasks:

Vinaya : Creating the UI for login page and developing the logic for authentication.

Leela: Creating the UI for registration page and developing the logic to load the user input into database.

Testing: Leela(10%) Sravani(10%) Vaishnavi (10%)

Testing includes validation of data and ensuring that the page transitions are performed correctly.

2) API & Welcome Page

Description:

Implementation of google places api to get the nearby grocery stores. Creation of welcome page/home page that has all the features of the app.

Responsibility: Sravani , Vaishnavi

Time taken: 10 Hours

Contribution:

Database Connectivity: Sravani (40%) , Vaishnavi (30%)

Division of tasks:

Sravani : Creating the UI for welcome page and developing the logic for fetching data from google places api.

Vaishnavi: Implemented the marker logic to show the current location.

Testing: Vaishnavi(10%) Vinaya(10%) Leela (10%)

Issues/Concerns:

The android studio is very slow. Hence we had to invest more time than normally required for developing applications. Also there is less support online for android studio compared to eclipse for developing android applications.

PROJECT INCREMENT 2

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 #2

We have developed an application which helps the user do the following:

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.

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
New API's	Grocery list Import 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=9fcd3aa5746d2a423a350cee3ea4d57d

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

New Services:

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

Grocery List: This API is used to maintain the grocery lists. We have implemented two methods in this API.

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 a 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

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

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/vinaya2/RecipeIngredient/Service1.svc/data/

Use Case Template for Adding grocery items:

Use Case Name	Add grocery items
Primary Actor	User
Pre-Conditions	User has the app installed on his/her mobile. User should be logged into the app.
Guarantee(Post Conditions)	Grocery items are stored in the database
Main Success Scenario	User clicks on “Add Grocery” button in the home page. The system transitions the UI to the “Add Grocery” layout. User enters the grocery item name, quantity, selects the measuring unit from the drop down menu and clicks on “Add”. System adds the grocery information to the database. System sends a notification that the grocery item is successfully added to the database. User can go back to the home page by clicking on return button.
Extensions	3a. User enters incorrect value for quantity. Systems throws an error asking the user to add proper value. 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. 5a. User can add as many groceries items as he needed to by following steps 3 & 4 and then go back to the home page.

Use Case Template to View the grocery list:

Use Case Name	View grocery list
Primary Actor	User
Pre-Conditions	User has the app installed on his/her mobile. User should be logged into the app.
Guarantee(Post Conditions)	User can view the list of groceries. User can update the quantity of grocery items.
Main Success Scenario	<p>User clicks on “View Grocery List” button in the home page.</p> <p>The system transitions the UI to the “View Grocery” layout.</p> <p>System fetches the grocery list of the user from the database.</p> <p>4. System displays the grocery list to the user.</p> <p>5. User can update the quantity of any of the grocery items.</p> <p>6. System updates the database to reflect the changes made by the user.</p> <p>7. System sends a notification saying the update is successful.</p>
Extensions	None

Use Case Templates for searching and importing a recipe:

Use Case Name	Search & import a recipe
Primary Actor	API
Pre-Conditions	The user must be a valid user to use the app
Post-Conditions	Retrieves the results for the search

Main Success Scenario	If the user wants to search for a recipe, then, the user enters the keyword to search. User clicks on the search recipes. The system displays ten recipe names. If the user wants to see the ingredient list, then the user can click on the recipe name. System displays ingredient list along with quantity. If the user wants to import the ingredient list then click on import. The system stores the ingredient list in the database.
Extensions	None
Special Requirements	None

UML CLASS DIAGRAM:

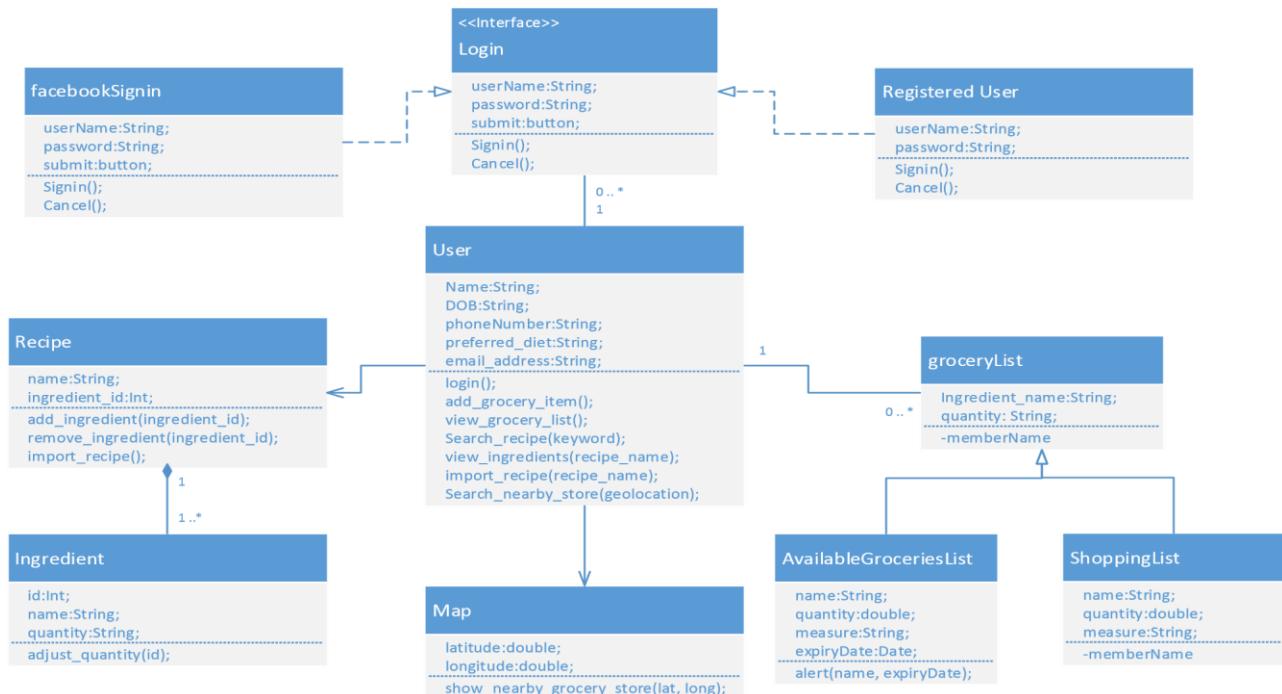
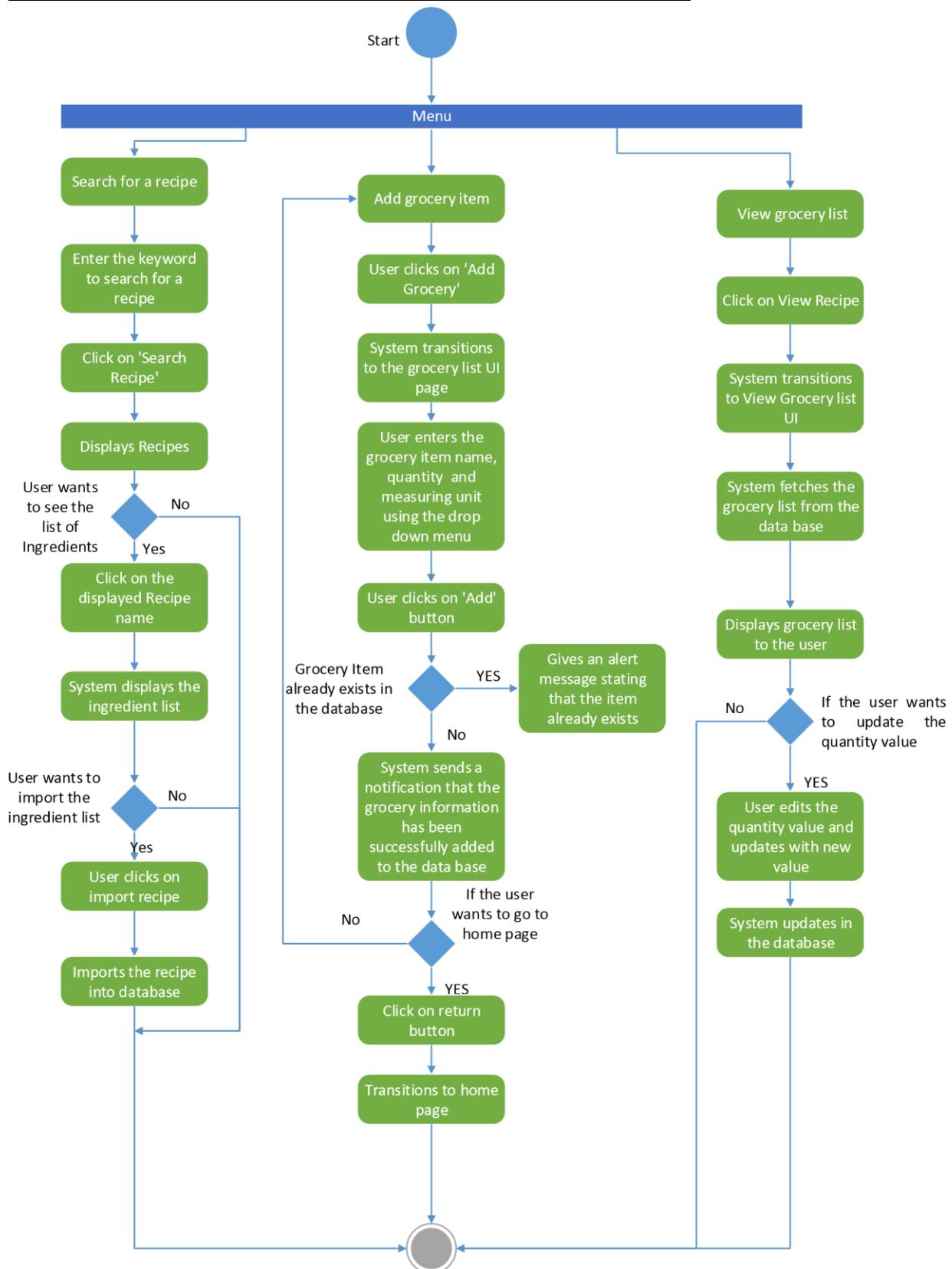


Figure 1: UML Class Diagram of the entire system

Activity Diagram for the functionality implemented in Increment 2:



Sequence Diagram:

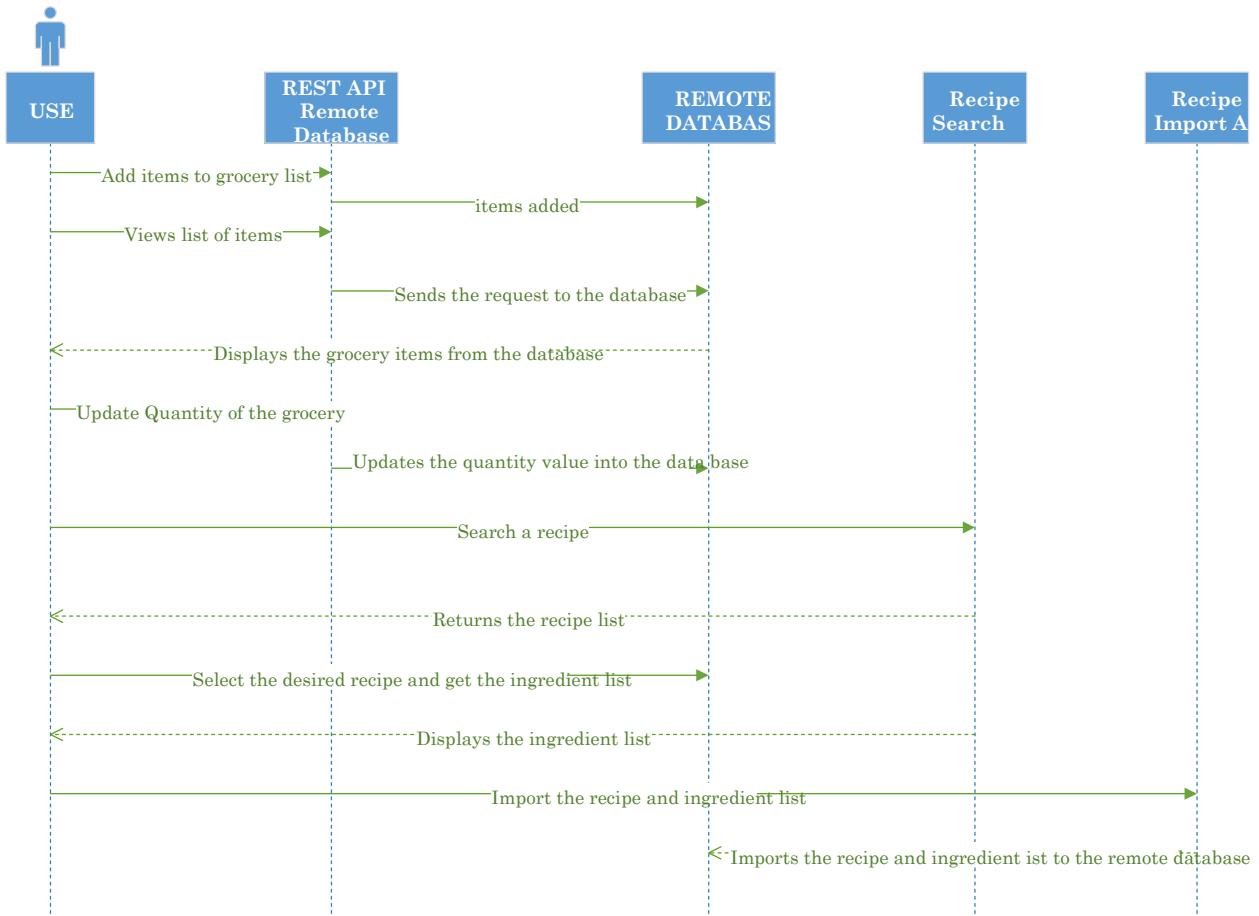


Figure 3: Sequence Diagram

Increment#2 Implementation:

Implementation of User Interface:

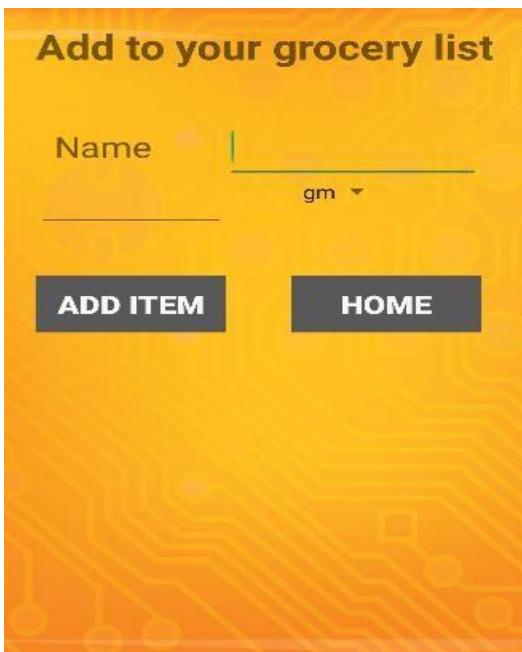
We have developed four layouts as part of UI implementation.

- Home Page
- Add Grocery
- View Grocery List
- Search Recipe

Screenshots:



Screenshot 1: Home Page page.



Screenshot 2: Add Grocery page

Home Page

User is provided with three buttons:

Add Grocery Button: User can click this button If he wants to add grocery items to his list.

View Grocery List: If user wants to view the grocery list, he/she could click on this button.

Search Recipe button: If user wants to search for a recipe, then he/she could click on this button.

Add Grocery

When the user clicks on Add grocery button, the UI is transitioned to add grocery item

User enters the grocery item, name, quantity.

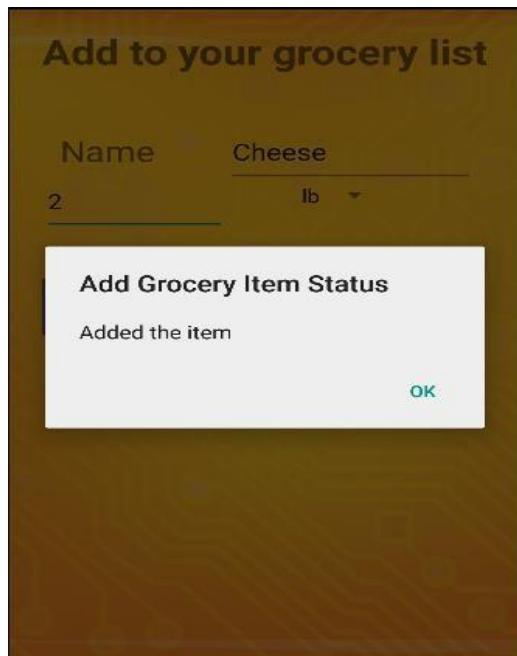
User can select the measuring unit from the drop down menu

Technical Details:

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



Screenshot 3: Add Grocery page



Screen shot 4: Add Grocery notification

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.

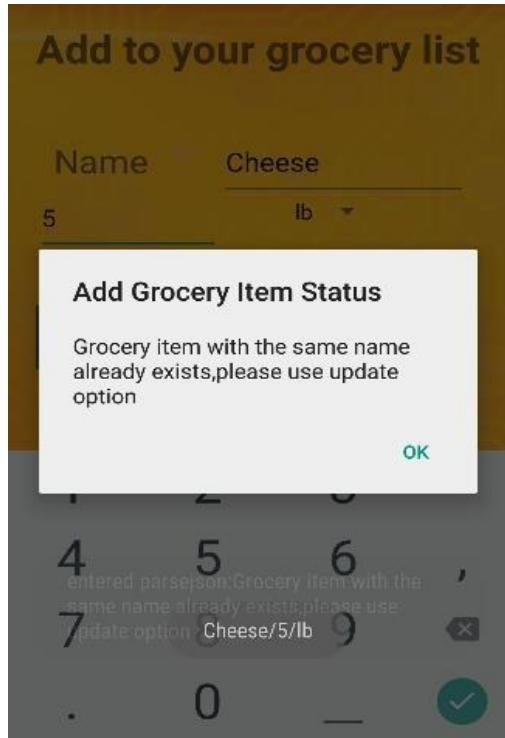
After user clicks ok the add grocery UI is refreshed, the user can add more grocery items or use the home button to go to the home page.

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.



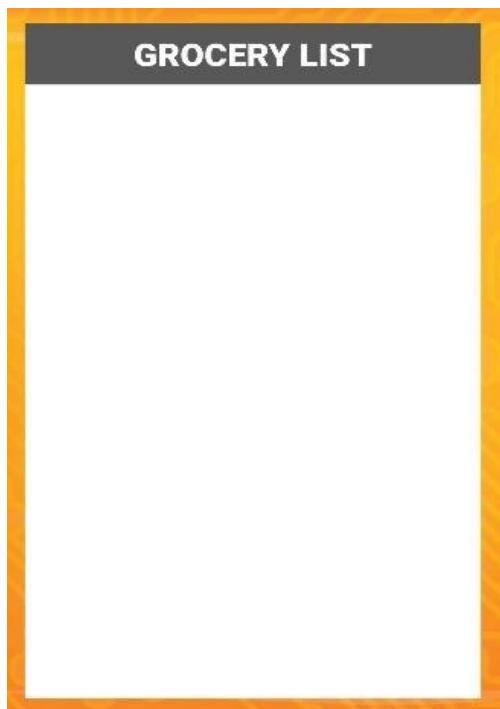
Screen shot 5: Add Grocery update alert

Add Grocery

If the user tries to add the same name again, then the system throws an alert saying “a grocery item with the same name already exists in the list.

Technical Details:

Before insert, we are implemented a logic to check if a record is already present in the grocery_list table with the same name.



View Grocery List

If the user clicks on “View grocery list” button in home page then the UI transitions to this page.

Technical Details:

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

Screen shot 6: View Grocery list



View Grocery List

Once the user clicks on “Grocery List” button, the web page is loaded into the web view.

Technical Details:

Used JQuery to implement the client side logic.

Screen shot 7: View Grocery list

GROCERY LIST		
View Grocery List		
Name	Quantity	Units
Onion	200	gm
honey	20	gm
butter	30	gm
banana	5	kg
orange	100	gm
lime	20	gm
milk	1	l
spinach	1	lb
grapes	2	lb
pineapple	30	gm
potato	5	lb

View Grocery List

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

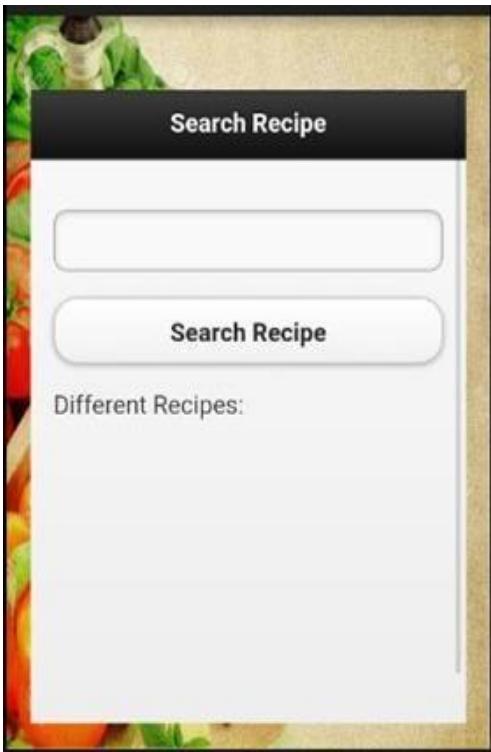
Technical Details:

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.

Screen shot 8: View Grocery list



Screenshot 9: Search Recipe

Search Recipe

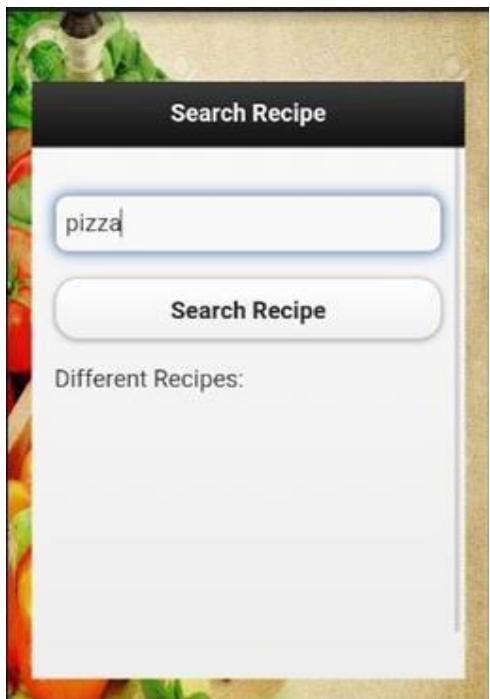
The user gets the search recipe page when he clicks the search recipe button in home page.

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.



Screenshot 10: Search Recipe

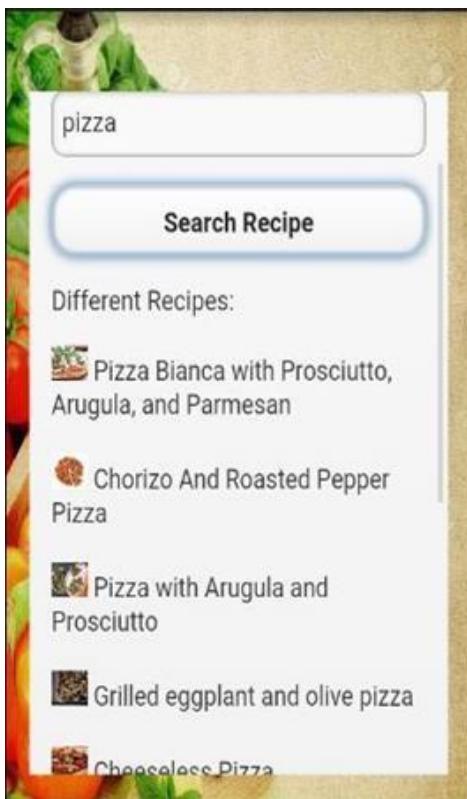
Search Recipe

User enters pizza as a search key to get various recipe on pizza

User clicks the search recipe button

Technical Details:

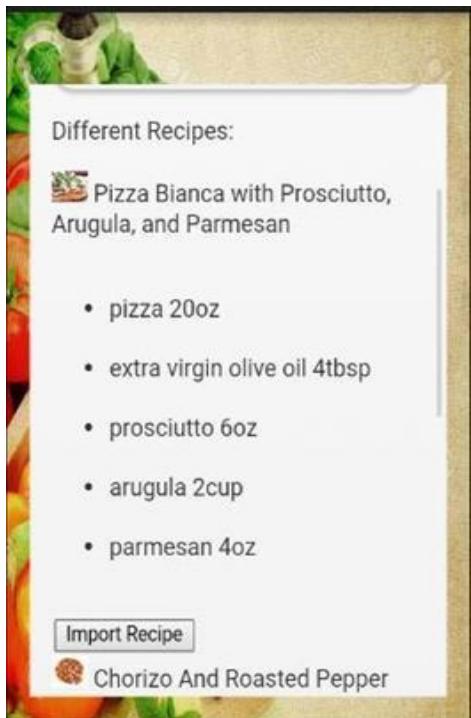
We are using the recipe & diet api to fetch the recipe information based on keyword search



Screenshot 11: Recipe List

Search Recipe

② System displays the list of recipes for pizza.



Screenshot 12: Ingredient List

3) Database Implementation

For this increment we created a database “calorie2grocery” in SQL server database in remote server. In this database we created three tables. They are:

User_profile

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

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.

Recipe

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

User_profile

The screenshot shows the SQL Server Management Studio interface. On the left, the Object Explorer displays the database structure, including the 'Calorie2Grocery' database with its tables (User_profile, grocery_list, etc.). On the right, the Results tab shows the data for the User_profile table:

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

Fig: User_profiles table containing user data

Grocery_list Insert Procedure:

The screenshot shows the SQL Server Management Studio interface with the 'Object Explorer' on the left and the 'Script Editor' on the right. The 'Script Editor' contains the code for the 'Add_Grocery' stored procedure:

```
SET QUOTED_IDENTIFIER ON
GO
CREATE Procedure [dbo].[Add_Grocery] (@username varchar(50), @name varchar(50), @quan float, @measure varchar(10), @msg int out)
AS
BEGIN
    DECLARE @id int;
    DECLARE @date date;
    DECLARE @dtd date;
    DECLARE @uid int;
    Declare @exist varchar(50);
    SELECT @gid=CONVERT(varchar, COALESCE(MAX(g_id), 0)+1),
    @idt=CONVERT(VARCHAR(10), GETDATE(), 20),
    @edt=CONVERT(varchar(10), GETDATE() + 7, 20)
    FROM grocery_list;

    select @exist=g_name from grocery_list where g_name=@name;

    select @uid=user_id from user_profile where user_name=@username;

    IF (@exist IS NOT NULL)
        set @msg='2';
    else
        begin
            IF (@name IS Not NULL)
                Begin
                    Insert into grocery_list(g_id, user_id, g_name, g_quantity, g_measure, g_load_dt, g_update_dt, g_expiry_dt)
                    values(@gid, @uid , @name , @quan, @measure , @idt, @idt , @edt);
                    set @msg='0';
                End;
            else
                SET @msg = '1';
        END;
    END;
    GO
```

The above screenshot shows the logic which is used to validate the grocery items before inserting into the database. If the grocery item is already present in the database, it will give an output message stating that particular grocery item already exists in the data base.

Grocery List table:

Results

	g_id	user_id	g_name	g_quantity	g_measure	g_load_dt	g_update_dt
1	1	1	Onion	200	gm	2015-03-07	2015-03-07
2	2	1	honey	20	gm	2015-03-07	2015-03-07
3	3	1	butter	30	gm	2015-03-07	2015-03-07
4	4	1	banana	5	kg	2015-03-14	2015-03-14
5	5	1	orange	100	gm	2015-03-14	2015-03-14
6	6	1	lime	20	gm	2015-03-14	2015-03-14
7	7	1	milk	1	l	2015-03-14	2015-03-14
8	8	1	spinach	1	lb	2015-03-14	2015-03-14
9	9	1	grapes	2	lb	2015-03-14	2015-03-14
10	10	1	pineapple	30	gm	2015-03-14	2015-03-14
11	11	1	potato	5	lb	2015-03-14	2015-03-14
12	12	1	apple	20	gm	2015-03-14	2015-03-14
13	13	1	Beans	100	gm	2015-03-15	2015-03-15
14	14	1	coffee	8	gm	2015-03-17	2015-03-17
15	15	1	papaya	2	lb	2015-03-17	2015-03-17
16	16	1	mango	250	gm	2015-03-17	2015-03-17
17	17	1	Cheese	2	lb	2015-03-18	2015-03-18

Fig : Table containing the grocery item information

Recipe

Results

	recipename	ingredientlist
1	Roasted Pepper, Olive, Feta, and Artichoke Pizza	pizza20oz.extra virgin olive oil4tbsp.prosciutto6oz,arugul...
2	Roasted Pepper, Olive, Feta, and Artichoke Pizza	pizza20oz.extra virgin olive oil4tbsp.prosciutto6oz,arugul...
3	Roasted Pepper, Olive, Feta, and Artichoke Pizza	chorizo2oz,roasted red peppers0.5cup,cheese pizza14...
4	Roasted Pepper, Olive, Feta, and Artichoke Pizza	pizza20oz.extra virgin olive oil4tbsp.prosciutto6oz,arugul...
5	Pizza Bianca with Prosciutto, Arugula, and Parme...	pizza20oz.extra virgin olive oil4tbsp.prosciutto6oz,arugul...
6	Pizza Bianca with Prosciutto, Arugula, and Parme...	pizza20oz.extra virgin olive oil4tbsp.prosciutto6oz,arugul...
7	Pizza Bianca with Prosciutto, Arugula, and Parme...	pizza20oz.extra virgin olive oil4tbsp.prosciutto6oz,arugul...
8	Spinach-Cucumber-Celery Juice	spinach4oz,cucumber1cucumber,celery1stalk,
9	dfv	sdfgv
10	The Ultimate Burger	skirt steak2.5lb,burger buns1bun,burger buns1bun,
11	dfv	sdfgv
12	Keema With Peas	piece ginger1piece,garlic2clove,onion1large,yogurt2lbs...
13	Keema With Peas	piece ginger1piece,garlic2clove,onion1large,yogurt2lbs...
14	Deep Fried Fish Bones	fish8small,vegetable oil4cup,
15	hg	hgv
16	Pizza Bianca with Prosciutto, Arugula, and Parme...	pizza20oz.extra virgin olive oil4tbsp.prosciutto6oz,arugul...
17	Pizza Bianca with Prosciutto, Arugula, and Parme...	pizza20oz.extra virgin olive oil4tbsp.prosciutto6oz,arugul...
18	Stir-Fried Luffa Gourds with Eggs	vegetable oil1tbsp,garlic2tsp,luffa1lb,eggs2large,fish sa...
19	Pizza Bianca with Prosciutto, Arugula, and Parme...	pizza20oz.extra virgin olive oil4tbsp.prosciutto6oz,arugul...
20	recipename	ingredients

Fig: Table containing the imported recipe information

Design Of Unit Test Cases

UI Testing:

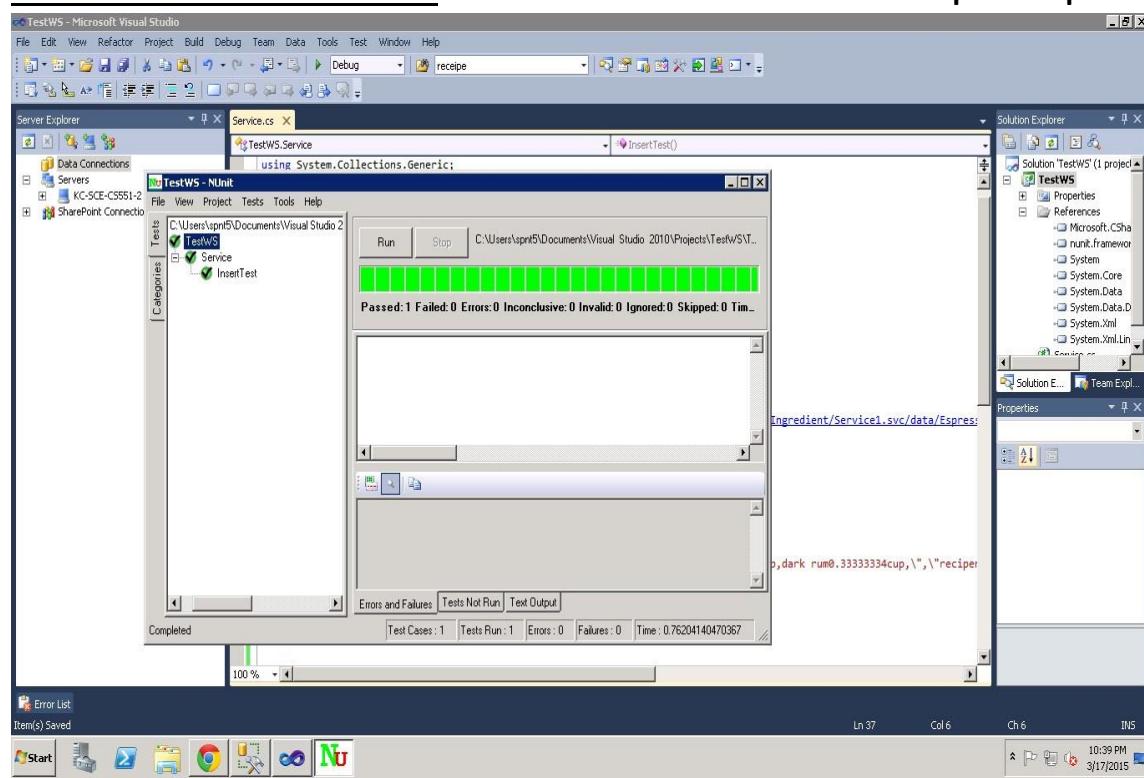
The following test cases has been written for UI testing and the actual output is validated with expected output. All of the below test cases which have been written obtained positive results.

Page	Button	Expected	Actual
Home	Search Recipes	Open browser for search recipe	Open browser for search recipe
	View Groceries list	Display the My groceries page	Displays the My groceries page
	Add Groceries	Display the Add groceries page	Display the Add groceries page
Search recipe	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
	Import recipe	The ingredients list of the selected recipe should be added to the remote database	The ingredients list of the selected recipe is added to the remote database
My Groceries	Add groceries	The list of items should be added to the database	The list of items are added to the database
	View my grocery list	Should display the list of grocery items present	Displays the list of grocery items present
	Update the quantities of an item	Should allow us to update the quantity of an item	Allowing us to update the quantity of an item

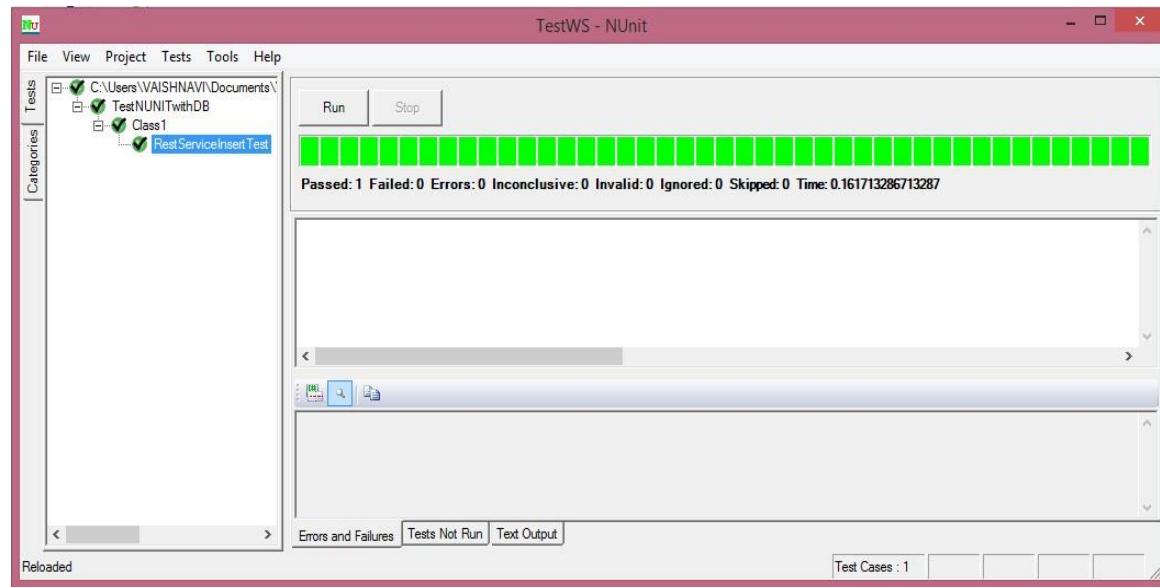
Data Validation:

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:



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/122931#>

Screenshots for Increment#2 user stories:

All the user stories planned for increment #2 have been successfully implemented.

#35 As a developer, i need to test the import recipe API to ensure that the recipe information is inserted appropriately.				
Done Tasks 0 Comments				
#34 As a developer, i need to test the API for adding groceries to the database to validate the data.				
Done Tasks 0 Comments				
#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.				
Done Tasks 0 Comments				
#32 As a system, i should display the UI for viewing the grocery list				
Done Tasks 0 Comments savani				
#31 As a system,I should display the UI for searching recipes.				
Done Tasks 0 Comments				
#30 As a system, i should display the UI for adding groceries to the database.				
Done Tasks 0 Comments savani				
#29 As a system,i should be able to fetch the grocery items data from the database and display it to the user.				
Done Tasks 0 Comments savani				
#26 As a user i should be able to import my favorite recipes for customization.				
Done Tasks 0 Comments				
#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				
Done Tasks 0 Comments Story7				3
#8 As a user i should be able to search for recipes based on some keywords.				
Story8				
Done Tasks 0 Comments Story8				3
#22 As a user i should be able to add grocery items available at home to a list in the application				
Done Tasks 0 Comments savani				
#25 As a user, i should be able to view the list of ingredients required for a selected recipe.				
Done Tasks 0 Comments Story#25 savani				
#21 As a system, i should be able to store the grocery information provided by the user in the database				
Done Tasks 0 Comments savani				
#23 As a system, i should display a list of recipes based on the keywords given by the user in the search box.				
Story#23				
Done Tasks 0 Comments				
#24 As a developer i need to search for a suitable API for recipe search for my application				
Done Tasks 0 Comments Story#24				

Increment #3 user stories

Increment#3 - March 19, 2015 - April 8, 2015				
Todo	Doing	Reviewing	Done	
#27 As a user I should be able to delete some of the entries in the grocery list 0 Comments - Tasks	#28 As a system, I should send notifications to the user about the expiry date of the grocery items. 0 Comments - Tasks			
#12 As a user I should be able to add/remove ingredients from the recipe 0 Comments - Tasks	#9 As a user I should be able to update the quantity of the grocery items 0 Comments - Tasks			

Increment #4 user stories

Todo	Doing	Reviewing	Done
#16 As a developer I need to develop the UI for showing the calorie count of the ingredients in the recipe 0 Comments - Tasks			
#15 As a developer I need to develop the UI for getting the shopping list. 0 Comments - Tasks			
#13 As a user I need a shopping list so that I wont miss something needed for preparing the recipe 0 Comments - Tasks			
#11 As a user I should get the calorie count for each ingredient of the recipe 0 Comments - Tasks			

User Story#	User Story	Hours Worked	Team Member	Status
30	As a system, I should create the UI for adding groceries to the database.	3	Sravani	Done
22	As a user I should be able to add grocery items available at home to a list in the application	8	Sravani	Done
21	As a system, I should be able to store the grocery information provided by the user in the database	8	Sravani	Done
24	As a developer I need to search for a suitable API for	10	Vaishnavi, Vinaya	Done

	recipe search functionality in my application			
31	As a system, I should display the UI for searching recipes	8	Vinaya, Vaishnavi	Done
8	As a user I should be able to search for recipes based on some keywords.	12	Vinaya, Vaishnavi	Done
23	As a system, I should display a list of recipes based on the keywords given by the user in the search box.	10	Vinaya, Vaishnavi	Done
25	As a user, I should be able to view the list of ingredients required for a selected recipe.	4	Sravani	Done
26	As a user I should be able to import my favorite recipes for customization.	22	Leela	Done
33	As a system, I should store the details like ingredients, quantity, and preparation method etc. for the recipes imported by the user in the database.	20	Leela	Done
32	As a system, I should display the UI for viewing the grocery list	2	Sravani	Done
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	5	Sravani	Done
29	As a system, I should be able to fetch the grocery items data from the database and display it to the user.	12	Sravani, Vaishnavi	Done

34	As a developer, I need to test the add groceries API to validate the data inserted into the database	2	Vaishnavi	Done
35	As a developer , I need to test the import recipe api to ensure that the recipe information is inserted appropriately	2	Vinaya	Done

GIT Hub URL:

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

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

Bibliography:

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

API: http://api.bigoven.com/recipes/47725api_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.

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...

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

API:https://api.edamam.com/search?q=chicken&app_id=51aba909&app_key=9fcd3aa5746d2a423a350cee3ea4d57d

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.

PROJECT INCREMENT 3

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

	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=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

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](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](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/Recipelngrredient/Service1.svc/data/](http://kc-sce-cs551-2.kc.umkc.edu/aspnet_client MPG12/vinaya2/Recipelngrredient/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/gjh](http://kc-sce-cs551-2.kc.umkc.edu/aspnet_client MPG12/Leela/ViewRecipeList/Service1.svc/viewRecipeList/gjh)

Use Case Template to update grocery item:

Use Case Name	Update grocery items
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)	Quantity of grocery items is updated in 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. 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	<ol 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

	<ol style="list-style-type: none"> 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 button2. 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 button3. User gets all the recipe name, ingredient list , quantity and label from the database.4. User can add, remove, update the ingredient , quantity and label5. 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:

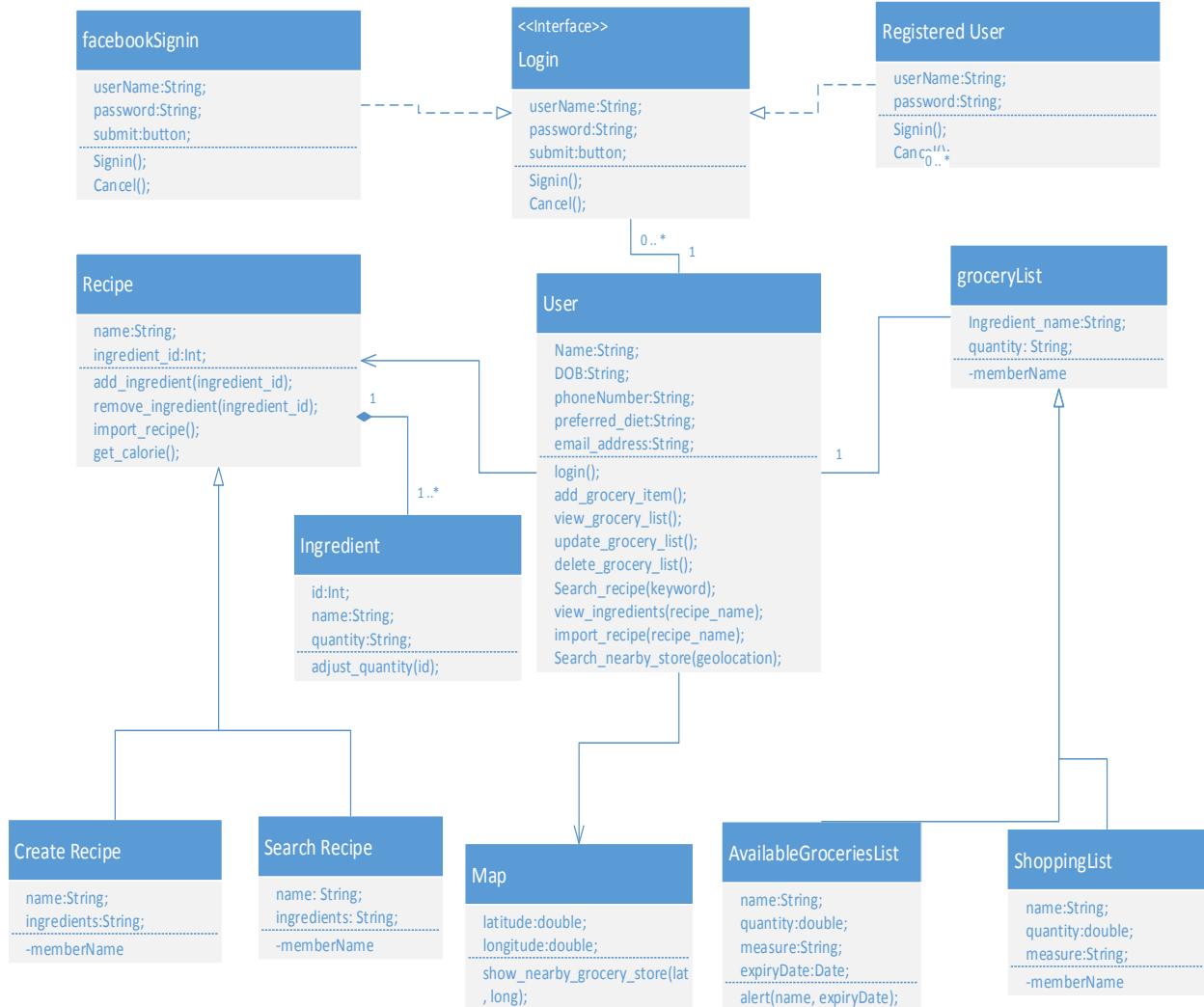
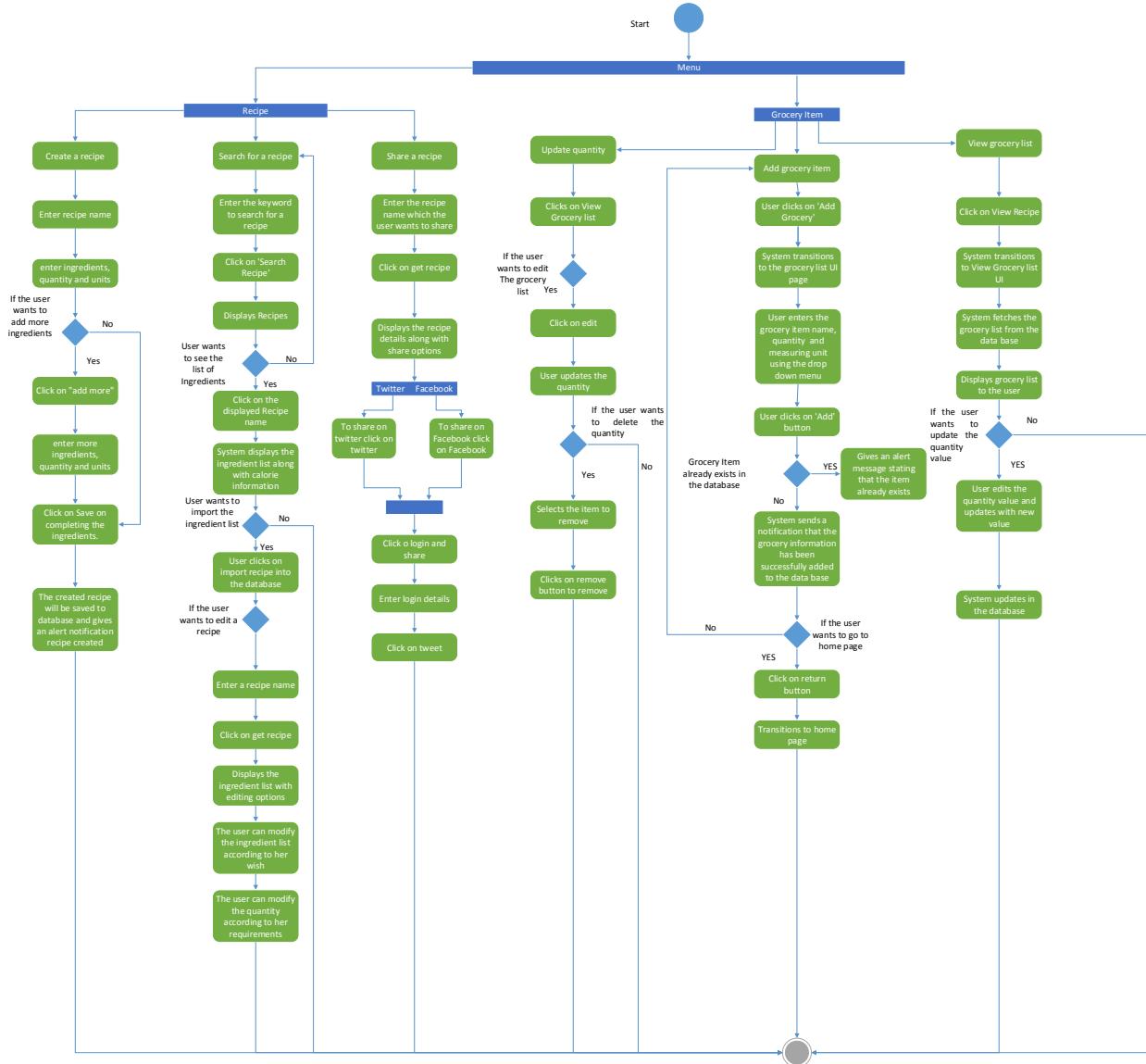


Figure 1: UML Class Diagram of the entire system

Activity Diagram for the functionality implemented in Increment 3:



Sequence Diagram:

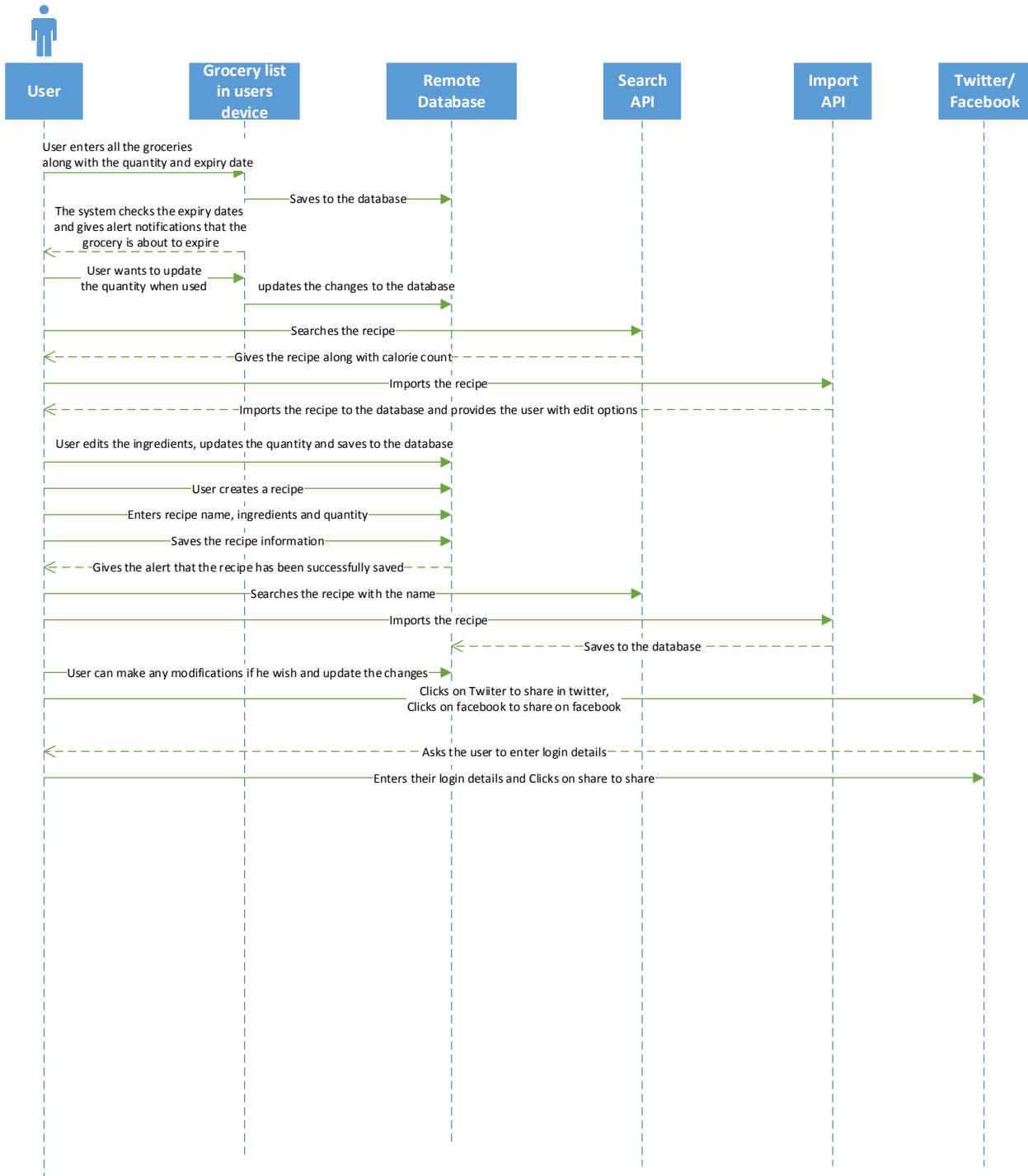


Figure 3: Sequence Diagram

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:



Screenshot 1: Home Page

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 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

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.

Screenshot 3: View/Edit Grocery page

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

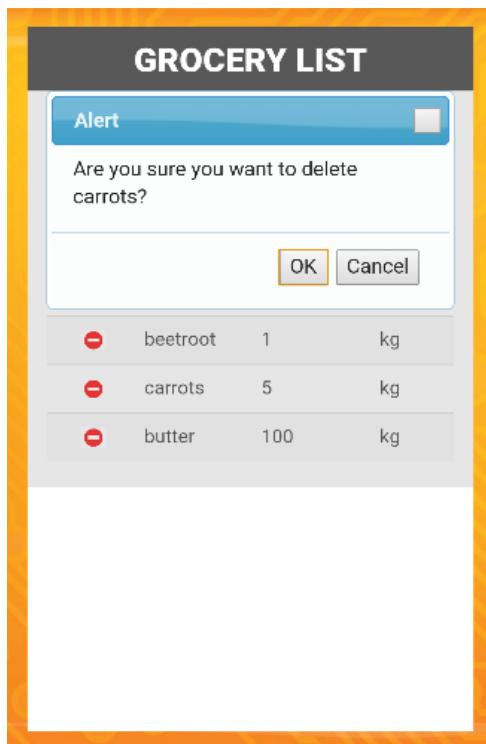
View/Edit Grocery List

- 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



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

Screen shot 6: Delete Grocery

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.

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	
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.

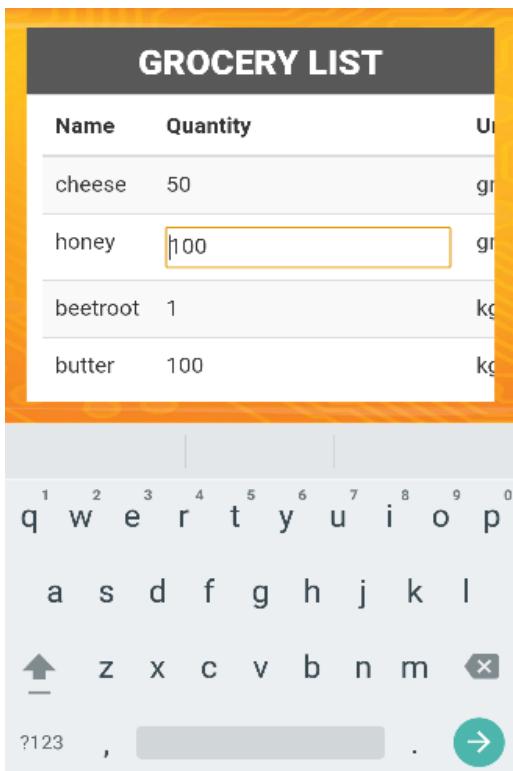
Screen shot 7: Delete Grocery

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

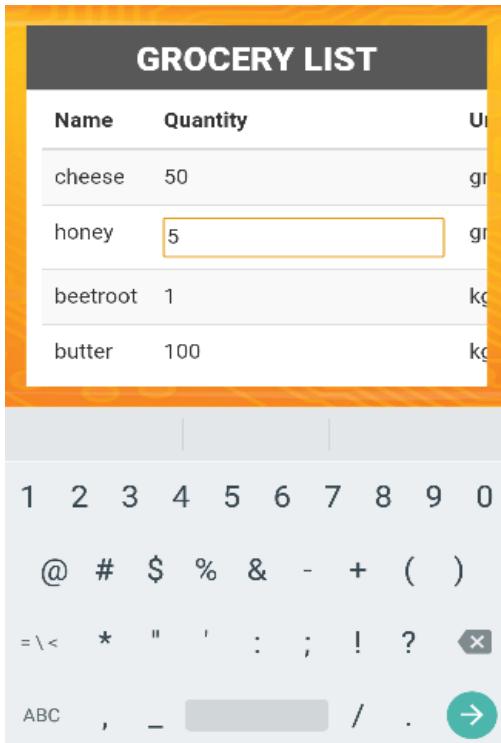
View Grocery List

- 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.

Screen shot 8: View Grocery list



Screenshot 9: Update Grocery



Screenshot 10: Update Grocery

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.

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.

GROCERY LIST		
Name	Quantity	Units
cheese	50	gm
honey	5	gm
beetroot	1	kg
butter	100	kg

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.

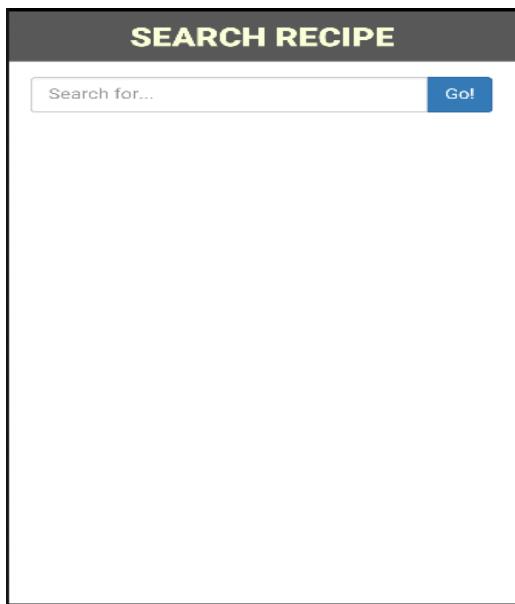
Screenshot 11: Update Grocery



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 12: Recipes Page



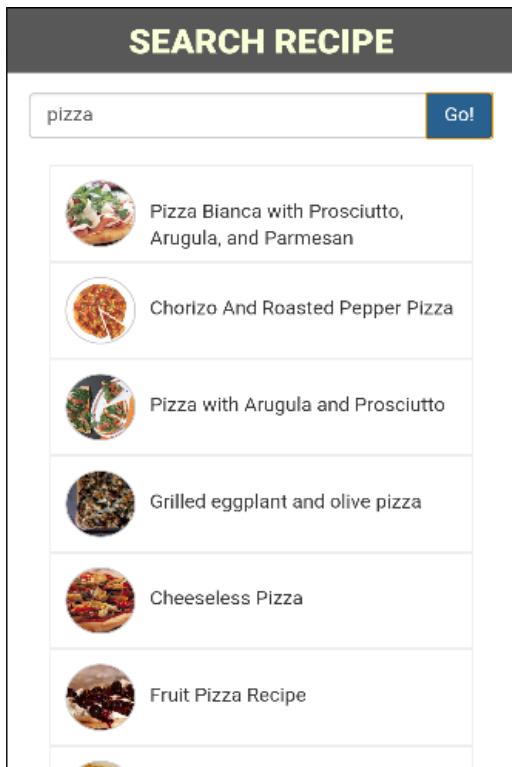
Screenshot 13: Search Recipe

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

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.

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,	14oz	279 Kcal
Regular Crust, Frozen, Cooked		
Parsley, Fresh	2tbsp	23 Kcal

Screenshot 15: Ingredients with calorie info

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.



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.

Screenshot 17: Create Recipe



Create Recipe Page

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

Screenshot 17: Adding ingredient

The screenshot shows a 'Create New Recipe' form. The main fields contain 'strawberry smoothie', '2', and 'cups'. Below this, an 'Add More' section contains entries for 'milk' (200 ml), 'sugar' (100 grams), and 'water' (50 ml). A 'Save' button is at the bottom.

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.

Screenshot 17: Adding more ingredients

The screenshot shows the SSMS interface with a query window containing the following code:

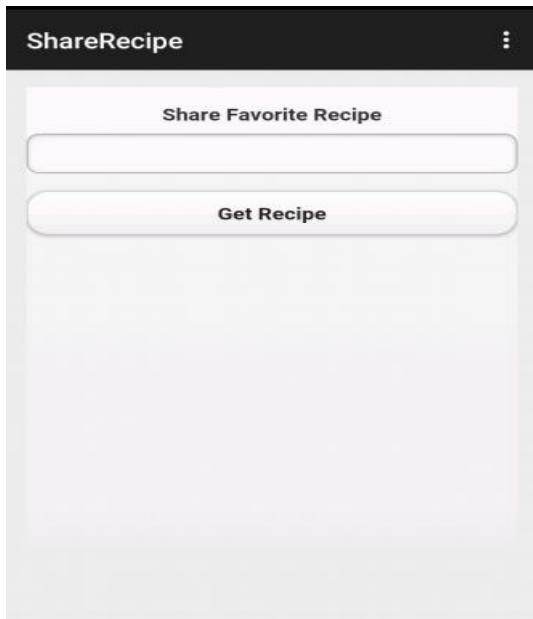
```
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP 1000 [item_name]
      ,[ingredient]
      ,[quantity]
      ,[label]
  FROM [Calorie2Grocery].[dbo].[Ingredients]
```

The results grid displays the following data:

	item_name	ingredient	quantity	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

At the bottom, a message says 'Query executed successfully.'

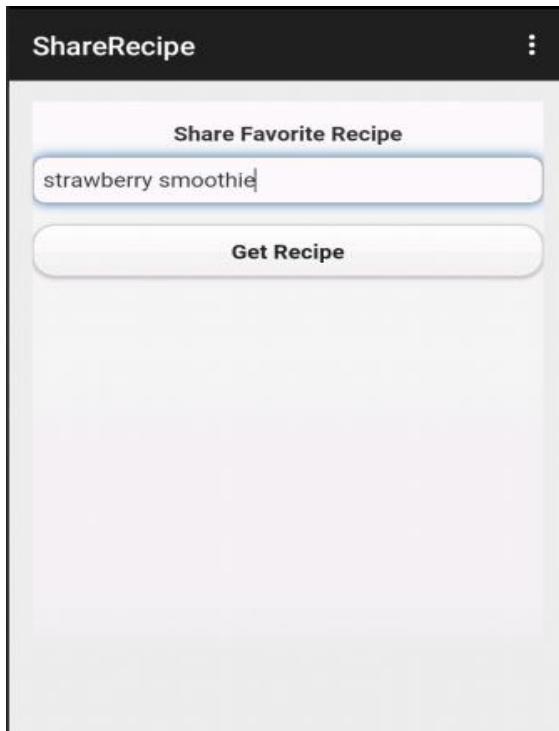
Screen Shot: 18 shows that the above created recipe has been added to the database successfully.



Screen Shot: 19 Share Recipe Page

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.



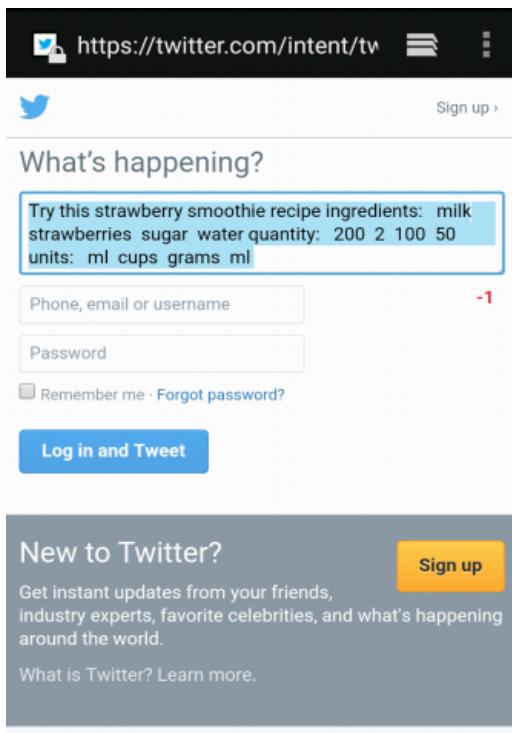
Screen Shot: 20 Share Recipe

Share Recipe Page

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



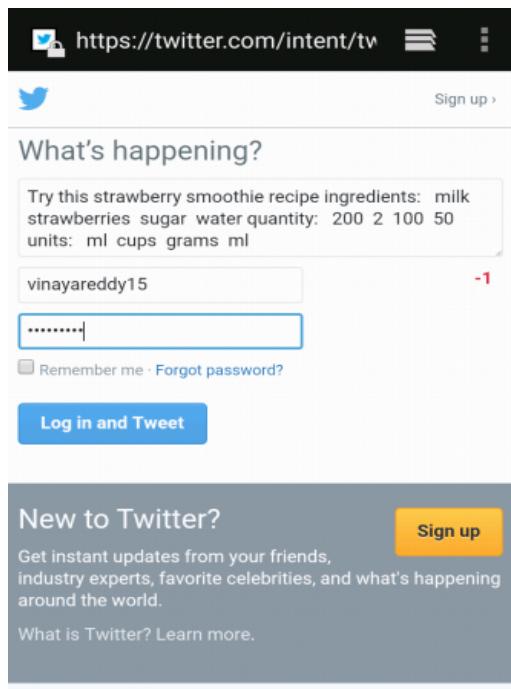
Screen Shot: 21 Showing the recipe details



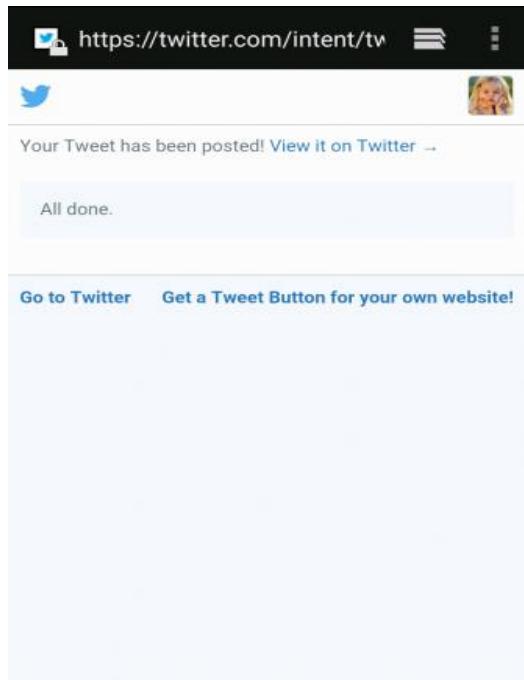
Screenshot: 22

Share Recipe Page

- Screenshot shows displaying the details of the strawberry recipe.



Screenshot : 23



Screenshot : 24

Share Recipe using Twitter

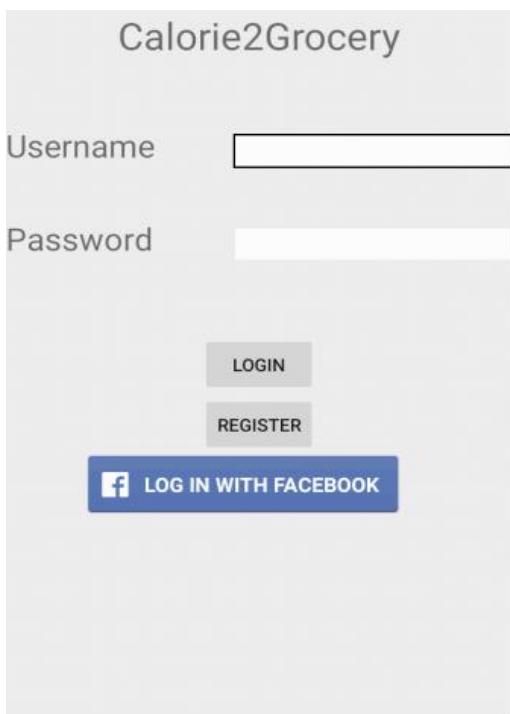
- Screenshot showing success message after posting the tweet.



Screenshot :25 Twitter post

Share Recipe using Twitter

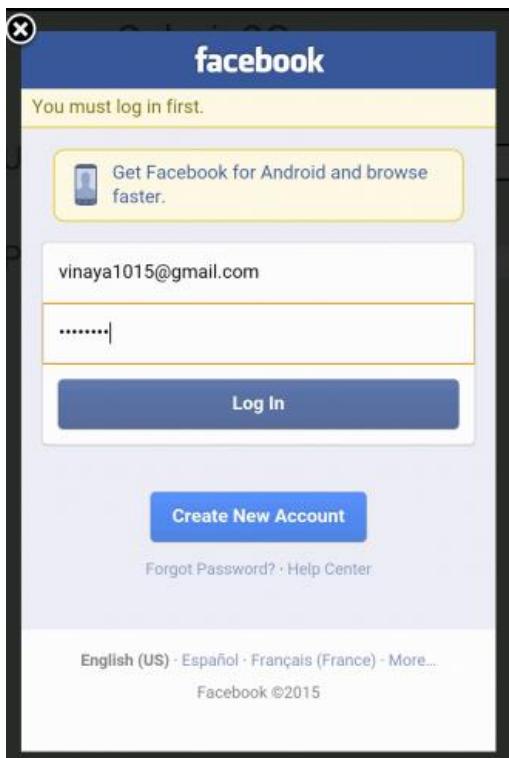
- Screen shot showing the post in twitter page.



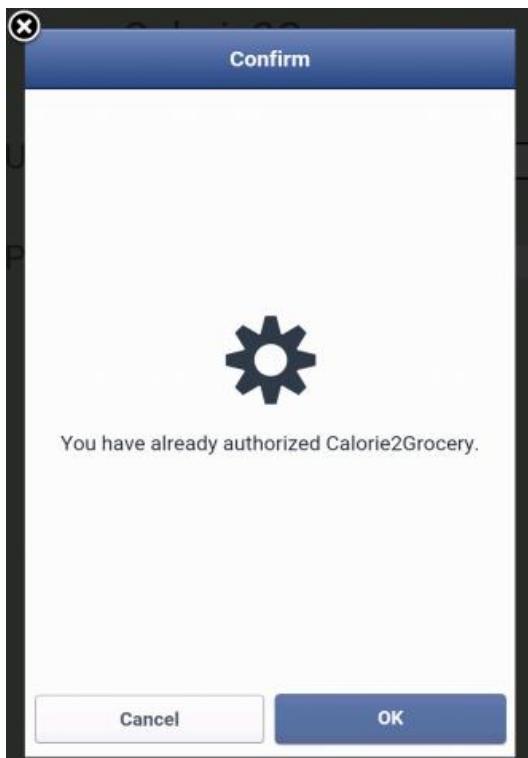
Screen shot: 26 Feacebook login

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.



Screen shot :27 Facebook login



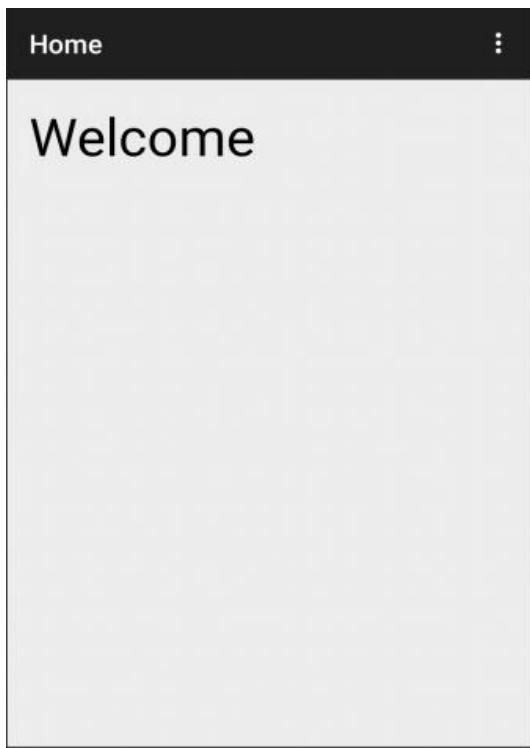
Screen shot:28 Facebook login

Facebook Login

- This screenshot shows entering the login details for Facebook.

Facebook Login

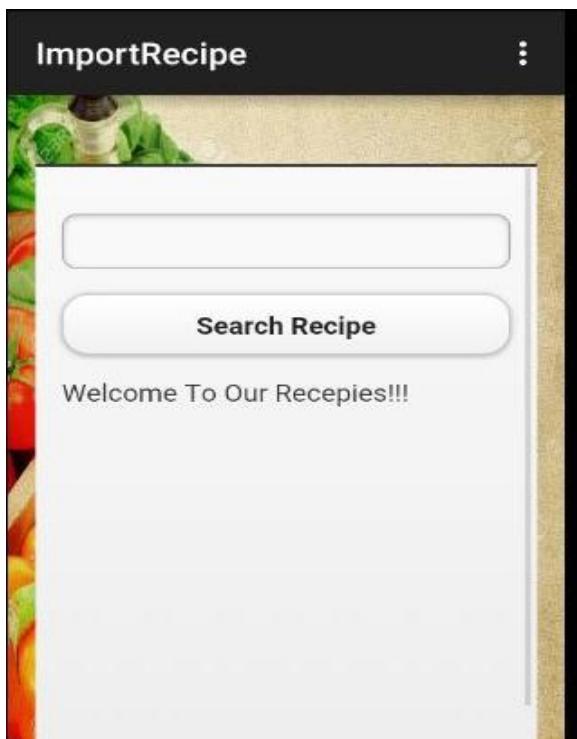
- This screenshot shows that the user has authorized the Calorie2Grocery application to access basic info from facebook.



Screen shot: 29

Facebook Login

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



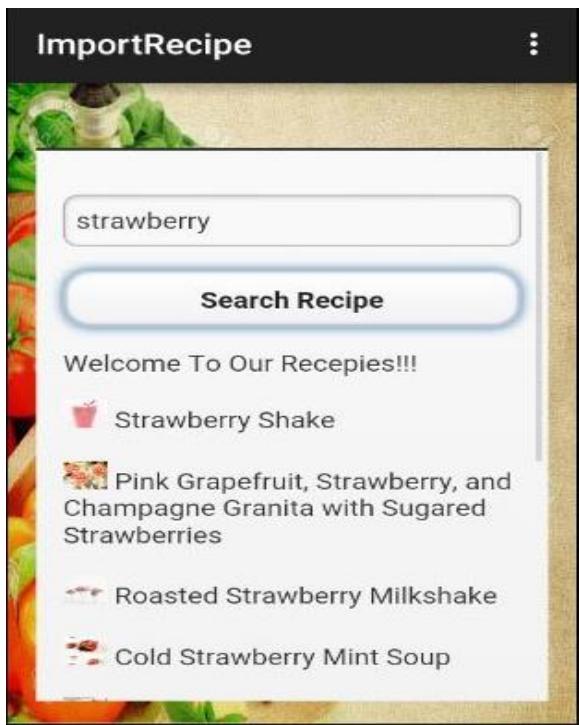
Screenshot: 30 Search for a recipe

Search Recipe

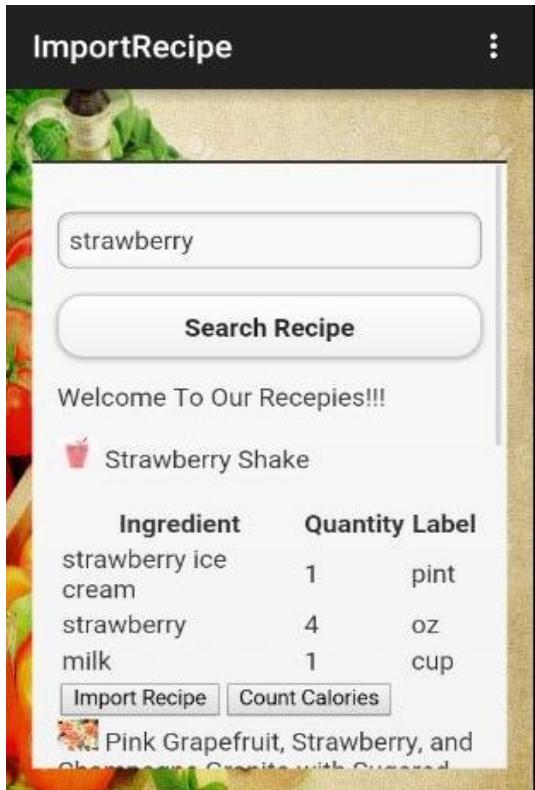
- The user gets the search recipe page when he clicks the search recipe button in home page.
- The user can search any recipe by giving the keywords in the search bar.

Technical Details:

- The client side logic is implemented using web technologies JavaScript, Jquery & Ajax



Screen shot: 31 Search recipe



Screen shot: 32 Import recipe

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
Connect master
SQLQuery5.sql - (local)\...\s...5) [SQLQuery4.sql - RSpnt5 (58)] [SQLQuery3.sql - SSpnt5 (57)*] [SQLQuery2.sql - (local)\...\...2)] [SQLQuery1.sql - (local)\...\...5)]
SELECT TOP 1000 [item_name]
, [ingredient]
, [quantity]
, [label]
FROM [Calorie2Grocery].[dbo].[Ingredients]
```

item_name	ingredient	quantity	label
382	strawberry smoothie	200	ml
383	strawberry smoothie	100	grams
384	strawberry smoothie	50	ml
385	Strawberry Shake	1	pint
386	Strawberry Shake	4	oz
387	Strawberry Shake	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 master
SQLQuery6.sql - (local)\...\s...5)* [SQLQuery5.sql - (local)\...\...5)] [SQLQuery2.sql - (local)\...\...2)] [SQLQuery1.sql - (local)\...\...5)]
select * from [Calorie2Grocery].[dbo].[Ingredients] where item_name='sweet';
```

item_name	ingredient	quantity	label
1	sweet	laggey	0
2	sweet	milk1	6
3	sweet	sugar	50
4	sweet	water	1

Query 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.

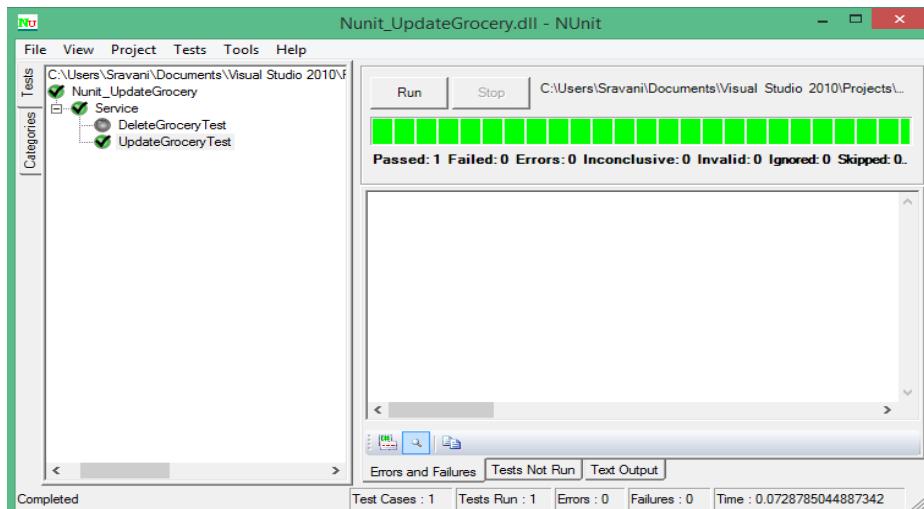
Page	Button	Expected	Actual
View Grocery list	view	Should display grocery list	Displaying grocery list
	edit	Should modify the ingredients, quantity and label	Able to modify the ingredients, quantity and label
	delete	Should delete the grocery items	Deleted the grocery items
Calorie count	Browse for 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

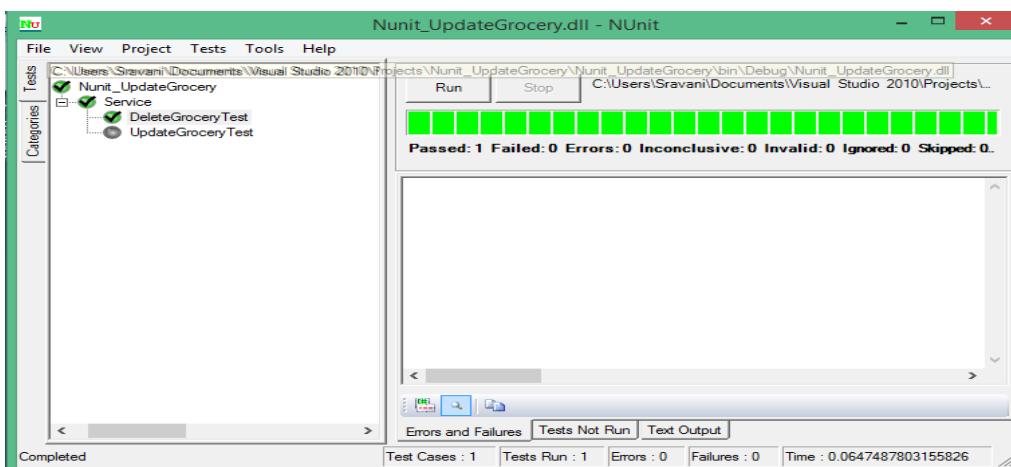
Data Validation:

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.

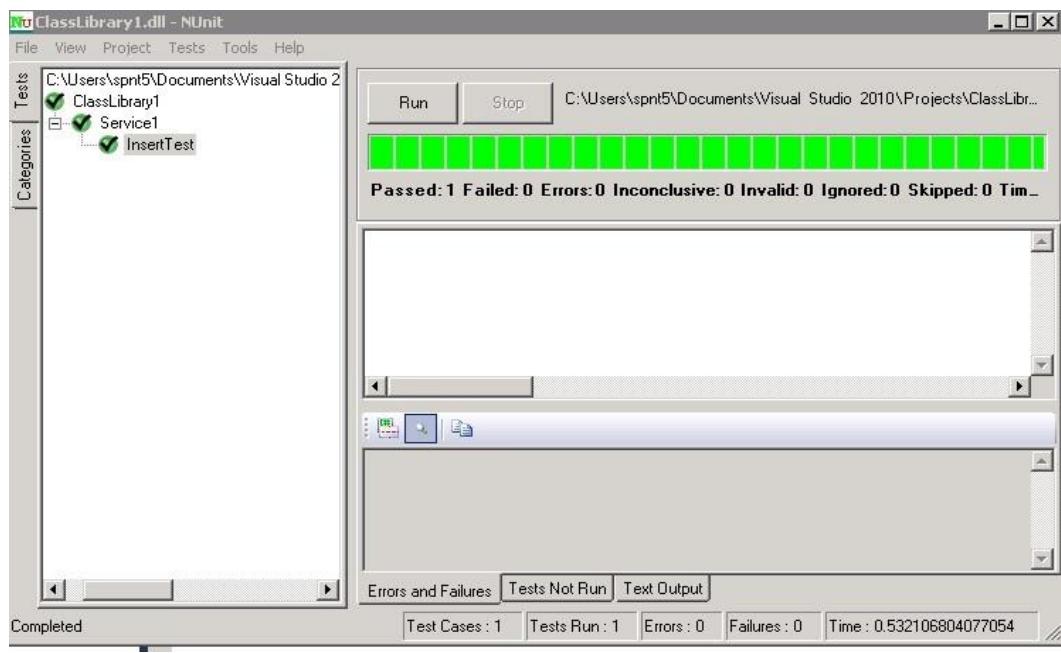
Screenshot of N-Unit validation: This is used to validate the data for update grocery item method in the rest service.



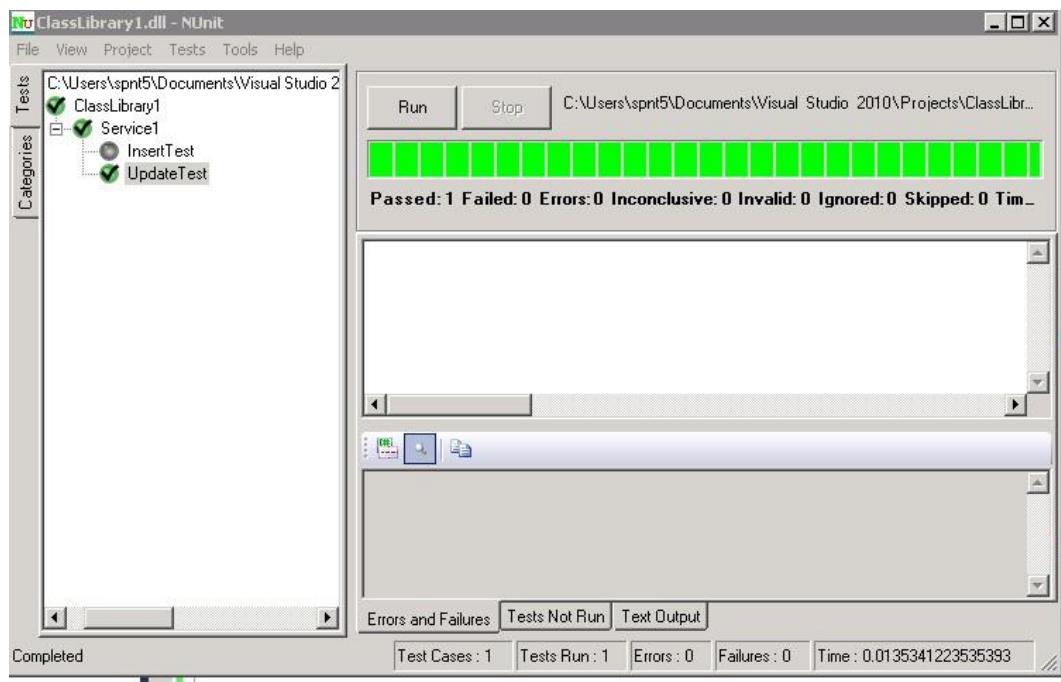
Screenshot for N-Unit validation of delete groceries API:



Screenshot for N-Unit validation of Import groceries API:



Screenshot for N-Unit validation of Update groceries API:



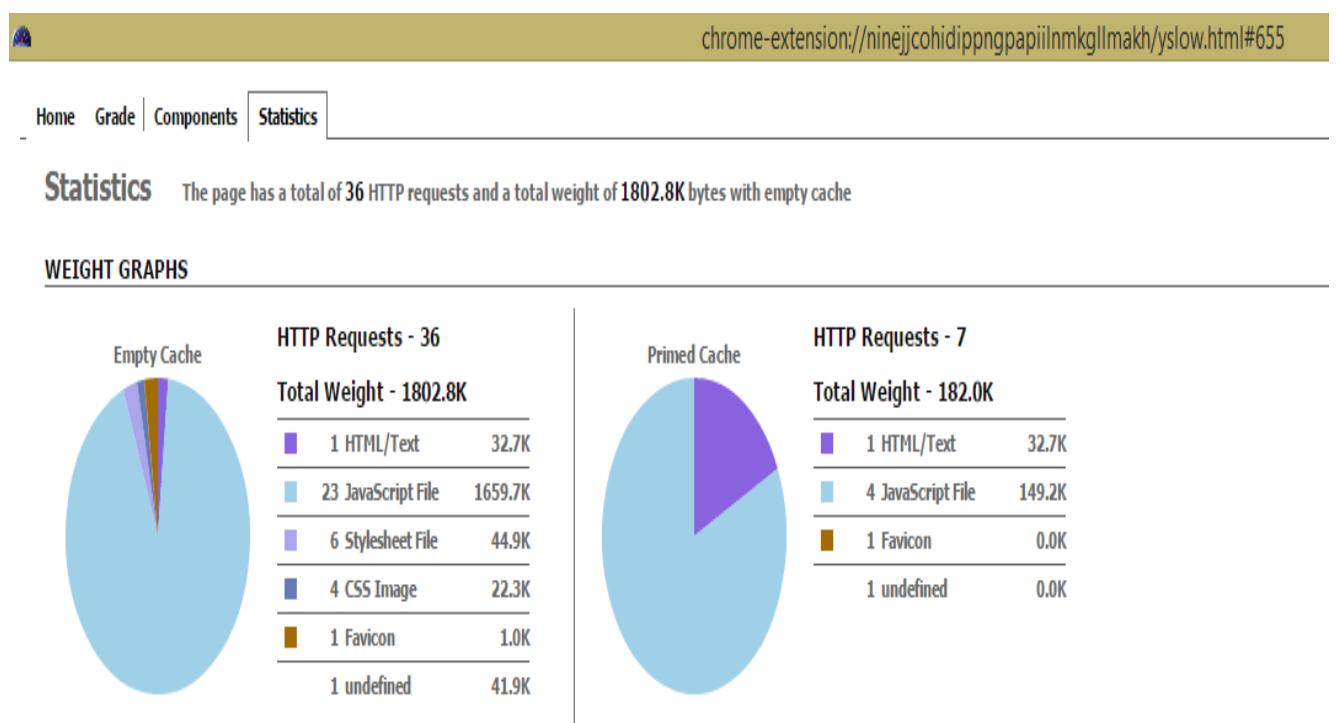
Performance Testing:

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://ninejjcohidipppngpapiinmkglmakh/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 \(6\)](#)

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

F Make fewer HTTP requests

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

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://ninejjcohidipppngpapiinmkglmakh/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](#)

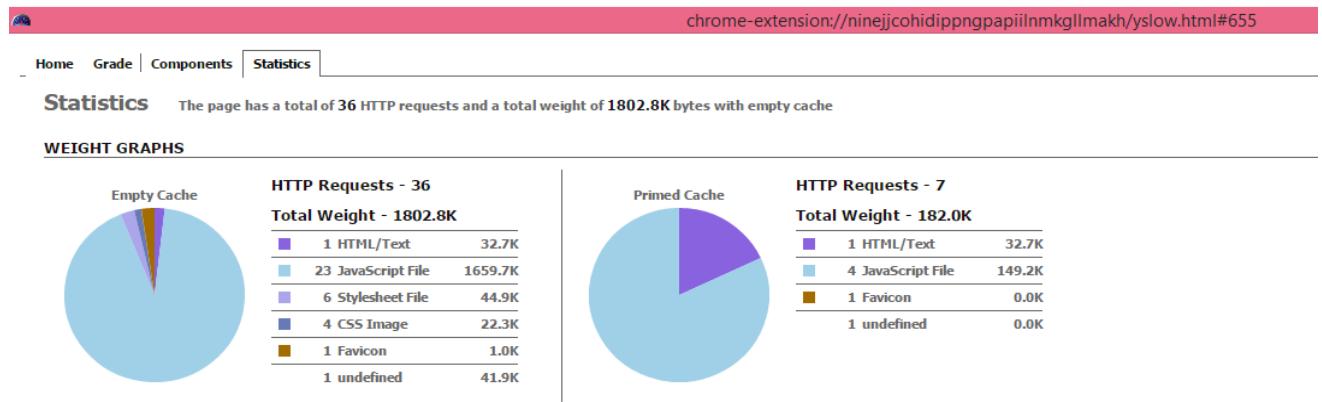
TYPE	SIZE (KB)	GZIP (KB)	COOKIE RECEIVED (bytes)	COOKIE SENT (bytes)	HEADERS	URL	EXPIRES (Y/M/D)	RESPONSE TIME (ms)	ETAG	ACTION
+ doc (1)	32.7K									
+ js (23)	1684.5K									
+ css (6)	44.9K									
+ cssimage (4)	22.3K									
+ favicon (1)	1.0K									
+ font (1)	41.9K									

* 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 page displaying a table of network requests.

Components: The page has a total of 37 components and a total weight of 1802.8K bytes.

TYPE	SIZE (KB)	GZIP (KB)	COOKIE RECEIVED (bytes)	COOKIE SENT (bytes)	HEADERS	URL	EXPIRES (Y/N/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://ninejcohidippingpapiiinmkglmakhy/yslow.html#655

Home Grade Components Statistics

Grade C Overall performance score 74 Ruleset applied: YSlow(V2) URL: http://jsfiddle.net/laGh6/yfsdp3mq/1/

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

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:

[Read More](#)

Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 3

Testing for a web page for Creating a recipe

chrome-extension://ninejcohidippingpapiiinmkglmakhy/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 Total Weight - 1802.8K <ul style="list-style-type: none"> 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 	HTTP Requests - 7 Total Weight - 182.0K <ul style="list-style-type: none"> 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://ninejjcohidipngpapiinmkglmakh/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://ninejjcohidipngpapiinmkglmakh/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/vp813/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.
This page has 6 external stylesheets. Try combining them into one.

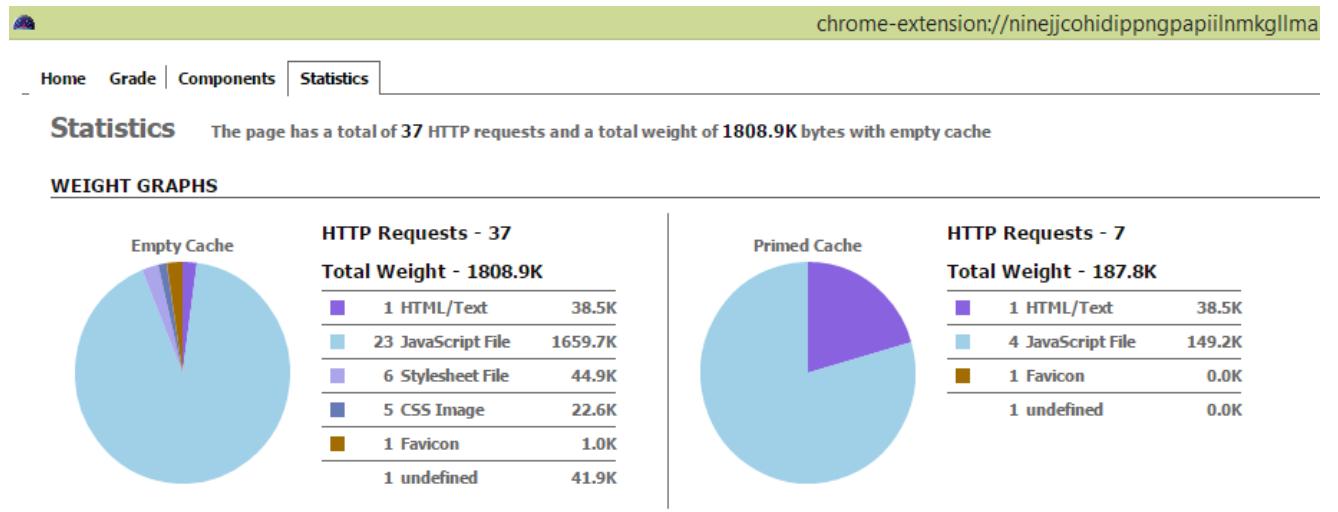
Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet,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

[Read More](#)

Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 3

Web Page for displaying a recipe long with calorie count



Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 1

The screenshot shows the 'Components' tab of the WebPageTest interface. It lists 37 components with the following 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)	38.5K									
doc	38.5K	38.5K		665	🔗	http://jsfiddle.net/b6h6/qfevzmq/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/clusive_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

Screen Shot: 2

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

Grade C Overall performance score 74 Rule set applied: YSlow(V2) URL: http://jsfiddle.net/la6h6/gfgy2smg/8/

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

!!/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

Empty Cache

Type	Weight
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

Type	Weight
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

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

TYPE	SIZE (KB)	GZIP (KB)	COOKIE RECEIVED (bytes)	COOKIE SENT (bytes)	HEADERS	URL	EXPIRES (Y/H/D)	RESPONSE TIME (ms)	ETAG	ACTION
doc (1)	34.7K									
doc	34.7K	34.7K		665	⌚	http://jsfiddle.net/spnt5/n6rl5r3s/2/	no expires	0		
+ 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 (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-48b"	
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: 2

chrome-extension://ninejcohidipngpapiinmkglmakh/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/spnt5/n6rl5r3s/2/>

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

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.

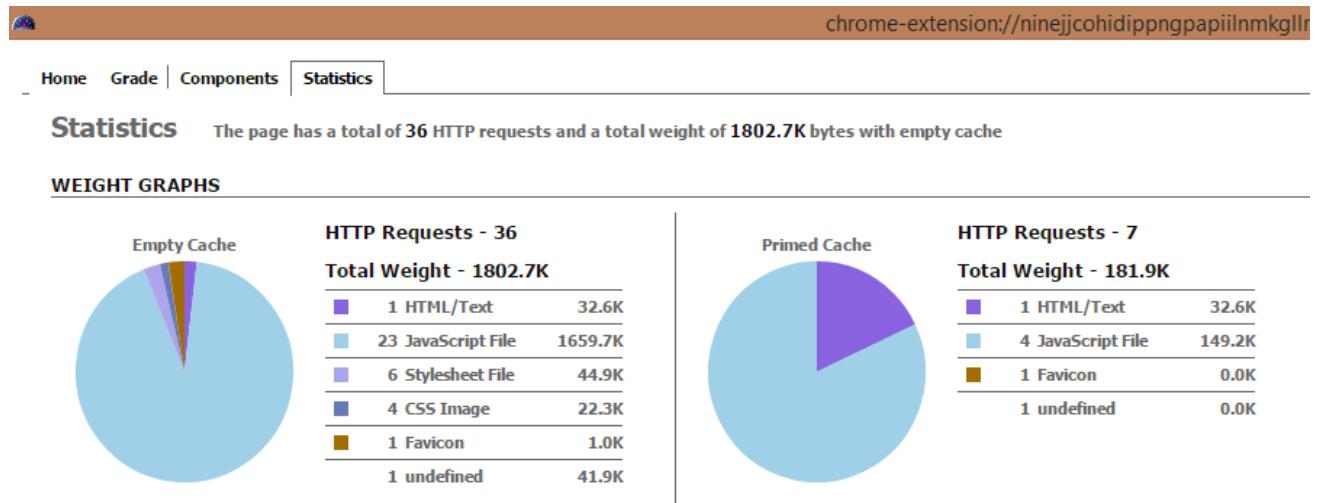
Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet,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

[»Read More](#)

Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 3

Testing a web page for updating a recipe



Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 1

The screenshot shows a table of components with columns for Type, Size (KB), GZIP (KB), Cookie Received (bytes), Cookie Sent (bytes), Headers, URL, Expires (Y/H/D), Response Time (ms), ETAG, and Action. The table includes rows for various file types like doc, js, css, and font.

TYPE	SIZE (KB)	GZIP (KB)	COOKIE RECEIVED (bytes)	COOKIE SENT (bytes)	HEADERS	URL	EXPIRES (Y/H/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/css/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			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.

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

Screen Shot: 2

chrome-extension://ninejjcohidippngpapiiinmkglmakh/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 \(6\)](#)

[Twitter](#) [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.

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://ninejjcohidippngpapiiinmkglmakh/yslow.html#757

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

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

Primed Cache

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

Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 1

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

Home Grade Components Statistics

Components The page has a total of **36** components and a total weight of **1801.3K 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)	31.2K									
js (23)	1684.5K									
css (6)	44.9K									
css	0.6K	0.6K		667	⌚	http://jsfiddle.net/css/normalize.css?...	2015/5/4	0		
css	0.2K	0.2K		667	⌚	http://jsfiddle.net/css/tricks.css?...	2015/5/4	0		
css	18.7K	18.7K		667	⌚	http://jsfiddle.net/css/elusive-webfont.css?...	2015/5/4	0		
css	8.0K	8.0K		667	⌚	http://jsfiddle.net/js/codemirror/lib/codemirror.css?...	2015/5/4	0		
css	17.2K	17.2K		667	⌚	http://jsfiddle.net/css/screen.css?...	2015/5/4	0		
css	0.0K			667	⌚	http://jsfiddle.net/css/light.css?...	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									
font	41.9K			667	⌚	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://ninejcohidipngpapiinmkglmakh/yslow.html#1008

Home Grade Components Statistics

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

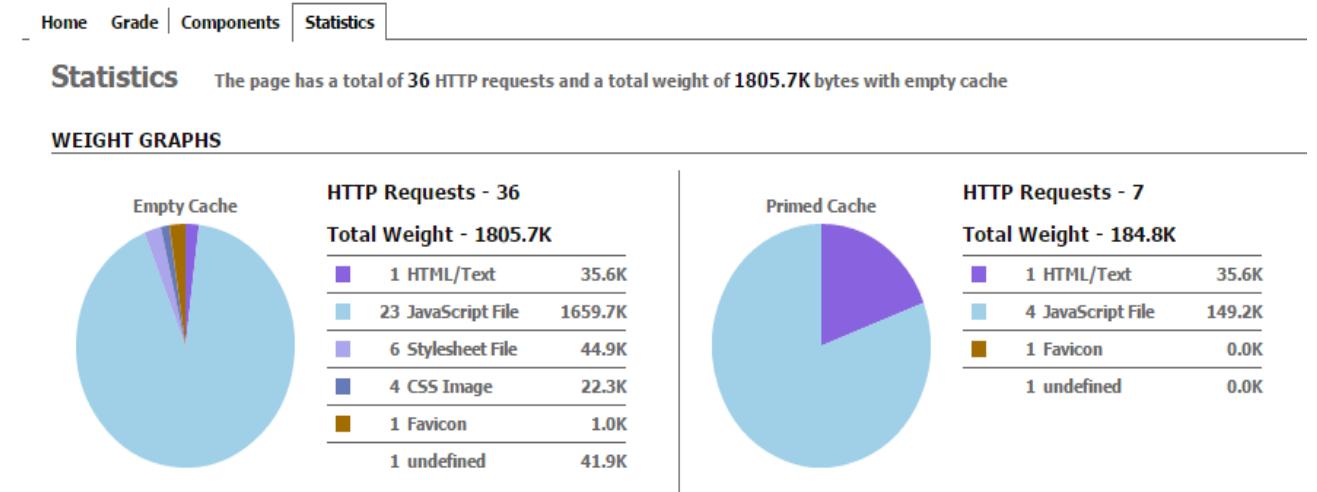
Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, 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://ninejjcohidippngpapiilmkgilmakh/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/M/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's performance analysis for a page on jsfiddle.net. The grade is C. Key findings include:

- Grade F** on "Make fewer HTTP requests". It lists 23 external JavaScript scripts and 6 external stylesheets.
- Grade A** on "Use a Content Delivery Network (CDN)".
- Grade F** on "Add Expires headers".
- Grade B** on "Compress components with gzip".
- Grade A** on "Put CSS at top".
- Grade F** on "Put JavaScript at bottom".
- Grade A** on "Avoid CSS expressions".
- n/a** on "Make JavaScript and CSS external".
- A** on "Reduce DNS lookups".
- F** on "Minify JavaScript and CSS".
- A** on "Avoid URL redirects".
- A** on "Remove duplicate JavaScript and CSS".
- A** on "Configure entity tags (ETags)".
- A** on "Make AJAX cacheable".
- A** on "Use GET for AJAX requests".
- A** on "Reduce the number of DOM elements".
- A** on "Avoid HTTP 404 (Not Found) error".
- A** on "Reduce cookie size".
- F** on "Use cookie-free domains".
- A** on "Avoid AlphaImageLoader filter".
- A** on "Do not scale images in HTML".
- A** on "Make favicon small and cacheable".

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 a browser window with multiple tabs open. The active tab is titled "Create Recipe" and contains a "Create New Recipe" form. The form fields include:

- veg fried rice
- boiled rice
- 1
- bowl

Below the form is a table with rows for carrots, onions, and spices, each with a quantity of 1 and a unit of cup/spoon. At the bottom of the form is a "Save" button.

A modal dialog box is displayed in the center of the screen with the message "recipe created!" and an "OK" button. Below the dialog is a checkbox with the text "Prevent this page from creating additional dialogs".

At the bottom of the browser window, the Firebug toolbar is visible, showing tabs for Console, HTML, CSS, Script, DOM, Net, and Cookies. The "Console" tab is selected. The status bar at the bottom right shows the date and time: 10:54 PM 4/13/2015.

Screenshot of a web browser showing a recipe creation form. The form includes fields for ingredients like 'veg fried rice', 'boiled rice', and quantities like '1 bowl'. A modal dialog box in the center says 'recipe created!' with an 'OK' button. Below the form is a 'Save' button. At the bottom, a 'Net' panel shows network activity with four requests made to 'kc-sce-cs551-2.kc.umkc.edu' and one to 'maps.googleapis.com'. The total load time is 402ms.



2. Share Recipe.html

Screenshot of a web browser showing a 'Share Favorite Recipe' form. The user has entered 'sweet' in the ingredient field. Below the form is a 'Get Recipe' button. Underneath the button, there is some pre-filled text: 'recipename : sweet', 'ingredients : jaggery sugar water', 'quantity : 0 50 1', and 'label : lbs lb gallon'. At the bottom, a 'Console' panel shows a successful GET request to 'http://kc-sce-cs551-2.kc.umkc.edu/aspnet_client/M...inaya2/displayrecipe/Service1.svc/data?sweet?url'. The response body contains the JSON object: { "ingredient": "jaggery:sugar:water", "label": "lbs/lb:gallon", "quantity": "0:50:1" } followed by the string "jaggery:sugar:water", "jaggery sugar water", "0 50 1", and "lbs lb gallon". The console also lists several stack frames for 'jquery-1.9.1.js' and 'sharerecipe.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 gallon

[Tweet](#)

[Facebook](#)

Console HTML CSS Script DOM Net Cookies

Clear Persist All HTML CSS JavaScript XHR Images Plugins Media Fonts

URL	Status	Domain	Size	Remote IP	Timeline
GET sweet?url	200 OK	kc-sce-cs551-2.kc.umkc.edu	83 B	134.193.136.128:80	137ms
1 request			83 B		137ms

10:50 PM 4/13/2015

3. viewRecipe.html

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

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

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

Console HTML CSS Script DOM Net Cookies

Clear Persist Profile All Errors Warnings Info Debug Info Cookies

jQuery-1.9.1.js (line 8526)
 ViewRecipe.html (line 48)
 ViewRecipe.html (line 48)

10:51 PM 4/13/2015

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

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

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

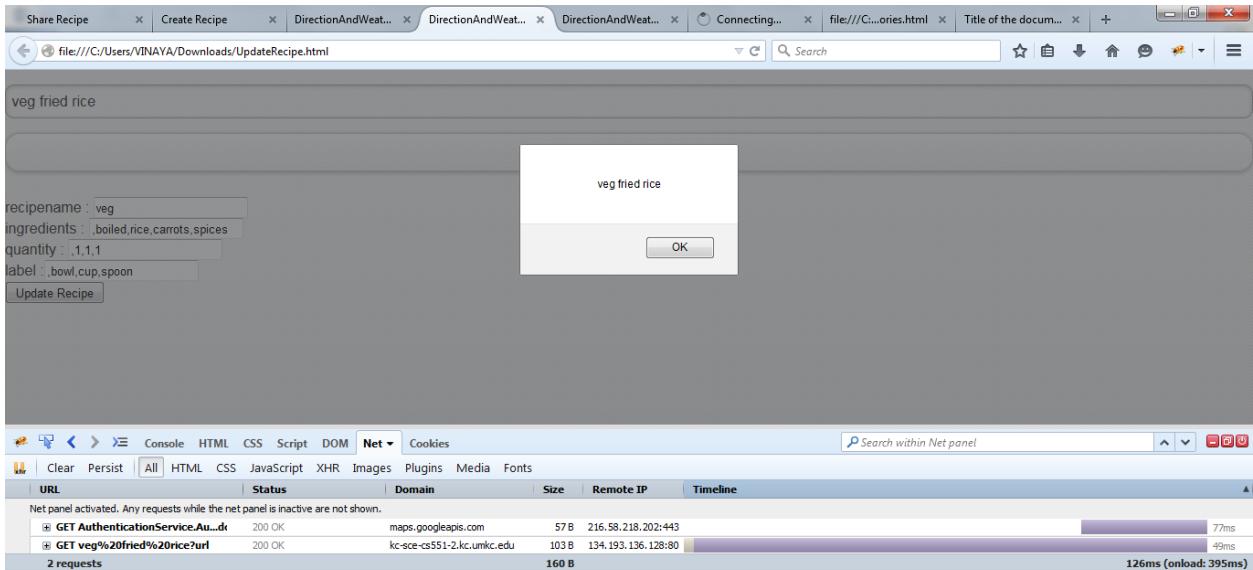
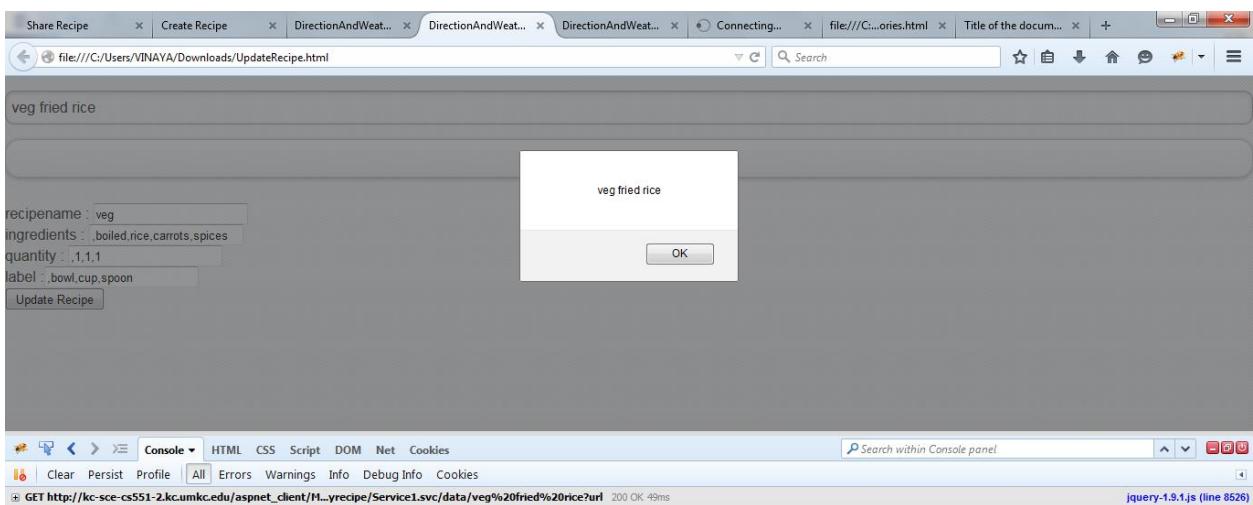
Console HTML CSS Script DOM Net Cookies

Clear Persist All HTML CSS JavaScript XHR Images Plugins Media Fonts

URL	Domain	Size	Remote IP	Timeline
Net panel activated. Any requests while the net panel is inactive are not shown.				
GET gfh?url	kc-sce-cs551-2.kc.umkc.edu	2.3 KB	134.193.136.128:80	10 ms
GET AuthenticationService.Au...C	maps.googleapis.com	57 B	216.58.218.202:443	49ms
2 requests		2.3 KB		150ms (load: 442ms)

10:51 PM 4/13/2015

4. Updaterecipe.html



5. Import Recipe.html

The screenshot shows a web browser window with multiple tabs open. The active tab displays a search result for "yogurt". A modal dialog box is centered on the screen with the message "recipe imported successfully!!!". Below the message, there is a checkbox labeled "Prevent this page from creating additional dialogs" and an "OK" button. The main content area shows a welcome message "Welcome To Our Receipties!!!", a thumbnail image of a yogurt container, and a table titled "Ingredient Quantity Label" with two rows: "milk 4 cup" and "yogurt 15 gram". Below the table are two buttons: "Import Recipe" and "Count Calories". There is also a link to "Maple Banana Frozen Yogurt". At the bottom of the page, there is a navigation bar with links like "Console", "HTML", "CSS", "Script", "DOM", "Net", and "Cookies", along with a search bar and a toolbar.



The screenshot shows a web browser window with the same search results for "yogurt" as the previous screenshot. A "Search Recipe" button is visible below the search bar. The "Net" tab of the developer tools is selected, showing a list of network requests and their details. The requests listed are:

- GET search?q=apple&app_id=51... 200 OK api.edamam.com 20.2 kB 107.20.173.119:443 825ms
- GET AuthenticationService.Au...dc 200 OK maps.googleapis.com 57 B 216.58.218.170:443 45ms
- GET 8bb76a6852aa383785f54c24 200 OK edamam.com 24.0 kB 107.20.173.119:443
- GET e507a640dc013a573d2d87cd 200 OK edamam.com 10.7 kB 107.20.173.119:443
- GET c75a65078289a1c1c74057923 200 OK edamam.com 0 B 107.20.173.119:443
- GET 31d8fd3f977560d2209be6ca 200 OK edamam.com 13.0 kB 107.20.173.119:443
- GET feff415120db72c7290dc62e 200 OK edamam.com 9.5 kB 107.20.173.119:443
- GET 47c75bbe333d0337fd5b84315 200 OK edamam.com 18.2 kB 107.20.173.119:443
- GET 9811c168c13ab9daa0e66716 200 OK edamam.com 7.4 kB 107.20.173.119:443
- GET 50319d4ec6fd12fa10f7b02f48 200 OK edamam.com 12.1 kB 107.20.173.119:443
- GET ea7d9d38b512c1dea759df1f5 200 OK edamam.com 9.1 kB 107.20.173.119:443
- GET 0775818ccfd07cc52d0b00b 200 OK edamam.com 13.5 kB 107.20.173.119:443
- GET search?q=yogurt&app_id=5... 200 OK api.edamam.com 18.2 kB 107.20.173.119:443 2.22s
- GET 13f3d6891c17b73ea39398f75 200 OK edamam.com 64.0 kB 107.20.173.119:443 1.41s

At the bottom of the page, there is a navigation bar with links like "Console", "HTML", "CSS", "Script", "DOM", "Net", and "Cookies", along with a search bar and a toolbar. The system tray on the right shows the date and time as "10:42 PM 4/13/2015".

6. Calorie.html

strawberry

Strawberry Shake

Total Calories: 692

Ingredient

Ice Creams, Strawberry
Strawberries, Raw
Milk, Whole, 3.25% Milkfat, With Added Vitamin D

	Quantity	Calories
1pint	353 Kcal	
4oz	400 Kcal	
1cup	84 Kcal	

Pink Grapefruit, Strawberry, and Champagne Granita with Sugared Strawberries

Net panel

Search within Net panel

2.73s (onload: 222ms)

10:45 PM 4/13/2015

Console HTML CSS Script DOM Net Cookies

18 requests 214.1 KB

Request details:

- GET 087c32b6f70b90168ce568r 200 OK edamam.com 10.4 KB 107.20.173.119:443 431ms
- GET c0d075b898cd446ec60e1c 200 OK edamam.com 14.3 KB 107.20.173.119:443 528ms
- GET 2495c263291ad4ff8ec13afcc 200 OK edamam.com 13.0 KB 107.20.173.119:443 415ms
- GET 3057774c22a554674e6bb3b 200 OK edamam.com 18.2 KB 107.20.173.119:443 529ms
- GET a475e9c073aaace4dc51cb2; 200 OK edamam.com 14.4 KB 107.20.173.119:443 644ms
- GET b7dc59fb463cf15e4427fd9d9c 200 OK edamam.com 19.6 KB 107.20.173.119:443 738ms
- GET 79d39475d96e8b0493b73aa 200 OK edamam.com 20.7 KB 107.20.173.119:443 840ms
- GET ?format=json&q=strawber 200 OK api.nal.usda.gov 24.6 KB 54.165.187.228:80 399ms
- GET ?format=json&q=strawber 200 OK api.nal.usda.gov 217B 54.165.187.228:80 391ms
- GET ?format=json&q=milk+sort 200 OK api.nal.usda.gov 209B 54.165.187.228:80 390ms
- GET ?ndbno=18547&type=b&for 200 OK api.nal.usda.gov 1.1 KB 54.165.187.228:80 167ms
- GET ?ndbno=03681&type=b&for 200 OK api.nal.usda.gov 1.5 KB 54.165.187.228:80 292ms
- GET ?ndbno=35001&type=b&for 200 OK api.nal.usda.gov 955B 54.165.187.228:80 166ms

strawberry

Strawberry Shake

Total Calories: 692

Ingredient

Ice Creams, Strawberry
Strawberries, Raw
Milk, Whole, 3.25% Milkfat, With Added Vitamin D

	Quantity	Calories
1pint	353 Kcal	
4oz	400 Kcal	
1cup	84 Kcal	

Pink Grapefruit, Strawberry, and Champagne Granita with Sugared Strawberries

Console

Search within Console panel

10:46 PM 4/13/2015

Console HTML CSS Script DOM Net Cookies

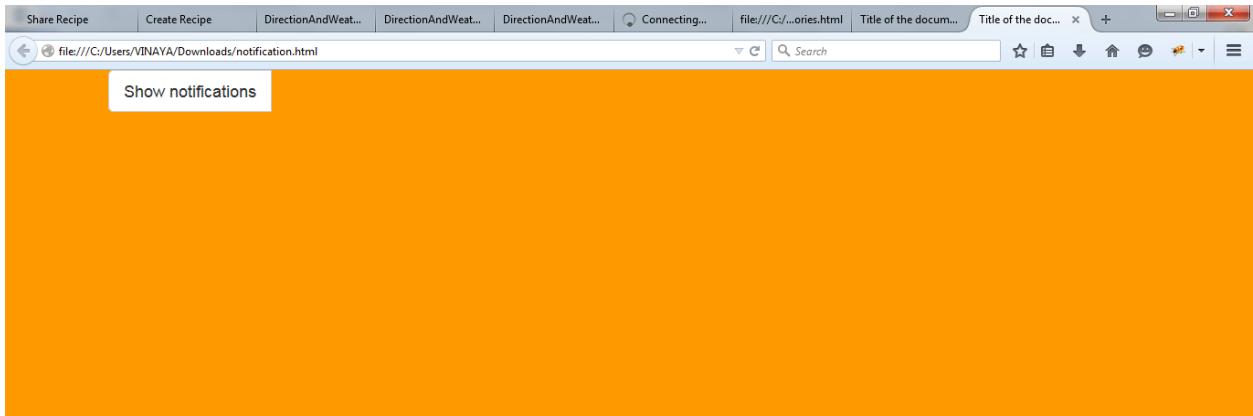
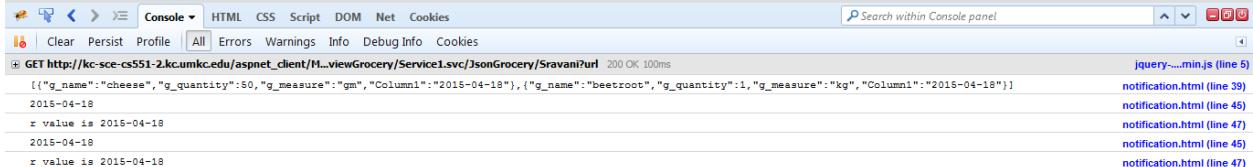
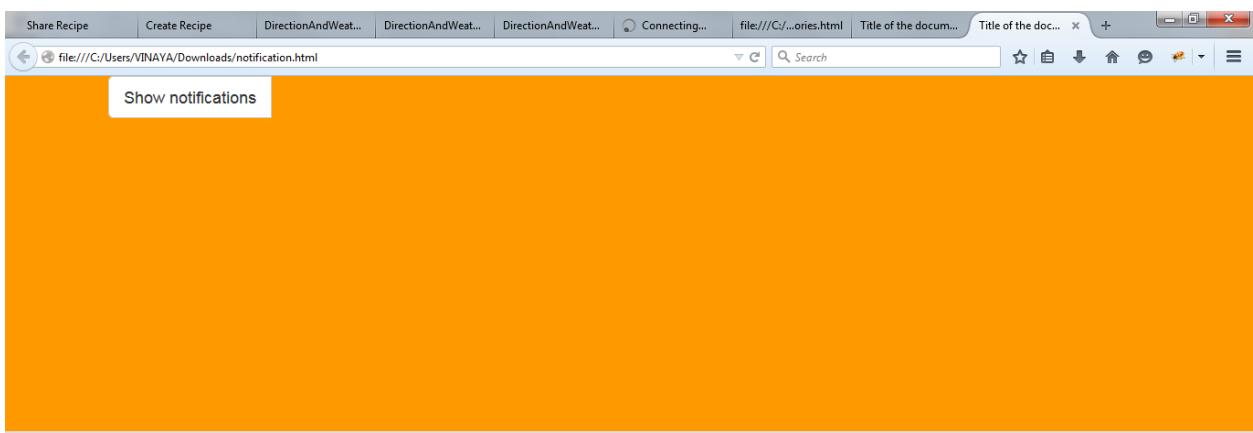
1 Errors Warnings Info Debug Info Cookies

Request details:

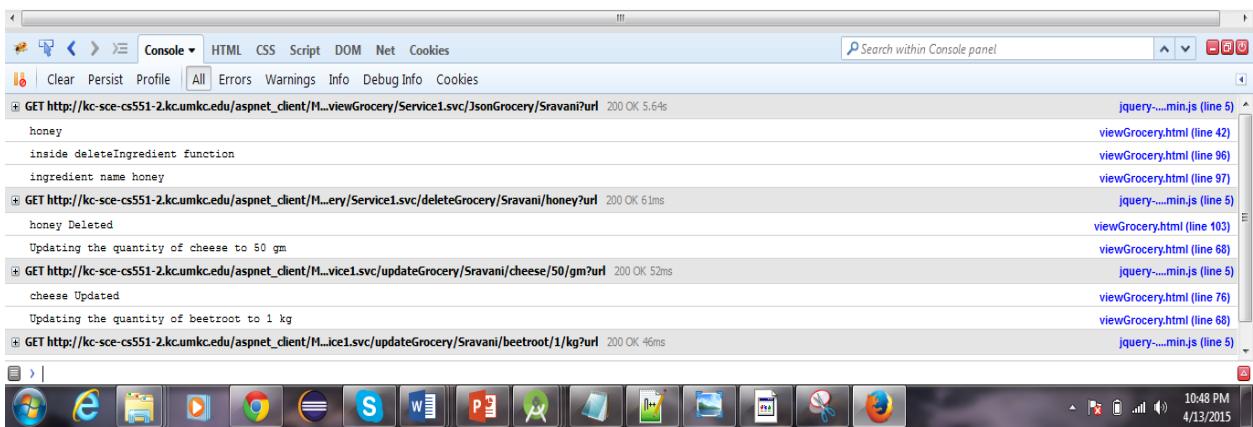
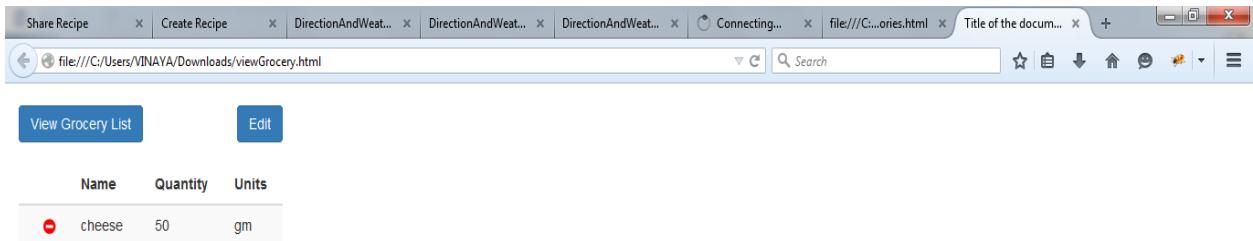
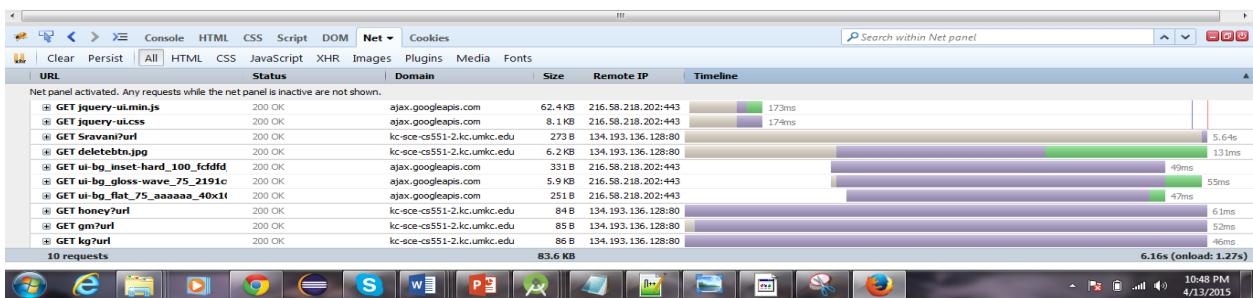
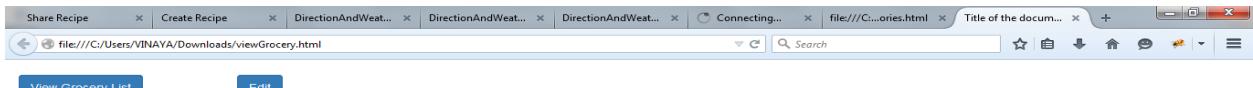
- GET http://api.nal.usda.gov/usda/ndb/search/?format=j...key=DgxqZYmXG8YsyCIEflkJaAklH4ybQTEHORaKedx&url 200 OK 399ms jquery....min.js (line 4)
- GET http://api.nal.usda.gov/usda/ndb/search/?format=j...key=DgxqZYmXG8YsyCIEflkJaAklH4ybQTEHORaKedx&url 200 OK 39ms jquery....min.js (line 4)
- GET http://api.nal.usda.gov/usda/ndb/search/?format=j...key=DgxqZYmXG8YsyCIEflkJaAklH4ybQTEHORaKedx&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



8. viewGrocery.html



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 srawani				
#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 srawani				
#39 As a user i should be able to get the calorie count of the complete recipe				
Done Tasks 0 Comments srawani				
#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				
<hr/>				
#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	3
#9 As a user i should be able to update the quantity of the grocery items in my list	   
Done Tasks 0 Comments  sravani	3

Increment #4 user stories

Increment#4 - April 9, 2015 - April 29, 2015    New Story  Filter Board 

Todo	Doing	Reviewing	Done
3 9	2 3		
#15 As a developer i need to develop the UI for getting the shopping list 0 Comments - Tasks 	#48 As a system, I should display the calorie count for the exact quantity of the ingredient. 0 Comments - Tasks 		
#13 As a user i need a shopping list so that i wont miss something needed for preparing the recipe 0 Comments - Tasks 	#16 As a team, we need to integrate the code developed by all the team members. 0 Comments - Tasks 		
#11 As a developer, we need to perform integration testing on the application. 0 Comments - Tasks 			

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 transition to the app - Fix on increment 1	5	Vinaya	Done
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/47725api_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=9fcd3aa5746d2a423a350cee3ea4d57d

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.

PROJECT INCREMENT 4

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 and 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 chose 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:

- 5) User can add tomatoes and onions to his currently available groceries list. User can update the list every time he purchases groceries.
- 6) User can search for a recipe of veg sandwich
- 7) The app displays the ingredients for the sandwich. Eg : Bread, Potatoes, Onions, Cucumber, Tomatoes, Lettuce
- 8) 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.
- 9) 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 and 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

Fresh Box	Fooducate	Anylist	Proposed System
User can add the items along with expiration date and	User can add the items based on barcodes but	User can just add the groceries list	User can add the items along with expiration date and

get notifications whenever the product expired.	cannot add products which doesn't have barcodes.	depending upon the category.	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

- 10) Register or login securely through Facebook
- 11) Maintain a list of groceries available in their home
- 12) Search for recipes and save them for quick reference
- 13) Get the detailed information of calories per each ingredient required for the recipe

- 14) Add/remove the ingredients and adjust the quantity of the ingredients
- 15) Get the final count of calories per serving of the recipe
- 16) Based on the ingredients, get a list of groceries to buy in order to prepare the recipe.
- 17) Get the nearest grocery stores based on the current location
- 18) 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:

Requirement	Tools
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

Rest Services

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=9fcd3aa5746d2a423a350cee3ea4d57d

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

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

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/

- 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/

- 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/

- 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

- 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/useShoppingList

- 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/

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.

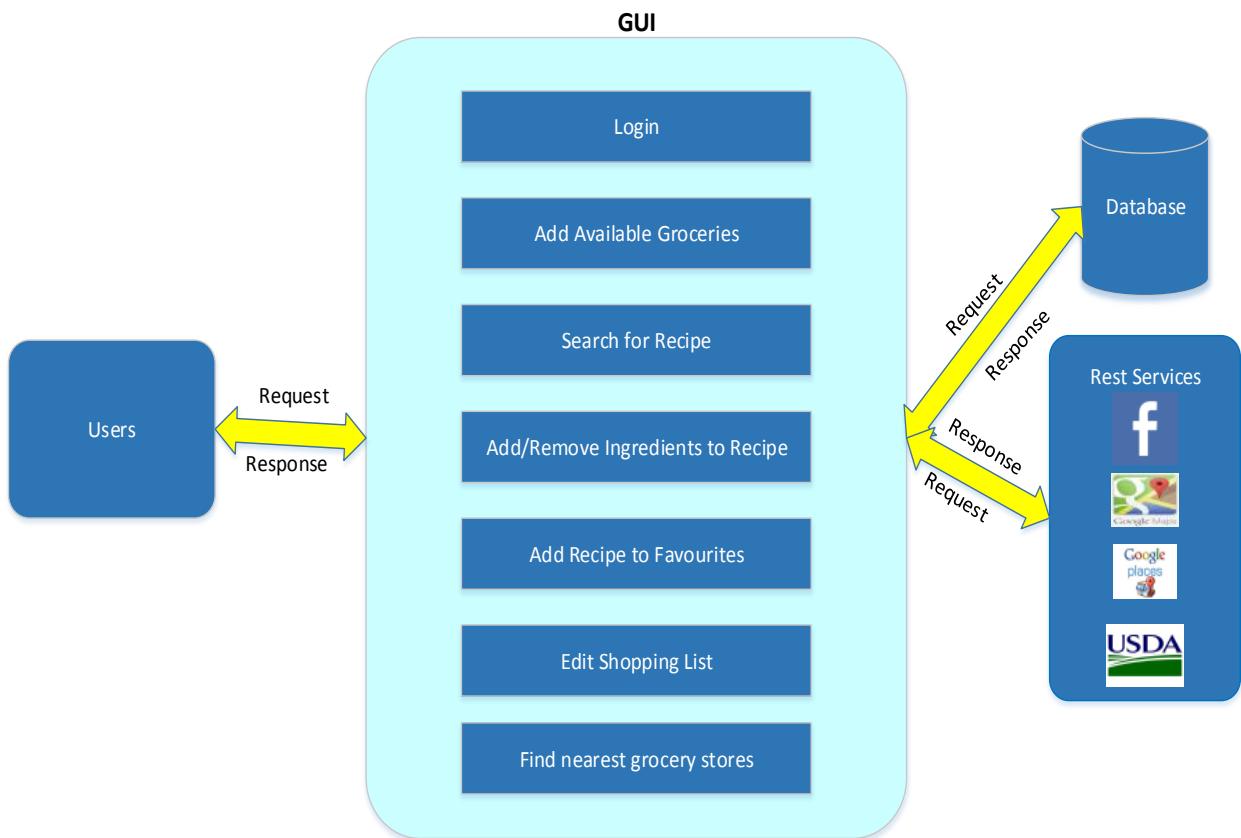
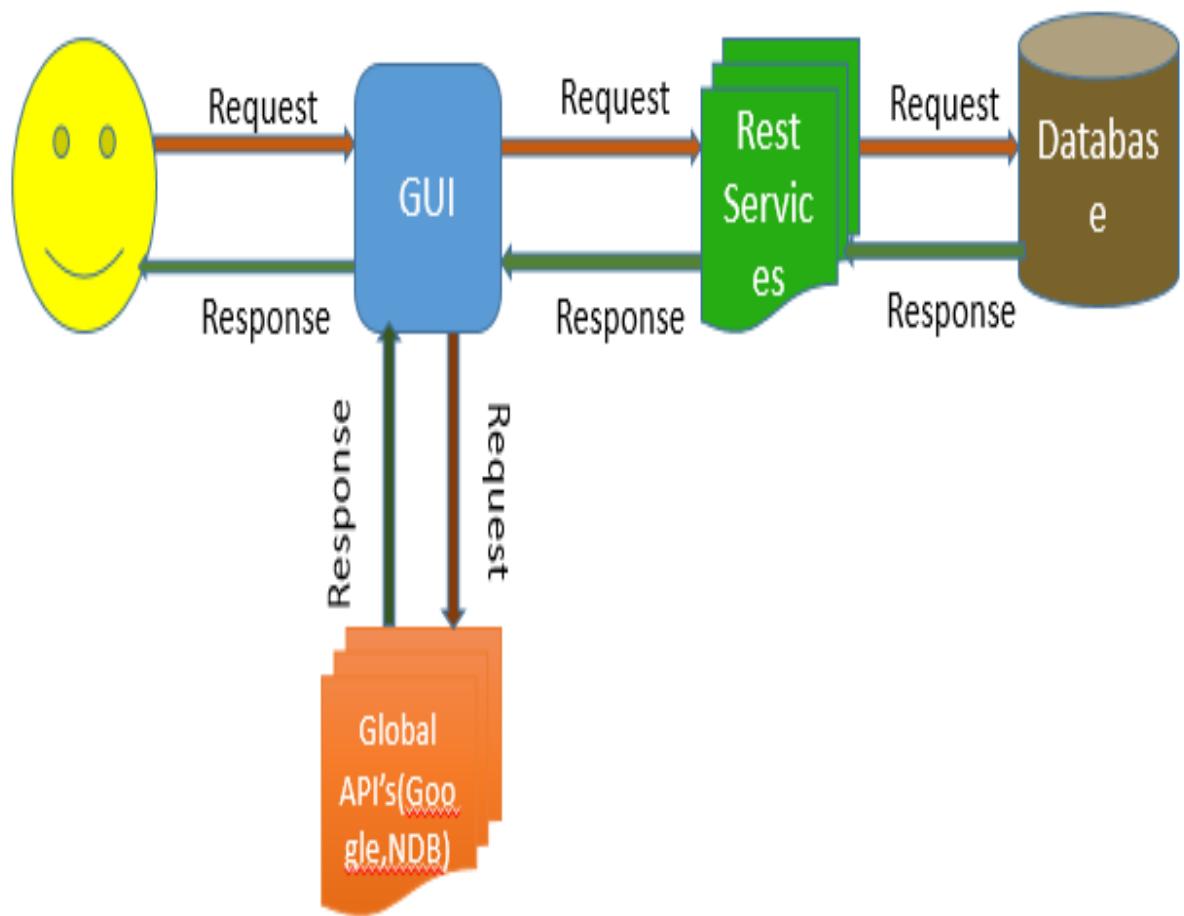


Figure 1.2 System Model

System Architecture Diagram:



Class Diagram:

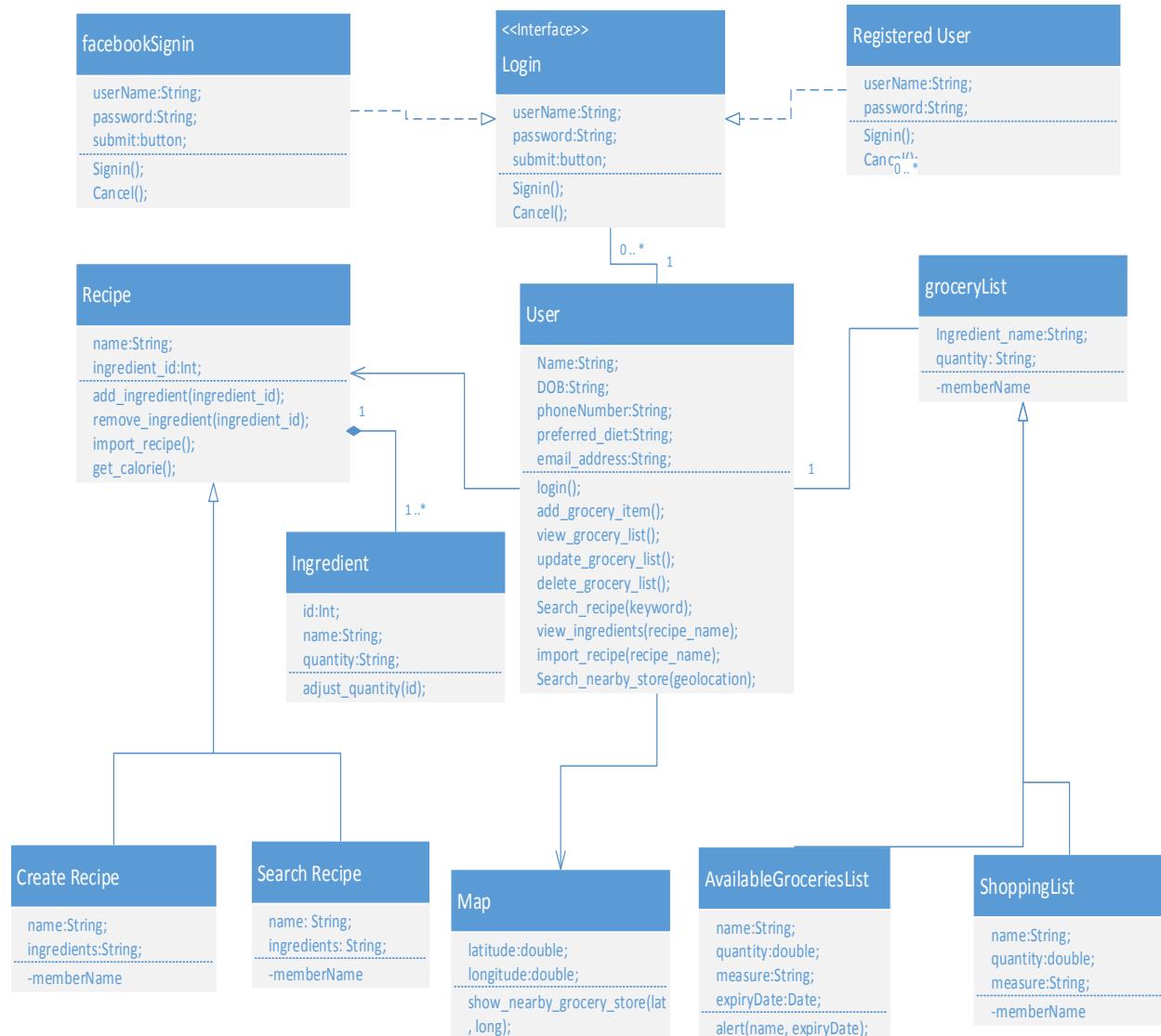


Figure 1.1 Class Diagram

Activity Diagram:

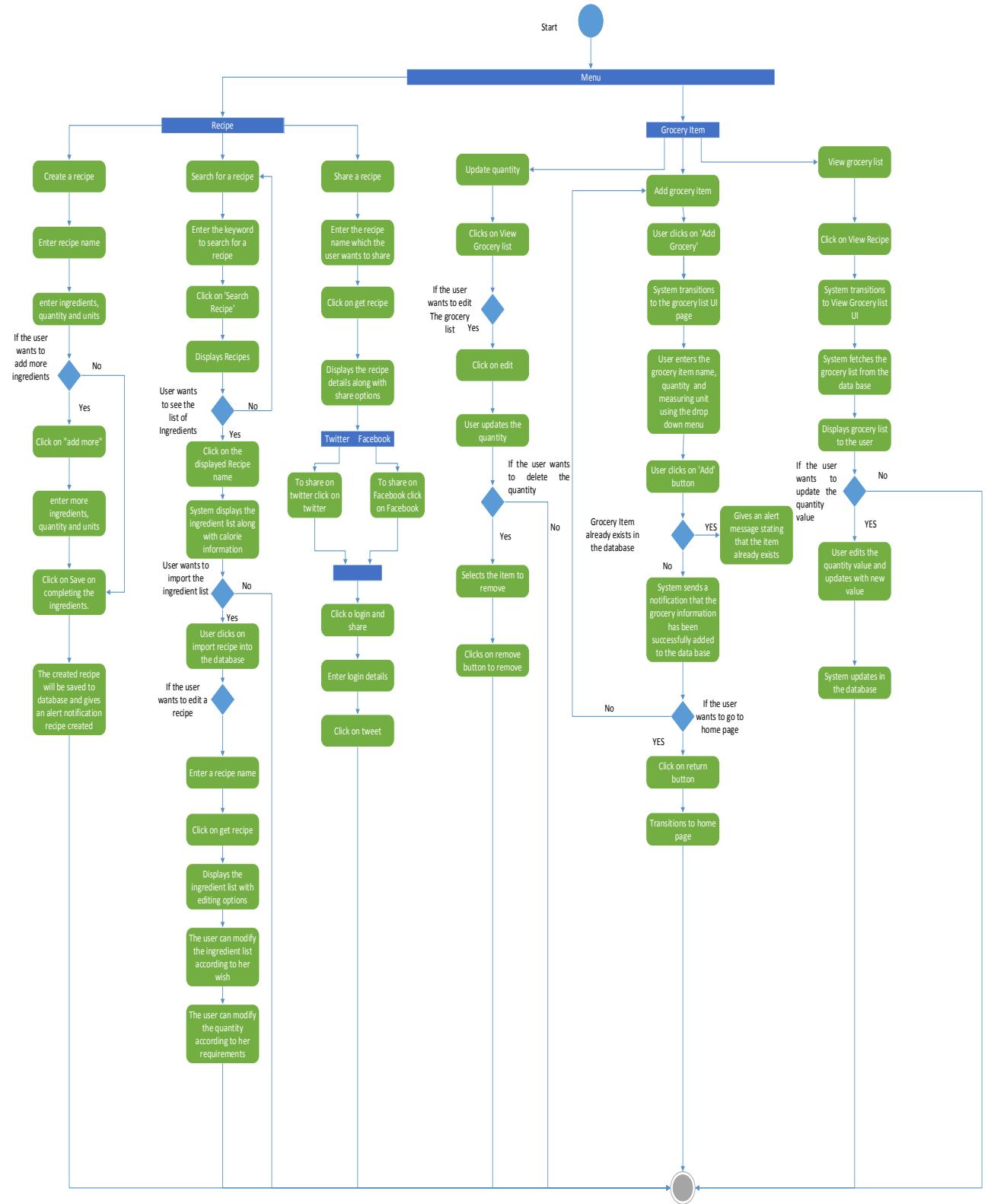
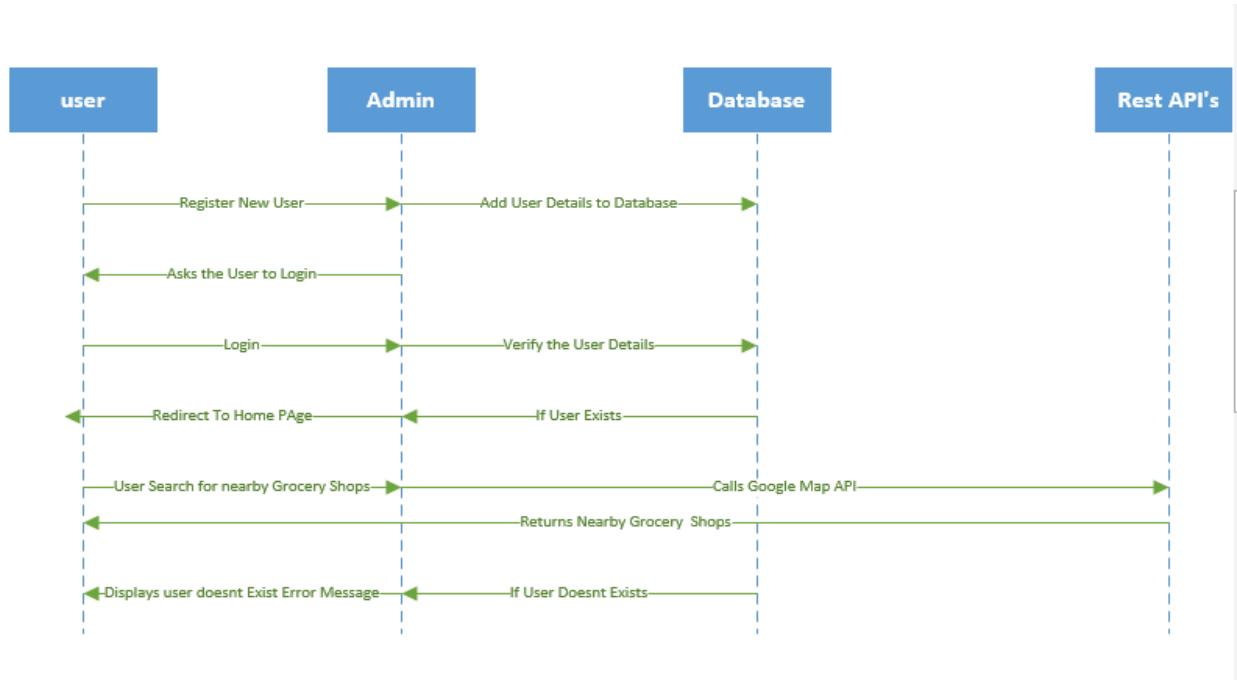


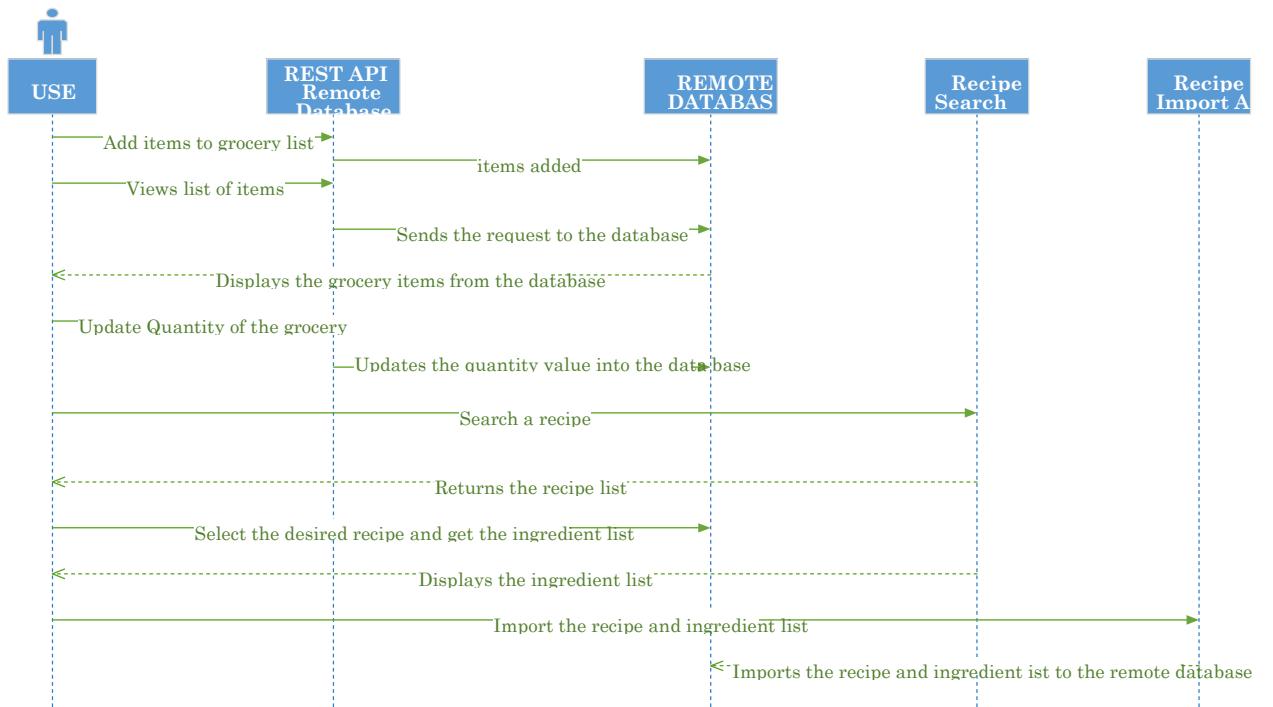
Figure 1.4 Activity Diagram

Sequence diagrams

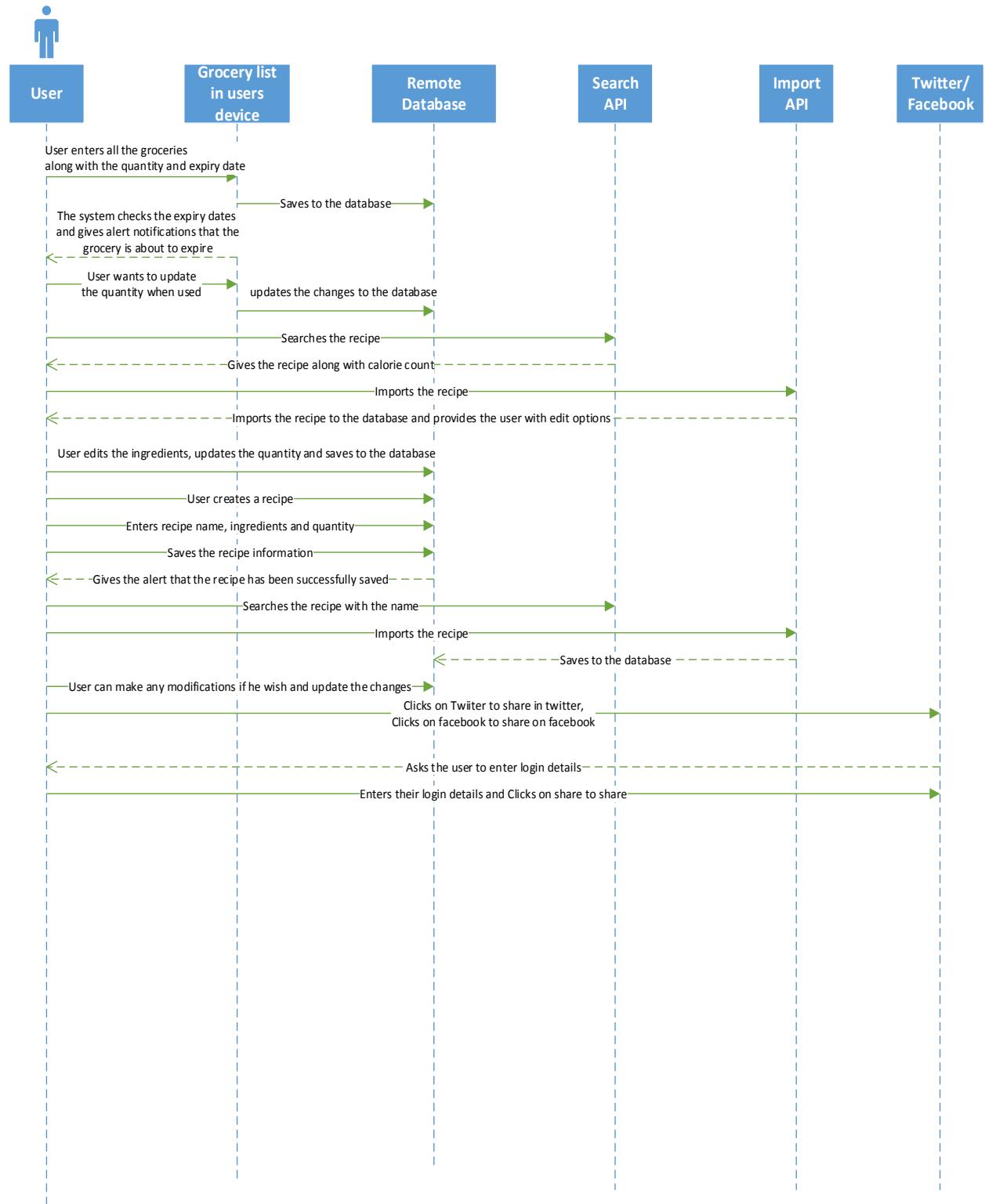
1. For Login Page



2. Adding Grocery Items



3. Importing recipes along with calorie count



Use case Templates

1) Use Case Template for Adding grocery items:

Use Case Name	Add grocery items
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 items are stored in the database
Main Success Scenario	<ol style="list-style-type: none"> 1. User clicks on “Add Grocery” button in the home page. 2. The system transitions the UI to the “Add Grocery” layout. 3. User enters the grocery item name, quantity, selects the measuring unit from the drop down menu and clicks on “Add”. 4. System adds the grocery information to the database. 5. System sends a notification that the grocery item is successfully added to the database. 6. User can go back to the home page by clicking on return button.
Extensions	<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 & 4 and then go back to the home page.</p>

2) Use Case Template to View the grocery list:

Use Case Name	View grocery list
Primary Actor	User
Pre-Conditions	<ol style="list-style-type: none"> 3. User has the app installed on his/her mobile. 4. User should be logged into the app.
Guarantee(Post Conditions)	User can view the list of groceries. User can update the quantity of grocery items.

Main Success Scenario	<ul style="list-style-type: none"> 3. User clicks on “View Grocery List” button in the home page. 4. The system transitions the UI to the “View Grocery” layout. 5. System fetches the grocery list of the user from the database. 6. System displays the grocery list to the user. 7. User can update the quantity of any of the grocery items. 8. System updates the database to reflect the changes made by the user. 9. System sends a notification saying the update is successful.
Extensions	None

3) Use Case Template to update grocery item:

Use Case Name	Update grocery items
Primary Actor	User
Pre-Conditions	10. User has the app installed on his/her mobile. 11. 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"> 8. User clicks on “View/Edit grocery list” button in the home page. 9. The system transitions the UI to the “View Grocery” layout. 10. System fetches the grocery list of the user from the database. 11. User clicks on the quantity of the grocery item he/she needs to update. 12. System changes the quantity cell to editable mode. 13. User updates the quantity and hits enter. 14. System updates the data in the remote database.
Extensions	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.

4) Use Case Template to delete grocery item:

Use Case Name	Delete grocery item
Primary Actor	User
Pre-Conditions	12. User has the app installed on his/her mobile. 13. User should be logged into the app.
Guarantee(Post Conditions)	Grocery item is deleted from the database.
Main Success Scenario	<p>5. User clicks on “View/Edit Grocery List” button in the home page.</p> <p>6. The system transitions the UI to the “View Grocery” layout.</p> <p>14. System fetches the grocery list of the user from the database.</p> <p>15. System displays the grocery list to the user.</p> <p>16. User can click on “Edit” button</p> <p>17. System display minus sign in front of each grocery item in the list.</p> <p>18. User clicks on the minus sign in front of the grocery item he/she wants to delete.</p> <p>19. System opens an alert dialog asking the user to confirm deletion.</p> <p>20. User hits “OK”.</p> <p>21. System deletes the grocery item from the database.</p> <p>22. System removes the row in the display</p>
Extensions	<p>9a. User hits “Cancel”.</p> <p>9b. System doesn’t delete the grocery item.</p>

5) Use Case Templates for searching and importing a recipe:

Use Case Name	Search & import a recipe
Primary Actor	API
Pre-Conditions	The user must be a valid user to use the app
Post-Conditions	Retrieves the results for the search
Main Success Scenario	<p>8. If the user wants to search for a recipe, then, the user enters the keyword to search.</p> <p>9. User clicks on the search recipes.</p>

	10. The system displays ten recipe names. 11. If the user wants to see the ingredient list, then the user can click on the recipe name. 12. System displays ingredient list along with quantity. 13. If the user wants to import the ingredient list then click on import. 14. The system stores the ingredient list in the database.
Extensions	None
Special Requirements	None

6) Use Case Templates to 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	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

7) Use Case Templates to 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	<ul style="list-style-type: none"> 4. If the user wants to get the favorite recipe then user clicks the view favorite recipe button 5. User clicks on the favorite recipes. 6. User gets all the favorite recipe list from the database.
Extensions	User can update the recipe

8) Use Case Templates to Update 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	<ul style="list-style-type: none"> 7. If the user wants to get the favorite recipe then user clicks the recipe name. 8. User clicks on the get recipe button 9. User gets all the recipe name, ingredient list , quantity and label from the database. 10. User can add, remove, update the ingredient , quantity and label 11. User can click the update button.

	12. System uploads the changed the recipe list to the database.
--	---

9) Use Case Template to Create 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	<ul style="list-style-type: none"> 5. If the user wants to create the favorite recipe then user enters the recipe name , ingredients list and their quantity and labels. 6. User clicks on the add more button for more ingredients to the recipe. 7. User adds all the ingredients and quantity and their labels and clicks the save recipe button. 8. System uploads the recipe tot the database.
Extensions	User can Share the recipe in twitter and facebook.

10) Use Case Templates to 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	Users should be able to share the recipe using twitter or facebook.
Main Success Scenario	<ul style="list-style-type: none"> 5. User can search the recipe he would like to search from his list of recipes in database.

	<ul style="list-style-type: none"> 6. System displays the ingredients list, quantity and label for the recipe. 7. User can share the recipe in the twitter by providing twitter login 8. Recipe information will be shared in the users twitter account as a tweet 9. Similarly, User can share the recipe in the facebook using the facebook link.
--	---

11) Use Case Template to View & Edit shopping list:

Use Case Name	View shopping list
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)	<p>User can view the shopping list</p> <p>User can edit the shopping list</p>
Main Success Scenario	<ul style="list-style-type: none"> 1. User clicks on “Shopping List” button in the home page. 2. The system transitions the UI to the “Shopping List” layout. 3. System fetches the list of grocery items that need to be purchased from the database. 4. System displays the shopping list to the user. 5. User can delete any of the items from the list by clicking on edit button. 6. System displays minus sign against each item in the list. 7. User can click on the minus sign in the same row as the shopping item to be deleted. 8. System deletes the item from the database.
Extensions	None

12) Use Case Template to Generate Notifications:

Use Case Name	Generate Notifications
Primary Actor	System
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.3. User should enter the grocery items along with expiry dates.
Guarantee(Post Conditions)	User can check the expiry dates of the grocery items
Main Success Scenario	<ol style="list-style-type: none">1. User clicks on show notifications2. The system checks the expiry dates of all the items and generates notifications which are about to expire within 1 day.3. The system generates notification “grocery expiry dates” in the notification bar of the android device.4. User clicks on the notification to get the complete details.5. The system transitions to other page which lists the grocery items which are about to expire.
Extensions	None

Implementation

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:

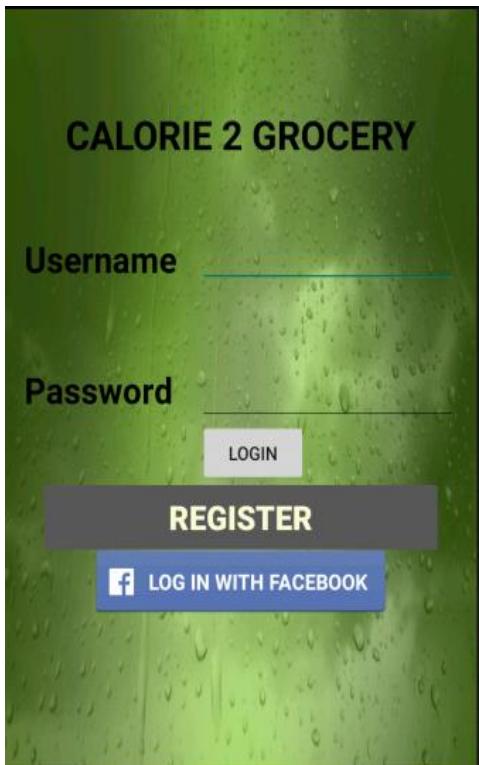
Provides the necessary buttons to navigate to different functions.

3) 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:
 - c) Add Grocery
 - d) View/Edit Grocery
- 4) Recipes: We have created the following buttons in this page:
 - d) Search Recipe
 - e) Create Recipe
 - f) Share Recipe
 - g) Update Recipe
- 5) Show Notifications: Notifies user about the expiry dates of the grocery items.
- 6) Shopping List
- 7) Grocery Stores

Screenshots:

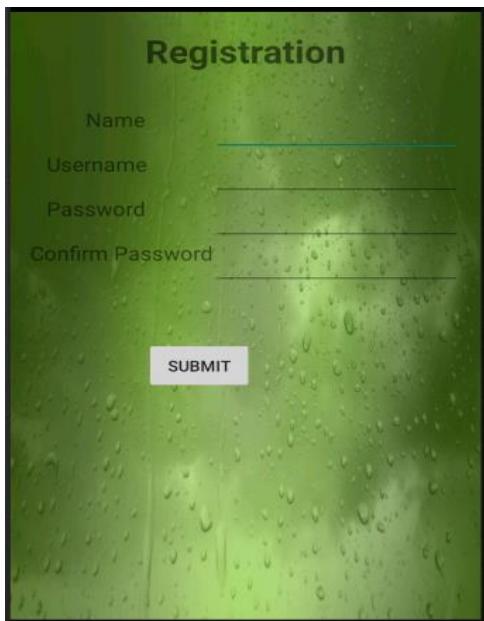


Login Page:

- 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.

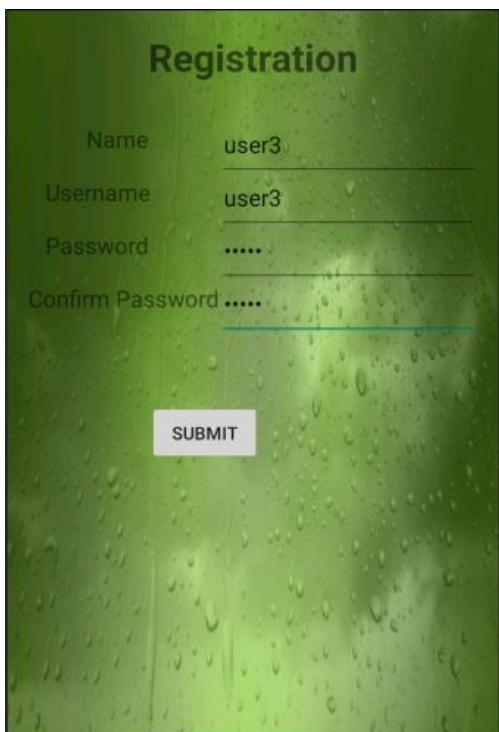


Registration Page:

- 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.

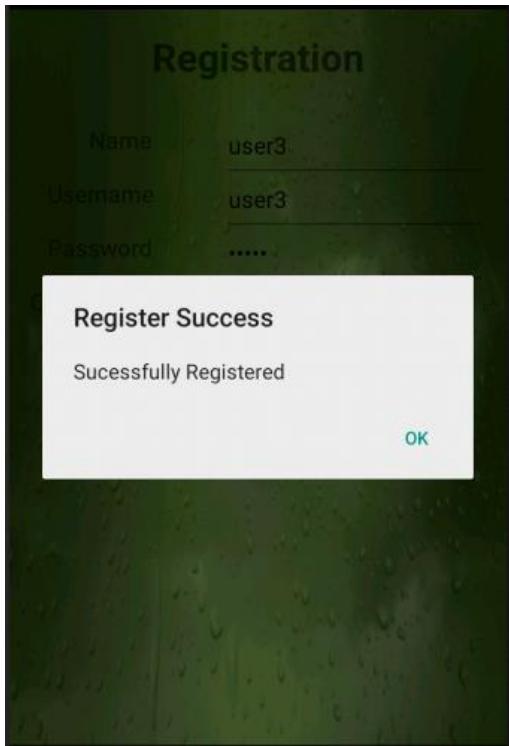


Registration page:

- The form filled with the details by the new user

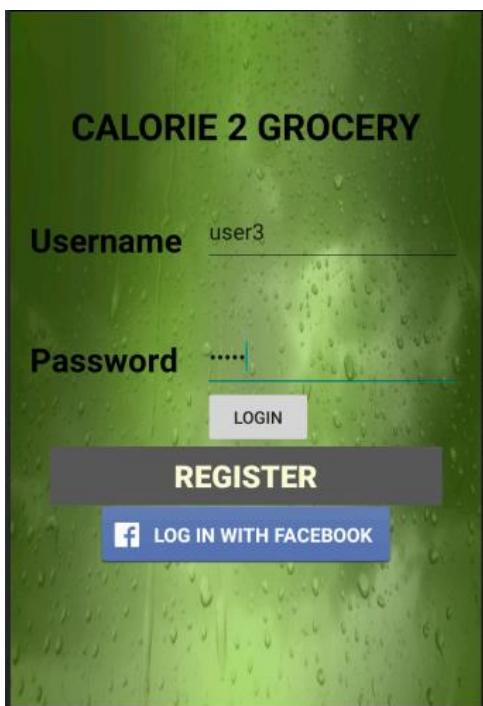
Technical Details:

- 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

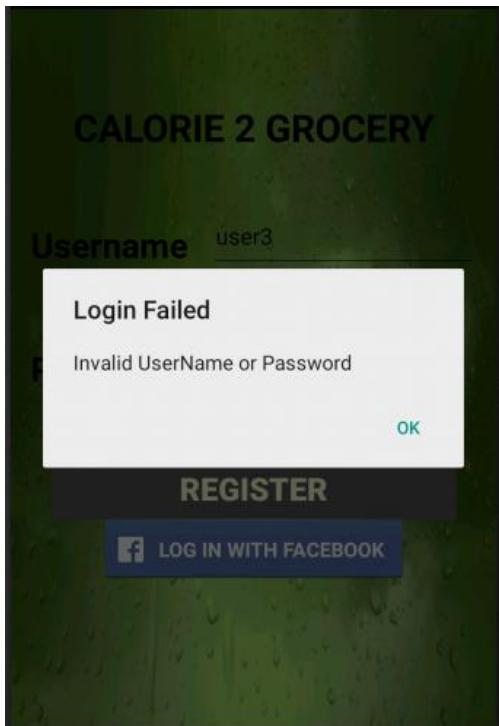


Home page:

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

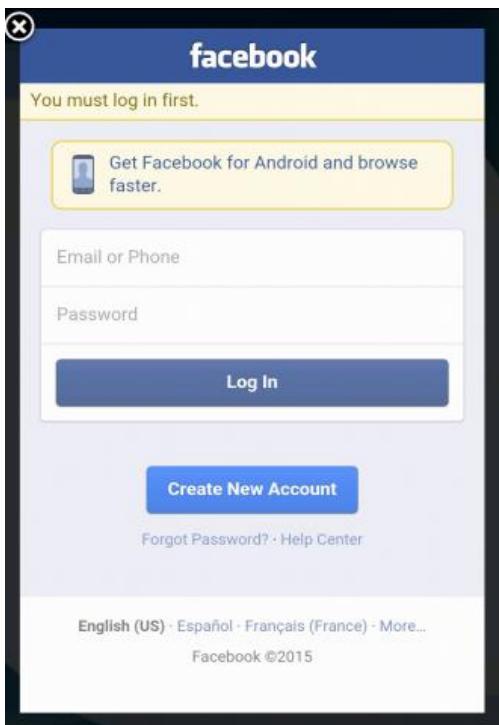
Technical Details:

- 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”

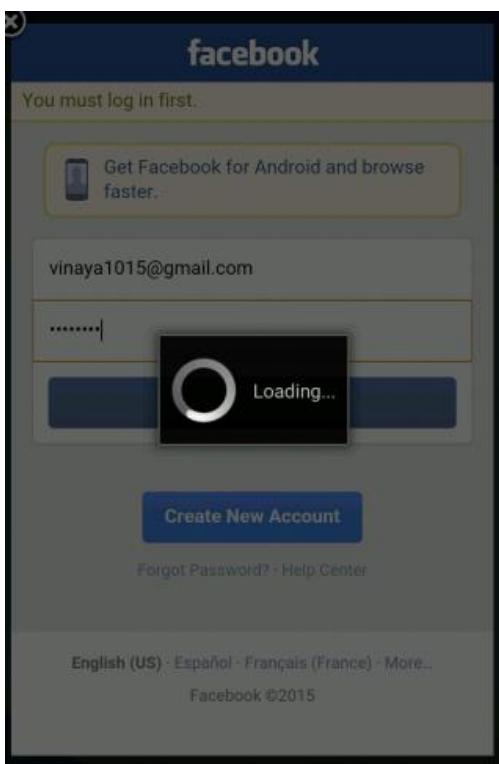


Facebook Login page:

- 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.



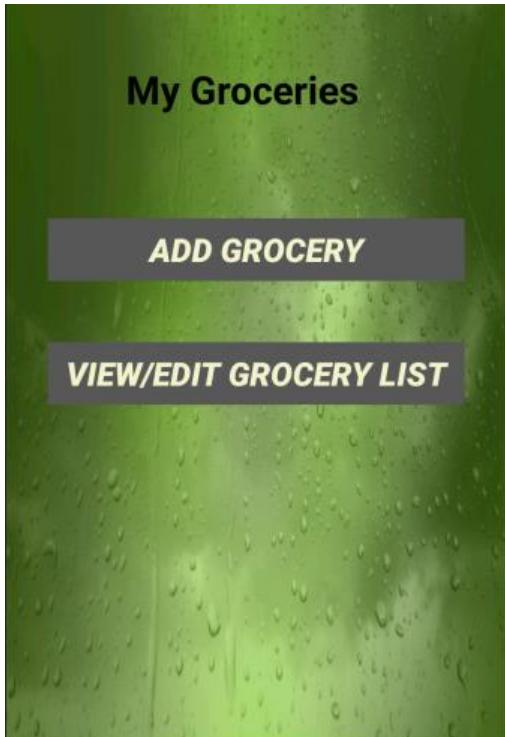
Facebook Login page:

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



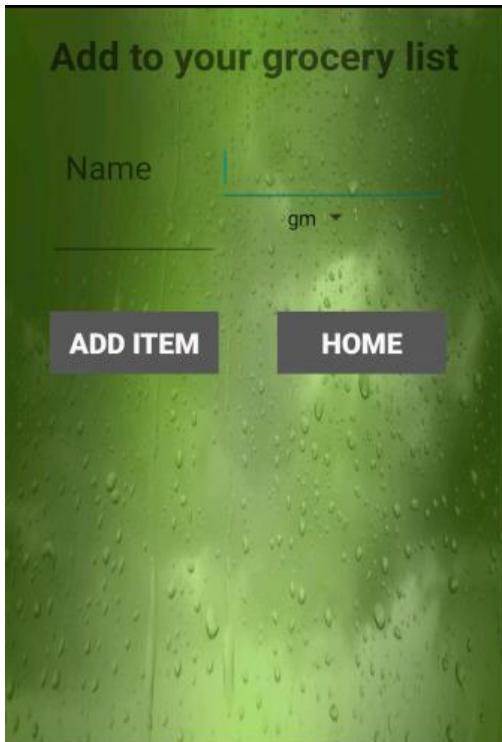
Home page:

- 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 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

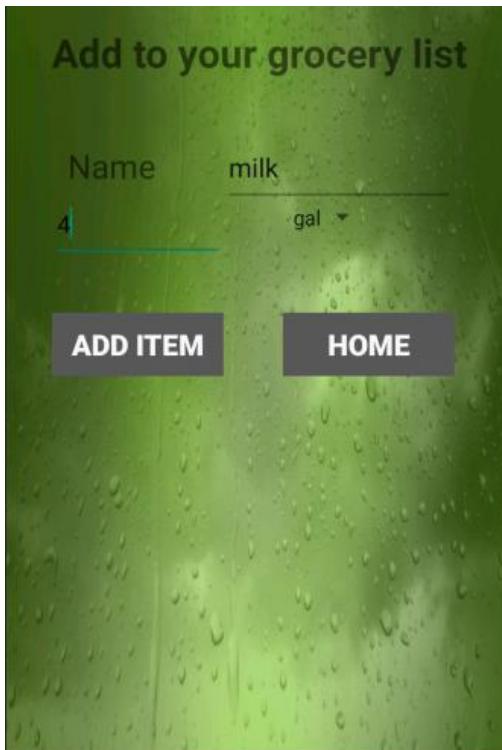


Add Grocery page:

- 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.

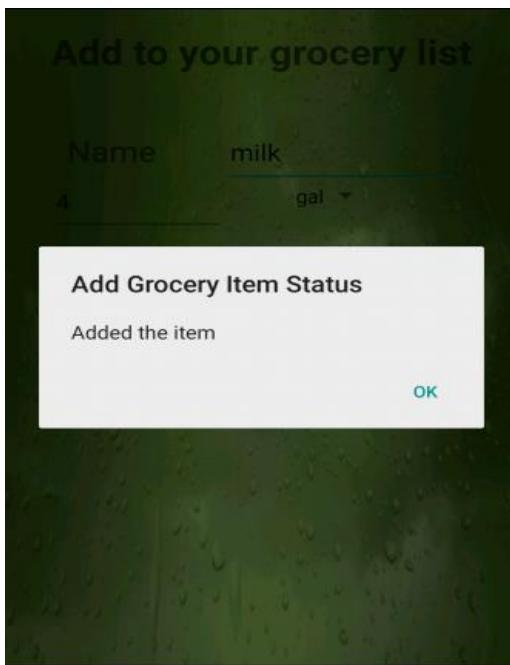


Add Grocery

- 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.

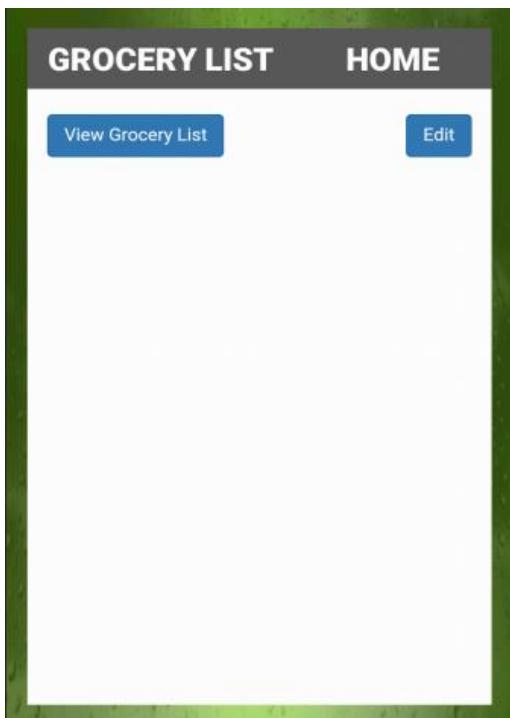


Add Grocery

- 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.

Technical Details:

- Web service to add the grocery items to database sends the result of the insert operation in the form of JSON.
- 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.

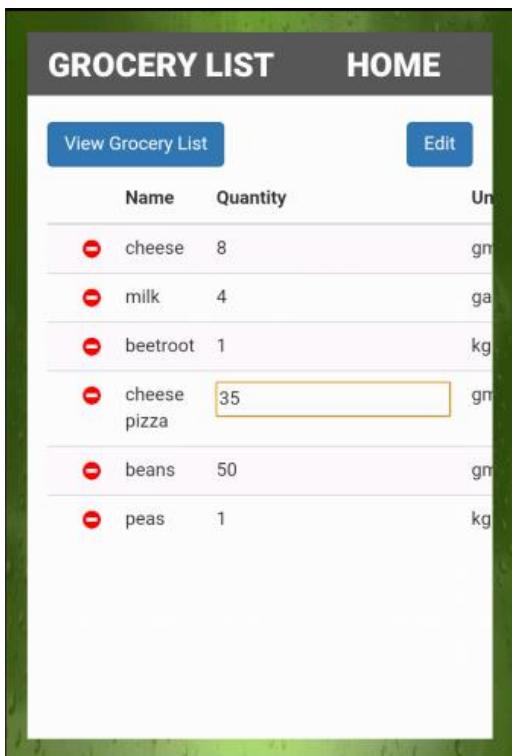
GROCERY LIST			HOME
View Grocery List		Edit	
Name	Quantity	Units	
cheese	8	gm	
milk	4	gal	
beetroot	1	kg	
cheese pizza	40	gm	
beans	50	gm	
peas	1	kg	

View Grocery List

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

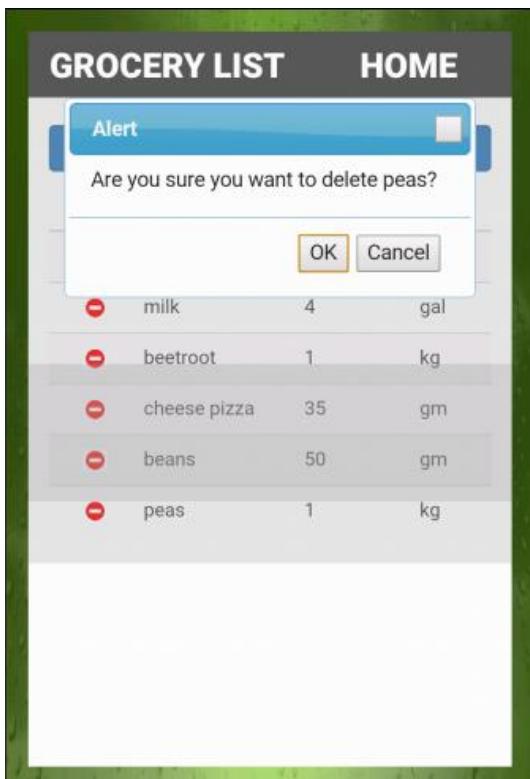
Technical Details:

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



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.

GROCERY LIST		HOME	
View Grocery List		Edit	
Name	Quantity	Units	
cheese	8	gm	
milk	4	gal	
beetroot	1	kg	
cheese pizza	35	gm	
beans	50	gm	

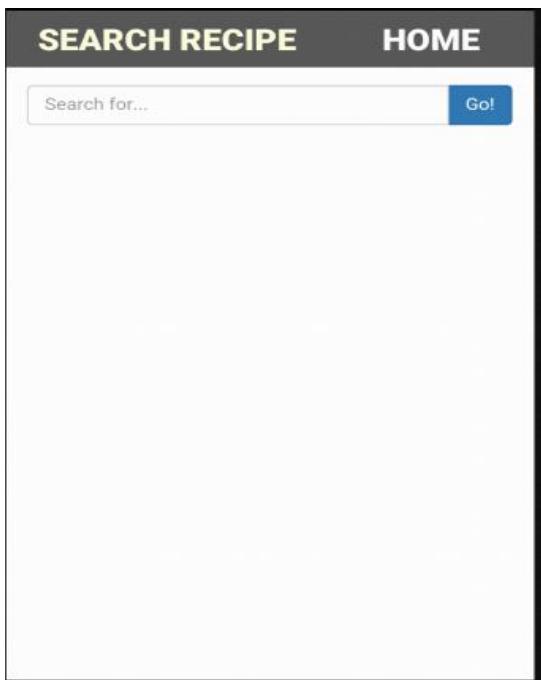
View Grocery List

- 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.

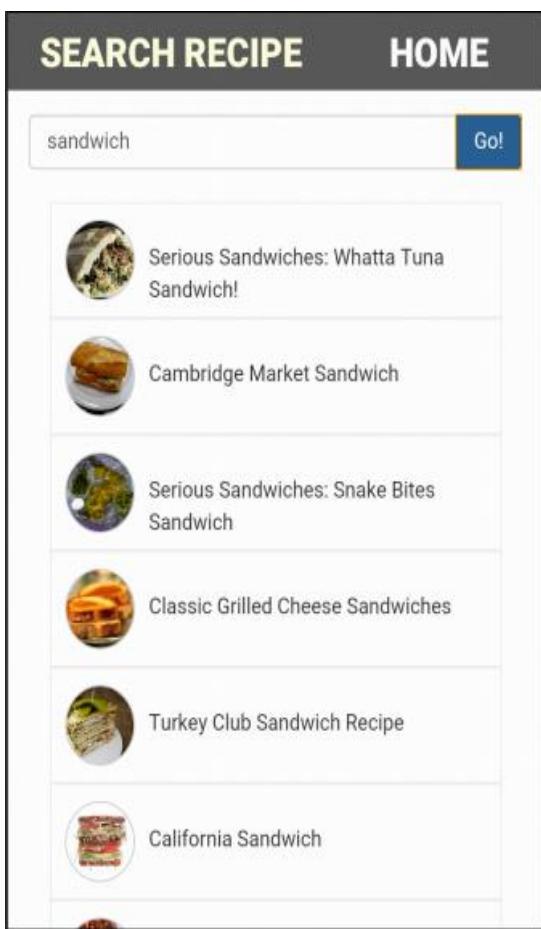


Search Recipe page:

- 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.

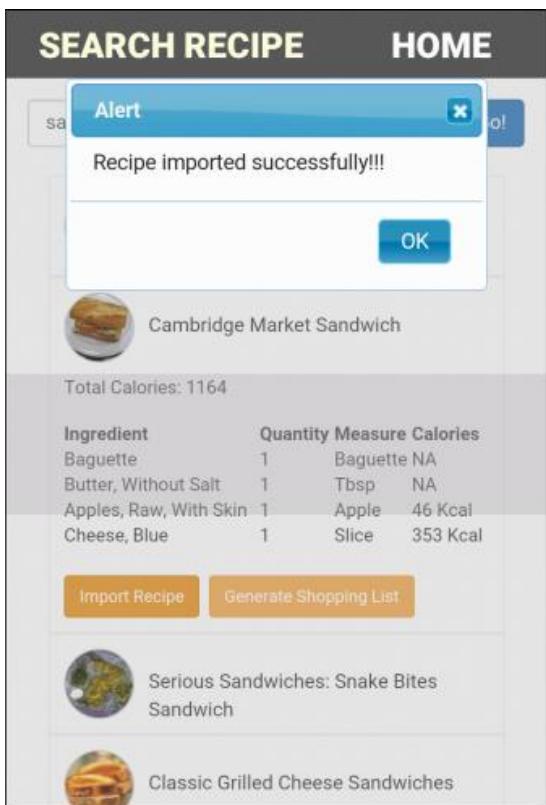


Search Recipe

- User enters sandwich as a search key to get various recipe on sandwich and clicks on GO button.
- It displays all the recipes and when user selects a particular recipe, the two buttons are displayed.
- Import recipe button: The user can store the particular recipe 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

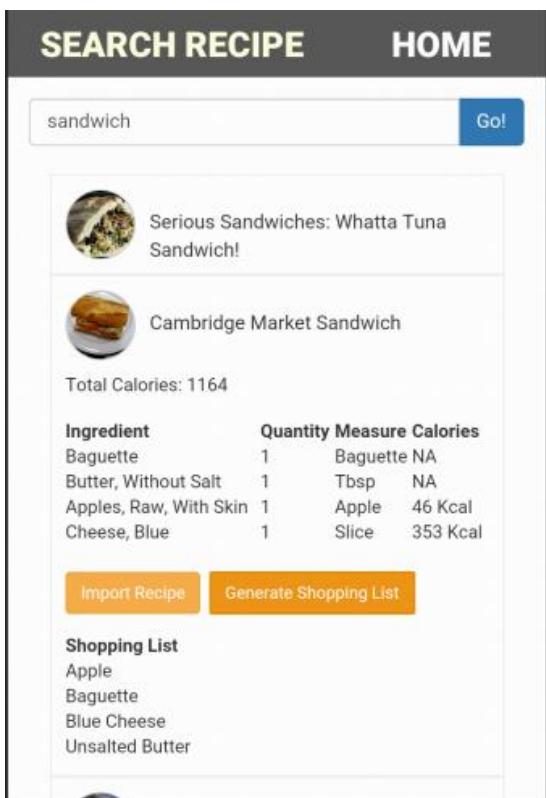


Search Recipe

- 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.



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 user interface for generating a shopping list. At the top left is a blue button labeled "View Shopping List". At the top right is a blue button labeled "Edit". Below these buttons is a section titled "Name" with a list of grocery items. The items listed are: black olives, blue cheese, butter, charred, cumin seeds, eggplant, garlic, ginger, lemon juice, oil, and olive oil. Each item is preceded by a small horizontal line.

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.

The screenshot shows a "CREATE RECIPE" form. At the top left is a button labeled "CREATE RECIPE" and at the top right is a button labeled "HOME". Below these buttons is a section titled "Create New Recipe". This section contains four input fields: "Recipe Name", "ingredient", "quantity", and "units". Below these fields is a button labeled "Add More". At the bottom of the section is a button labeled "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 **HOME**

Create New Recipe

orange juice
oranges
2
pieces

Add More

Save

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

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

CREATE RECIPE **HOME**

Create New Recipe

orange juice	
oranges	
2	
pieces	
Add More	
milk	100
ml	
sugar	50
gm	
water	20
ml	

Save

SHARE RECIPE **HOME**

Get Recipe

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.

Technical Details:

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

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

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.

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.



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.

A screenshot of a web-based application titled "UPDATE RECIPE" at the top left. To its right is a "HOME" link. Below the title, there is a text input field labeled "RecipeName". At the bottom of the screen, there are three blue rectangular buttons with white text: "Get Recipe", "Add", and "Delete".

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

UPDATE RECIPE **HOME**

mango juice

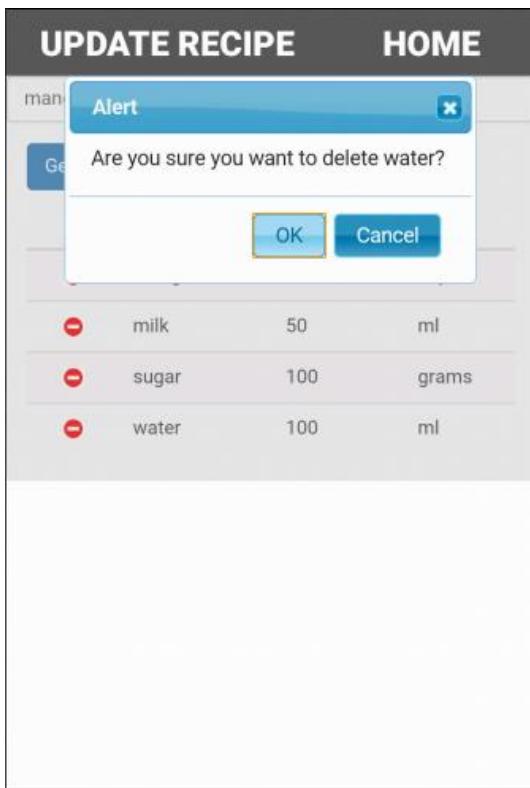
Get Recipe	Add	Delete
Ingredient	Quantity	Units
➡ mangoes	2	cups
➡ milk	50	ml
➡ sugar	100	grams
➡ water	100	ml

Update Recipe

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

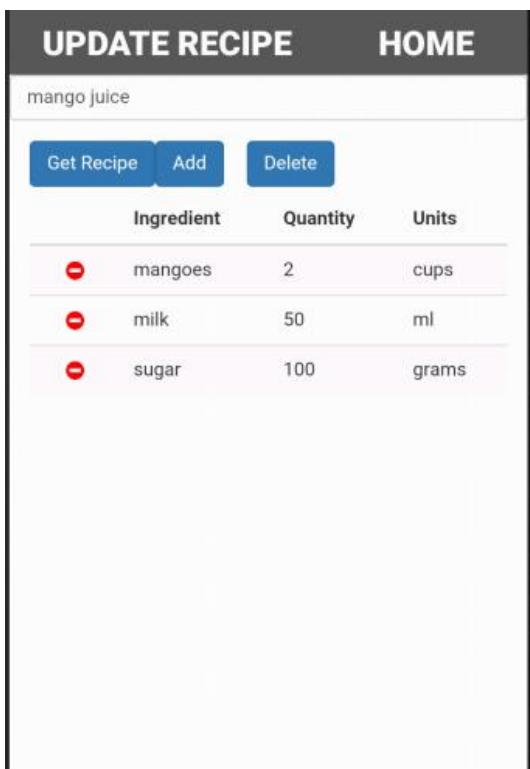
Technical Details:

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



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

- 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	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

Get Recipe	Add	Delete
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

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

Update Recipe

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

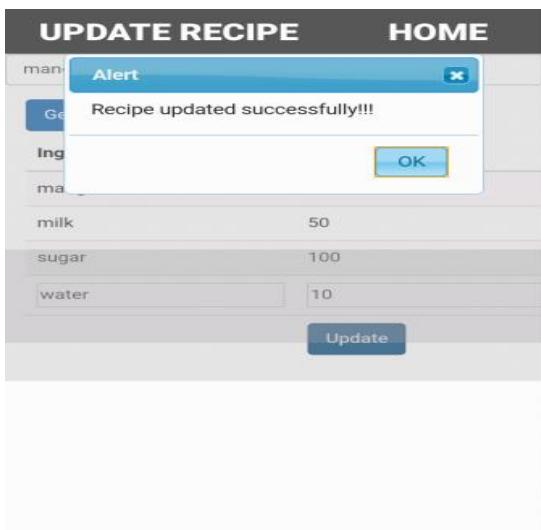
UPDATE RECIPE **HOME**

mango juice

<input type="button" value="Get Recipe"/>	<input type="button" value="Add"/>	<input type="button" value="Delete"/>
Ingredient	Quantity	
mangoes	2	
milk	50	
sugar	100	
<input type="text" value="water"/>	<input type="text" value="10"/>	
<input type="button" value="Update"/>		

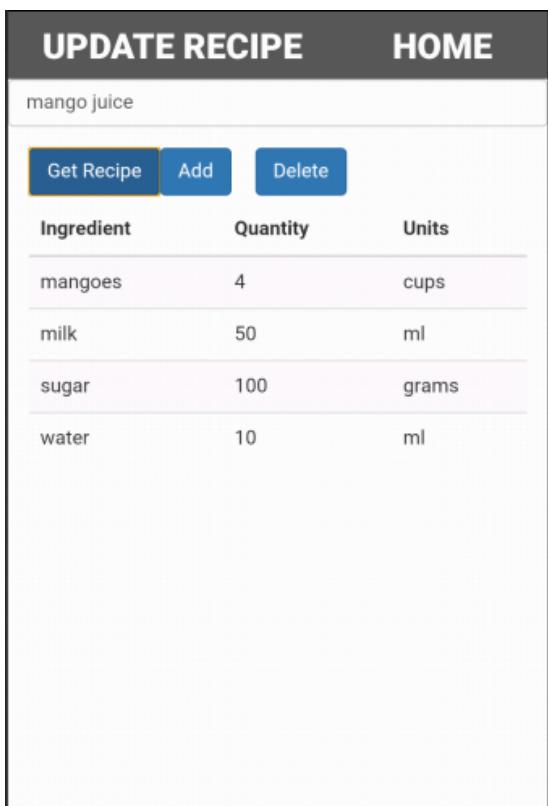
Update Recipe

- The ingredients are entered in the fields displayed.



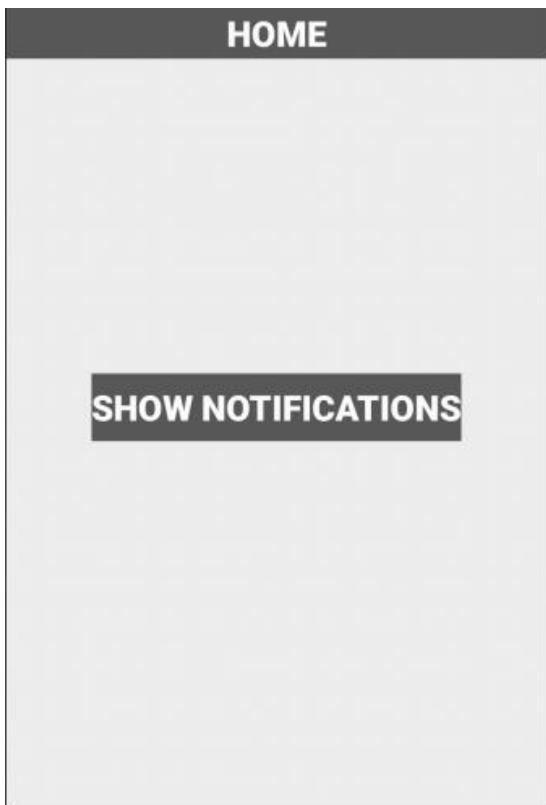
Update Recipe

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



Update Recipe

- The updated ingredients list for the recipe are displayed.

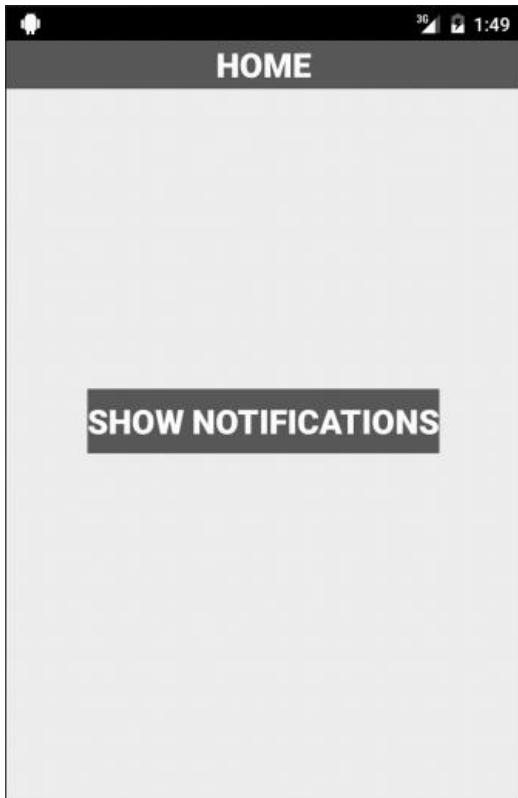


Show Notifications Page

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

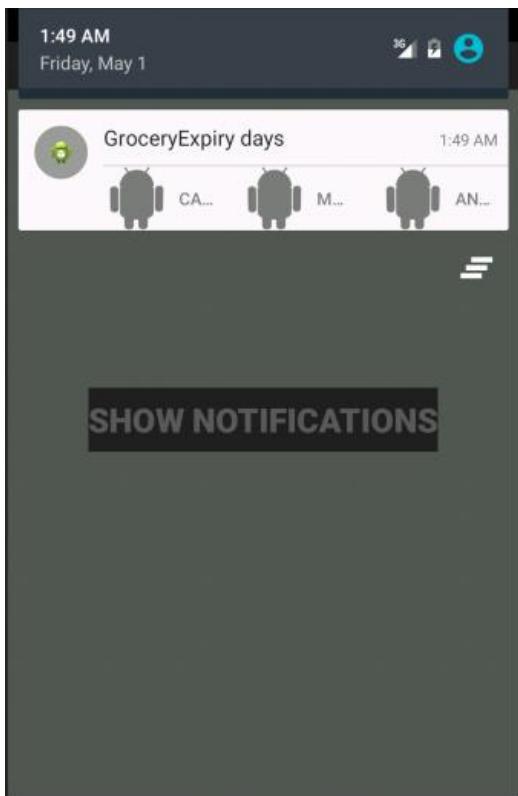
Technical Details:

- 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.



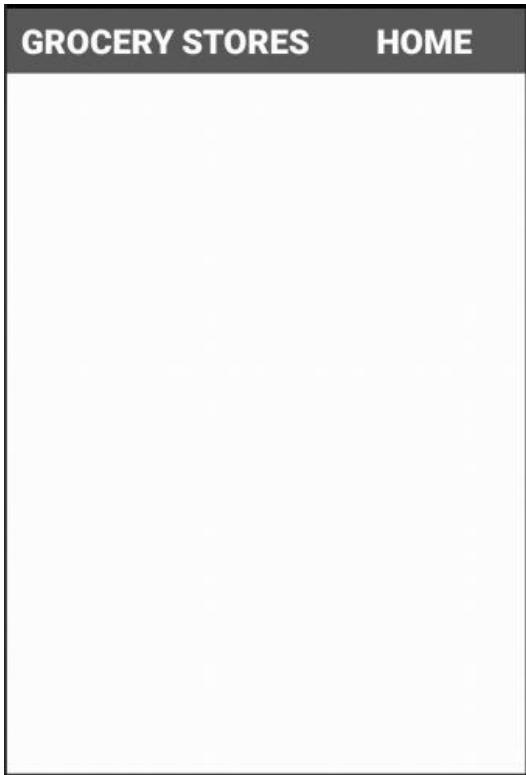
Show Notifications Page

- 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.

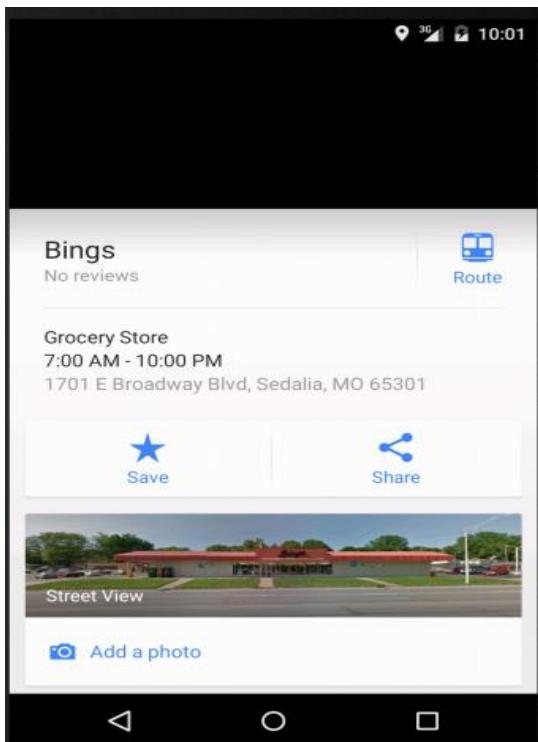


Grocery Stores Page

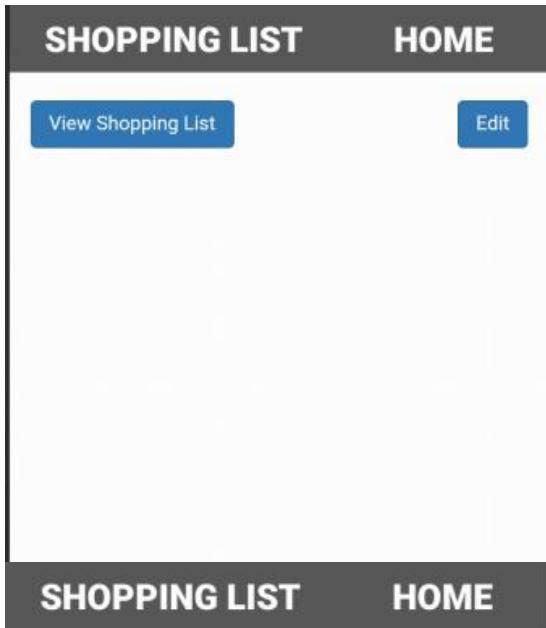
- The nearby grocery stores could be found by clicking the Grocery Stores button in Home page.

Technical Details:

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

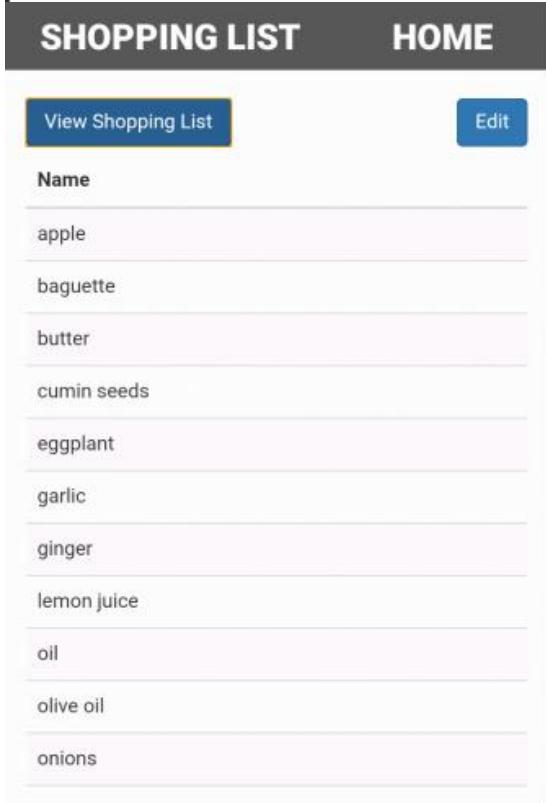


By Clicking on grocery stores, the application displays the nearby grocery stores as shown in the screenshot.



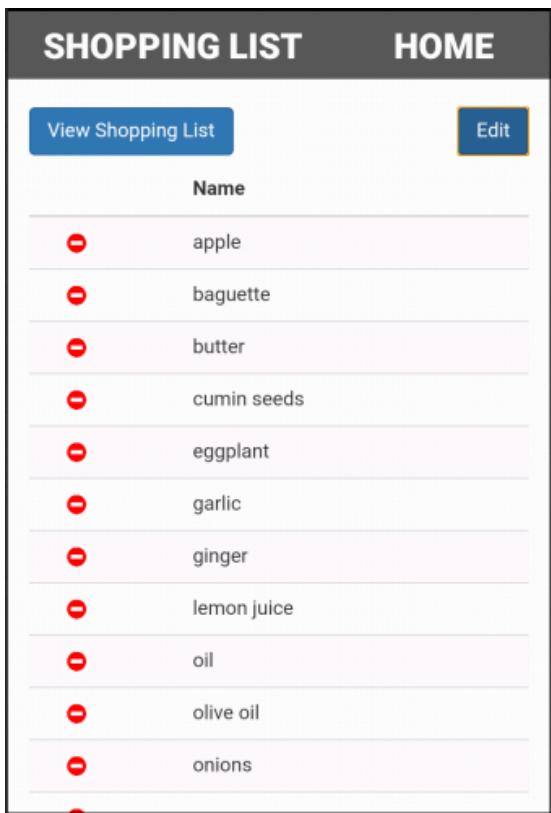
Shopping List Page

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



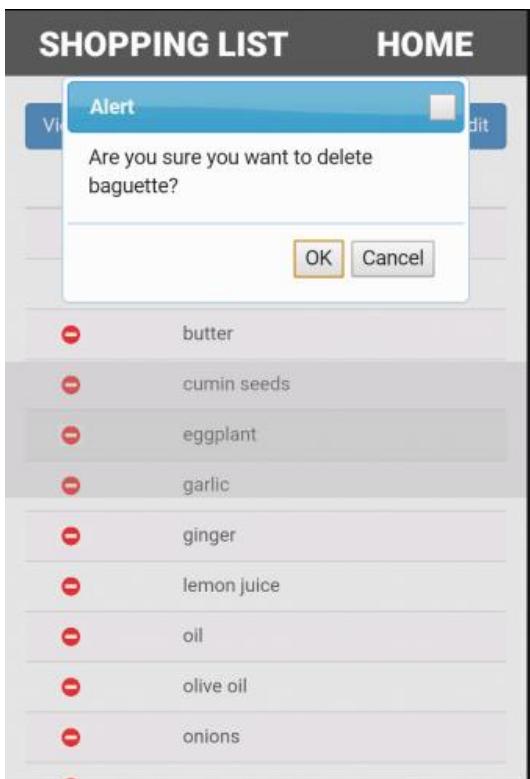
Shopping List Page

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



Shopping List 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.

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

Shopping List Page

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

3) 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.

	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.

```
SQLQuery1.sql...S\spnts5 (52)*
select * from grocery_list;
```

The screenshot shows a SQL query window with the command "select * from grocery_list;". Below it is a results grid with 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\spnts5 (52)*
select * from recipes;
```

The screenshot shows a SQL query window with the command "select * from recipes;". Below it is a results grid with the following data:

	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\spnts5 (52)*
select * from shopping_list;
```

The screenshot shows a SQL query window with the command "select * from shopping_list;". Below it is a results grid with the following data:

	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 chili 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.

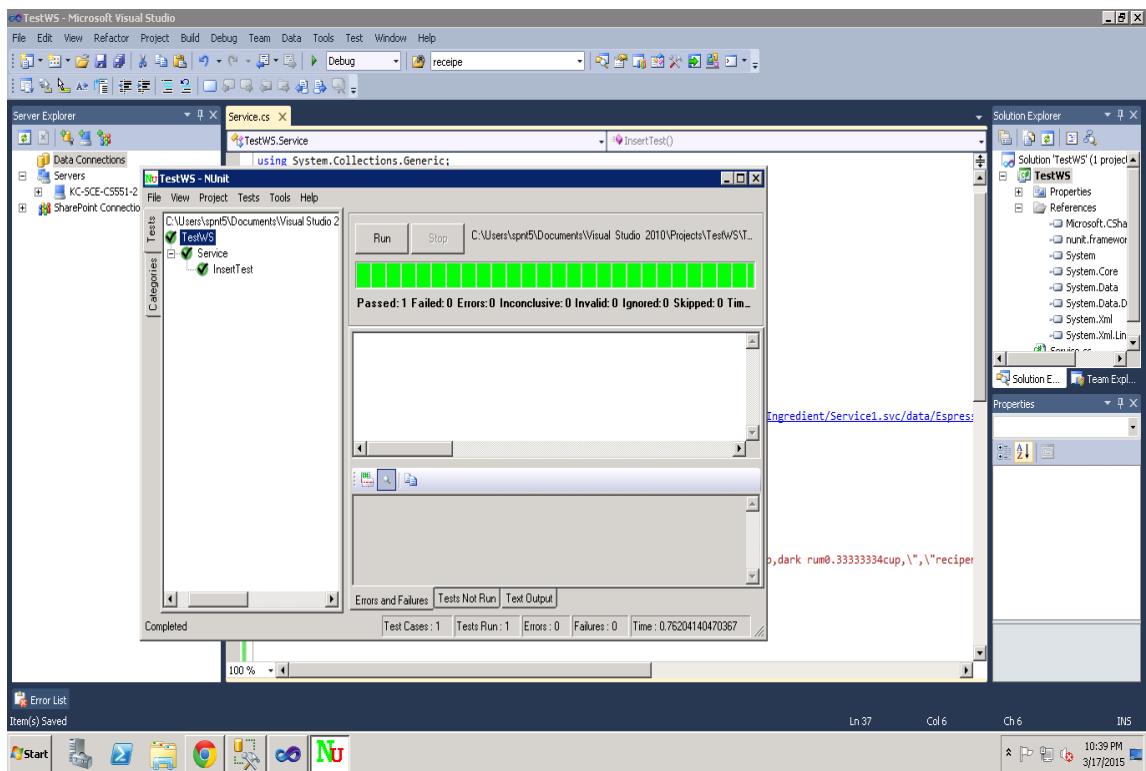
Page	Button	Expected	Actual
View Grocery list	view	Should display grocery list	Displaying grocery list
	edit	Should modify the ingredients, quantity and label	Able to modify the ingredients, quantity and label
	delete	Should delete the grocery items	Deleted the grocery items
Calorie count	Browse for 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

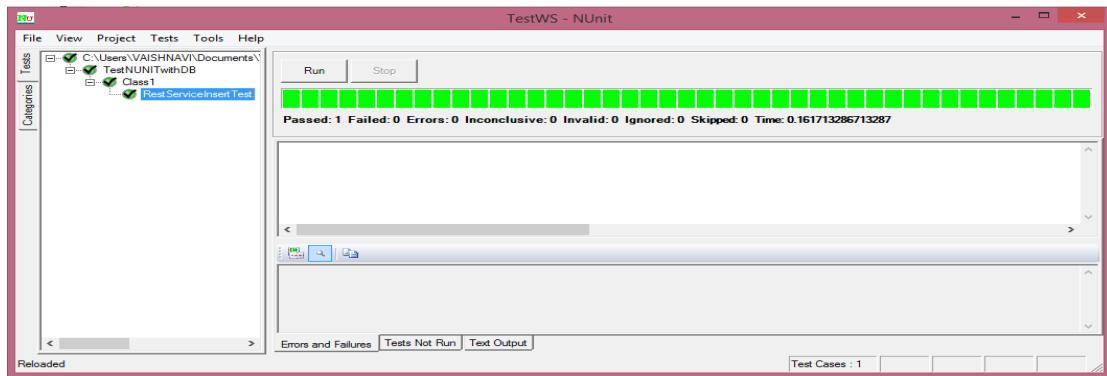
2) Data Validation:

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.

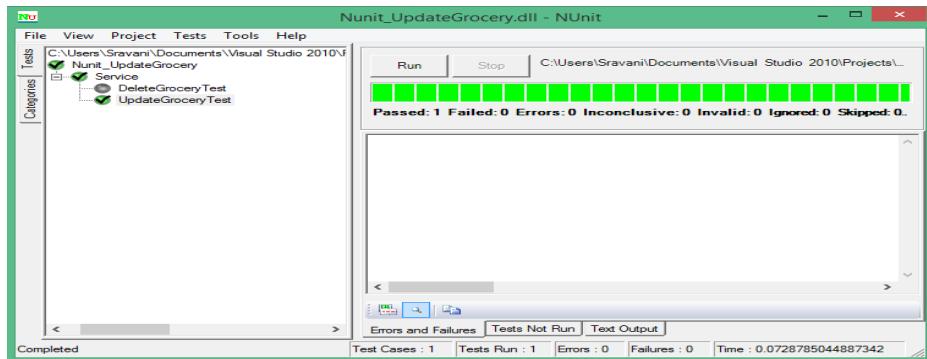
Screenshot of N-Unit validation: This is used to validate the data for Import recipe API:



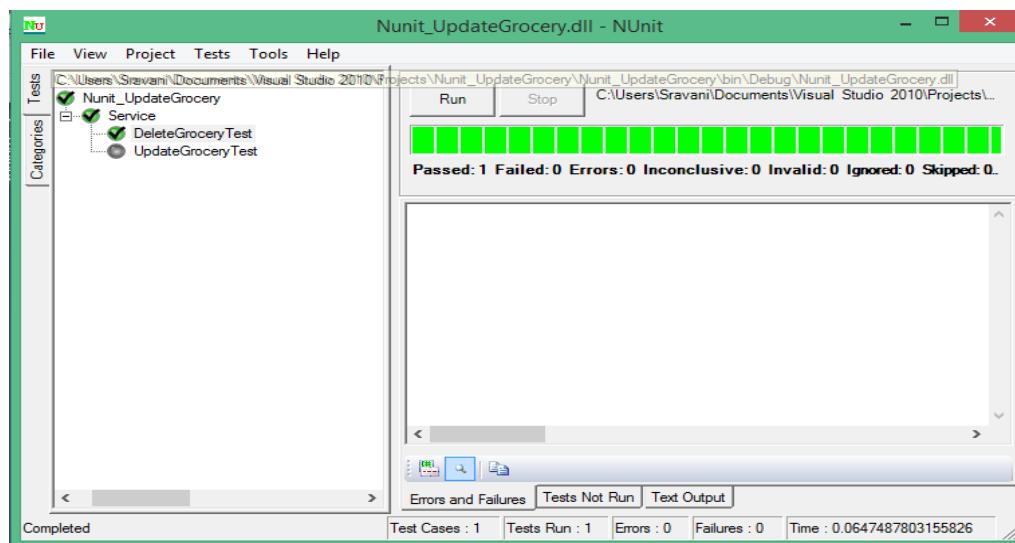
Screenshot for N-Unit validation of Add groceries API:



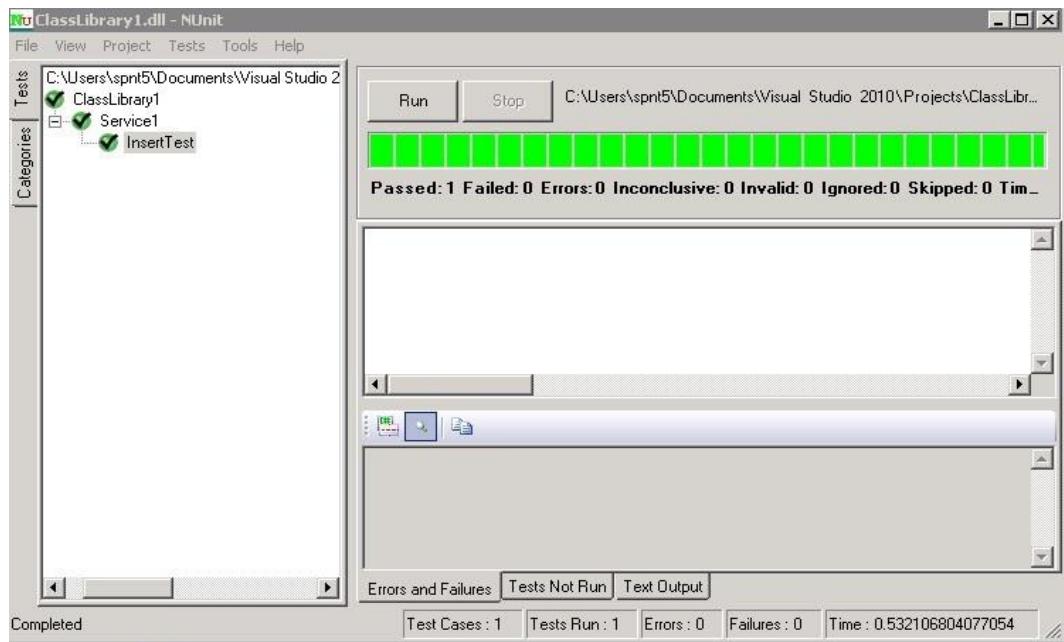
Screenshot of N-Unit validation: This is used to validate the data for update grocery item method in the rest service.



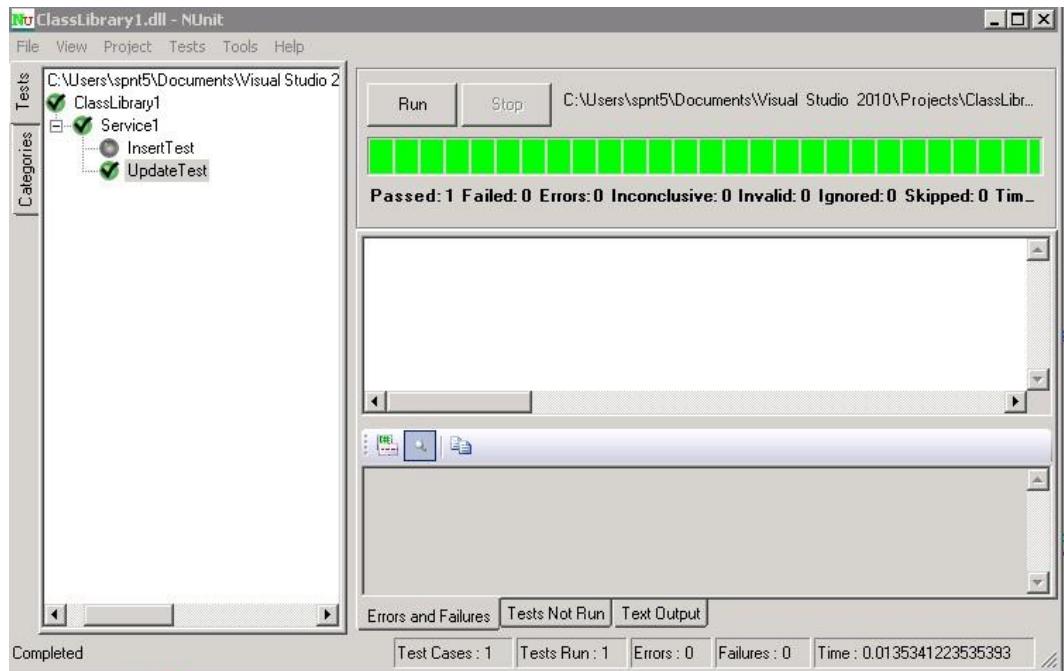
Screenshot for N-Unit validation of delete groceries API:



Screenshot for N-Unit validation of Import groceries API:



Screenshot for N-Unit validation of Update groceries API:

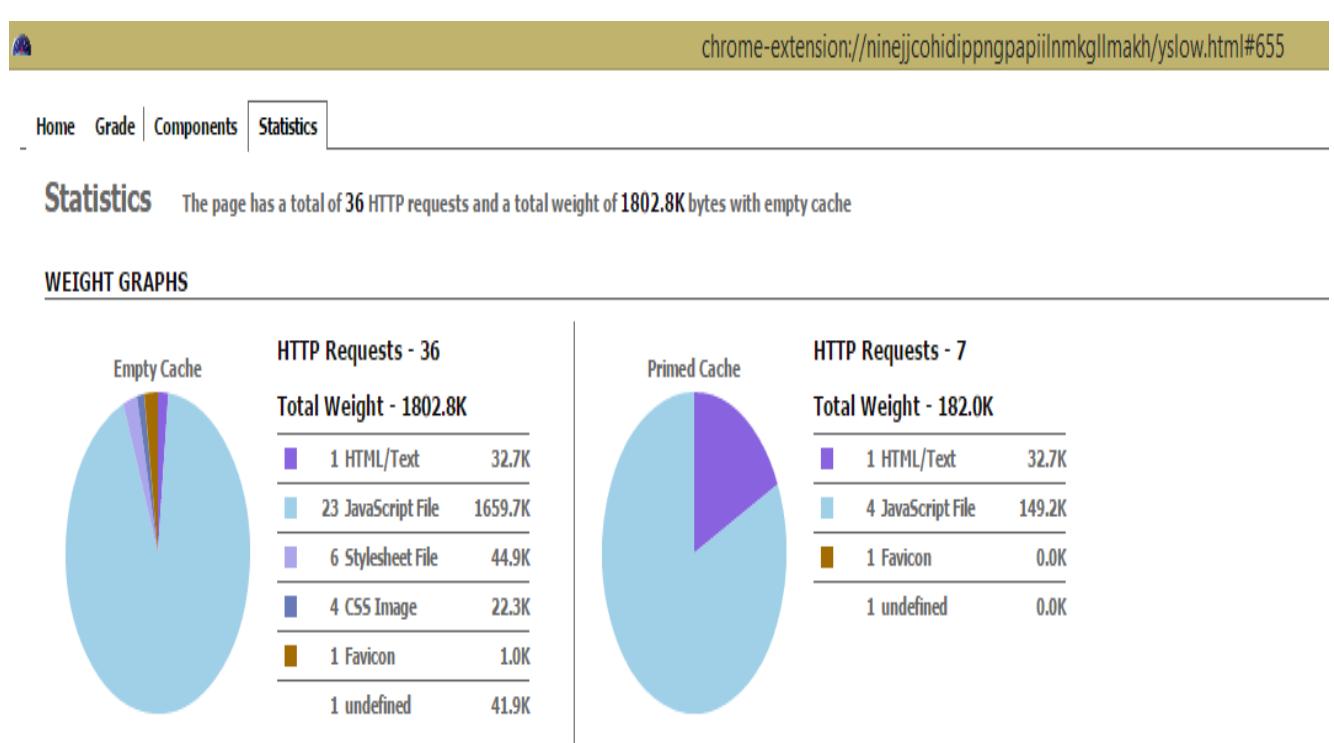


Performance Testing:

The following are the tools we used for performance testing:

- YSlow analyzer
 - Firebug
2. 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://ninejjcohidipppngpapiiinmkglmakh/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 \(6\)](#)

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

F Make fewer HTTP requests

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

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://ninejjcohidipppngpapiiinmkglmakh/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](#)

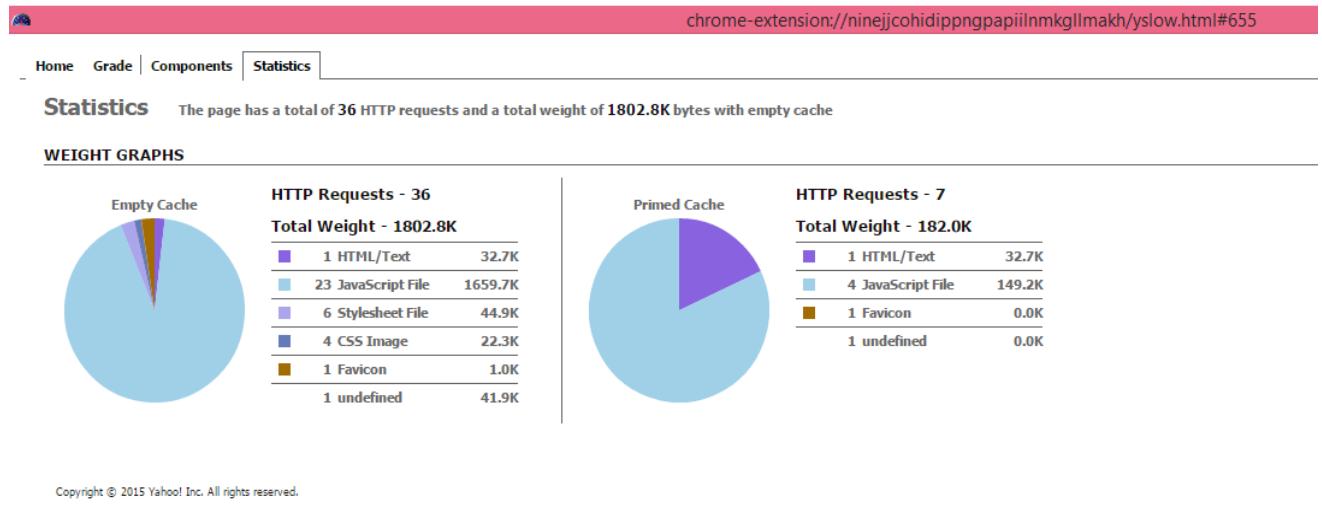
TYPE	SIZE (KB)	GZIP (KB)	COOKIE RECEIVED (bytes)	COOKIE SENT (bytes)	HEADERS	URL	EXPIRES (Y/H/D)	RESPONSE TIME (ms)	ETAG	ACTION
+ doc (1)	32.7K									
+ js (23)	1684.5K									
+ css (6)	44.9K									
+ csimage (4)	22.3K									
+ favicon (1)	1.0K									
+ font (1)	41.9K									

* 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



Screen Shot: 1

The screenshot shows the YSlow Components report for a web page. At the top, it says "chrome-extension://ninejjcohidippngpapiilnmkgllmakh/yslow.html#655". Below that, there are tabs: "Home", "Grade", "Components", and "Statistics". The "Components" tab is selected.

Components The page has a total of 37 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)	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://ninejcohidippingpapiiinmkglmakhy/yslow.html#655

Home Grade Components Statistics

Grade C Overall performance score 74 Ruleset applied: YSlow(V2) URL: http://jsfiddle.net/laGh6/yfsdp3mq/1/

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

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:

[Read More](#)

Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 3

Testing for a web page for Creating a recipe

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

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

Empty Cache	Primed Cache
HTTP Requests - 36 Total Weight - 1802.8K <ul style="list-style-type: none"> 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 	HTTP Requests - 7 Total Weight - 182.0K <ul style="list-style-type: none"> 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://ninejjcohidipngpapiinmkglmakh/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)	32.5K									
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://ninejjcohidipngpapiinmkglmakh/yslow.html#655

Home Grade Components Statistics

Grade C Overall performance score 74 Ruleset applied: YSlow(V2) URL: http://jsfiddle.net/vp813/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.
This page has 6 external stylesheets. Try combining them into one.

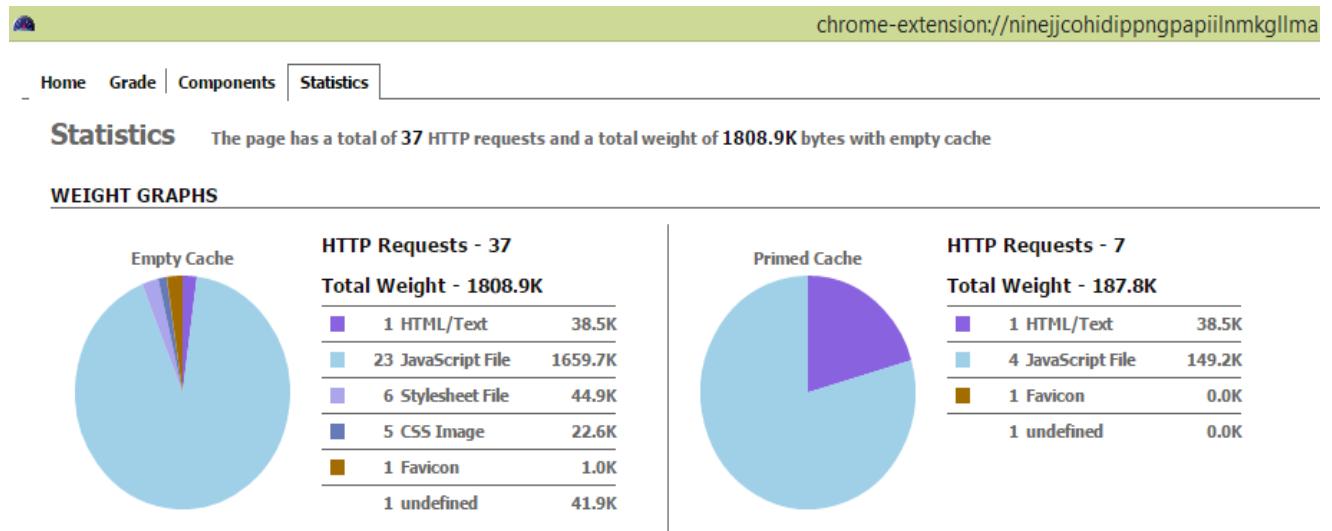
Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet,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

[»Read More](#)

Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 3

Web Page for displaying a recipe long with calorie count



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)	38.5K									
doc	38.5K	38.5K		665	ρ	http://jsfiddle.net/b6h6/q6y2smg/8/	no expires	0		
js (23)	1684.5K									
css (6)	44.9K									
css	0.6K	0.6K		665	ρ	http://jsfiddle.net/css/normalize.css?_	2015/5/3	0		
css	0.2K	0.2K		665	ρ	http://jsfiddle.net/css/tricks.css?_	2015/5/3	0		
css	18.7K	18.7K		665	ρ	http://jsfiddle.net/css/elusive-webfont.css?_	2015/5/3	0		
css	8.0K	8.0K		665	ρ	http://jsfiddle.net/css/codemirror/lib/codemirror.css?_	2015/5/3	0		
css	17.2K	17.2K		665	ρ	http://jsfiddle.net/css/screen.css?_	2015/5/3	0		
css	0.0K			665	ρ	http://jsfiddle.net/css/light.css?_	2015/5/3	0	"5120150e-0"	
cssimage (5)	22.6K									
cssimage	20.8K			665	ρ	http://jsfiddle.net/img/initializing.png	2015/5/3	0	"5128cfbb-515f"	
cssimage	1.1K			665	ρ	http://jsfiddle.net/img/logo.png	2015/5/3	0	"5120150e-488"	
cssimage	0.2K			665	ρ	http://jsfiddle.net/img/remove-resources.png	2015/5/3	0	"5120150e-120"	
cssimage	0.2K			665	ρ	http://jsfiddle.net/img/handle-h.png	2015/5/3	0	"5120150e-ca"	
cssimage	0.1K			665	ρ	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://ninejjcohidippngpapiilmkgllmakh/vslow.html#757

Screen Shot: 2

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

Grade C Overall performance score 74 Rule set applied: YSlow(V2) URL: http://jsfiddle.net/la6h6/gfory2smg/8/

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

!/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://ninejjcohidippngpapiiIlnmkglm

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

Empty Cache

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

Primed Cache

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

Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 1

TYPE	SIZE (KB)	GZIP (KB)	COOKIE RECEIVED (bytes)	COOKIE SENT (bytes)	HEADERS	URL	EXPIRES (Y/H/D)	RESPONSE TIME (ms)	ETAG	ACTION
doc (1)	34.7K									
doc	34.7K	34.7K		665	⌚	http://jsfiddle.net/spn15/n6rl5r3s/2/	no expires	0		
+ 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 (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-48b"	
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: 2

Home | Grade | Components | Statistics
Rulesets: YSlow(v2) | Edit | Help |

Grade C Overall performance score 74 Ruleset applied: YSlow(v2) URL: <http://jsfiddle.net/spn15/n6rl5r3s/2/>

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
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.

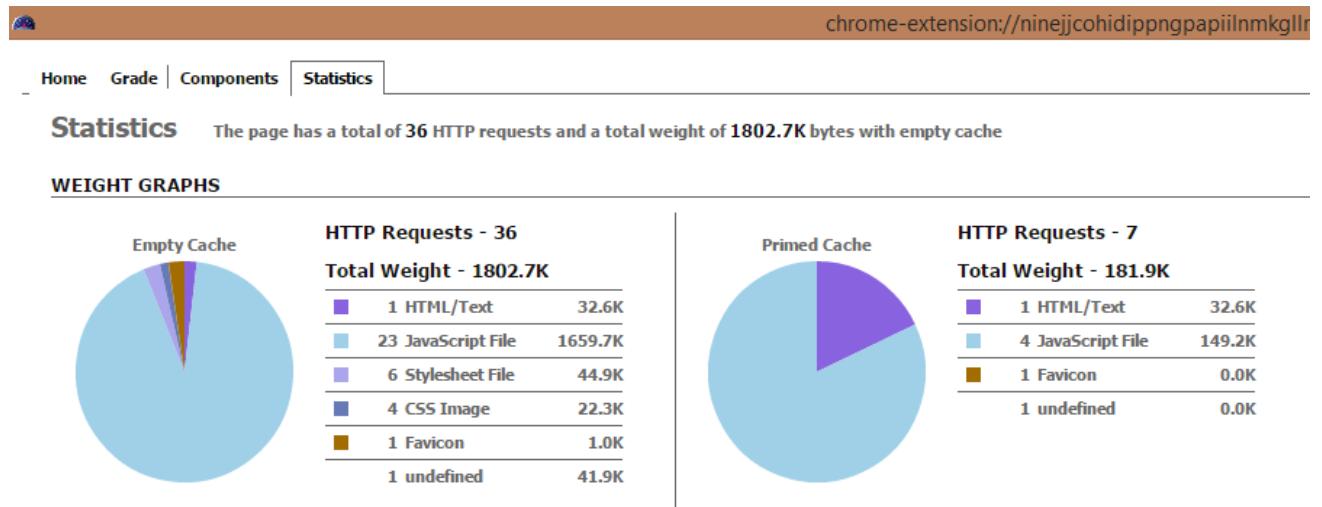
Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet,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

[Read More](#)

Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 3

Testing a web page for updating a recipe



Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 1

The screenshot shows the YSlow extension's Components tab. It lists 36 components with the following details:

TYPE	SIZE (KB)	GZIP (KB)	COOKIE RECEIVED (bytes)	COOKIE SENT (bytes)	HEADERS	URL	EXPIRES (Y/H/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/css/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.

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

Screen Shot: 2

chrome-extension://ninejjcohidippngpapiilmkgllm/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/la6h6/9ywq2u2n/1/

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

[Twitter](#) [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.

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://ninejjcohidippngpapiilmkgllm/yslow.html#757

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

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

Primed Cache

Category	Count	Total Weight
HTML/Text	1	31.2K
JavaScript File	4	149.2K
Favicon	1	0.0K
undefined	1	0.0K

Copyright © 2015 Yahoo! Inc. All rights reserved.

Screen Shot: 1

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

Home Grade Components Statistics

Components The page has a total of **36** components and a total weight of **1801.3K** 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)	31.2K									
js (23)	1684.5K									
css (6)	44.9K									
css	0.6K	0.6K		667	⌚	http://jsfiddle.net/css/normalize.css?...	2015/5/4	0		
css	0.2K	0.2K		667	⌚	http://jsfiddle.net/css/tricks.css?...	2015/5/4	0		
css	18.7K	18.7K		667	⌚	http://jsfiddle.net/css/elusive-webfont.css?...	2015/5/4	0		
css	8.0K	8.0K		667	⌚	http://jsfiddle.net/js/codemirror/lib/codemirror.css?...	2015/5/4	0		
css	17.2K	17.2K		667	⌚	http://jsfiddle.net/css/screen.css?...	2015/5/4	0		
css	0.0K			667	⌚	http://jsfiddle.net/css/light.css?...	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									
font	41.9K			667	⌚	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://ninejcohidipngpapiinmkglmakh/yslow.html#1008

Home Grade Components Statistics

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

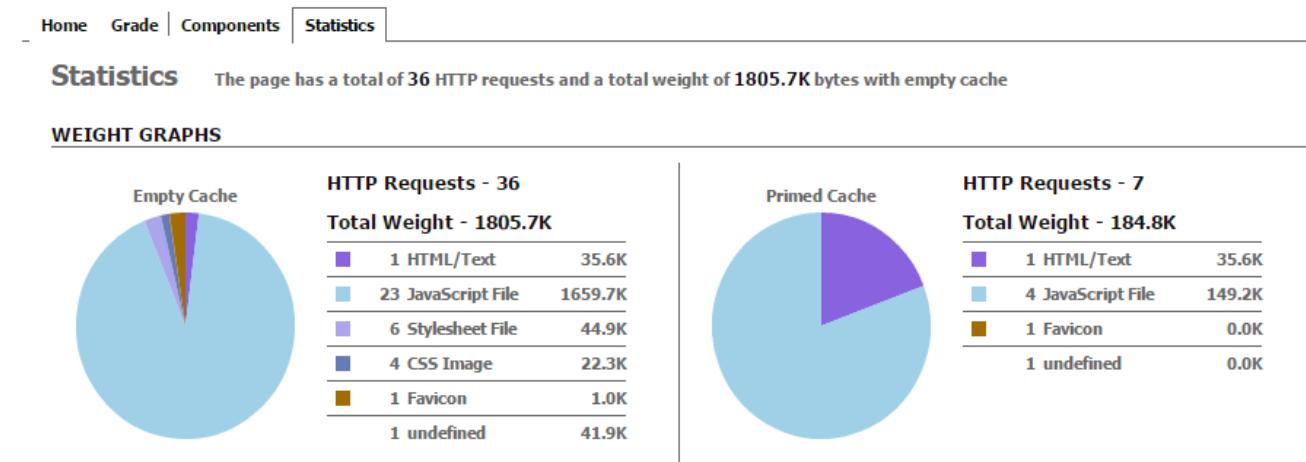
Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, 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://ninejcohidippngpapiinmkglmakh/yslow.html#1008

Rulesets YSlow(V2) Edit Help »Expand All

Home Grade Components Statistics

Components The page has a total of 36 components and a total weight of 1805.7K 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)	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 for a page with an overall performance score of 74. The audit results are categorized by component: Content (2), Cookie (2), CSS (6), Images (2), JavaScript (4), and Server (6). The main section is titled 'Grade F on Make fewer HTTP requests' and lists various recommendations for reducing the number of requests, such as using a Content Delivery Network (CDN), avoiding empty src or href attributes, and compressing components with gzip. A note states that the page has 23 external JavaScript scripts and 6 external stylesheets, suggesting they be combined. The interface includes a sidebar with navigation links like Home, Grade, Components, and Statistics, and a top bar with options like Rulesets, Edit, and Help.

Screen Shot: 3

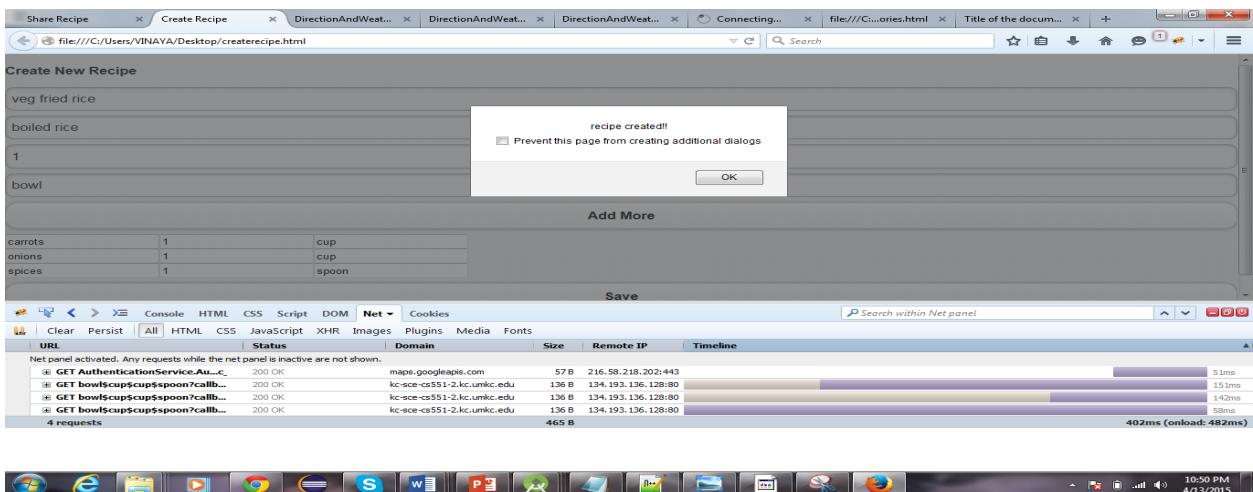
Fire Bug tool:

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.

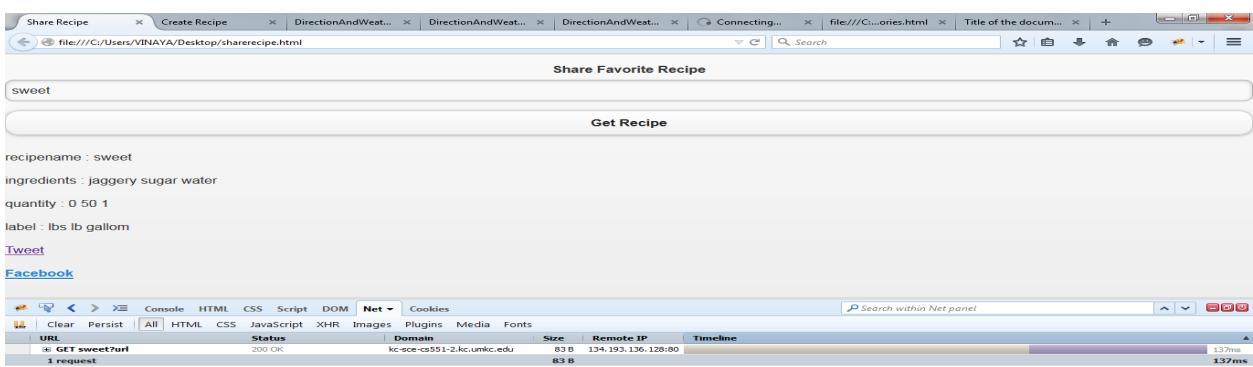
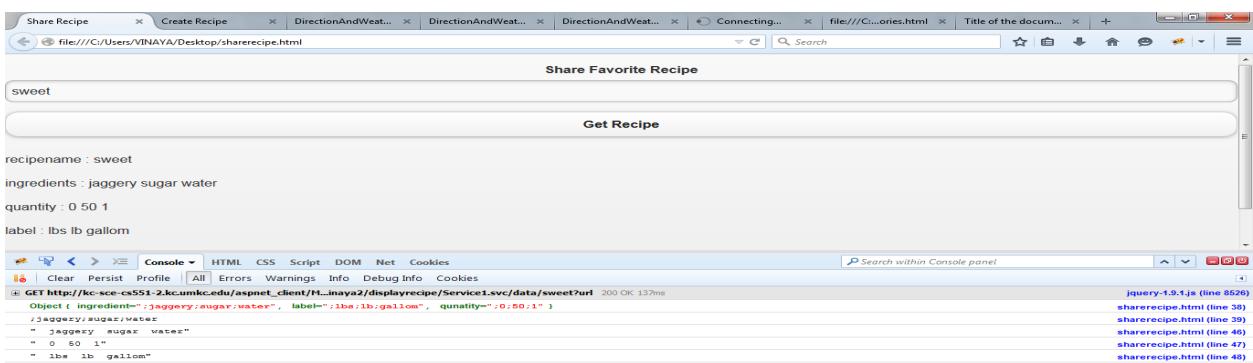
9. Createrecipe.html

The screenshot shows a browser window with multiple tabs open. The active tab displays a form titled 'Create New Recipe' for creating a dish. The form fields include 'veg fried rice' (name), 'boiled rice' (description), '1' (quantity), and 'bowl' (unit). Below the form is a table for ingredients: carrots (1 cup), onions (1 cup), and spices (1 spoon). A 'Save' button is visible at the bottom right of the form area. A confirmation dialog box is overlaid on the page, stating 'recipe created!' with an 'OK' button. At the bottom of the browser window, the Firebug toolbar is visible, showing tabs for Console, HTML, CSS, Script, DOM, Net, Cookies, and various status indicators like Errors, Warnings, Info, Debug Info, and Cookies.

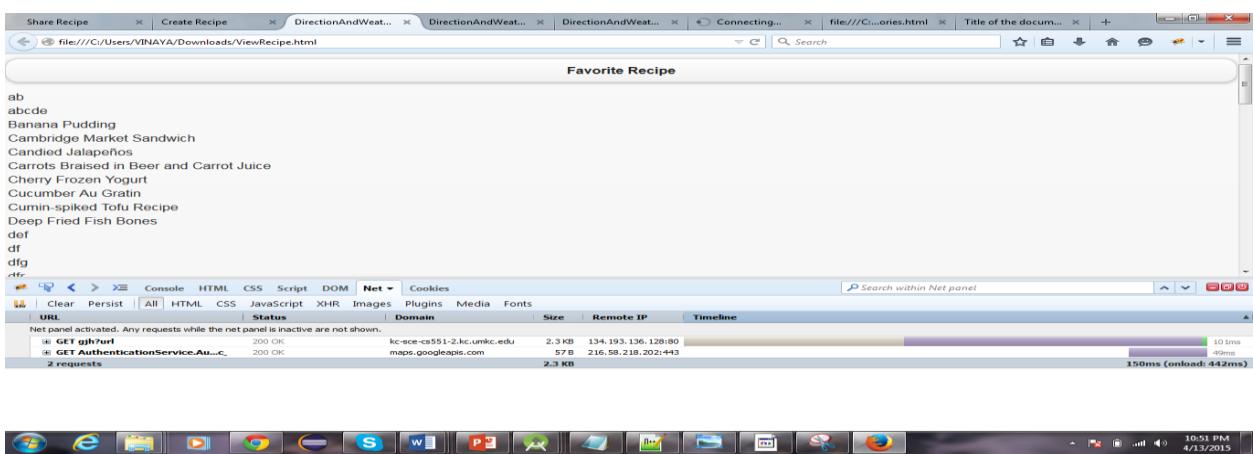
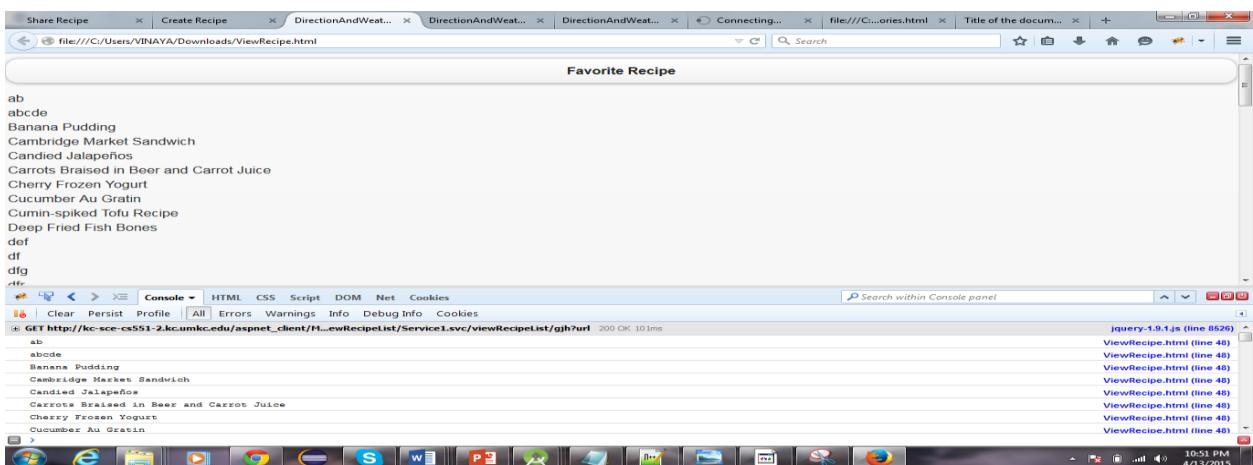




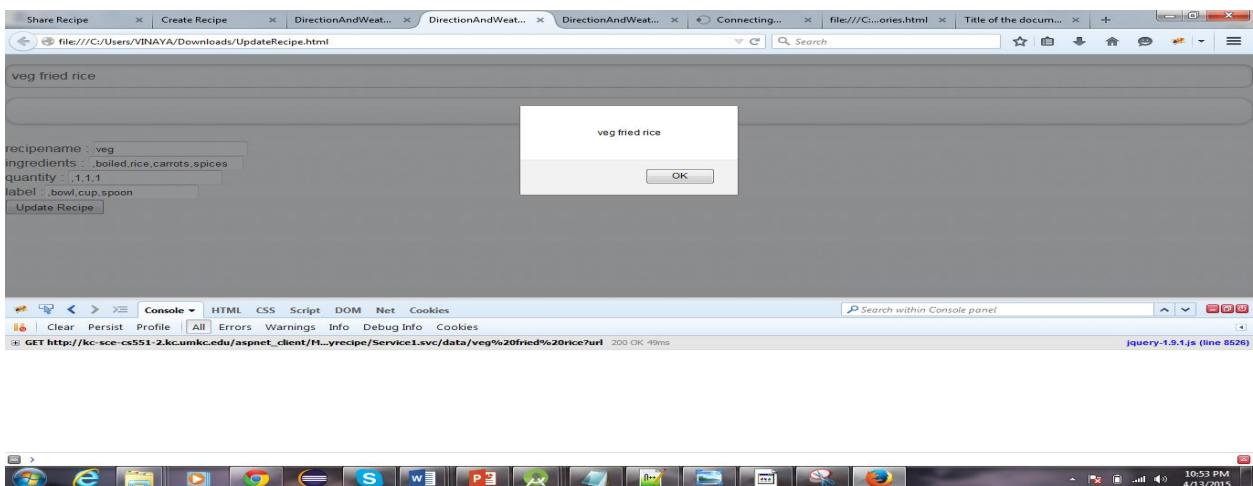
10. Share Recipe.html

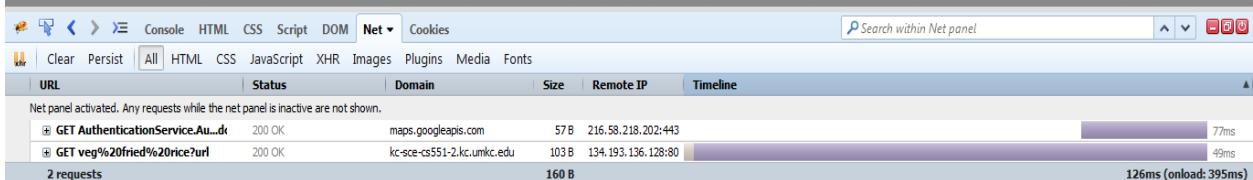
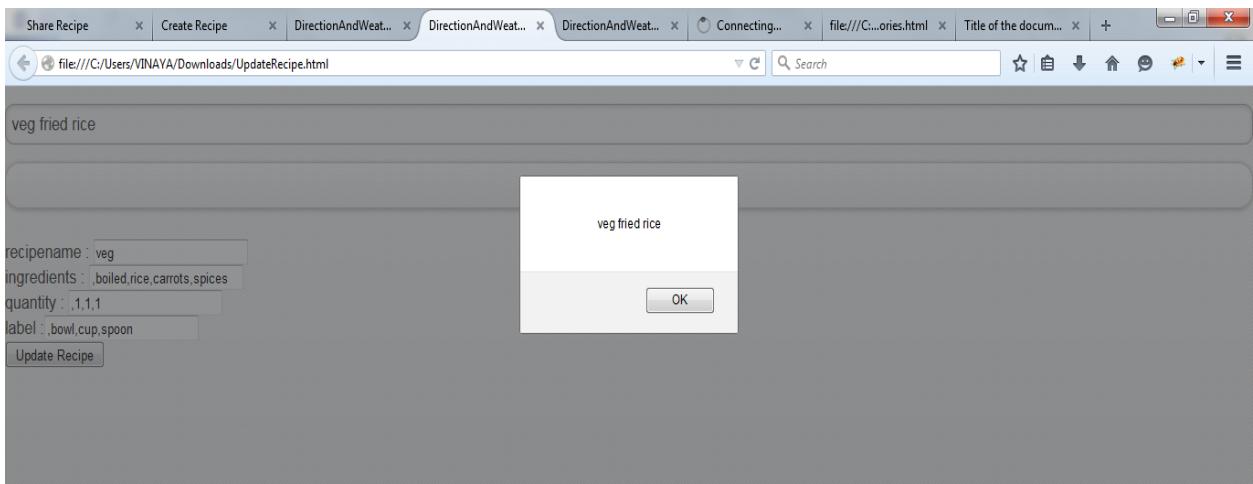


11. viewRecipe.html

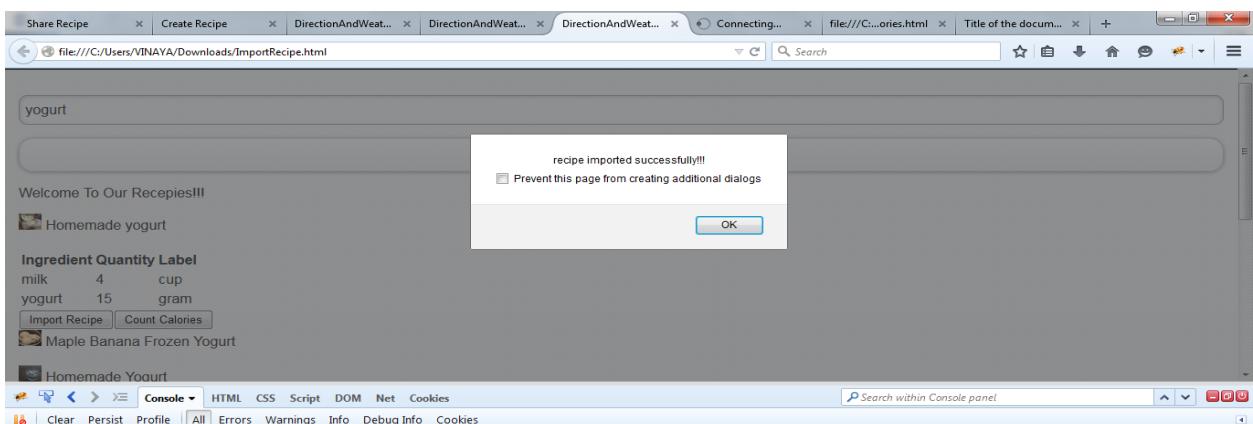


12. Updaterecipe.html





13. 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 Recepies!!!

Homemade yogurt

Ingredient	Quantity	Label
milk	4	cup
yogurt	15	gram

Import Recipe **Count Calories**

Console HTML CSS JavaScript DOM Net Cookies

Search within Net panel

Clear Persist All HTML CSS JavaScript XHR Images Plugins Media Fonts

825ms 45ms 340ms 56.5ms 57.2ms 56.2ms 439ms 80.5ms 92.3ms 69.3ms 68.9ms 2.22s 1.41s

10:42 PM 4/13/2015

14. Calorie.html

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

strawberry

Go!

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

Console HTML CSS JavaScript DOM Net Cookies

Search within Net panel

Clear Persist All HTML CSS JavaScript XHR Images Plugins Media Fonts

431ms 528ms 415ms 529ms 644ms 738ms 840ms 399ms 391ms 390ms 167ms 292ms 166ms 2.73s (onload: 222ms)

18 requests 214.1 KB

10:45 PM 4/13/2015

Strawberry Shake

Total Calories: 692

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

Pink Grapefruit, Strawberry, and Champagne Granita with Sugared Strawberries

Console panel showing network requests:

- GET http://api.nal.usda.gov/usda/ndb/search?format=json&key=DgqxZYmxGBYsycIEFLKJaAkH4ybQTHORakdx&url 200 OK 399ms
- GET http://api.nal.usda.gov/usda/ndb/search?format=json&key=DgqxZYmxGBYsycIEFLKJaAkH4ybQTHORakdx&url 200 OK 391ms
- GET http://api.nal.usda.gov/usda/ndb/search?format=json&key=DgqxZYmxGBYsycIEFLKJaAkH4ybQTHORakdx&url 200 OK 390ms
- GET http://api.nal.usda.gov/usda/ndb/reports/?ndbno=1...&key=DgqxZYmxGBYsycIEFLKJaAkH4ybQTHORakdx&url 200 OK 167ms
- GET http://api.nal.usda.gov/usda/ndb/reports/?ndbno=0...&key=DgqxZYmxGBYsycIEFLKJaAkH4ybQTHORakdx&url 200 OK 293ms
- GET http://api.nal.usda.gov/usda/ndb/reports/?ndbno=3...&key=DgqxZYmxGBYsycIEFLKJaAkH4ybQTHORakdx&url 200 OK 165ms



15. Notification.html

Show notifications

Console panel showing network requests:

- GET http://kc-sce-cs551-2.kc.umkc.edu/aspxt_client/M...viewGrocery/Service1.svc/JsonGrocery/Sravani?url 200 OK 100ms


```
{"q_name": "cheese", "q_quantity": 50, "q_measure": "gm", "Column1": "2015-04-18"}, {"q_name": "beetroot", "q_quantity": 1, "q_measure": "kg", "Column1": "2015-04-18"}  
x value is 2015-04-18  
2015-04-18  
x value is 2015-04-18
```



Show notifications

Net panel showing network requests:

Status	Domain	Size	Remote IP	Timeline
200 OK	kc-sce-cs551-2.kc.umkc.edu	184 B	154.193.136.128:80	100ms (onload: 167ms)

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



16. ViewGrocery.html

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "ViewGrocery.html". The page content displays a table with two rows:

	Name	Quantity	Units
●	cheese	50	gm
●	beetroot	1	kg

Below the table, there are "View Grocery List" and "Edit" buttons.

At the bottom of the browser window, the Firebug Network tab is visible, showing a list of 10 requests made by the page. The requests include various CSS and JavaScript files from ajax.googleapis.com, along with local files like "Sravani?url" and "honey?url". The timeline shows the duration of each request, with a total load time of 1.27s.

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "ViewGrocery.html". The page content displays a table with two rows:

	Name	Quantity	Units
●	cheese	50	gm
●	beetroot	1	kg

Below the table, there are "View Grocery List" and "Edit" buttons.

At the bottom of the browser window, the Firebug Console tab is visible, showing a list of network requests and their details. The requests are identical to those shown in the Network tab of the previous screenshot. The console also displays log messages such as "honey Deleted", "Updating the quantity of cheese to 50 gm", "cheese Updated", "Updating the quantity of beetroot to 1 kg", and "beetroot Deleted". The log entries are timestamped and linked to specific lines in the source code.

Screen Shots for shopping list generation:

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

Home Grade Components Statistics

Grade C Overall performance score 79 Ruleset applied: YSlow(V2) URL: http://jsfiddle.net/spt5/Lpvcbjdo/3/

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

F 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

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.

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

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

Empty Cache

Type	Weight (bytes)
HTML/Text	35.2K
JavaScript File	1644.9K
Stylesheet File	44.9K
CSS Image	22.3K
Favicon	1.0K
undefined	41.9K

Primed Cache

Type	Weight (bytes)
HTML/Text	35.2K
JavaScript File	4.0K
Favicon	0.0K
undefined	0.0K

Copyright © 2015 Yahoo! Inc. All rights reserved.

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

Home Grade Components Statistics

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									
is (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://ninejjcohidippngpapiiinmkglmakh/yslow.html#1

Home Grade Components Statistics

Grade C Overall performance score 79 Ruleset applied: YSlow(V2) URL: http://jsfiddle.net/sptn5/yL2vq7fp/19/

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

F 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

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.

chrome-extension://ninejjcohidippngpapiiinmkglmakh

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

Empty Cache

Type	Weight
HTML/Text	35.2K
JavaScript File	1644.9K
Stylesheet File	44.9K
CSS Image	22.3K
Favicon	1.0K
undefined	41.9K

Primed Cache

Type	Weight
HTML/Text	35.2K
JavaScript File	0.0K
Favicon	0.0K
undefined	0.0K

Copyright © 2015 Yahoo! Inc. All rights reserved.

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

Home Grade Components Statistics

Grade C Overall performance score 79 Ruleset applied: YSlow(V2) URL: http://jsfiddle.net/sptn5/yL2vq7fp/19/

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

F 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

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.

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.

5. Registration page
6. Login page
7. Welcome/Home page
8. 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:

4. Designing graphical user interface.
5. Connecting to the database and implementing web services.
6. Testing the entire application and bug fixing.

Scrum Do

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 throught the project as we add functionalities to the application Done Tasks 0 Comments Story4 sravani	3
#6 As a develop 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

#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		Tasks 0 Comments	Story7	
#8 As a user i should be able to search for recipes based on some keywords.				
Story8		Tasks 0 Comments	Story8	
#22 As a user i should be able to add grocery items available at home to a list in the application				
		Tasks 0 Comments	sravani	
#25 As a user, i should be able to view the list of ingredients required for a selected recipe.				
		Tasks 0 Comments	sravani	
#21 As a system, i should be able to store the grocery information provided by the user in the database				
		Tasks 0 Comments	sravani	
#35 As a developer, i need to test the import recipe API to ensure that the recipe information is inserted appropriately.				
		Tasks 0 Comments		
#34 As a developer, i need to test the API for adding groceries to the database to validate the data.				
		Tasks 0 Comments		
#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.				
		Tasks 0 Comments		
#32 As a system, i should display the UI for viewing the grocery list				
		Tasks 0 Comments	sravani	
#31 As a system,I should display the UI for searching recipes.				
		Tasks 0 Comments		
#30 As a system, i should display the UI for adding groceries to the database.				
		Tasks 0 Comments	sravani	
#29 As a system,j should be able to fetch the grocery items data from the database and display it to the user.				
		Tasks 0 Comments	sravani	
#26 As a user i should be able to import my favorite recipes for customization.				
		Tasks 0 Comments		

User stories for increment#3:

#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 srawani				
#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 srawani				
#39 As a user i should be able to get the calorie count of the complete recipe				
Done Tasks 0 Comments srawani				
#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 srawani				
#12 As a user i should be able to remove ingredients from the recipe				
Done Tasks 0 Comments				3
#9 As a user i should be able to update the quantity of the grocery items in my list				
Done Tasks 0 Comments srawani				3

User stories for Increment#4:

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

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	20	Vinaya, Sravani	Done

	developed by all the team members.			
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

Youtube URL: <http://youtu.be/tT-zoL18Qd8>

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/>

FINAL PROJECT EVALUATION

As part of our project completion we would like to evaluate our system. Our main aim in taking up this project is to provide a single platform to the users in managing Groceries and maintain proper diet. The tasks we thought to implement in our project are successfully accomplished.

Through UML diagrammatic representations we could make a clear picture of design model of our system in development phase. This gave us a complete overview on how we need to proceed in each phase. We have used agile process model in developing our project. One great thing about picking up agile model is that every person involved in the project gets to work in it learning to accomplish tasks in scheduled time.

In order to achieve our tasks completely, one should always have a better picture of how the output of each increment should be and design plans accordingly. Also in our future projects we would like to follow agile process model. As we followed this model and we had done each increment within a scheduled timeline, we never faced a problem. Everyone got involved in designing our project and so we were able to maintain timelines and deploy each increment as scheduled.

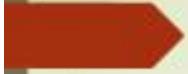
PRESENTATION MATERIALS





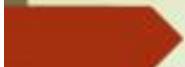
Goals & Objectives

- To manage groceries at home efficiently and reduce wastage by getting notifications about expiry dates.
- To provide calorie information about the ingredients used in preparing a recipe. This helps the user to take healthy choices about their meal.
- To automatically generate a shopping list so that the user doesn't miss something he needs to buy.



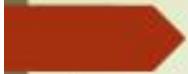
Why we choose this project?

- Grocery and diet management is an every day task. Our effort is to build an app that allows users to manage this task easily and effectively.
- Our app gives the calorie information for each ingredient in the recipe thus providing nutritional information to the user.



Key Features

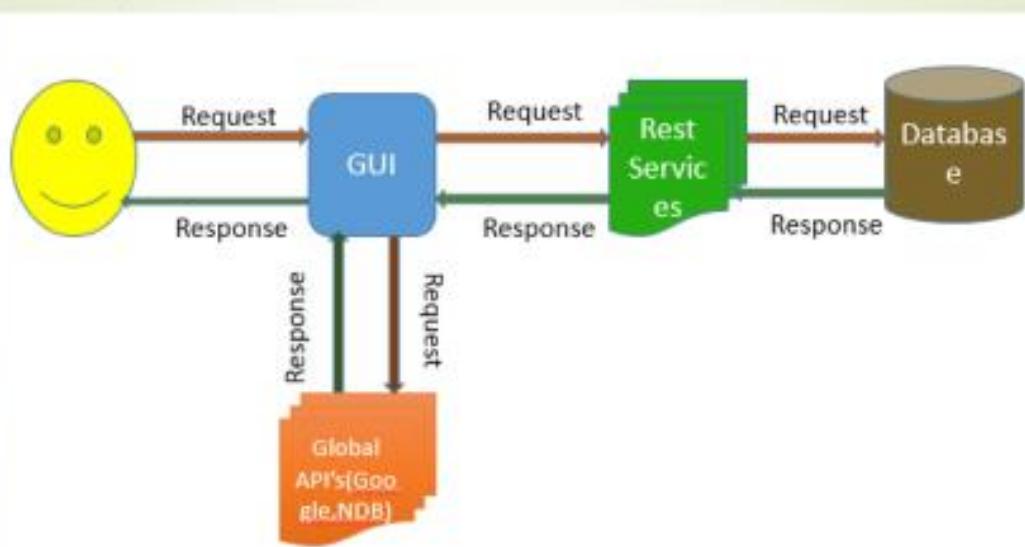
- Grocery Management
- Search Recipes
- Get Calorie information by ingredient for recipe
- Add/Remove Ingredients from recipes
- Generate Shopping List
- Create and Share Recipes
- Get Notifications
- Find Nearby Grocery stores.



Technologies used

- Existing web services
 - Recipe search & Diet API
 - USDA Nutrition API
 - Google Maps API
 - Twitter API
- New web Services
 - Recipes
 - Grocery list
 - Shopping list
- C#, ASP .NET
- Android Studio
- Ajax
- Jquery
- HTML, CSS, Bootstrap
- SQL Server Database

System Architecture

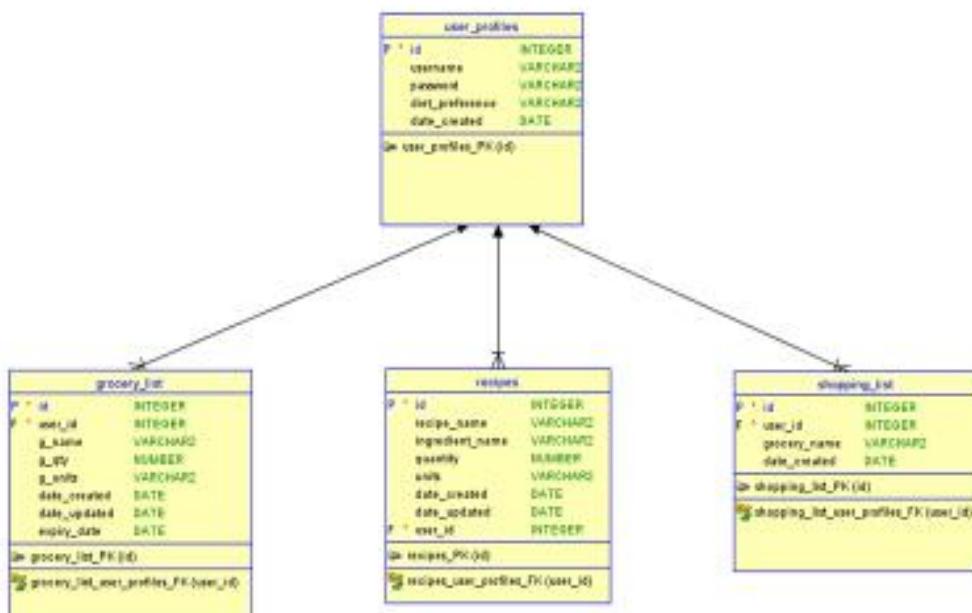




Challenges

- Android Studio
- Setting up Facebook sign-in
- Getting calorie count per ingredient
- Notifications

Data Model





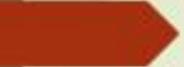
Testing

- UI Testing
- NUnit Testing
- Performance testing; Tools used
 - Firebug
 - ✓ The firebug tool will automatically show us if there are any errors in the JavaScript, CSS or HTML code and request URL's or XHR requests.
 - YSlow analyzer
 - ✓ This is a free, open source testing tool that analyzes web pages and suggests ways to improve their performance.



Future Enhancements

- Barcode Scanner
- Categorize groceries into different groups
- Recipe Suggestions based on user's diet preference



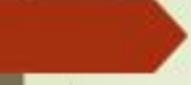
Project URLs

- ▶ GitHub URL

https://github.com/vaishnavi5054/AdvSoftEng/tree/master/ASE%20Increment_4%20PG_12

- ▶ Youtube URL

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



Thank You!

This work was done in partial fulfillment of the requirements of CS551:Advanced Software Engineering, CSEE Department, University of Missouri – Kansas City (Spring 2015).

Instructor: Dr. Yuguang Lee,

TA: Malathy Krishnan, Mayanka Chandrashekhar, Bharat Viswandum.

Special thanks to IBM Bluemix for providing the free cloud hosting service.