

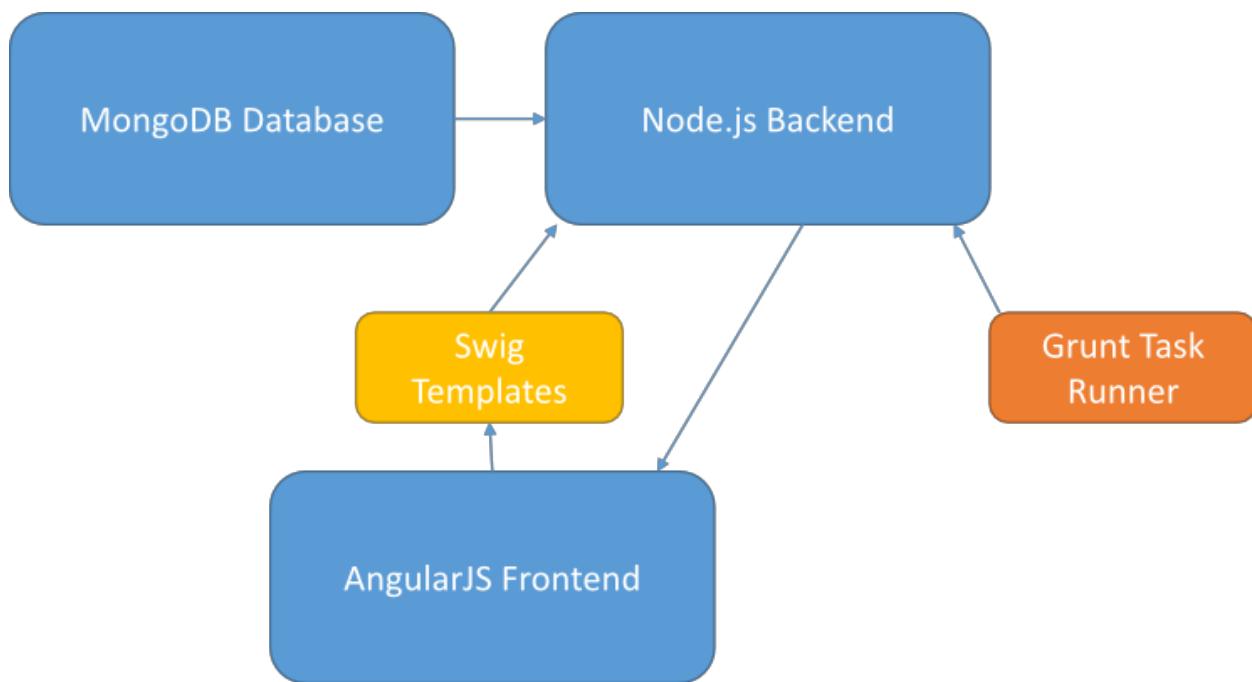
Group 7: Hestia

Developer Documentation - CannyKitchen

By Nathan Korzekwa, Hyun Kim, Anna Tran

Architecture

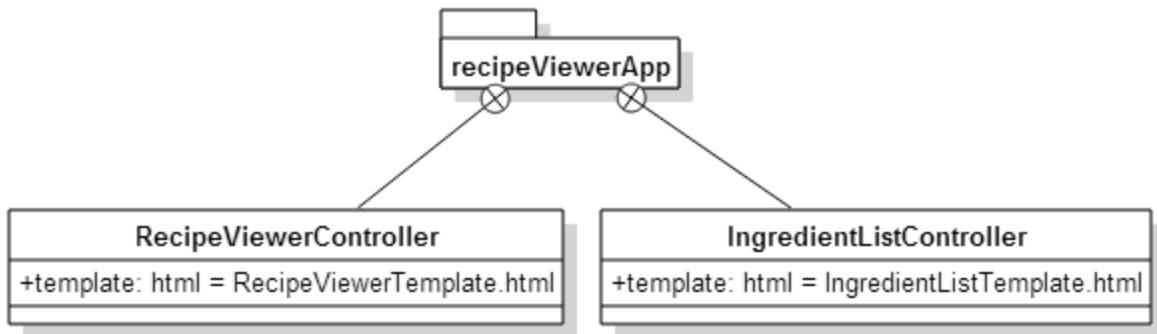
The high-level architecture of CannyKitchen conforms to an event-driven Client-Server style. As seen in the figure below, the two principal components of the application reside in the Node.js server backend and the AngularJS frontend. All data processing occurs in the backend, with the front end performing a primarily user-interface driven role.



The Swig templates straddle the gap between client and server, as they exist largely in the front-end but are influenced directly by variables (compiled by) in the Node.js Swig templating engine.

AngularJS Frontend

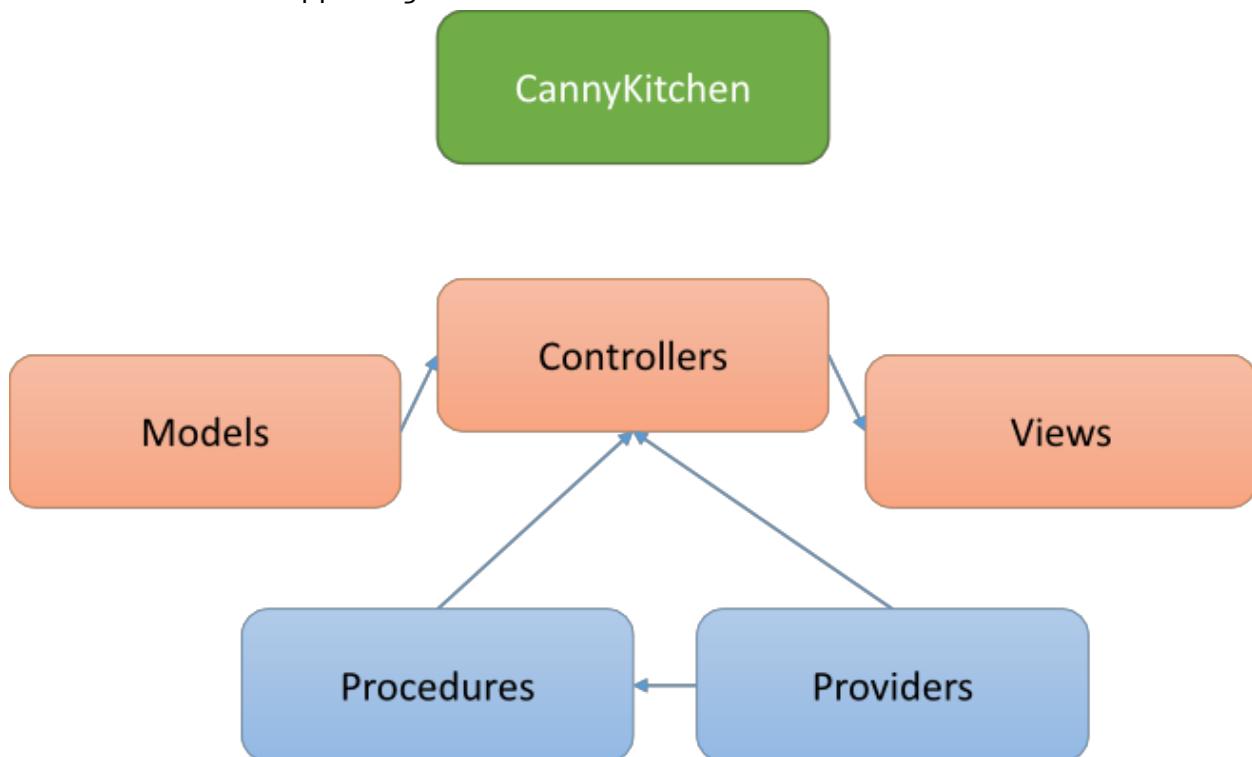
As an AngularJS-based application, the front-end is by necessity highly modularized. As the UI for this application that demands the rich interactivity that AngularJS enables is less present, the AngularJS application itself is rather simple.



As shown above, the entire application lives in the "recipeViewerApp" module, with there only being two different pieces of UI functionality. The "RecipeViewController" is invoked whenever its template file is included in the web-page, and it provides the interface allowing the user to request a meal-plan from the Node.js backend and then display the results. The "IngredientListController" is invoked whenever its template is included, and it provides the user the ability to request that a certain type and amount of ingredient be included in the generated meal-plan. These two modules are decoupled but must communicate, so they do so via the inherited scope of "recipeViewerApp."

Node.js Backend

The backend is partitioned into the standard MVC (Model-View-Controller) framework, as well as into various supporting classes.



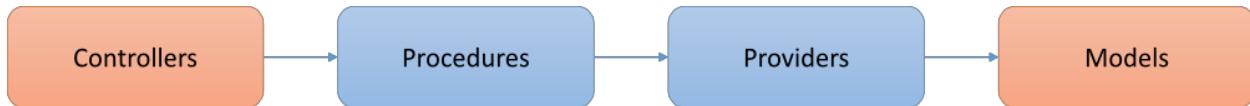
The Models component follows the structure of the database, outlined later, exactly. The Controllers facilitate the requests from the front-end, invoke methods in the Procedures

components to perform operations such as generating meal-plans, and compile the templates in the Views component for front-end consumption. The Providers component, perhaps a bit out of place, serves as a placeholder for more dynamic and less-defined data than in the Models component.

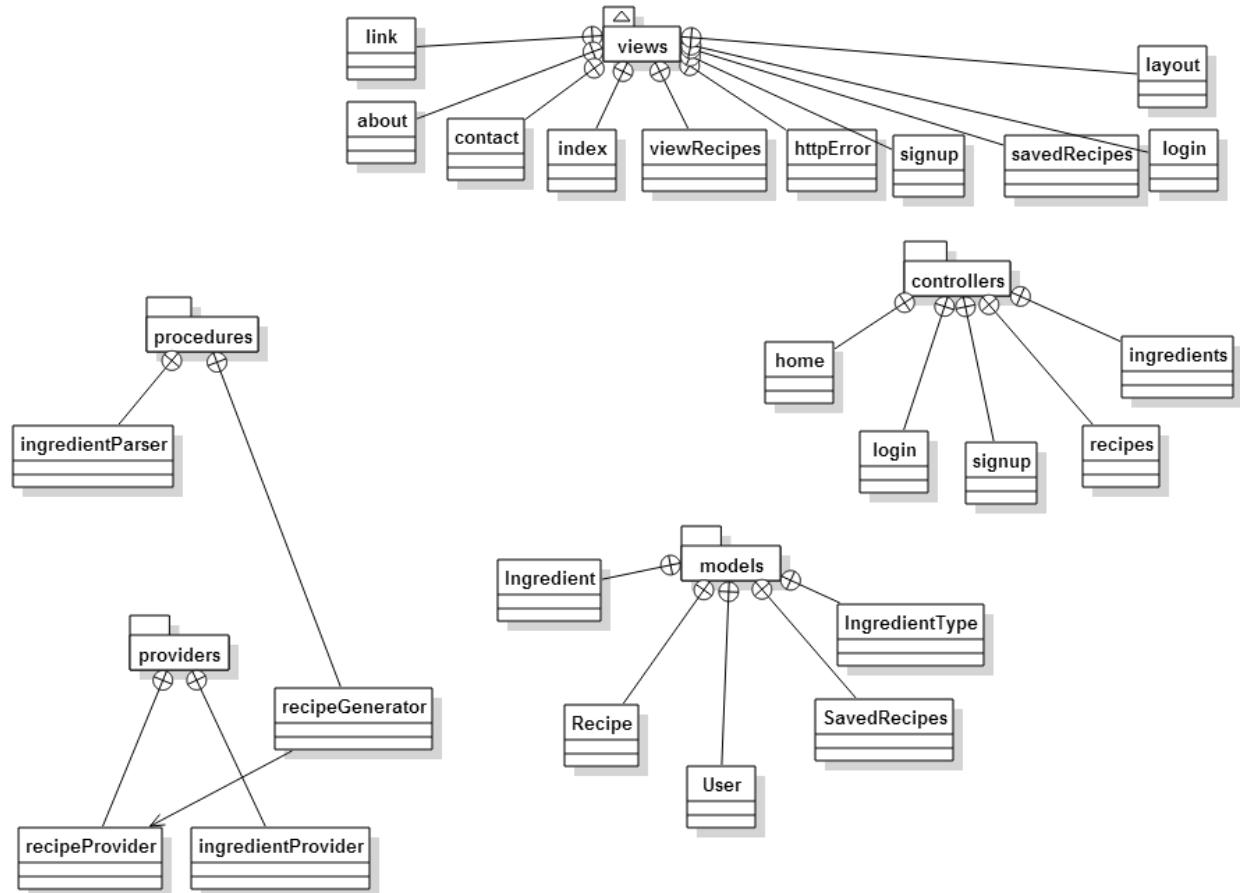
For most tasks, Grunt is used to help facilitate common tasks such as seeding an empty database full of test recipes and for running a local web-server.

To further illustrate this, consider a typical request made to the server. The AngularJS client makes a GET or POST request to the server, and hooks into any defined such request in the controllers. For the generate meal-plan request, a GET request is made to the endpoint "generate-recipes" along with URL-based parameters "beginDate," "endDate," "similarity," and "ingredients." The hook for this call in the "recipes" controller is invoked and the "generate-recipes" algorithm in the "Procedures" component is invoked. Using data from the "recip-provider" service in "Providers," it generates a mealplan and relays it back to the Controller.

This process is shown below.



Detailed Design



The general architecture for the server-side application is shown above. As mentioned in the previous section, it is split into several components. Many of these are self-explanatory, but the most important ones will be outlined below. Some of them, like "ingredientProvider" and "ingredientParser" are deferred features not currently used in the application.

recipeGenerator

This is the heart of CannyKitchen, as it generates the meal-plans that the user requests. Any documentation wouldn't be complete without a description of this algorithm and thus it will be described here.

In this algorithm, each recipe can be described as a vector, \vec{r} , where each component stands for an ingredient type and the number it contains represents how many of that type. Then, for an N -day period, and a preferred vector of ingredients, \vec{p} , attempt to

find a set of recipes $R = (\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N)$, such that $\sum_{i=0}^N \sum_{j=1}^i g(\|\vec{r}_1 - \vec{r}_2\|) + f(\vec{p} - \sum_{i=0}^N \vec{r}_i)$ is maximized,

where f and g are custom functions. For the currently distributed application, $g(t) = \frac{e^{x-u}}{(1+e^{x-u})^2}$ and $f(t) = 1/t$.

To approximate this optimization problem, local search is used. R is initialized to a random subsequence of recipes, and then perform the following update:

$$R_i^{t+1} \leftarrow \operatorname{argmax}_r \sum_{j=0, j \neq i}^N \|R_j^t - r\|$$

Thus R is updated from $i = 0$, to N for a given number of iterations and the result is returned.

recipeProvider

This provider queries the “Recipe” MongoDB collection and returns random and filtered samples from the database. The recipe generation algorithm depends on its randomness heavily since it is a local-search based problem intended to give different results each time it is run.

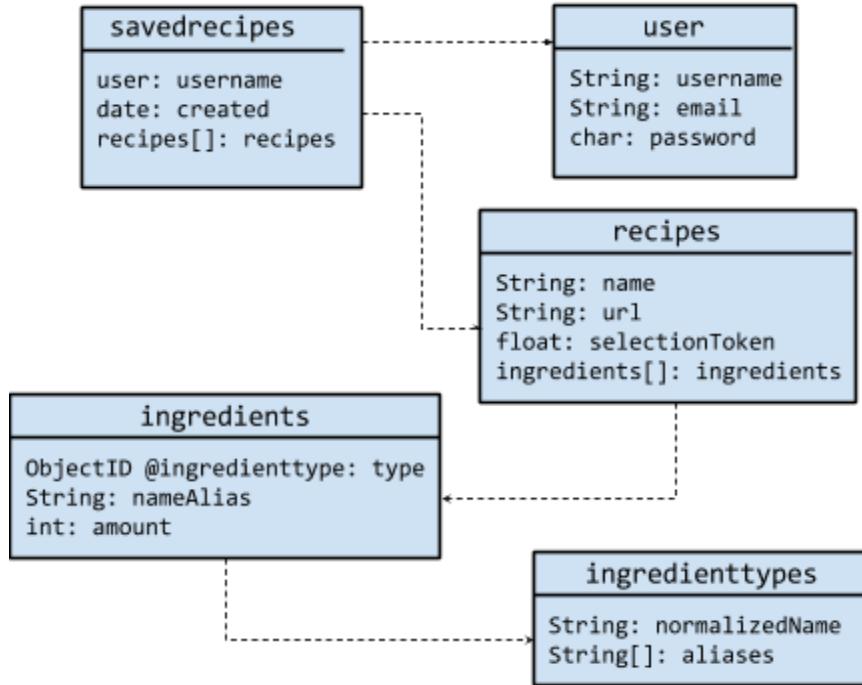
signup

This controller contains all the routines necessary for creating new accounts, including authentication, hashing, and database manipulation.

Data Storage

All data, including recipes and user accounts, are stored on MongoLab, a cloud database service that hosts MongoDB databases. The database consists of five collections--`ingredienttypes`, `ingredients`, `recipes`, `savedrecipes`, and `user`.

`ingredienttypes` stores all the ingredients parsed from the recipes. It is also from this collection that the ingredient search box datalist is populated with. As for `ingredients`, this collection has ingredient types along with amounts that will help to create recipes. The `recipes` collection stores all the recipes. The `savedrecipes` collection stores all the meal plans that users decide to save. Each document within `savedrecipes` has the username, date created, and recipes generated for the meal plan. As for the `user` collection, this holds all account information. Each `user` document holds a username, hashed password, and email address.



User Interface

All the user interface elements are set in the “view” section of our web application. Bootstrap is used for the front-end framework due to its sleek and modern design.

When the user first visits the website, they are greeted with the home page which consists of a navigation bar at the top, a left sidebar with instructions on how to use the service, and a main panel with date selectors with a similarity measure, and a button to generate meal plans based on the user inputs (Figure 1).

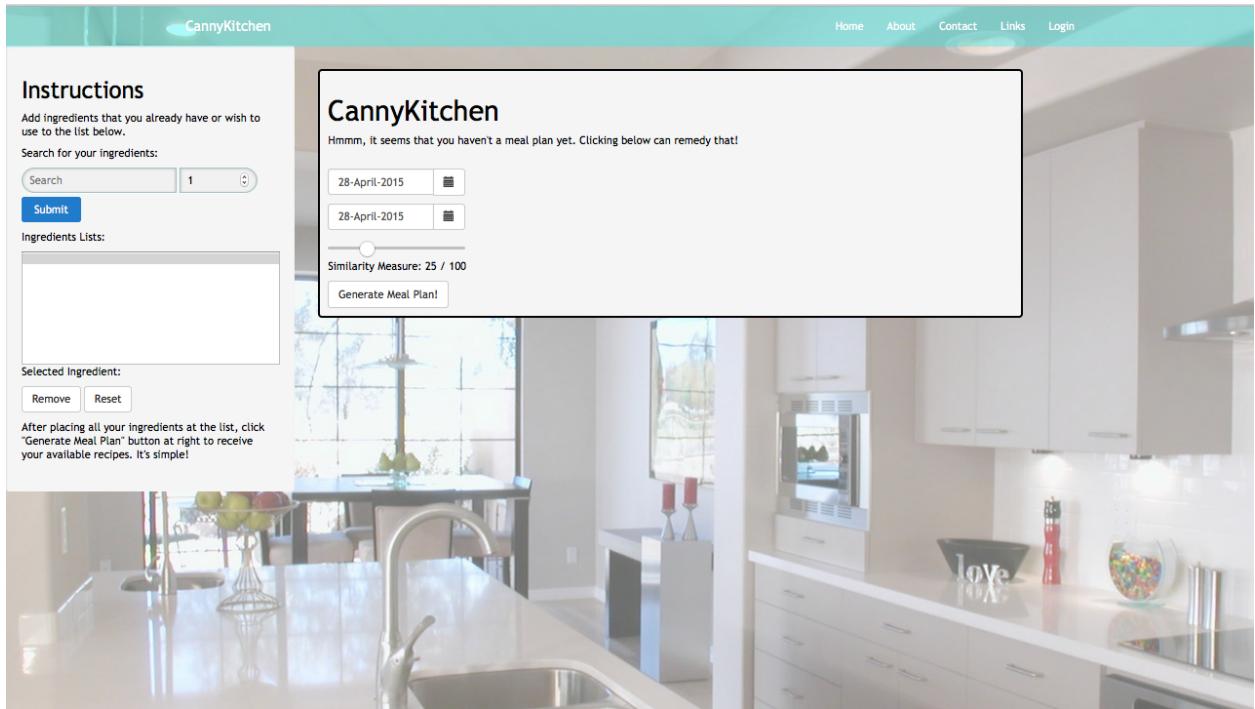


Figure 1. Homepage

The navigation bar consists of important links for users and is accessible from any page. A user can easily navigate back to the homepage by either clicking "Home" or "CannyKitchen" on the far left corner. From the homepage, users can also access the "About", "Contact", "Links", and "Login" pages.

Homepage

First time users are expected to initially read the "Instructions" panel to understand how the website works and then proceed onto the main panel to generate a meal plan. When searching for an ingredient, users are given suggestions of available ingredients as they type (Figure 2). Ingredients are based on recipes stored in the database. If a user tries to input an ingredient not available in the database or already on their list, then a modal dialog will pop up warning users of their invalid input. Adjacent to the ingredient search box is an input of the number of the selected ingredient the user has available or want to use. For example, if a user wants to use two (2) apples, he/she can type in apples and increase the amount to two. For now, specific units like gallons, tablespoon, etc. are not available. Everything is measured in units.

Instructions

Add ingredients that you already have or wish to use to the list below.

Search for your ingredients:

1
▲ ▼

app
apple
 apple (honeycrisp)
 apple juice
 musselman's apple butter
 pineapple
 pineapple chunk

Ingredients Lists:

2 units of apple

Selected Ingredient:
`{"name": "apple", "amount": 2, "id": 0}`

Remove
Reset

Figure 2. Ingredient suggestions

Figure 3. Ingredients lists

After submitting an ingredient and unit, those values will be added onto the Ingredients List (Figure 3). From there users will have the option to remove individual ingredients using the “Remove” button or clear the entire list using the “Reset” button. Pressing the “Remove” or “Reset” button will trigger a modal dialog asking users to confirm if he/she wants to remove the ingredient or reset the list, respectively.

CannyKitchen

Hmmm, it seems that you haven't a meal plan yet. Clicking below can remedy that!

28-April-2015
📅

28-April-2015
📅

Similarity Measure: 25 / 100

Generate Meal Plan!

Figure 4. Main panel

This main panel is where users can select a time range for the number of days he/she wants in the meal plan (Figure 4). Users pick the dates by clicking the calendar icon, which will then display a calendar starting at the current day. The first date input is for the start day, and the second is for the end day. Underneath the date pickers is a slider to adjust the similarity measure. Sliding to the left decreases the measure, which means recipes chosen will vary. Sliding to the right increases the measure, which means recipes chosen will be more similar. For example, sliding to the furthest right will set the scale at 100 / 100, so the meal plan will consist of all the same recipe. The default scale is 25 /100. Users can then click the

"Generate Meal Plan!" button to create a meal plan based on the chosen parameters. Meal plans can still be generated whether ingredients are selected or not. If no ingredients are selected, then a meal plan with random recipes will be created.

Meal Plan

Upon clicking the "Generate Meal Plan!" button on the homepage, a meal plan added onto the main panel with the sidebar still intact (Figure 5). The number of recipes shown is based on the date range selected. Only one recipe is generated per day. If a user is logged in, then the "Save Meal Plan" button will be active. Otherwise, a link will be shown suggesting users to either login or register. Because the original content of the homepage are still available, users can choose to create another meal plan using the same process as before.

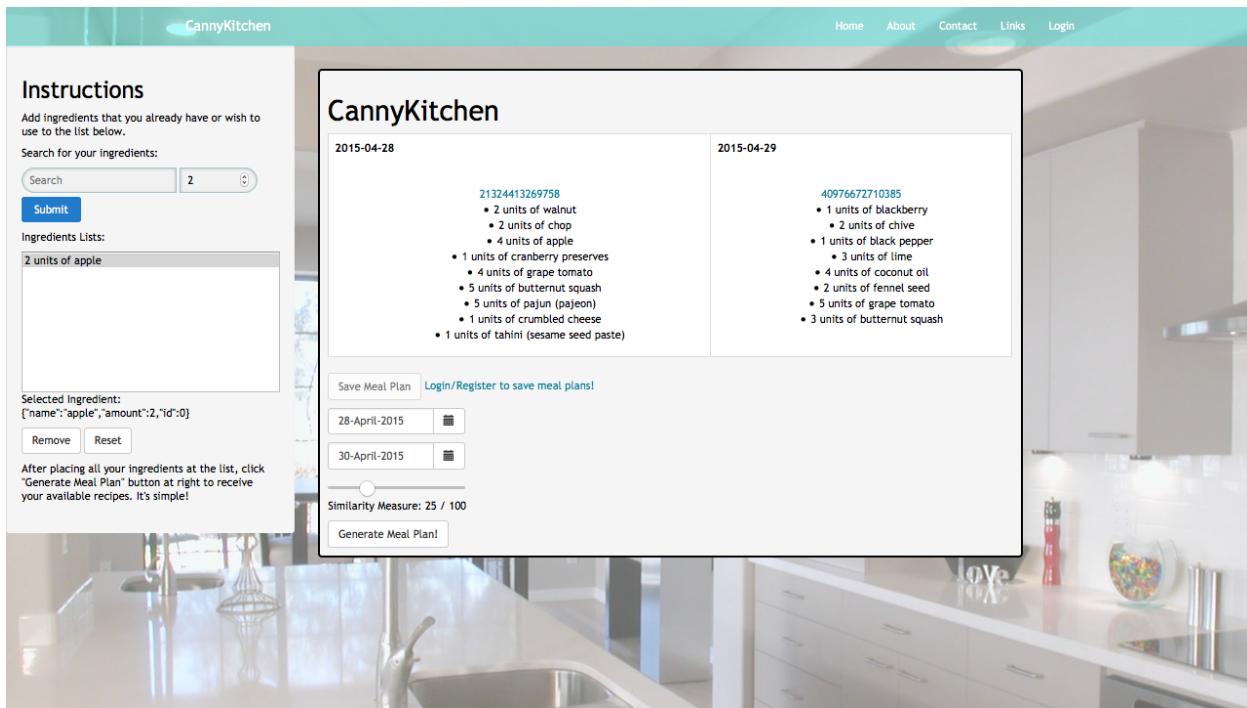


Figure 5. Generated meal plan

About

The "About" page can be accessed from any page and from any user (Figure 6). This page consists of CannyKitchen's mission along with its status on features.

Contact

Like the "About" page, the "Contact" page is viewable by any site visitors (Figure 7). This page provides CannyKitchen's developers' names and email addresses.

Links

The "Links" page consists of useful links pertaining to nutrition and cooking (Figure 8). It can be seen and accessed from any page on the site.

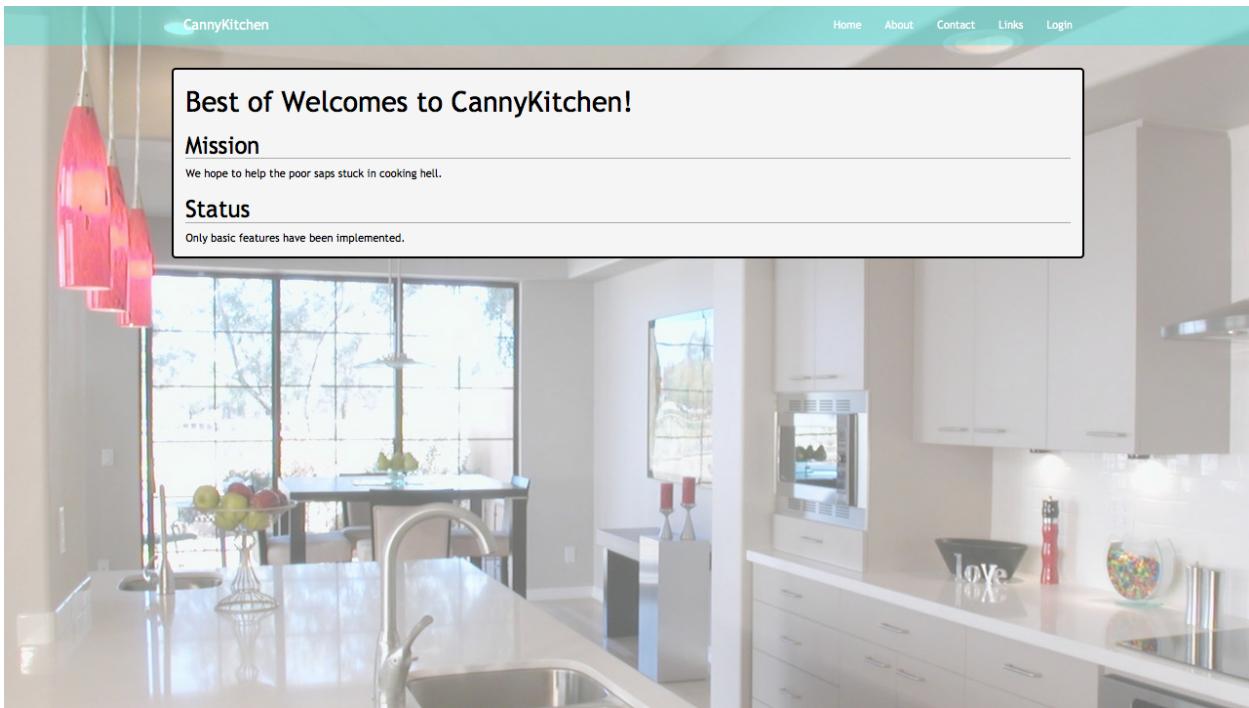


Figure 6. About page

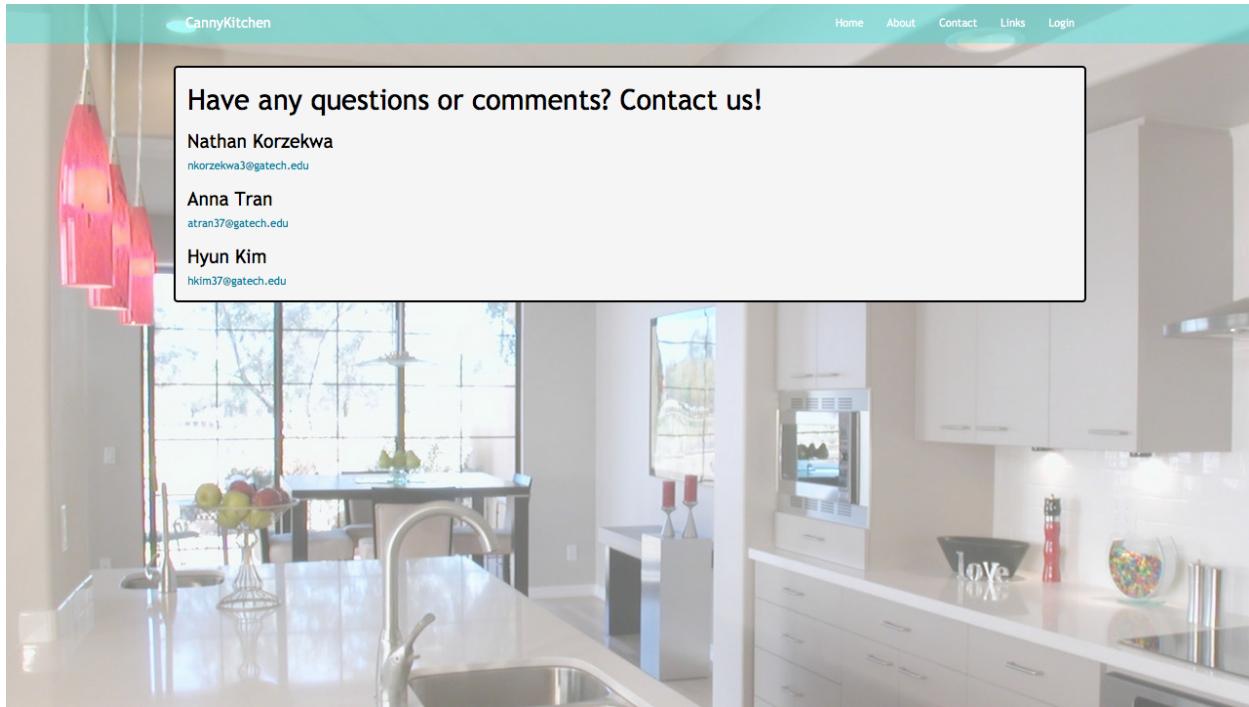


Figure 7. Contact page

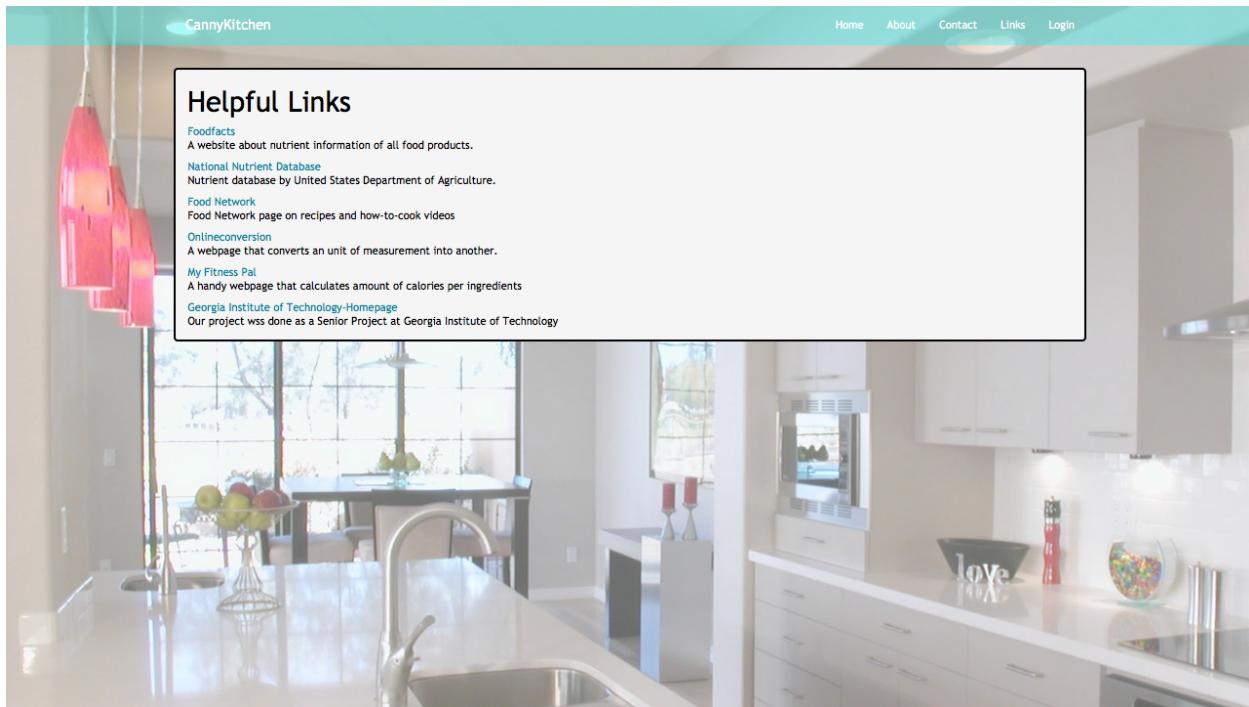


Figure 8. Links page

Login

If the user is not logged in, then the "Login" link will be visible on the navigation bar. Clicking the "Login" link will redirect the user to the login page where users have the option to either

sign in or register for a new account (Figure 9). Registered users can sign in by inputting his/her username and password in the form. If either the username or password is wrong, the user will be alerted with a banner (Figure 10).

The screenshot shows a 'Log In' page with a teal header. Below it is a white form area containing two input fields: 'Username' and 'Password'. Each field has a label to its left and a corresponding text input box. Below the fields is a blue 'Sign In' button. At the bottom of the form is a link: 'Not Registered? [Register here.](#)'

Figure 9. Login form

The screenshot shows the same 'Log In' page as Figure 9, but with a red banner at the top containing the text 'Username/password is incorrect!'. The rest of the page, including the input fields and 'Sign In' button, remains the same as in Figure 9.

Figure 10. Login error

Registration

The registration page can be accessed from either the login page or from the meal plan generation page (if not logged in). Once on the signup page, a user must input an email, username, password, and the same password again to confirm. Alerts will come up in a user inputs an email or username that already exists in the database or if the passwords do not match (Figure 11). If the user's registration is successful, the page redirects to the home page.

Sign Up		Sign Up		Sign Up	
Email already taken.		Username already taken.		Passwords do not match.	
Email	<input type="text"/>	Email	<input type="text"/>	Email	<input type="text"/>
Username	<input type="text"/>	Username	<input type="text"/>	Username	<input type="text"/>
Password	<input type="password"/>	Password	<input type="password"/>	Password	<input type="password"/>
Confirm Password	<input type="password"/>	Confirm Password	<input type="password"/>	Confirm Password	<input type="password"/>
Sign Up		Sign Up		Sign Up	

Figure 11. Registration errors

Logged In Users

If a user is logged in, he/she will see new links on the navigation bar (Figure 12). Registered users can additionally view their meal plans or logout. The logout button is located in the username drop down menu.

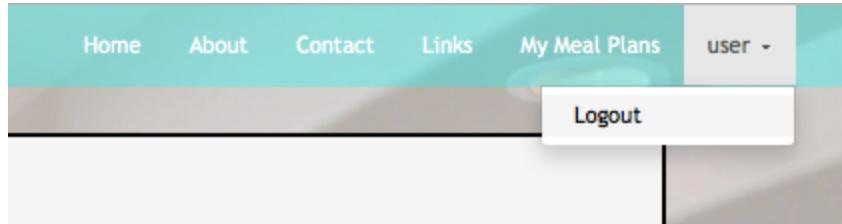


Figure 12. Navigation bar for logged in users

My Meal Plan

Registered users have the option to save meal plans generated. The "Save Meal Plan" button will no longer be disabled when viewing a meal plan (Figure 13). Clicking the button will save the meal plan into the database under the corresponding username and then redirect the user to the "My Meal Plans" link, where all saved recipes can be viewed. Logged in users can also view their meal plan by clicking on the "My Meal Plans" link in the navigation bar.

The screenshot shows a meal plan entry for '65497445357032'. The ingredients listed are: 1 units of pumpkin, 4 units of steak, 4 units of peach, 2 units of escarole, 4 units of cream of tartar, and 3 units of brandy. Below the list is a 'Save Meal Plan' button. Underneath the button are two date fields: '29-April-2015' and '30-April-2015', each with a calendar icon. A similarity measure slider is set at '25 / 100'. At the bottom is a 'Generate Meal Plan!' button.

Figure 13. Save Meal Plan button activated when logged in

On the "My Meal Plans" page, there will be an ordered list of the user's meal plans based on date creation with the most recent being on top (Figure 14). Each meal plan lists the number of days along with the recipe name and link. If a user decides he/she does not want a meal plan, the user can easily remove it by clicking "Delete Meal Plan" next to the corresponding meal plan.

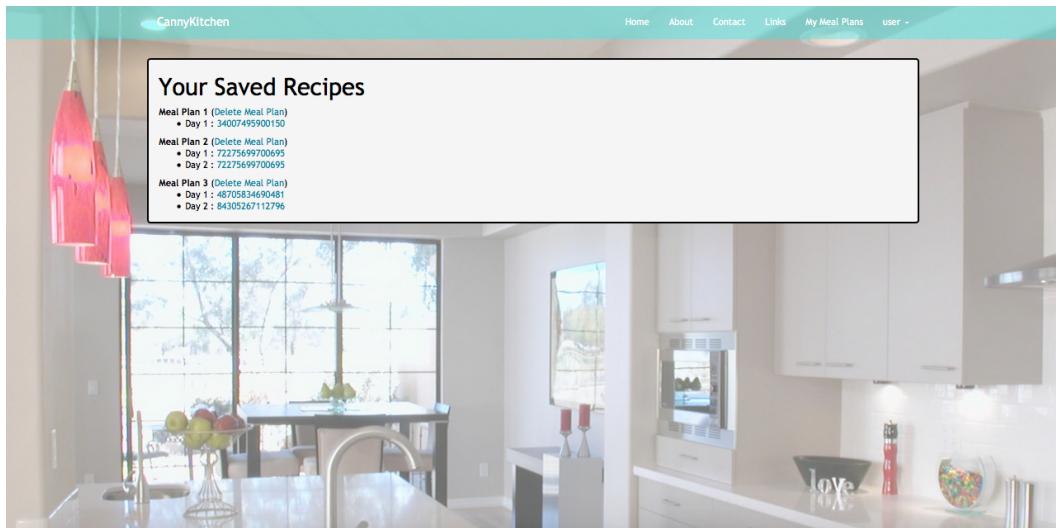


Figure 14. My Meal Plans page

Logout

Finally, if the user wants to log out, simply click the username and the link to logout will drop down (Figure 12). Clicking logout will then redirect back to the homepage.

