GroGro Project Proposal

ITWS 4200 Web Science

**By: Team 1**

[**Sarah Abrahamson**](https://www.facebook.com/sarahxabrahamson)**, Robert Russo, Peter Ryder, Jessie Sodolo**

## Project Description

GroGro aims to be a service for providing nutrition information for lists of food. As such a service, we aim to create an open RESTful API where anyone can send a formatted JSON list to our server, and receive nutrition information for those items. To show the power of this service we are creating an application on top of this to allow the creation of a grocery list. As items are added a graph will be generated, reflecting the nutritional value of the overall grocery list. In doing this, GroGro will be providing useful insight for shoppers so they can better balance their groceries. GroGro can also be used to track meals, or plan meals.

## Architecture:

GroGro is built on the MEAN stack. This starts with Node as our server. On top of Node we are running Express to simplify many commons server operations. Using express we have created two routes, or endpoints. One endpoint creates the frontend application, while another provides the data needed for the application. On the front end, we are using AngularJS to create a single page application. The front end pages are handled using ng-route to create the single page experience. As pages are called, Jade pages are rendered on the backend and then served to angular. These are then embedded in the primary view. This greatly improves performance, as all javascript files are loaded right off the bat. Switching pages only required loading small Jade files. All of our controllers are loaded and ready on the initial page load. For design, we are using SASS for more organized CSS. We decided not to use Bootstrap so as to retain a more original design for our website, and to showcase our abilities. Once code is added to our project, it is committed to our Github page. From there Travis CI runs a few tests to ensure the code doesn’t throw an error. Once the checks are completed, the code is immediately deployed to our Heroku instance, creating a continuous integration pipeline.

## 

## Development:

The team meets weekly to discuss the progress of the project and newer Github commits. Separate branches for both backend and frontend have been created and implementation of both ends happen simultaneously. Sarah and Peter focus on the backend, work with the USDA API, Facebook API and the underlying database. Rob and Jessie focus on the frontend, the angular controls that style the data received from the API calls, the styling of all pages and the Nutrient graphs for each grocery list. Currently the base structure of the application has been implemented including: the node server, jade pages for rendering the backend. The initial API call to the USDA API has been made enabling the user to search for an ingredient. The current search implementation will just return the ingredient names that match what the user typed in. The team has implemented pagination in an attempt to reduce the amount of results a user sees and chooses from at one time.

## 

## Moving forward the team has to work on the Facebook login for users, the mongodb that will store grocery list for users, the styling of each page, the API call for ingredients and adding more user functionality.

**Task responsibility breakdown:**

Sarah - Facebook login API, MongoDB setup, MongoDB implementation

Peter - USDA Nutrients API call, debugging backend functionality(Breaking the system)

Rob - Styling of the about and homepage, implementation of the nutrient charts

Jessie - Styling of the list page, user functionality to add to list.

## 