

# Motivate

Design Document

Version: 1.0.1

Date: April 5, 2015

Presented by: Gelatinous Robots

Nate Levesque

Ryan Hoffmann

Yura Kim

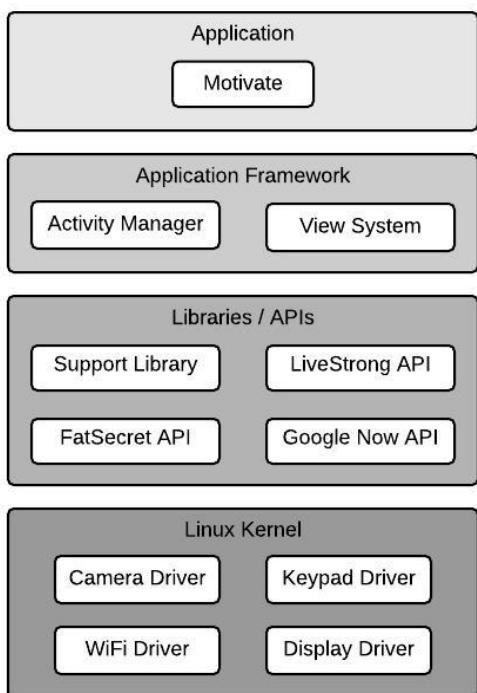
## Revision History:

Who	What	When	Version
Team	Create document skeleton	3/15	0.0.0
Team	Start filling out document	4/1	0.1.0
Team	Begin filling out section 1	4/3	0.2.0
Team	Import mock-ups and complete document	4/4	1.0.0
Team	Completed & Revised for final submission	4/5	1.0.1

# 1. Software Design

## 1.1 Program Structure

### 1.1.1 Software Architecture and Component Design Views



The Android operating system is made up of several layers. Our application, Motivate, will live in the top “Application” layer. We will rely on Android application framework components such as the activity manager and view system.

The back end of Motivate will rely on access to a number of external API’s including Livestrong, FatSecret, and Google Now. These will be used to populate Motivate with nutrition data and to provide notifications. The app will also rely on the standard Android storage system.

At its lowest level, we will make use of hardware drivers such as the camera driver, keypad driver, and WiFi driver. We will use Android API’s to gain access to hardware components.

### 1.1.2 Description for *Livestrong API*

The Livestrong API is a REST API provided by Livestrong for looking up nutrition information for foods. Motivate will access this API over standard HTTP, retrieving JSON data for the food being searched. Livestrong’s API provides access to a large database of food and associated nutrition information which can be consumed by applications.

### 1.1.3 Description for *FatSecret API*

The FatSecret API is made up of a JavaScript library as well as a REST API. It provides access to the FatSecret Platform which contains a database of food and nutrition information in addition to exercise and weight management tools. The FatSecret Platform also syncs with the FatSecret Brand Tools, a set of tools for owners of branded foods and beverages to submit, manage, and share nutrition related information.

#### **1.1.4 Description for *Support Library***

The Android Support Library provides backward-compatible versions of the Android framework APIs and certain features up to a certain point. Using this library ensures the application is compatible with devices running Android 1.6 (API 4) and up.

### **1.2 *Design Rationale***

The team is making use of more external API's than otherwise might be done while developing a new application. The intent is to use existing databases for nutrition data in order to avoid the need to build a database for the app, especially given the large data back end that is required for the application to be useful. In particular, the app will make use of one or more API's from services such as Livestrong and FatSecret in order to retrieve nutrition information about foods. Assuming that the API's have support for partial searches, the hope is to additionally use data from the API's in order to retrieve search suggestions. Using multiple API's to populate nutrition data can also make the app more valuable in that there may be food items in one database that are not available in the other.

Additionally, the application will use a number of Android system facilities rather than building custom ones, including using the Google Now API for sending notifications. Using Google Now for notifications eliminates the need for configuring custom push notifications and allows for much more useful alerts, as the Google Now cards are able to display larger amounts of information to the user in a style the user is already accustomed to.

## 2. User interface design

### 2.1 Description of the User Interface

The design is intended to use imagery in order to help motivate the user to stick to a recommended diet plan. On first launch of the app, the user is presented with an on-boarding process in order to gather basic personal information to better provide suggestions and tailor the app to their needs.

The first view in the on-boarding process (Figure 1) will collect information about the user, including current weight, height, gender, first name, and activity level. The activity level drop-down contains a list of three potential values: sedentary, active, very active.

The second view (Figure 3) is geared towards what the user wants to become. The user is presented with three cards, each containing an image and a button explaining to the user what the image purveys. In the image used in Appendix A, the user sees an image of a very fit woman, and the button below the image explains that selecting this “goal” means that he/she as the user wants to “get fit.” The user can swipe through three different goal types, which is visually explained by the three dots beneath the image cards. It’s important to note that depending on the gender the user selects in the previous view, the images used in this view will reflect the chosen gender. The text describing the goals will also change, as the goals for each gender are different. For example, the goal types for a male user would be to get: thinner, stronger, bigger. “Bigger” in the male instance refers to putting on muscle mass, and is described in the button below the accompanying image.

In the third view (Figure 4), the user is prompted to take their first weekly progress picture. The user is briefed on the reason for taking the first photo, and can do so immediately by tapping the prominent camera icon, but also has the option of doing it later if they’re in a public area.

Once the on-boarding process is complete and any subsequent times the user opens the app, they will be presented with a Summary view (Figure 6). This view, along with all other main navigational views in the app contain an image header which will randomly change each time the user navigates to the view. The image will always contain a picture of a person with a figure representative of what the user aspires to look like.

The Summary view contains information about the users current daily dietary progress. It features a card for each main building block: calories, protein, carbohydrates, and fats. Each card contains a progress bar as well as a numeric value showing the user how many more units they have to consume in order to hit their daily goal. If the user exceeds their goal for a specific unit, the progress bar will turn from the normal teal to a lighter hue of red, letting them know that they have eaten more than the application recommended. Each card also contains an accordion button that expands to show the user the number of units each food contributed to that units progress. For example, upon expanding the proteins card, after consuming a 8 ounce steak, the user would see the amount of protein that steak contained. The summary view also features a button in the bottom

right containing an image of a fork and knife, as well as a plus sign, telling the user that this button allows them to add a food to their log.

Upon tapping the “add food” button, the user is taken to the search view (Figure 9). This view contains a search bar as can be expected, but also features a list of recent foods and groups (which will be discussed later). Upon typing into the search bar, the user is immediately presented with a drop-down of food results that match the letters they’ve typed just below the search bar, also known as search auto-completion. If the user does not like any of the auto-complete suggestions they are able to hit the enter key on their keyboard and view a full list of all potentially matching foods. See (Figure 7) for a high level flow of the food logging process.

Upon choosing a food from the list, the user is taken to a view in which they’re able to choose the serving size of the selected food they consumed. The application chooses the most common measure of serving for the selected food, but the user has the option to select a different serving measurement via a drop-down next to the serving size drop-down (Figure 13). The serving size drop-down contains a scrollable list of numbers that depend on the chosen serving measurement and logically range depending on the chosen food.

After tapping the “Log food” button, they are immediately taken to their Food Log view (Figure 15), another global view. This view contains a list of cards for each day that has passed since the user began using the application, and to the left of each card is the day of the month. Each card shows the overall daily intake progression, which is a number based on each food unit (calories, protein, carbohydrates, fats) broken down in the summary view. The user also has the option of expanding the card to see each of the foods they have logged on a given day. The list is vertically ordered by the date, so the top-most view is always the current day. This means that whenever a user logs a food for the day, the Food Log views top-most card is by default expanded to show all of the foods the user has entered today, as well as the corresponding serving sizes.

Quite often, a user will have the same meal during the week, and will want to make the logging process easier. The meal usually contains the same foods and/or drinks, so the user is able to easily group foods together to make future logging quick and easy, given that these groups appear towards the top of the “recently added” foods list. See (Figure 16) for a high level flow of the food grouping process.

By simply long-pressing on a food item in any card in the Food Log view, check-boxes appear next to all foods in the card. The user is able to easily check whichever foods they want to group in the card, and can confirm the grouping by finding the button in the bottom right has changed to contain a check mark (Figure 18). Upon tapping the button, the user is presented with a modal that requires a name be given to the grouping, and also shows the user all foods as well as serving quantities to the user before they group the foods. This allows the user to confirm and completely understand what they’re grouping. The user is able to create or cancel the grouping by tapping the buttons below the summary of the groups components (Figure 19). Upon creating the group, the card, and all other cards containing those specific foods will simply show the name of the new group instead. The grouping text color is blue instead of black implying an interaction, and upon tapping the group

(Figure 23-24), the user is presented with a similar modal pop-up that allows them to change the name as well as remove any food item in the group by tapping the red “x” next to each food item.

As mentioned in the on-boarding process, the user is prompted to take a weekly progress photo to help measure progress over time. To do so, the user simply navigates to the “Progress” global view which contains all of the weekly images the user has taken since they first installed the app (Figure 25). The images are presented in an album-like fashion, in a two column grid view, beginning with the current weeks’ image. The two column view ensures that the user is comparing progress from the prior week, and each image mentions the week in which it was taken above, for example: “Week 8.” Just as can be found in the Summary and Food Log global views, the Progress view contains a button in the bottom right, but this button is for taking a weekly progression photo. Upon tapping this button, the user can immediately take a picture of themselves for the current week (Figure 26). It’s important to note that this button is not visible if the user has already added a picture for the current week. If the user has not added their weekly progression photo and the end of the week is approaching, they will be notified via a push notification, as well as in-app prompts. It’s important to note that the user can *always* opt-out of taking weekly photo’s but it is encouraged to do so.

Once a week, the user will be prompted to motivate another random user of Motivate. See (Figure 27) for a high level view of the motivating process. The user will receive what is called a “Snackbar notification,” reminding them that they need to motivate another user. The notification comes from the bottom of the users device, and is persistent. Upon tapping the motivate text on the right of the notification, the user is prompted with a modal pop-up window, which shows them the first name of the user they’re motivating. The user has the ability to not send the message by hitting the cancel button, but the notification will remain at the bottom of the screen until the user has motivated another user. Motivate contains a store of more than 100 motivational messages that the user can choose from, and auto-populates the motivational message body with a pre-built message (Figure 29). If the user doesn’t like the message, they have the ability to simply tap the message body to refresh the text with a new message. If the user desires, they are able to send their most recent weekly progression photo along with the message, but by default this check-box is not selected as many users will want to keep their initial progression photos private. It’s important to note that you as the user will always be motivating those who are “behind” you in terms of physical goal completion in the same goal type category.

On the opposite end, users will also be motivated during the week, see (Figure 32). They will randomly receive a “Snackbar” notification at the bottom of their screen which will again be persistent. Upon tapping the view button on the right of the notification, the user is shown a modal pop-up card of the motivational message (Figure 35). If the user who sent the message opted to include their most recent weekly progression image, that image is shown in a circular “chip,” according to the Material Design guidelines. After hitting the “okay” button, the user is able to dismiss the modal card and continue using the app.

## 2.2 Interface Design Rationale

Through our initial research, the team found that the majority of diet/weight tracking apps did not *effectively* motivate users to regularly meet recommended dietary goals, *consistently*. This was due to the fact that many users didn't see immediate changes in their physical appearance as quickly as they'd like to, and soon felt that the app was not serving a purpose other than making them feel less confident about their own figure. Solving this issue of habitual use was the main problem we wanted to address through our application's interactions and interface. We really wanted to develop an interface that was "habit forming," and would continually trigger users to interact with the app in order to achieve their physical goal.

To effectively build a habit forming experience, the team conceptualized a number of different interactions that would keep users coming back to our app, in a positive manner. The first technique we used was the "weekly progression photo."

Once a week if the user hasn't already, he/she will receive a notification saying something along the lines of, "Hey, this is what you looked like last week! Take a picture of yourself to see your progress." Leveraging notifications in a weekly manner means that we can build a weekly habit for the user to follow, and more importantly internally feel the need to accomplish. Theoretically after a number of weeks, users will look forward to comparing their current physique to the prior weeks. Once the user has reached that stage, we know that we have successfully built a positive habit which will ensure our users won't give up using the app, and more importantly, we believe they'll be less likely to give up on their physical goal.

The above interaction, however, is not enough to ensure users will stay motivated. Looking at what you've accomplished every week only goes so far and we confirmed this after finding that the majority of internal motivation comes from admiring the appearances of others (who are much further along in the process of what you want to physically look like). An example of this would be if you are interested in getting very fit, and you go to the gym and see a professional athlete working out near you. The athlete serves as a motivator in showing you what you could potentially look like if you work hard enough. Because of this, the team developed "motivational messages."

Like many existing diet tracking applications, Motivate recommends a daily calorie count as well as more fine grained metrics (protein, carbohydrates, fats). But how do we ensure that users stick to these metrics and eat as prescribed every day? Once a week, you're prompted to "motivate" another user of the app via a motivational message. The interesting part about the motivational messaging system is that you will always be motivating a user who isn't as far as you are in the physical transformation process. Imagine you have just started to work out and have told Motivate that you want to get stronger. There are a number of existing users who have made considerable gains and may now represent what you aspire to look like. These are the users who will be motivating you.

Motivate offers more than 100 default messages that you can choose from to quickly craft a message. The power of this interaction doesn't lie on the senders side, but on the receivers. When you receive a motivational message from another user, you see the weekly progress picture they

took last week if that user decided to include it. This works to visually inspire you, the receiver of the motivational message, to press on and keep working hard to ideally look similar to the very person that just motivated you.

Continuing on the idea of comparative visual motivation, Motivate leverages an image in the header of the dashboard view which can be seen in Appendix A. This image will randomly change whenever a top-level view is visited, and is specific to the archetype that the user identified they wanted to “look like” in the initial on-boarding process. The image not only conforms with Google’s Material guidelines, but also works to further motivate the user.

From a component standpoint, Google’s Material guidelines emphasize the use of “cards” which work to separate different information that is often within the same category. On the dashboard we chose to separate each of the food building blocks (calories, proteins, etc.) in their own smaller cards which keeps the app from looking overly busy, and ensures that it looks clean. Another common component in Material is the circular button in the bottom right corner of the view, which is often a view specific call to action. We chose to use this button as a simple way to let users know that tapping it would allow them to add a food from the Summary and Food Log views. Upon tapping the button, the user is immediately taken to the search view. Looking at existing food logging applications, one of the most popular apps in terms of usability happened to be the Fitbit app. The app smartly shows users previous searches and entries in a list underneath the top search bar before they have even selected the search field to type into (see the screenshot below). Given that many people eat on a weekly schedule, it’s highly likely that they will eat the same meal/food more than once a week, so providing these “pre-search” foods is extremely valuable in making the entry process almost effortless for those that follow a food plan that incorporates the same meals for each day of the week.

Upon typing in the search field, a by-character auto-complete is used to filter and recommend foods in real time in the list view below the search bar, again helping to make the entry process as easy as possible. It’s worth noting that after our user interviews, we found that a lot of people just didn’t feel like entering foods anymore because they felt it was annoying, so it was extremely important to us to make this process as seamless as possible.

The team also found that many users of existing weight-watching applications liked to have the ability to group foods together to form “meals” that they consume relatively frequently. Motivate implements this grouping functionality seamlessly into the Food Log view’s cards, allowing users to easily group food items together which can later be quickly selected in the above described “pre-search” list view without typing a single character.

In order to keep users in a positive mindset, we realized that an upbeat, bright color needed to be used. Again conforming with Google’s Material guidelines, we chose a brighter hue of teal that really brings life to the app. The team felt that the teal really made the app feel “more fun,” and would ideally help increase the moral of our users. The team understood the importance of not overloading the user with color associations, so we decided to use a teal color across all views, and a mixture of off-white, grays, and blacks for the cards on screen in order to keep everything as readable as possible. The team chose to stick with the recommended Roboto type face according to

the Material guidelines, which is in itself fun and quite readable. The headers all use a white font color that conforms with Material, while each of the cards contain fonts with an off-black color to maintain high readability and minimize contrast issues. It's important to note that the Android operating system offers a number of color-blind modes that help make the experience in any application much easier on the eyes through a simple toggle in the system settings.

## Appendix A: Prototype

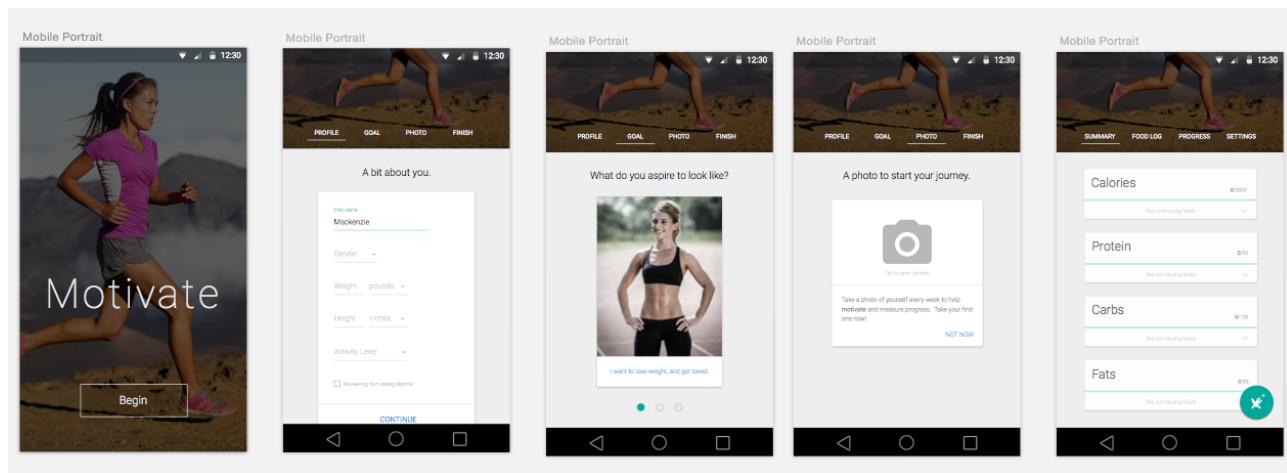


Figure 1: The initial on-boarding flow upon first app launch.

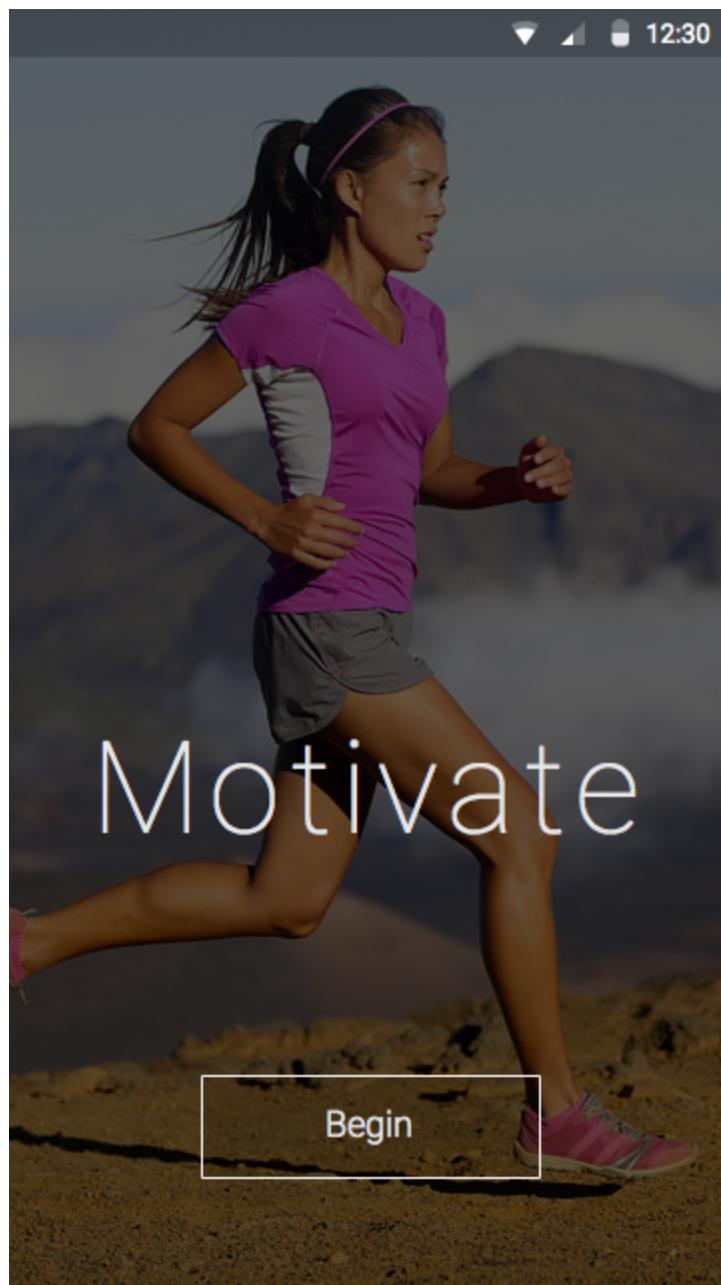


Figure 2: The introductory view in the on-boarding flow.

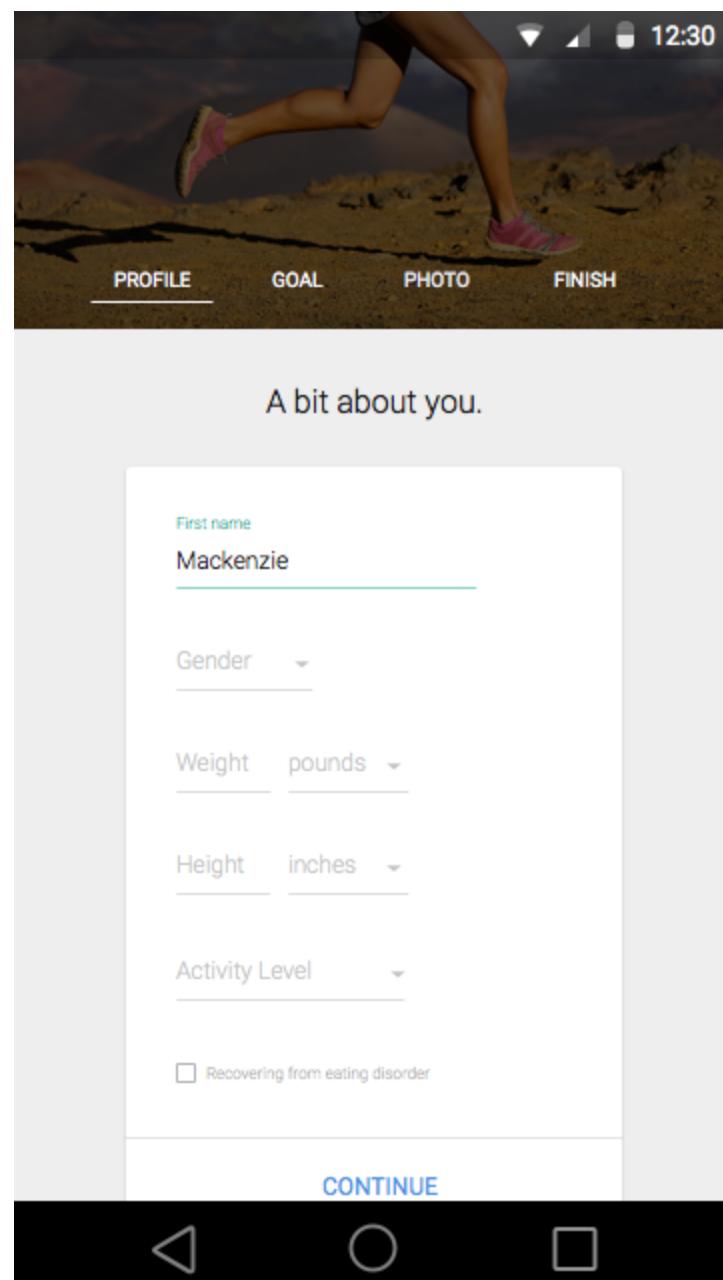


Figure 3: The first view in the on-boarding flow.  
Motivate asks the user to enter personal information.

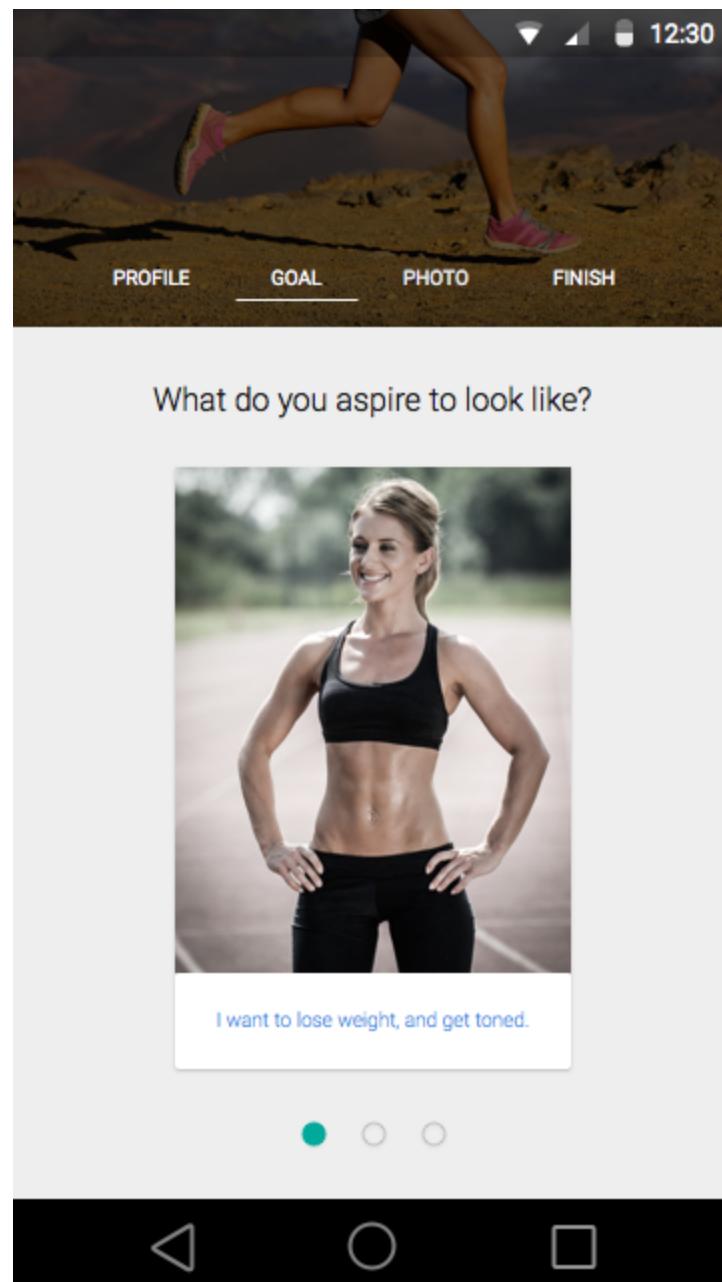


Figure 4: The second “goal” view in the on-boarding flow. The user is presented with 3 archetypes of what they aspire to look like. They swipe through the cards as denoted by the dots beneath the images card.

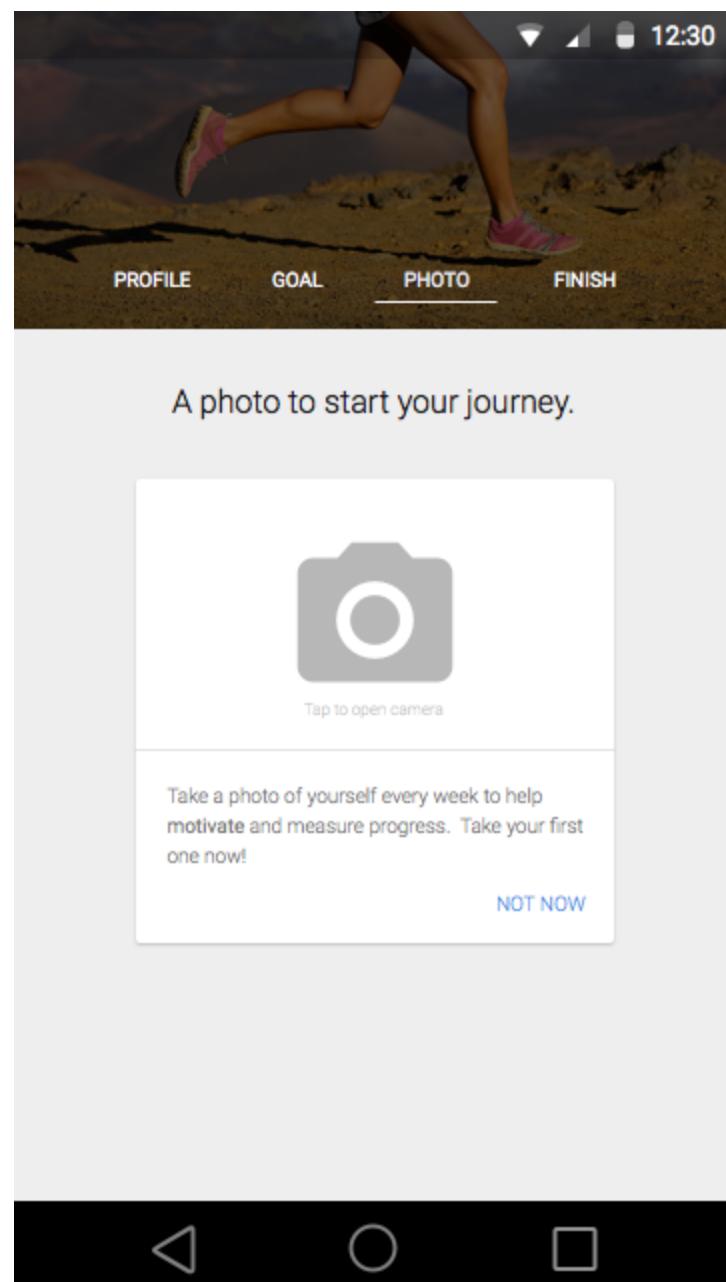


Figure 5: The third view in the on-boarding process. Motivate prompts the user to take their first weekly progression photo to kick things off.

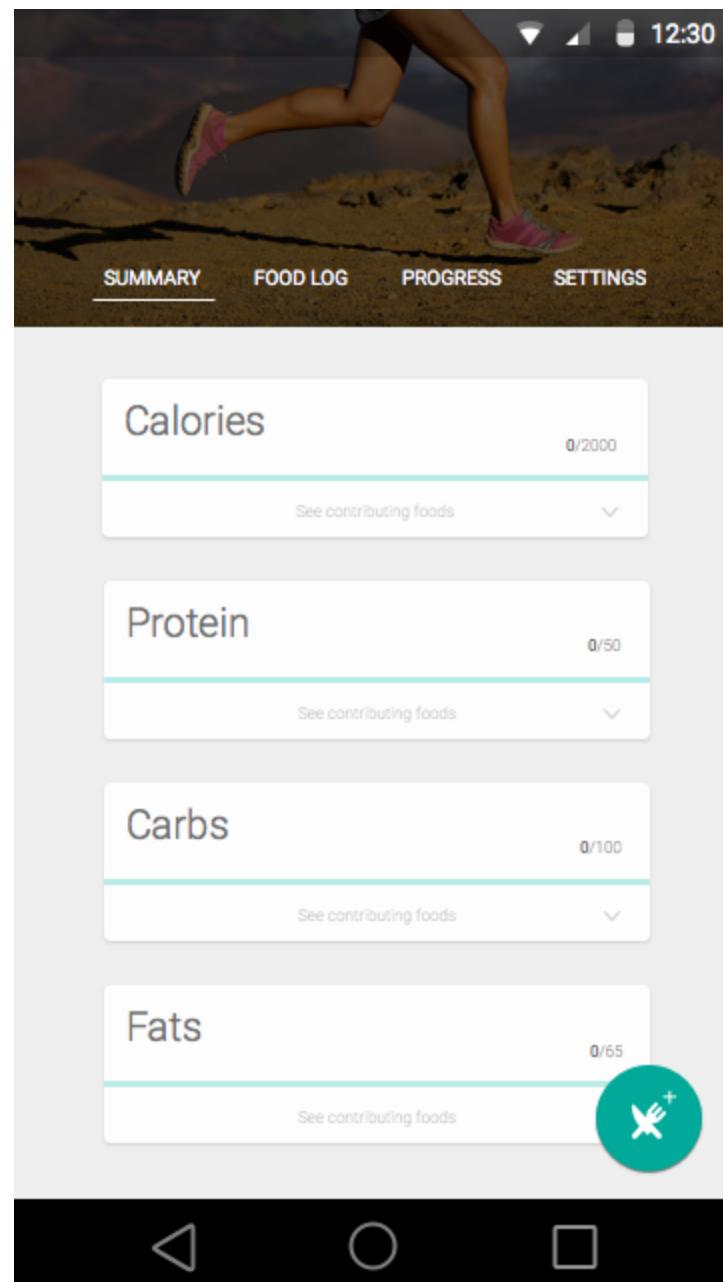


Figure 6: The on-boarding process is complete and the user is taken to the summary global navigation view. The view shows the users progress for each food component category (calories, etc.) as well as the contributing foods.

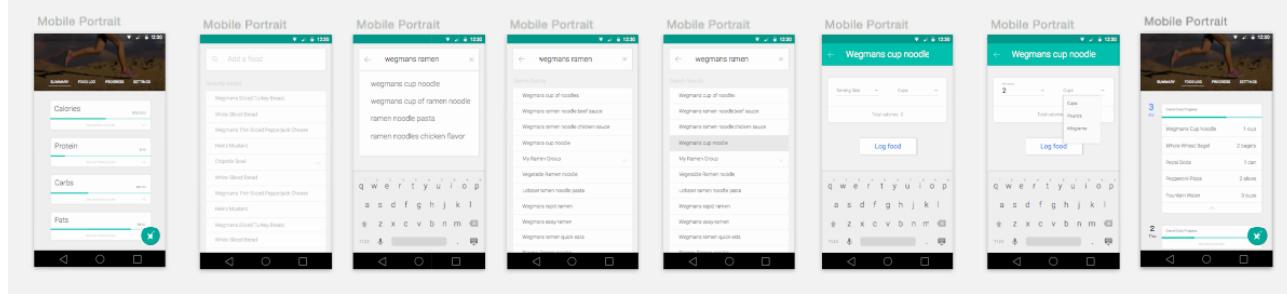


Figure 7: The Log Food Flow zoomed out. The user searches and logs a food into the system.

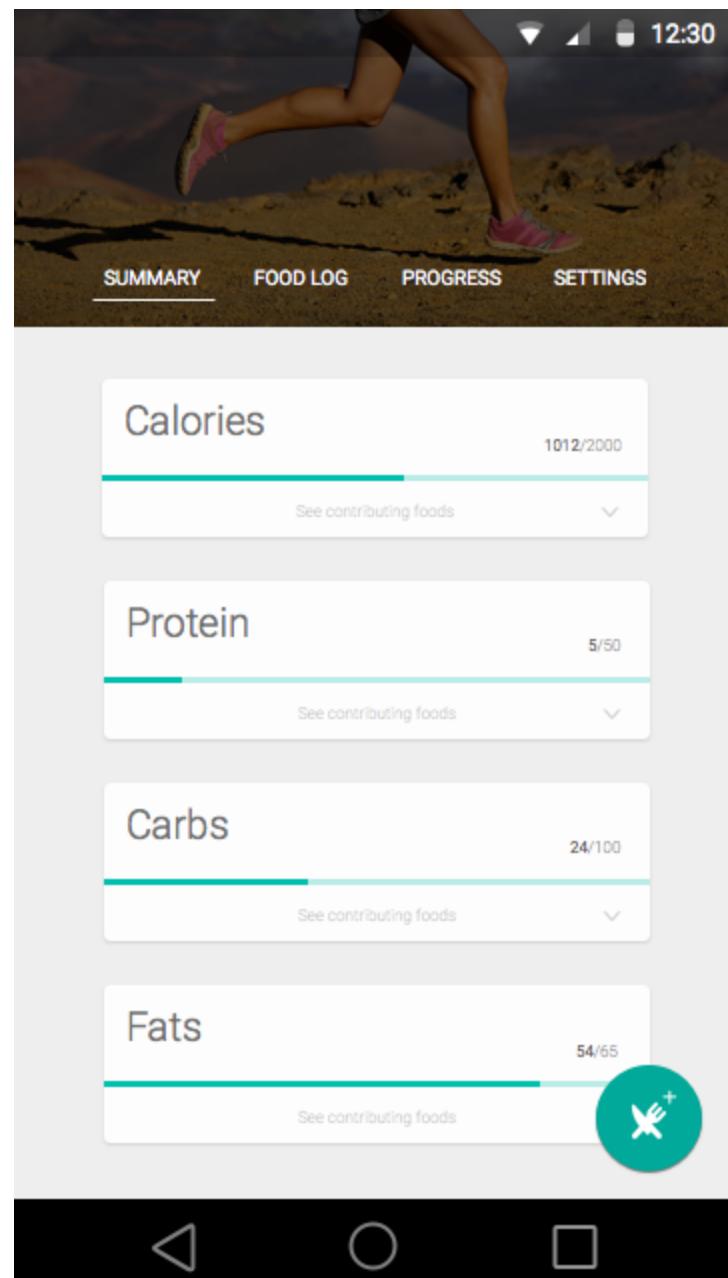


Figure 8: The recognizable Summary global view from which the log food interaction can begin.

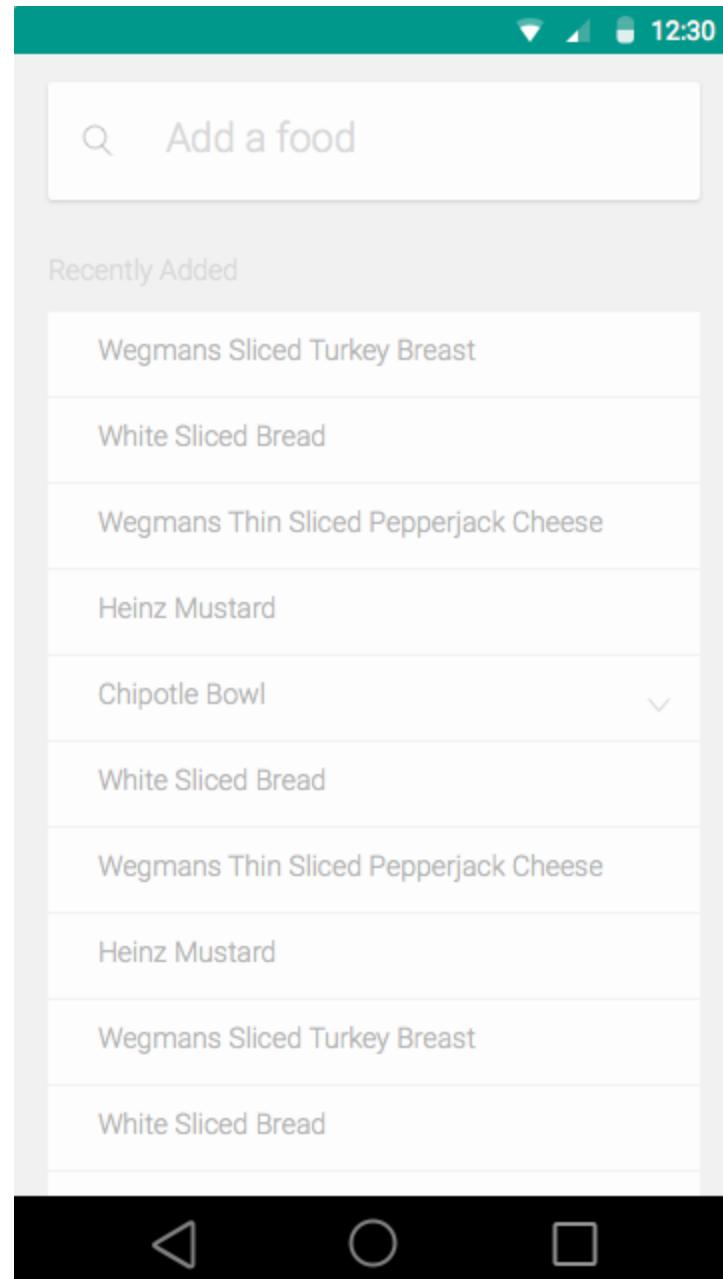


Figure 9: The user has tapped the add food button. A pre-search list shows the user his/her recently logged foods and groups of foods for quick selection.

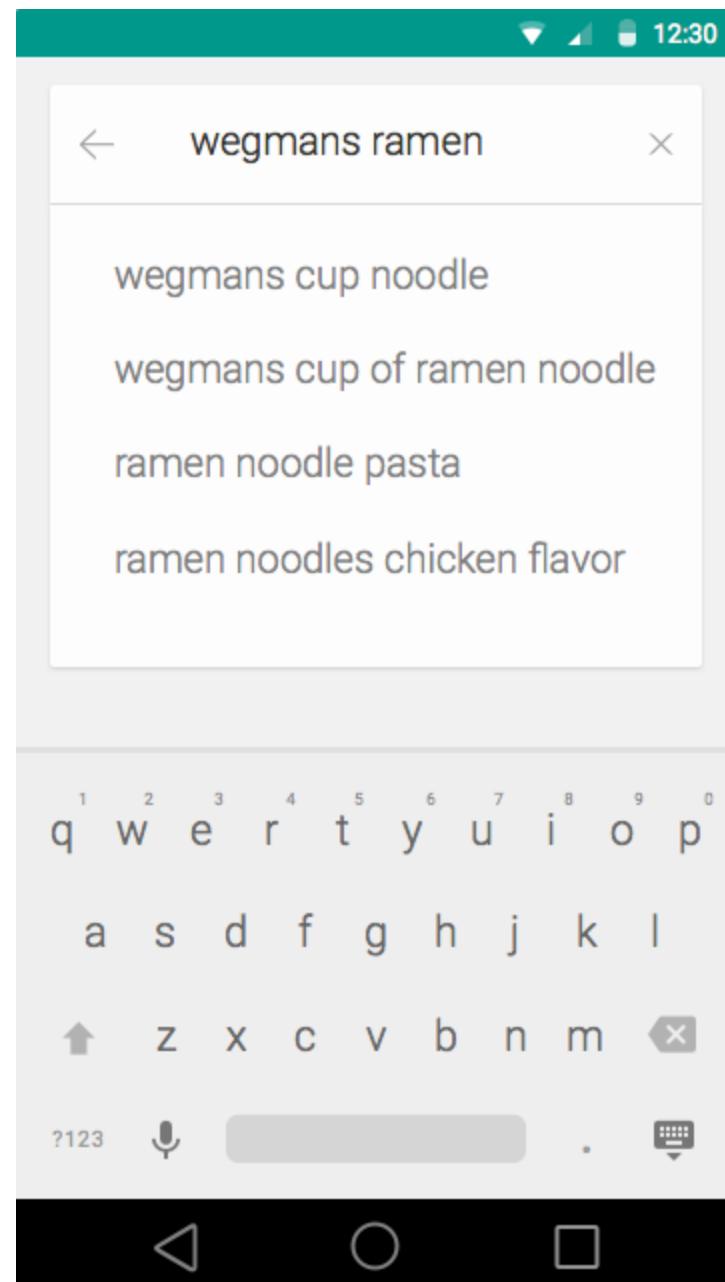


Figure 10: The user has began typing ramen noodles. The auto-complete suggestions appear below the search-bar for quick access.

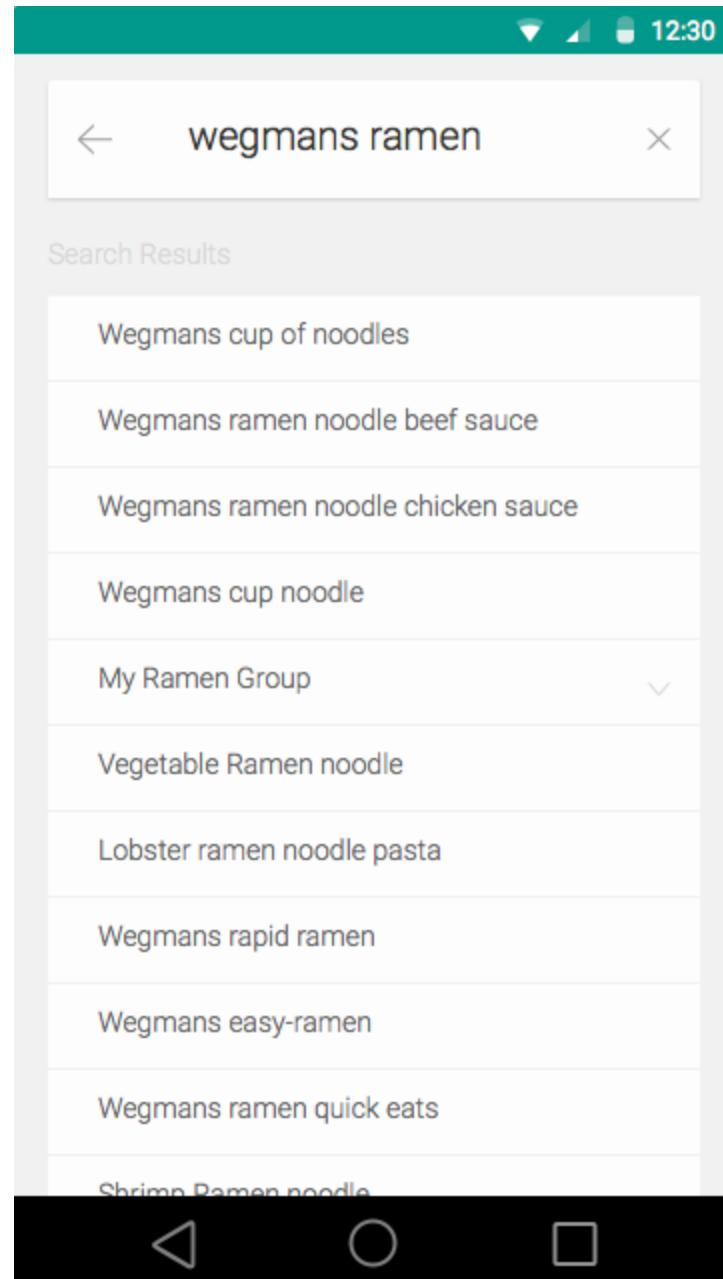


Figure 11: The user didn't select any of the suggestions and hit the search button instead on their keyboard. A list of results from the mentioned API's are shown.

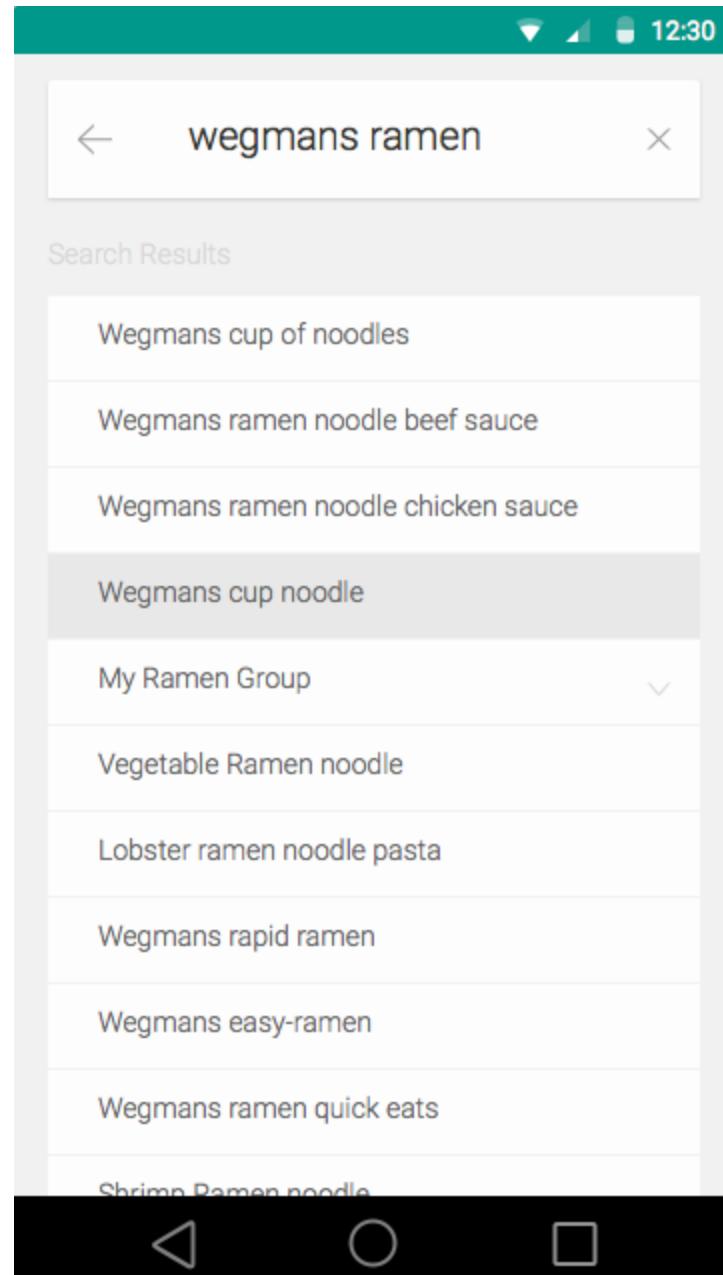


Figure 12: The user selects “wegmans cup noodle.”

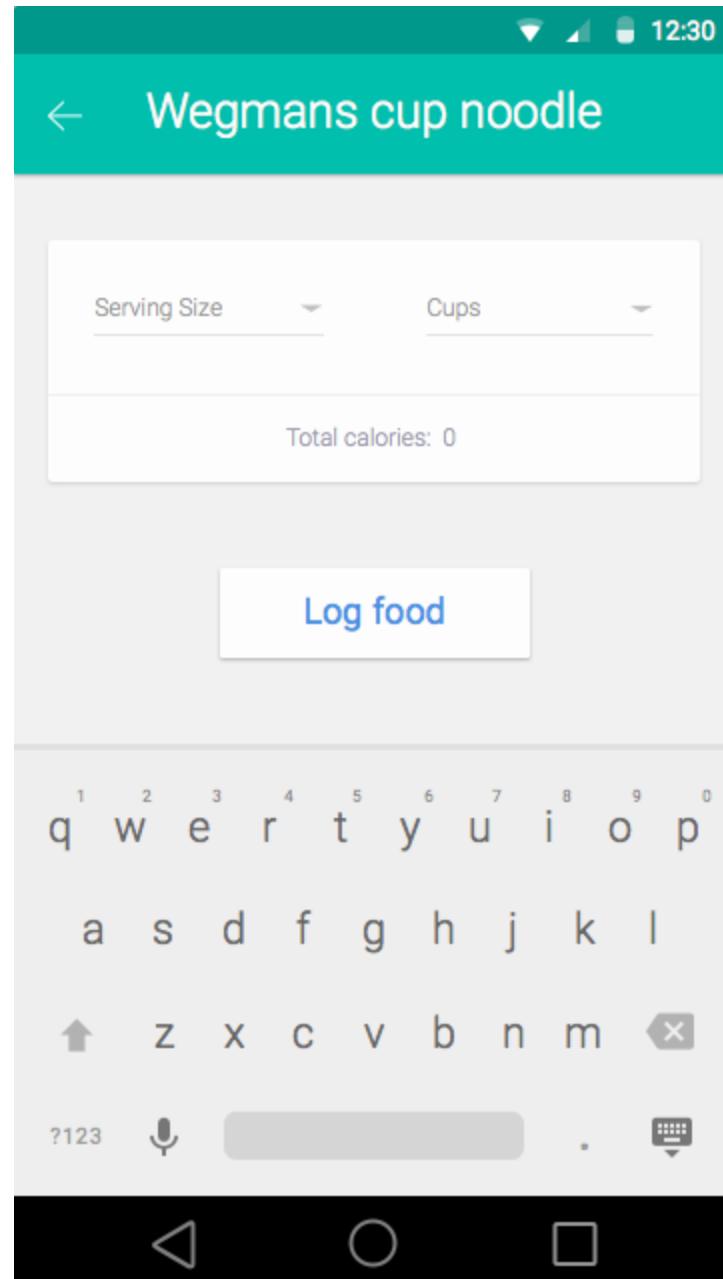


Figure 13: The user is taken to a new view to choose the serving size and measurement type. Motivate automatically chooses the most fitting measurement type (ex. “cups”)

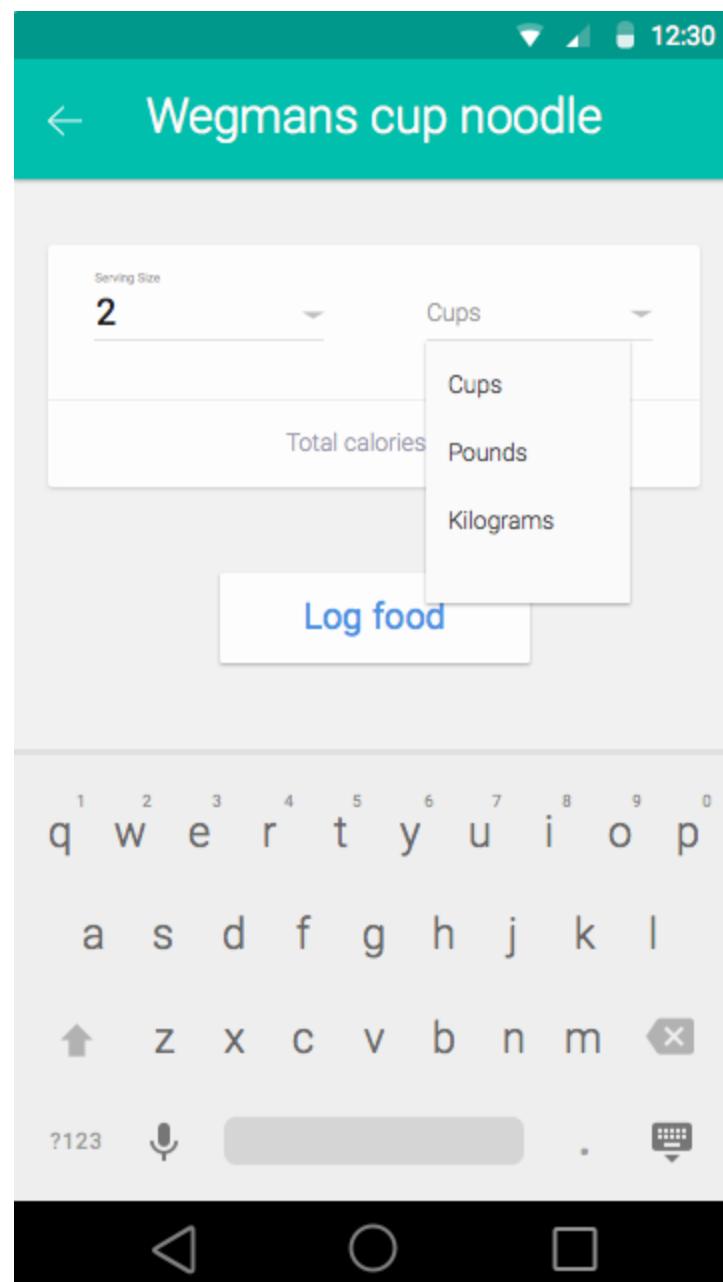


Figure 14: The user selects the measurements drop-down and selects cups. The user had 2 cups of ramen noodles. The user taps the button and logs the food.

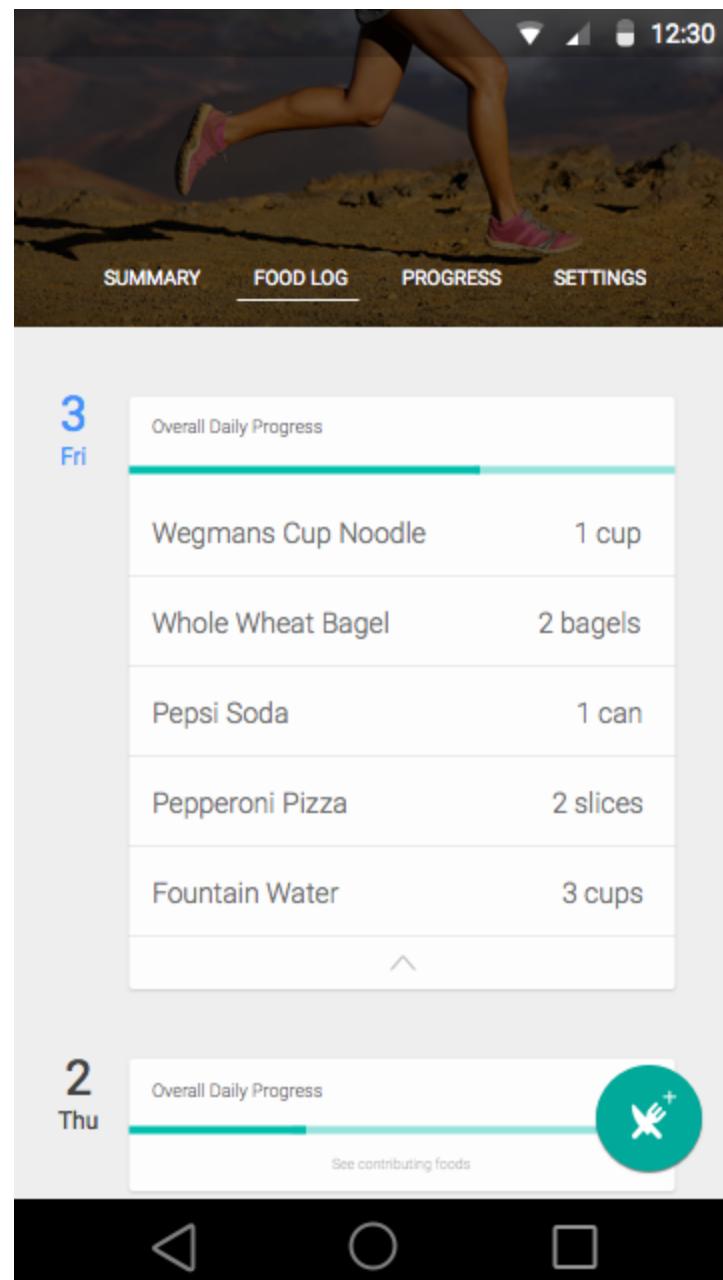


Figure 15: Upon logging the food, the user is taken to the Food Log view and can see their most recent addition as well as all other foods they've added today.

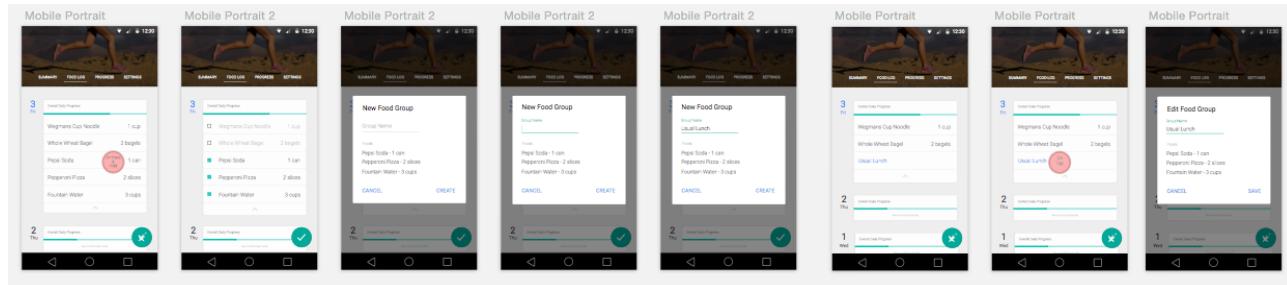


Figure 16: The Group Food high level flow.

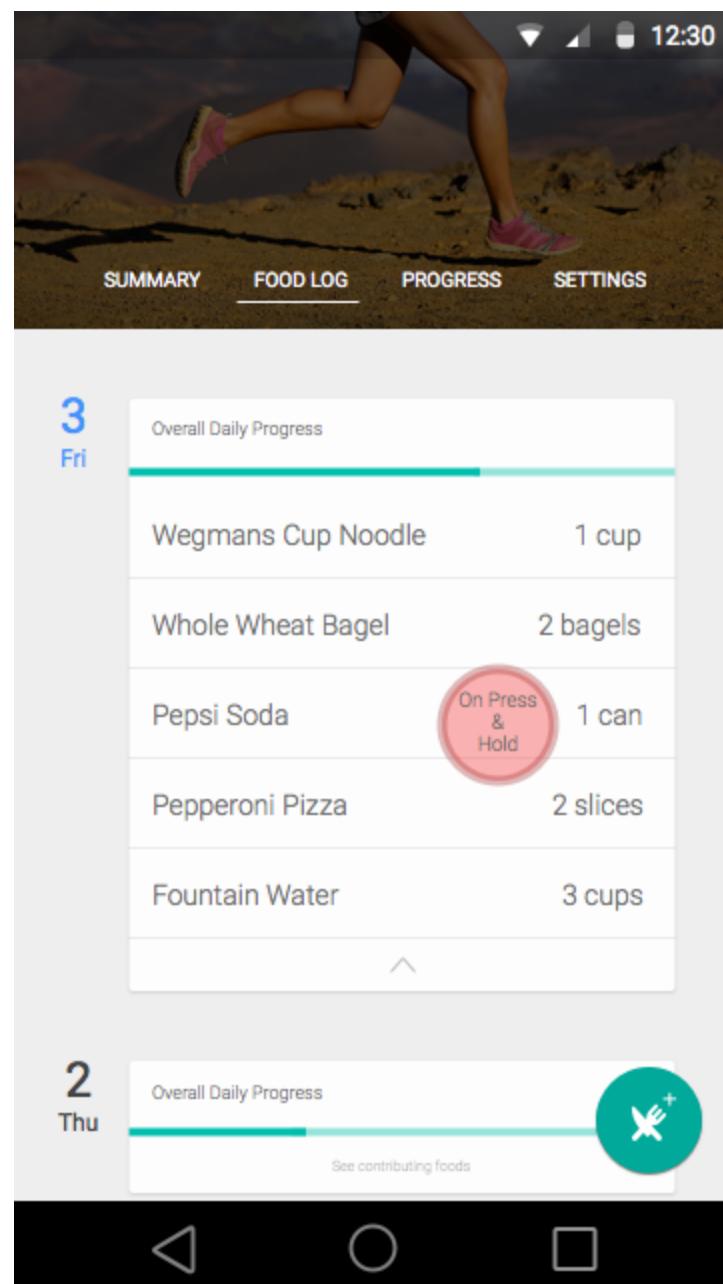


Figure 17: The user long-presses on a food to initiate the grouping interaction.

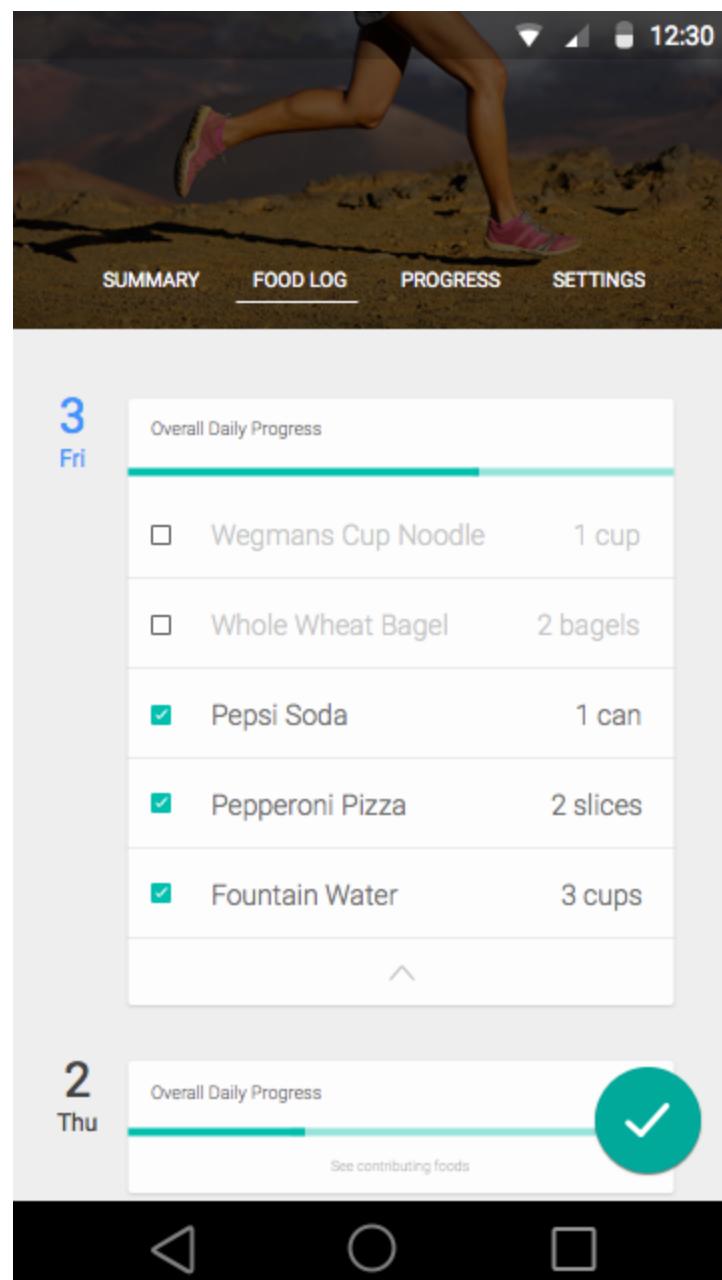


Figure 18: The user has initiated grouping, and sees check-boxes appear next to each food. All checked items can be confirmed as a grouping by tapping the morphed check-mark button in the bottom right corner.

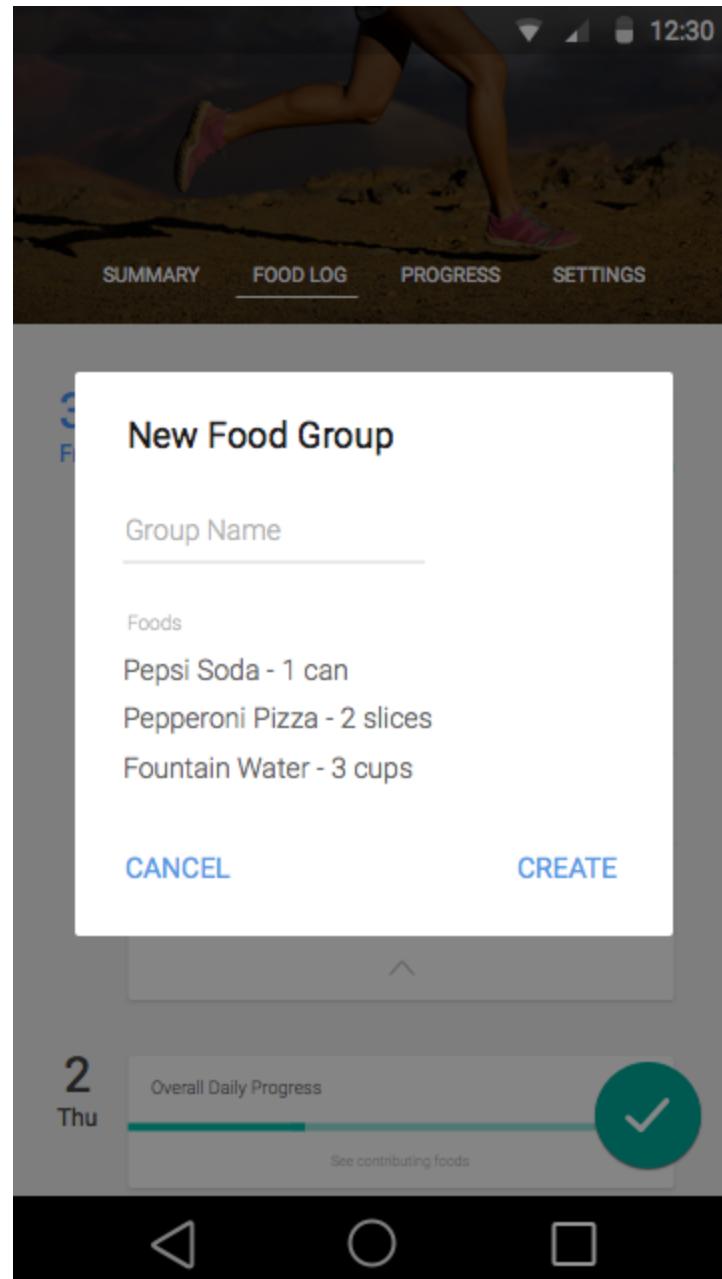


Figure 19: A modal pop-up is presented and the user can now name the group as well as confirm which items are in the grouping.

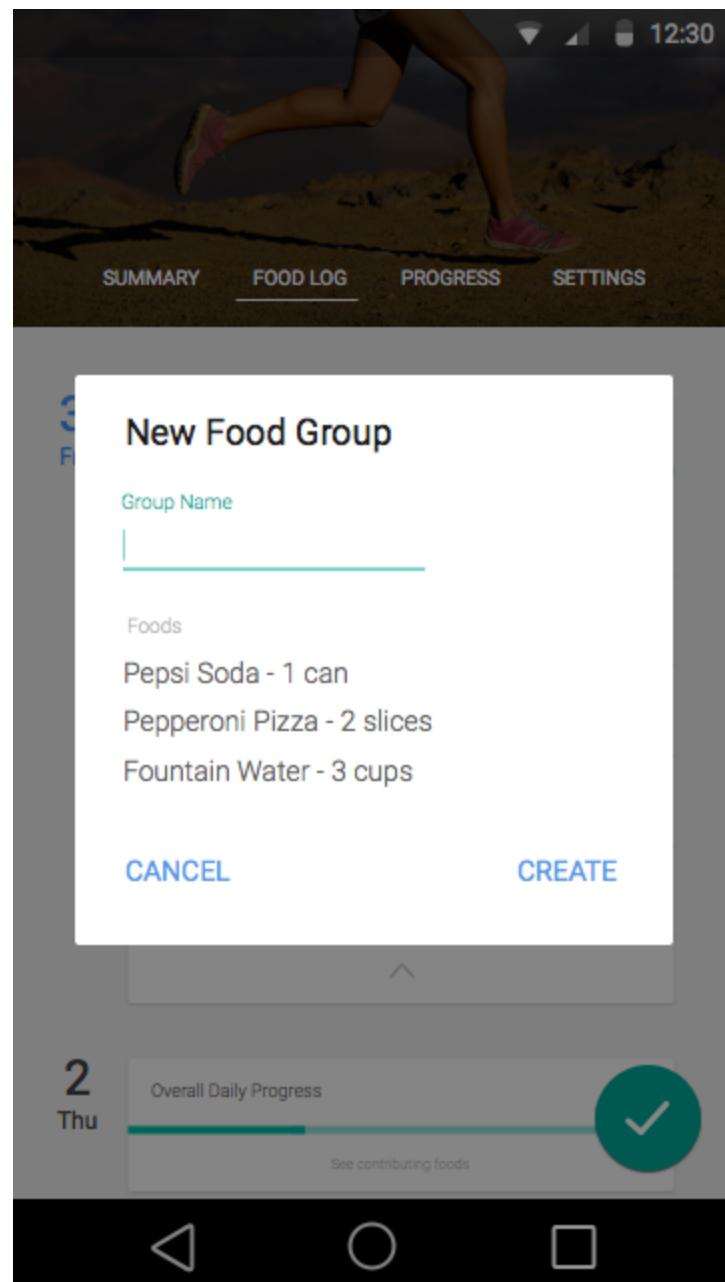


Figure 20: The user selects the name field.

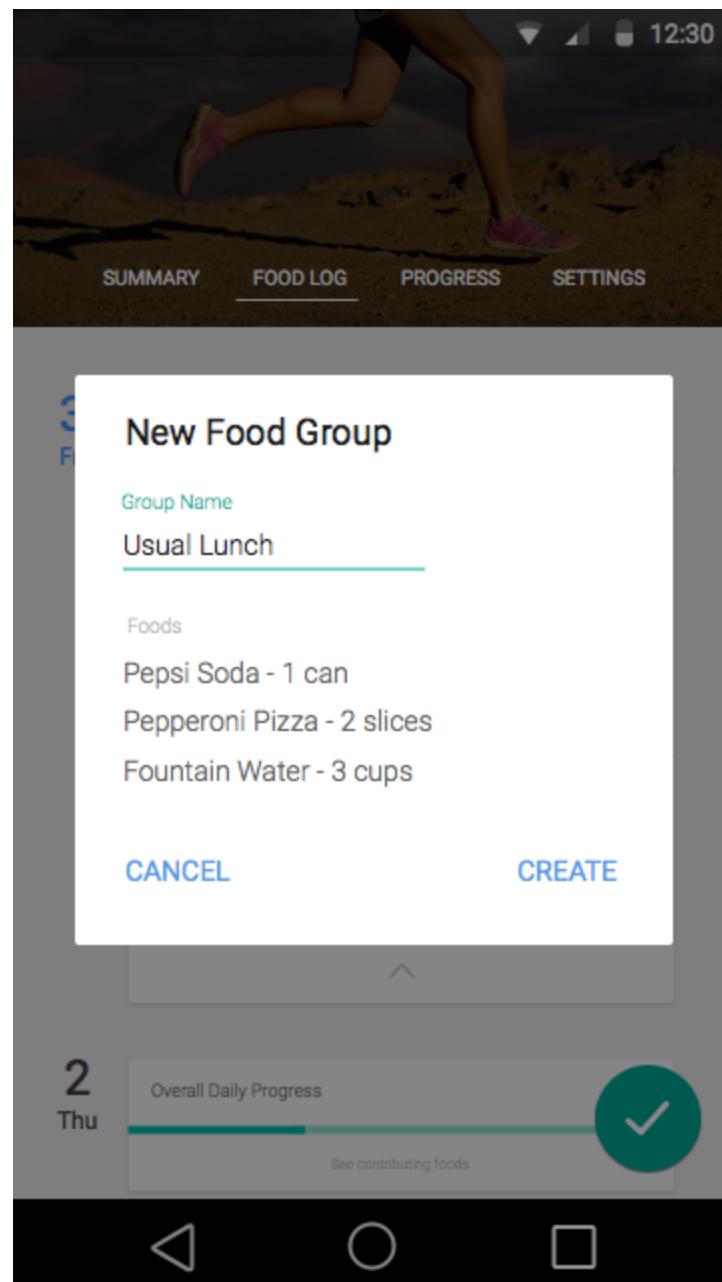


Figure 21: The user enters a new group name and hits the “create” button to group the items together.

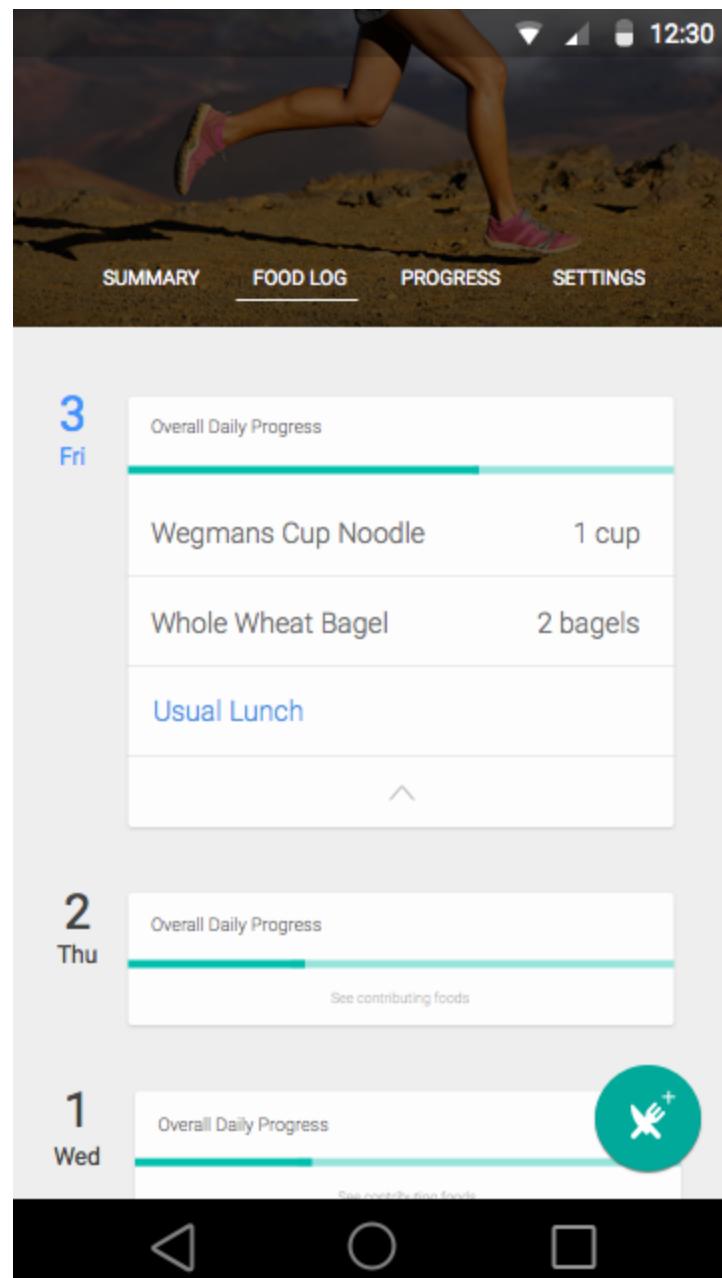


Figure 22: The user is shown the new grouping in the card. The grouping, and all others have blue text.

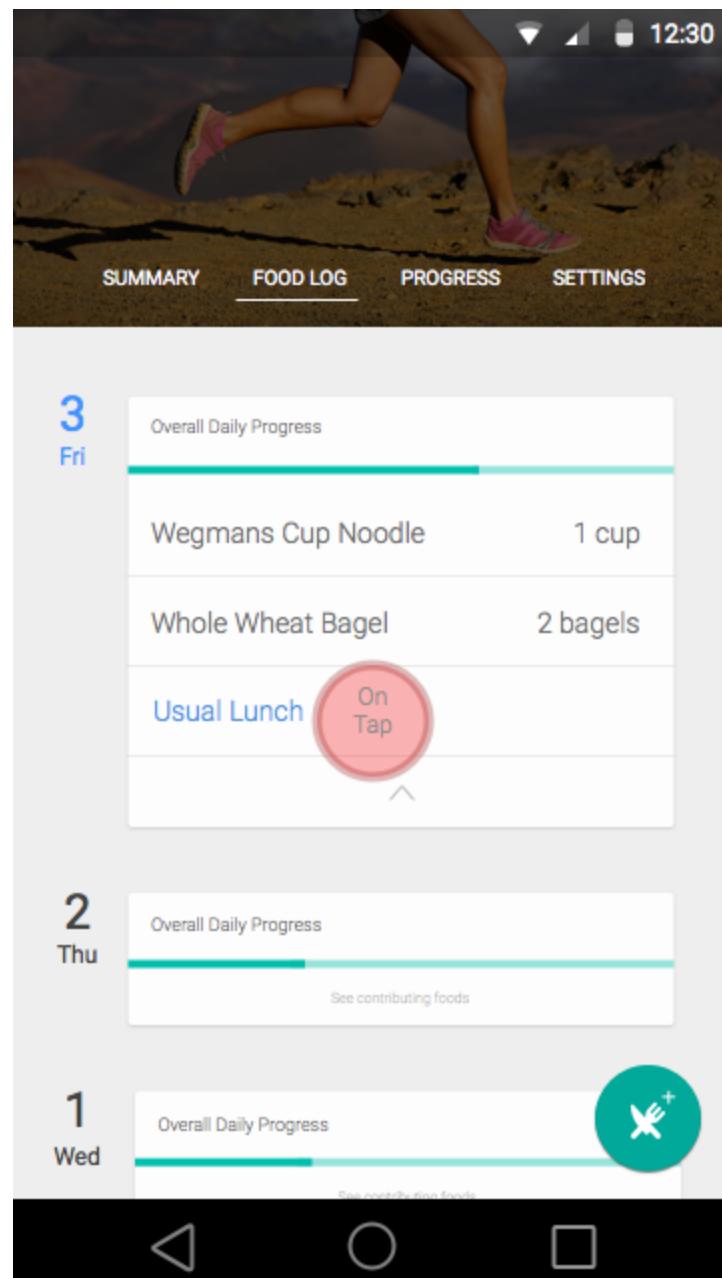


Figure 23: The user can tap the food group to edit the grouping, or rename it.

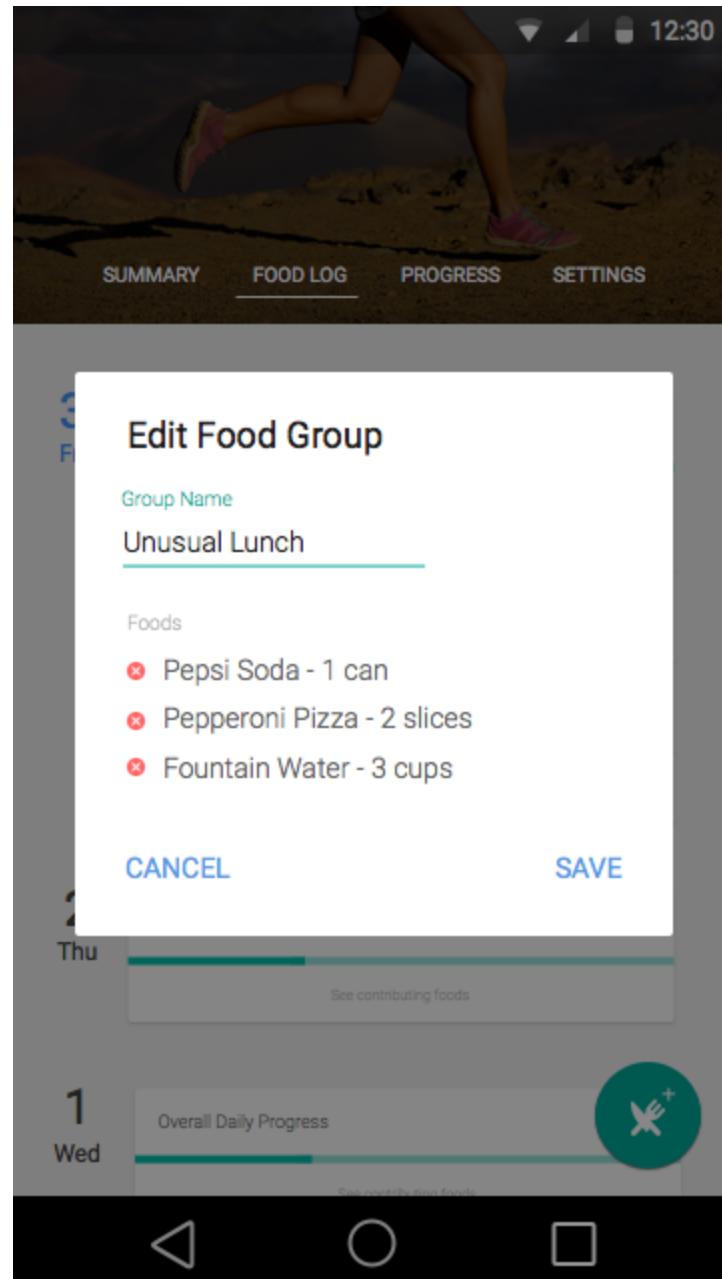


Figure 24: The user changes the name of the grouping, but decides to leave the foods in the group intact and taps the save button.

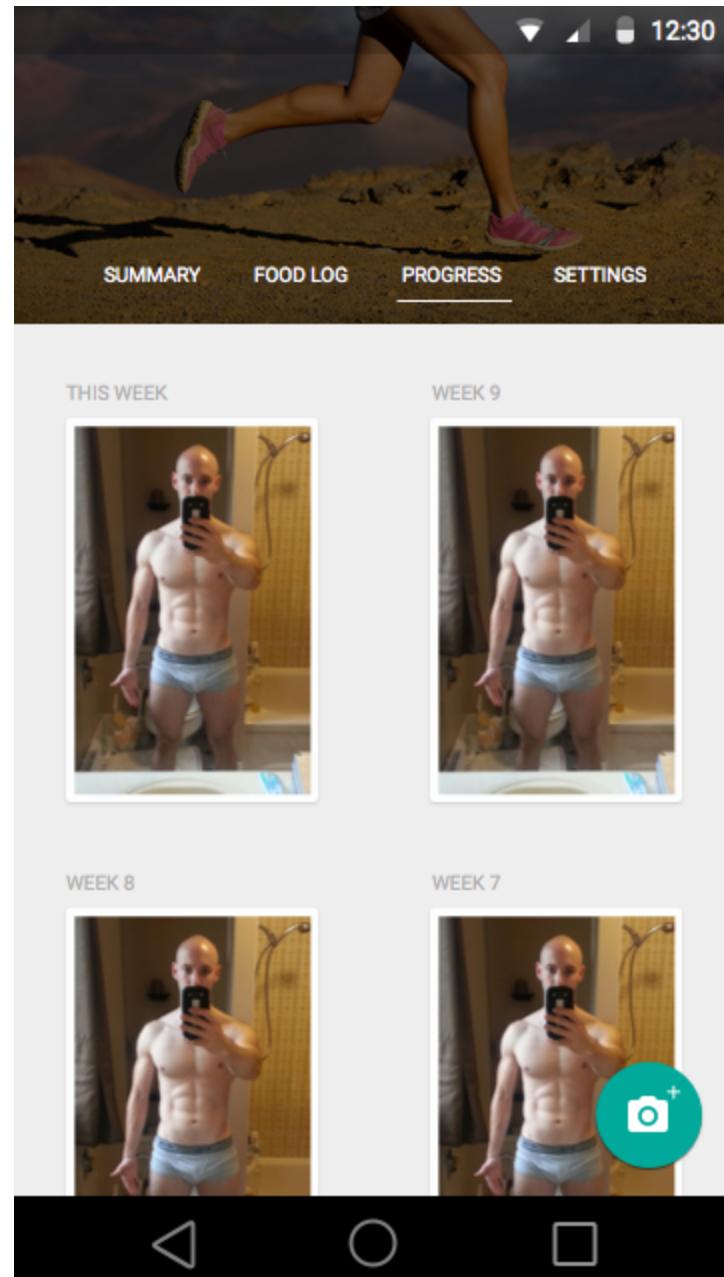


Figure 25: The global Progress view. An album of the user's weekly progression photos. If the user hasn't yet taken a photo for the current week, a button in the bottom right is shown.

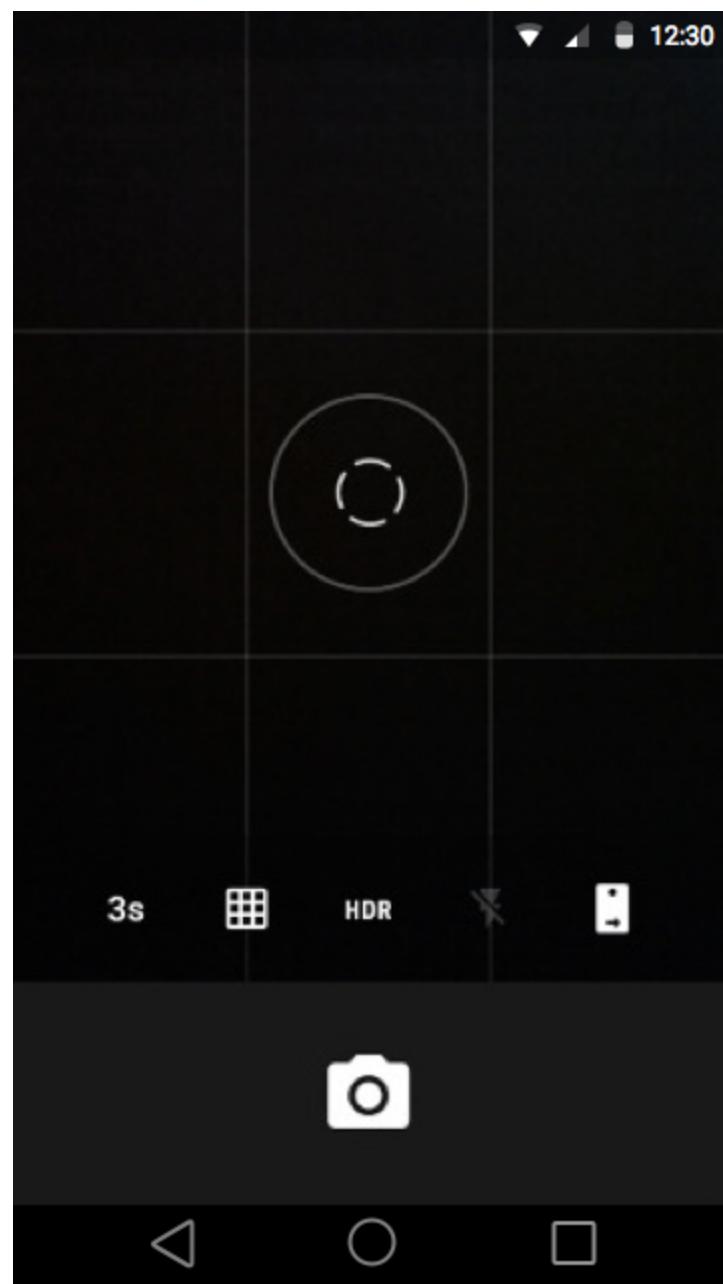


Figure 26: Upon tapping the above button, the user can immediately take a photo and return to the Progress view.

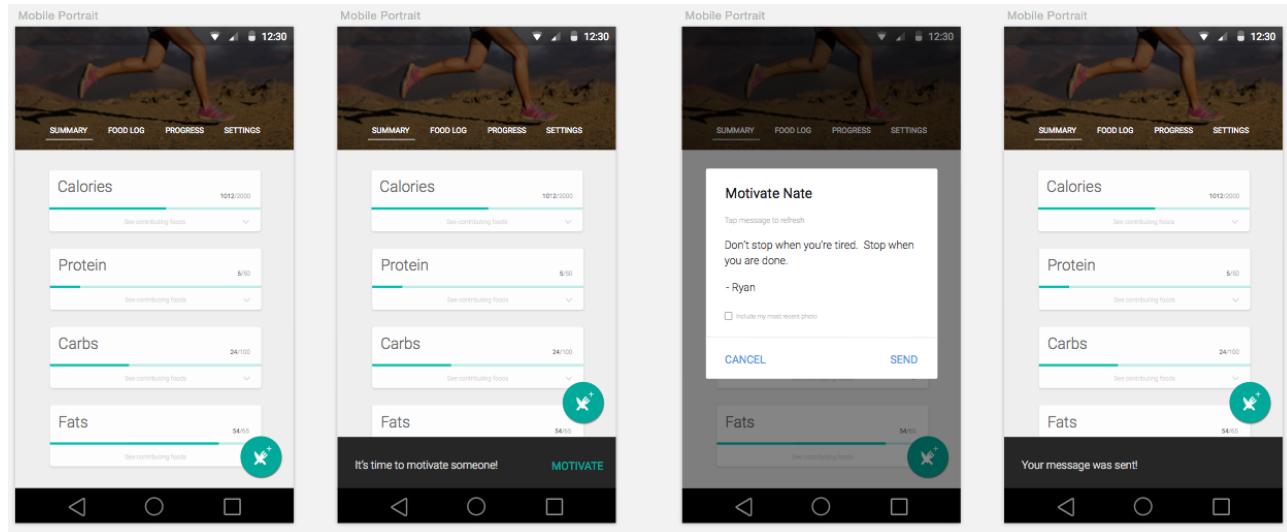


Figure 27: Motivate another user high level flow.

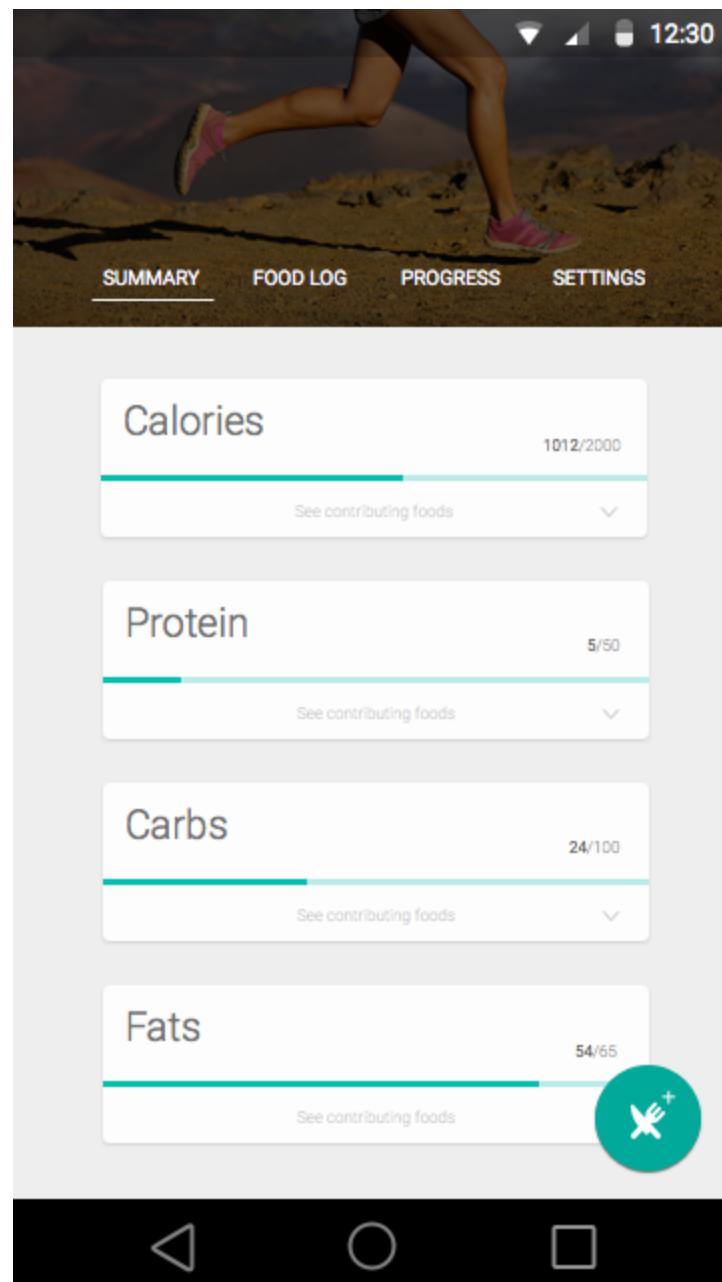


Figure 28: The user is currently on the summary view looking at their daily dietary progress.

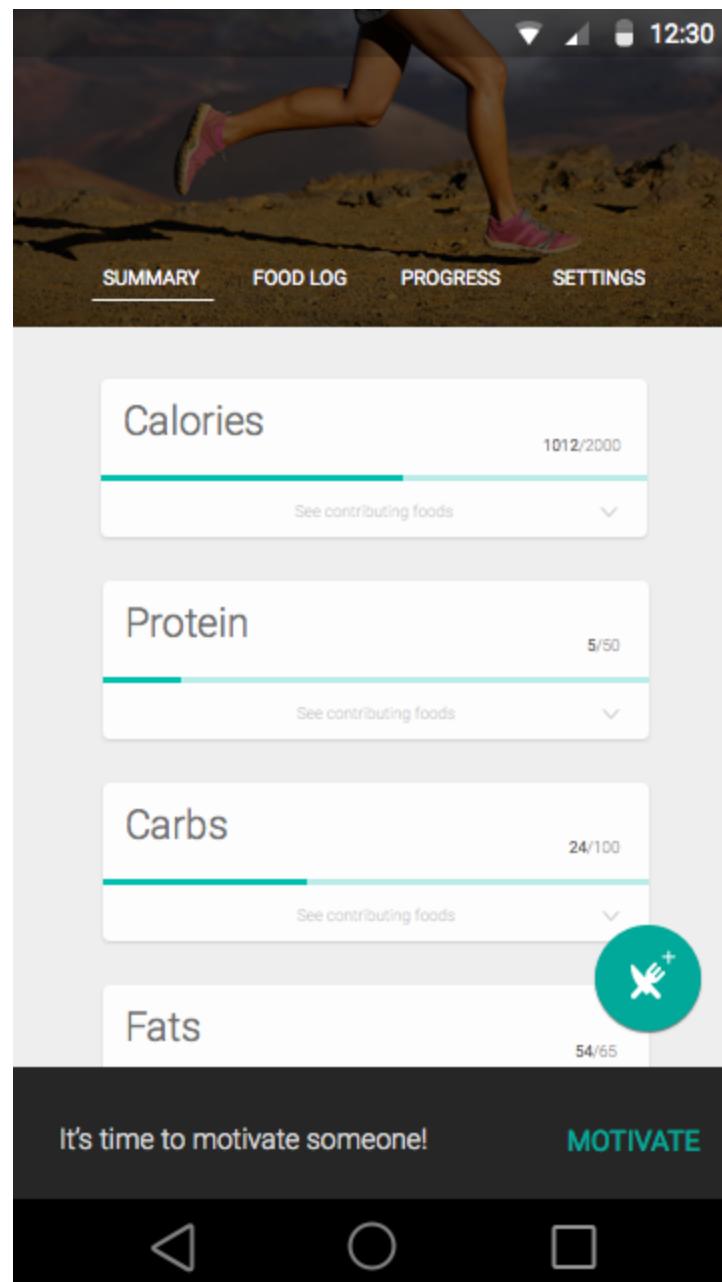


Figure 29: A persistent notification slides up from the bottom of the view, letting the user know they need to help motivate another user.

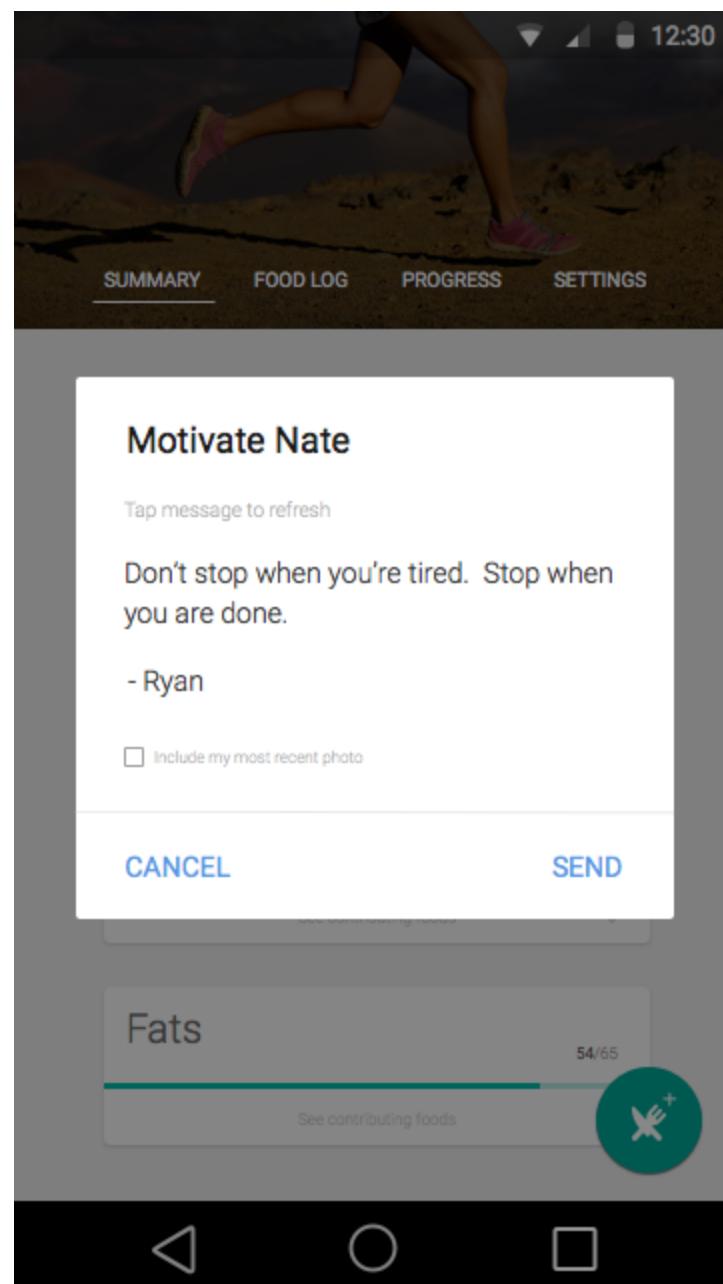


Figure 30: The user is presented with a modal pop-up containing one of hundreds of pre-built messages. He/she can easily tap the message body to generate a new message. The user also has the option to include their most recent weekly progression photo via the check-box towards the bottom.

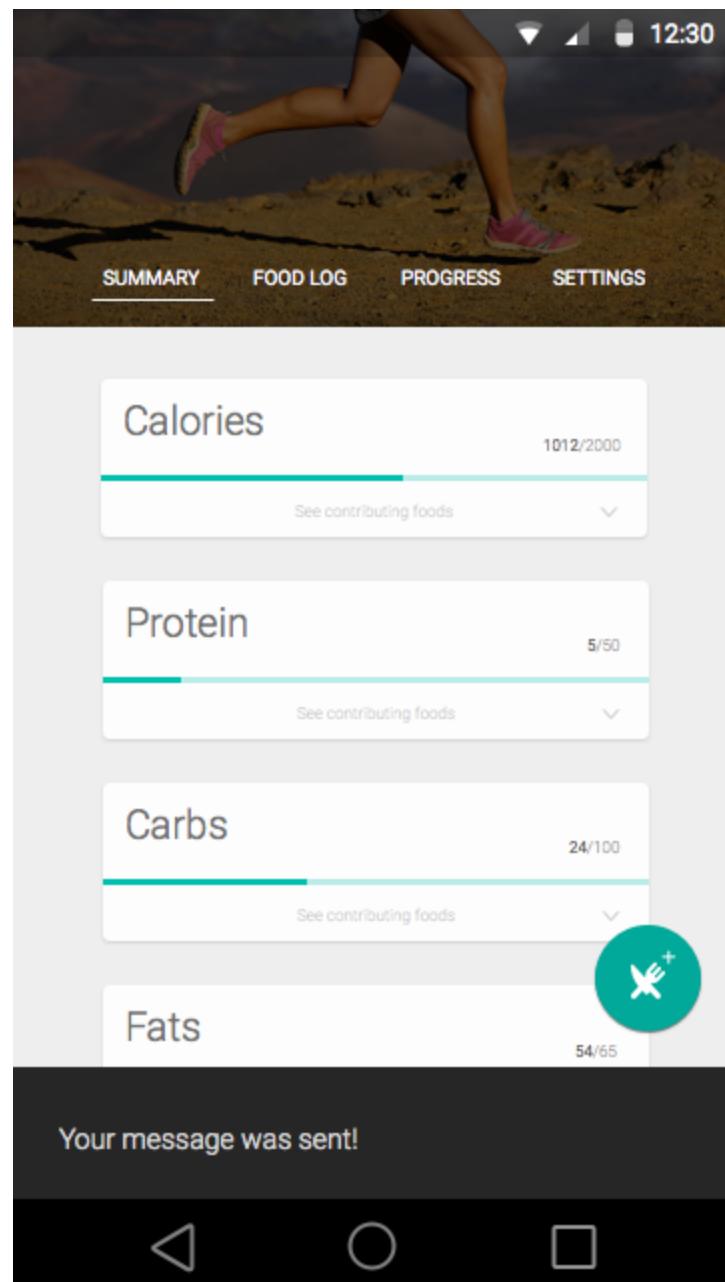


Figure 31: The user has sent the motivational message.

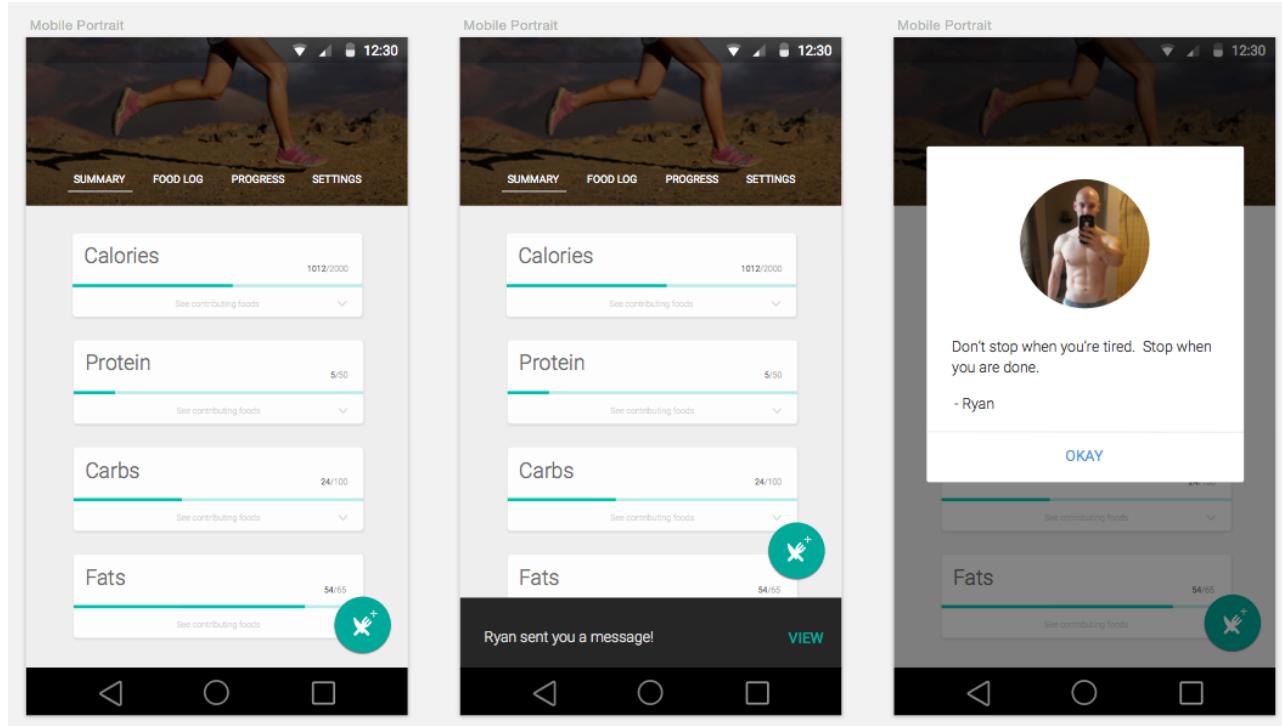


Figure 32: The user is being motivated by another user, high level flow.

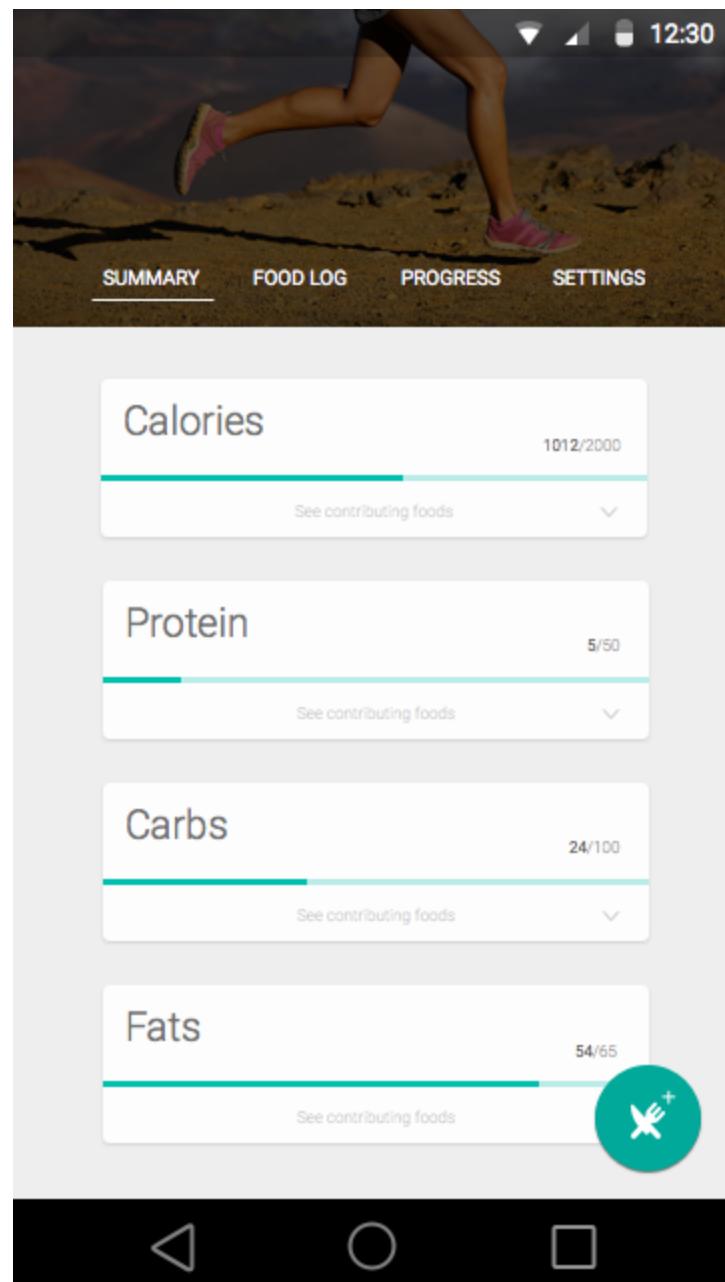


Figure 33: The interaction begins on any view. The user is currently on the Summary global view.

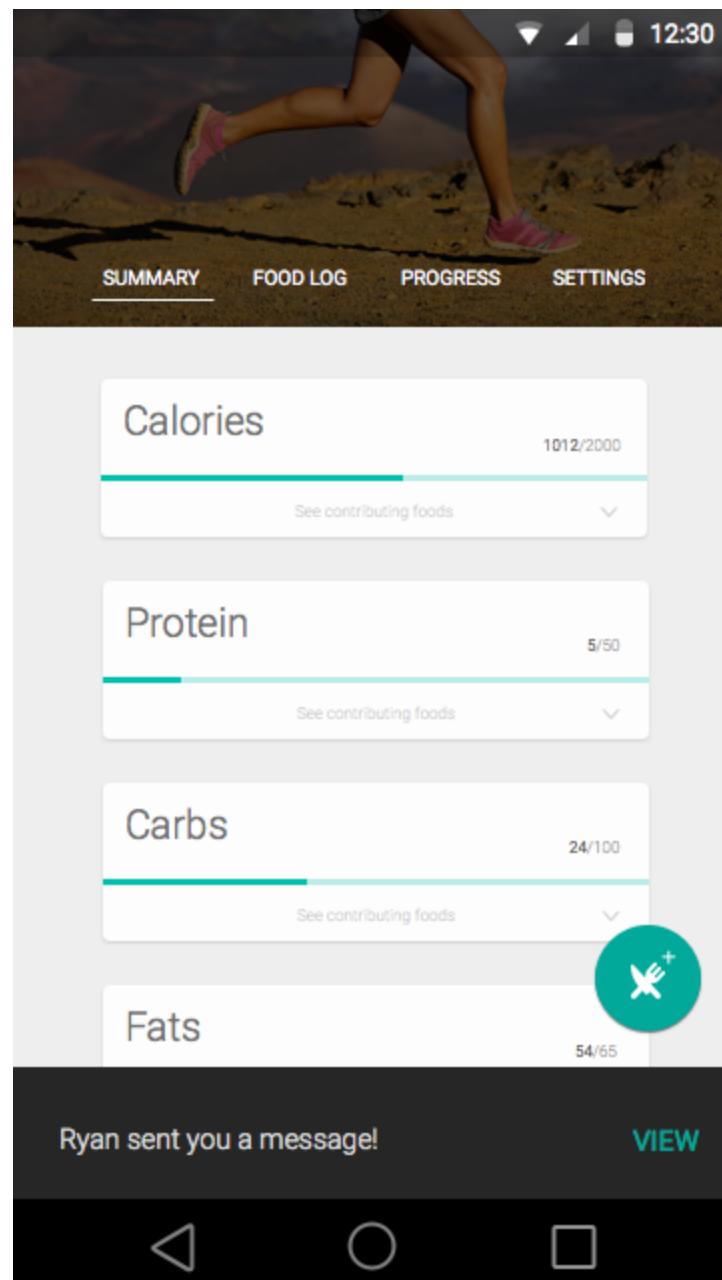


Figure 34: A persistent notification slides up from the bottom of the view. The user taps the “view” button.

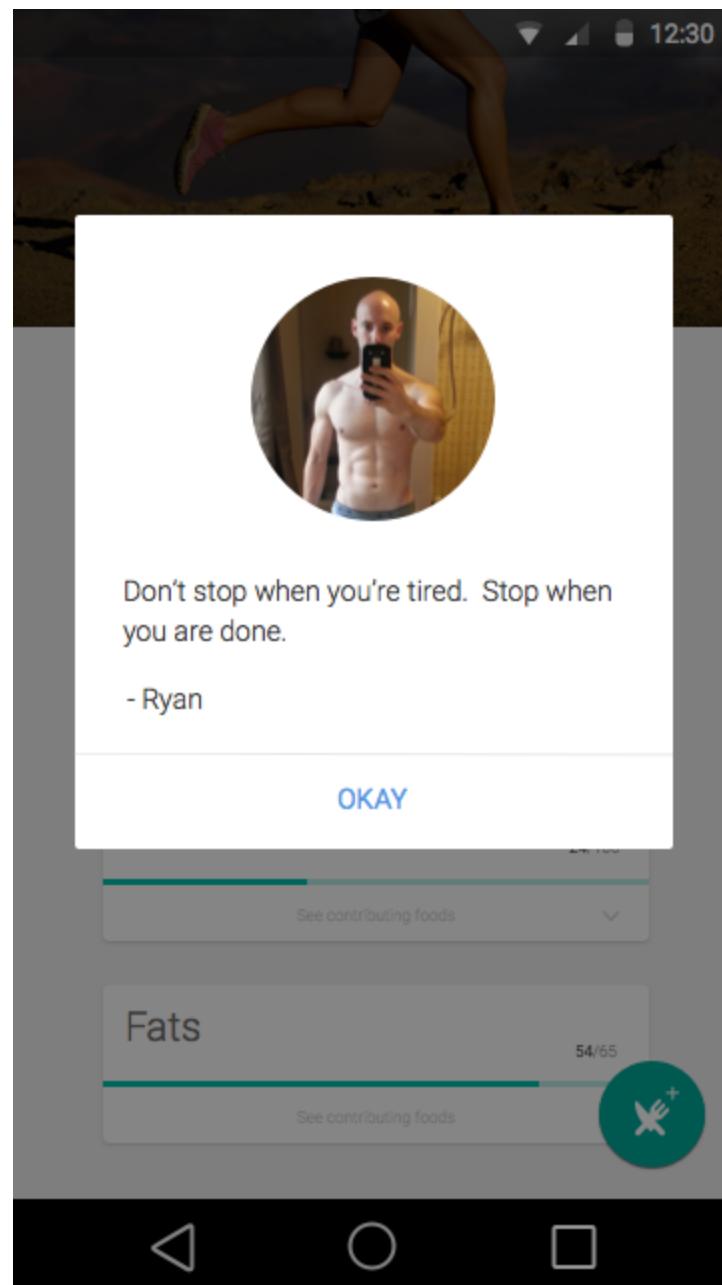


Figure 35: The user is shown a modal pop-up with the image of the motivating user, along with an accompanying message. The user dismisses the modal by tapping the “okay” button and returns to the Summary global view.