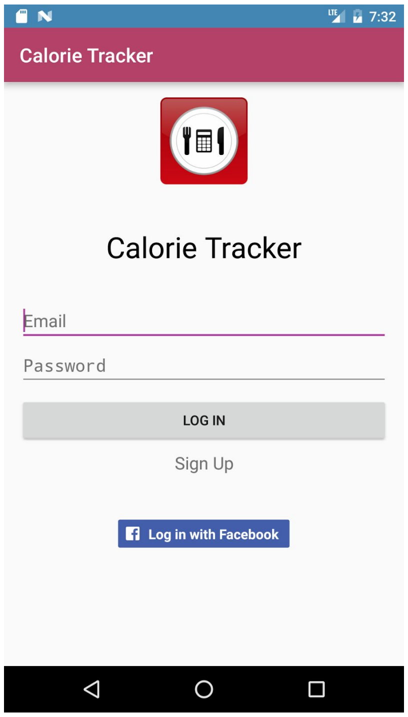
Calorie Tracker
User Manual

CIS 573 Team 4

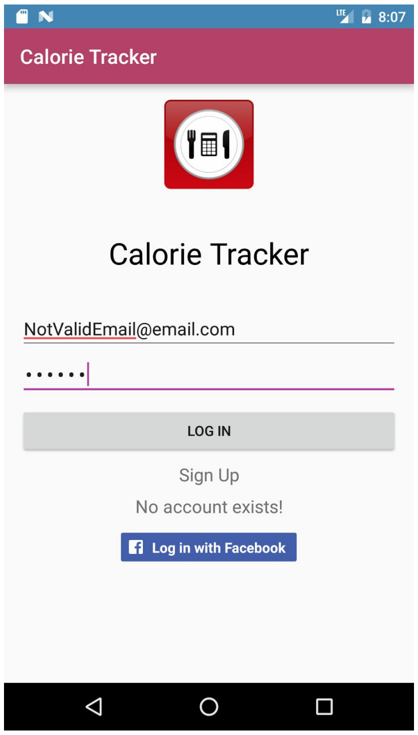
Anqi Chen
Li Huang
Tianxiang Dong
Yi Shang
Lijun Mao

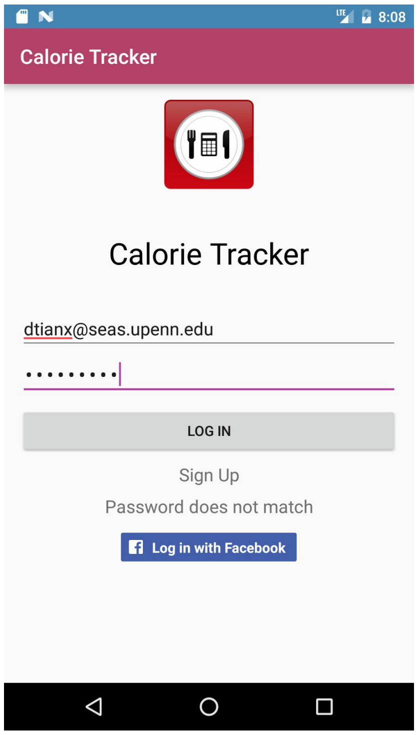
Log In


7:32
Calorie Tracker
Calorie Tracker
Email
Password
LOG IN
Sign Up
Log in with Facebook

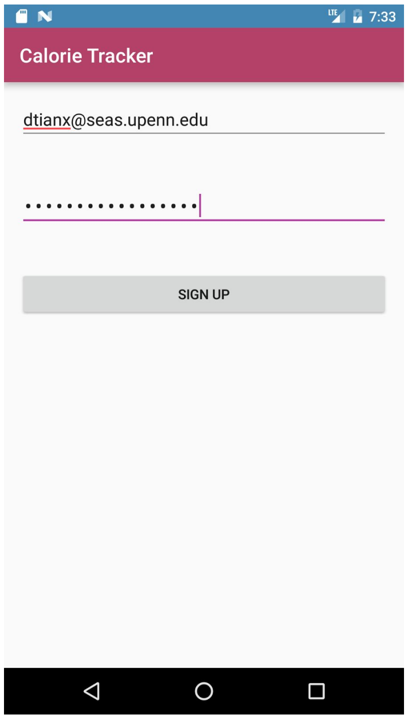
The login page is the default page which Calorie Tracker opens to. An existing user with known email account and password can login by typing those info in following two blank lines. .

The app will prompt an error message for incorrect password or show the user account doesn't exist.

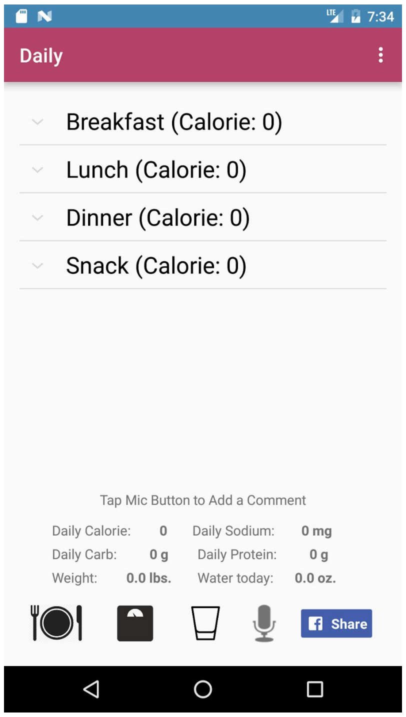

8:07
Calorie Tracker
Calorie Tracker
NotValidEmail@email.com
••••••
LOG IN
Sign Up
No account exists!
Log in with Facebook


8:08
Calorie Tracker
Calorie Tracker
dtianx@seas.upenn.edu
•••••••••
LOG IN
Sign Up
Password does not match
Log in with Facebook

Sign Up



If user doesn’t have an existing account, he can click the “Sign Up” button, which will redirect user to the sign up page. Then the user can register for a new account.



After user has successfully logged in or signed up, The page will redirect to our main view, the daily activity view. The menu to access other screens can be accessed by touching the three dots in the upper right corner.

It contains a daily listing of meals grouped by their type (Breakfast, Lunch, Dinner or Snacks). The groups can be collapsed and expanded by touching the labels. Any individual meal item can be edited by touching that meal.

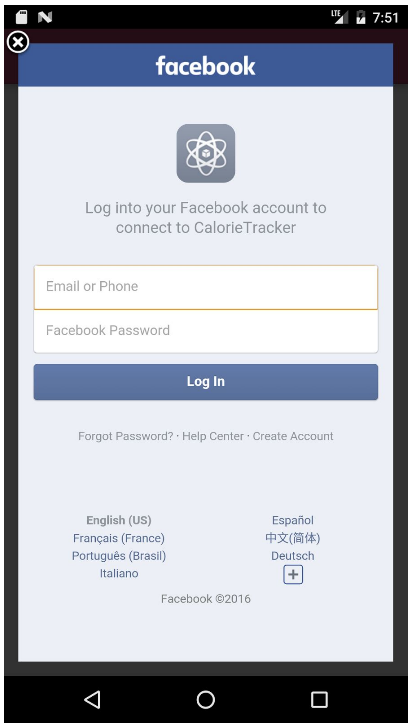
Below the daily summary, the app lists the most recently saved weight and water intake values for today. These values can be edited by clicking on the scale (weight) and glass buttons (water).

Login with Facebook

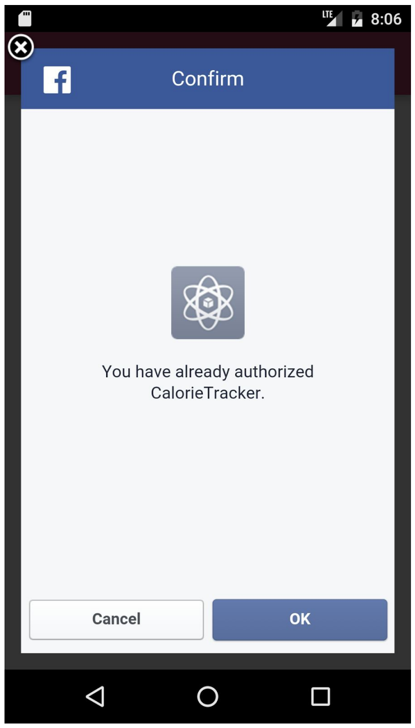
The user can log in to the Calorie Tracker with a facebook account by clicking on the "Log in with Facebook" button.

If username or password is wrong, the user is notified and directed to try again. Once facebook account is verified, a confirm page appears.

After checking "OK" to authorize Calorie Tracker, the user could be taken to the Daily page.



Log into your Facebook account to connect to CalorieTracker

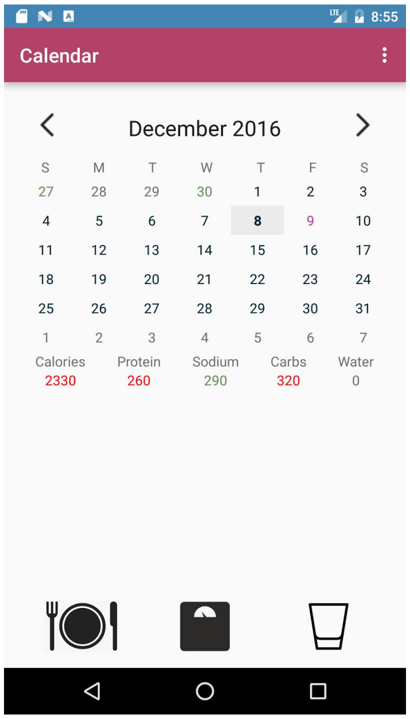


You have already authorized CalorieTracker.

Currently, this app is in development process. According to Facebook regulation, in development process only the designated developer can log in.

Previous Activity

When user clicks on a previous date on the calendar, a new page of that date is opened.



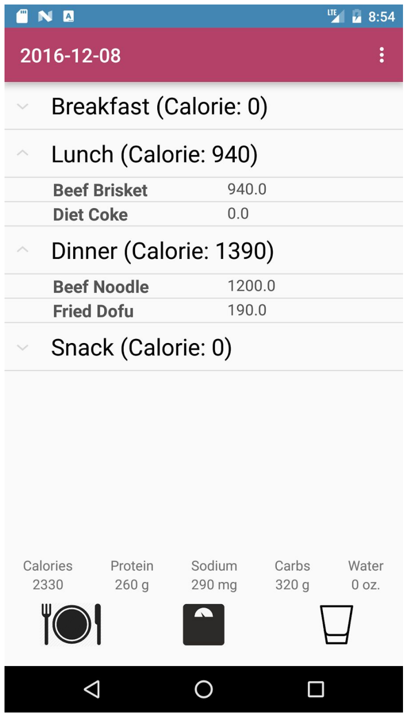
8:55

Calendar

December 2016

S	M	T	W	T	F	S
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Calories	Protein	Sodium	Carbs	Water
2330	260	290	320	0



8:54

2016-12-08

Breakfast (Calorie: 0)

Lunch (Calorie: 940)

Beef Brisket	940.0
Diet Coke	0.0

Dinner (Calorie: 1390)

Beef Noodle	1200.0
Fried Dofu	190.0

Snack (Calorie: 0)

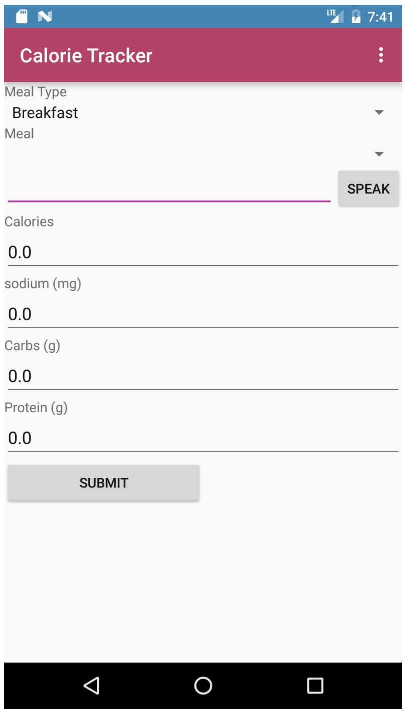
Calories	Protein	Sodium	Carbs	Water
2330	260 g	290 mg	320 g	0 oz.

Similar to the Daily View, the Previous View contains a daily listing of meals grouped by their type (Breakfast, Lunch, Dinner or Snacks). Meals with the same type are collected under the same label, and can be shown by touching the label.

Each meal shown in this page could be edited by clicking on the meal. The user can also add a new meal by clicking on the meal button at the bottom left. Both editing meal and adding meal are done in the Input View which would be introduced in detail later.

Beside the meal button, there are also a scale button to update weight information and a glass button to change water intake.

Frequent Meal and Auto Complete

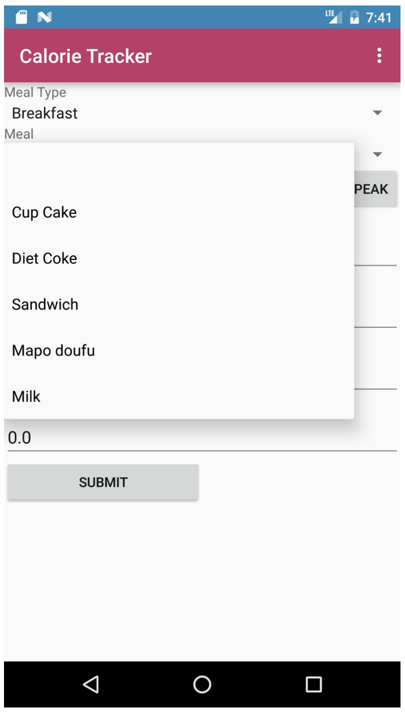


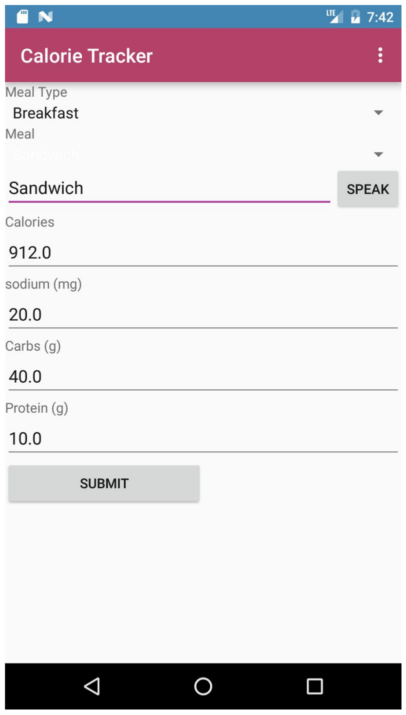
Meal Type
Breakfast
Meal

Calories
0.0
sodium (mg)
0.0
Carbs (g)
0.0
Protein (g)
0.0

In the Input Activity, the app recommends up to five frequent meals that the user previously took. The user can either choose a existed meal from the recommendations in a dropdown spinner, or edit a new meal step by step manually.

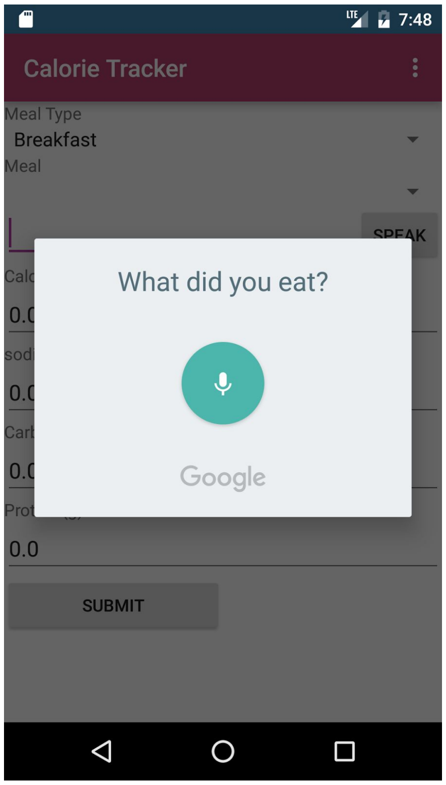
If the user clicks on a frequent meal from the spinner, all the fields for nutrition data are filled in automatically.



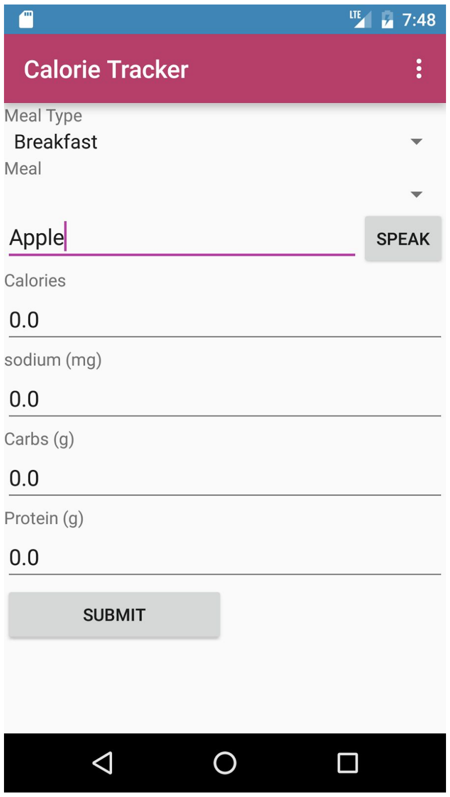


Meal Type
Breakfast
Meal
Sandwich
Calories
912.0
sodium (mg)
20.0
Carbs (g)
40.0
Protein (g)
10.0

Voice Input Meal Name

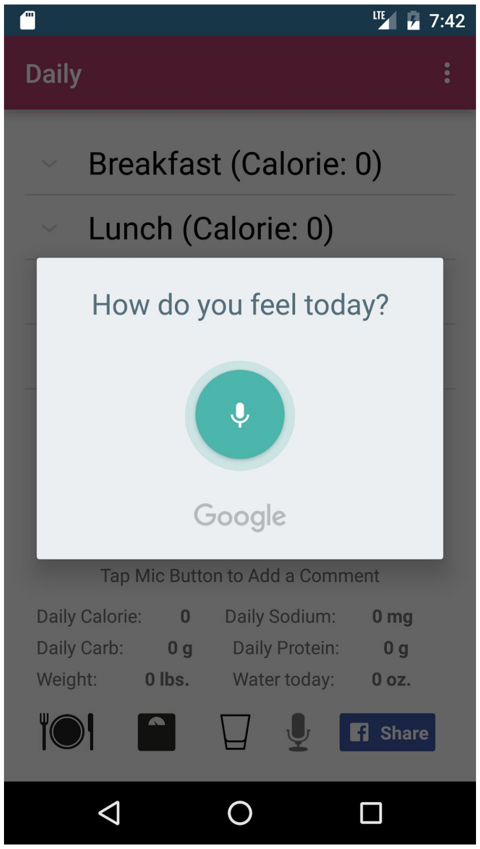


There is a Speak Button beside the input line of meal name. When you tap the button, it asks you ‘What did you eat?’. The app recognizes the user’s speech, and fills the meal name automatically.



e.g. When user says ‘apple’

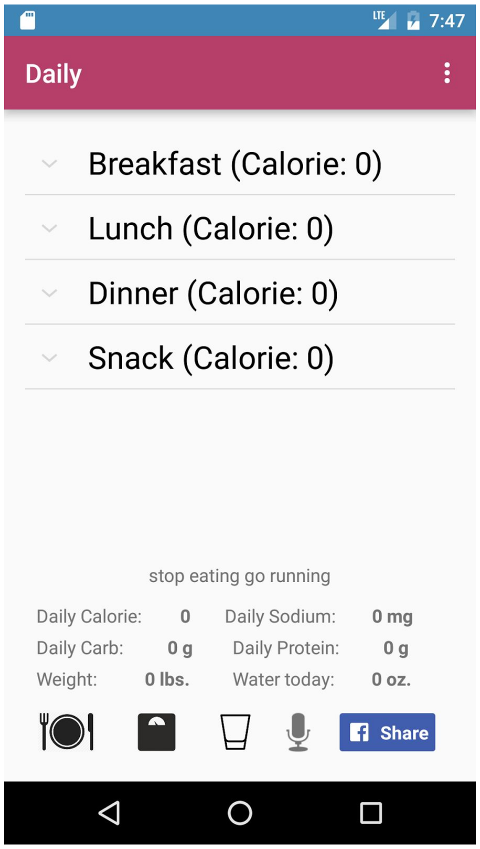
Add Comment



Tap Mic Button to Add a Comment

Daily Calorie:	0	Daily Sodium:	0 mg
Daily Carb:	0 g	Daily Protein:	0 g
Weight:	0 lbs.	Water today:	0 oz.

In daily main page, the user can add a comment. The default comment is ‘Tap Mic Button to Add a Comment’. When tap the mic button, it asks you ‘How do you feel today’. The app recognizes the user’s comment and replace the default text.

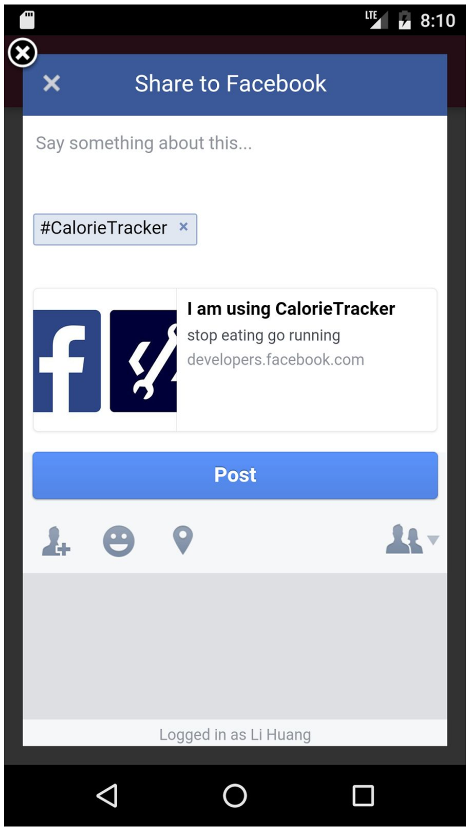


stop eating go running

Daily Calorie:	0	Daily Sodium:	0 mg
Daily Carb:	0 g	Daily Protein:	0 g
Weight:	0 lbs.	Water today:	0 oz.

e.g. When user speaks ‘stop eating go running’

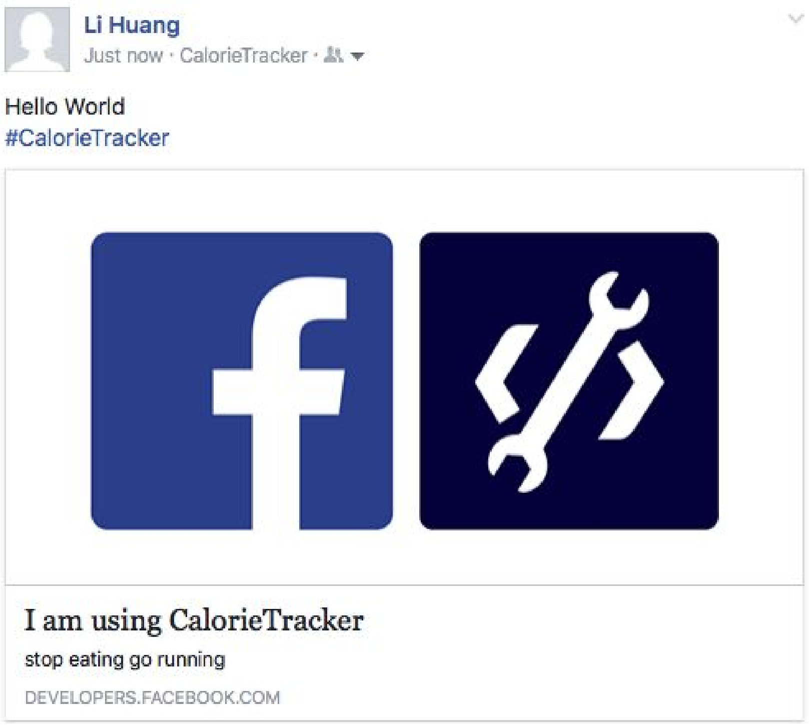
Facebook Sharing



I am using CalorieTracker
stop eating go running
developers.facebook.com

Logged in as Li Huang

In daily main page, there is Facebook sharing button. When tap the button, a Facebook Sharing window will pop up. The sharing content contains an URL link with a title, a description and a hashtag. The title is *'I am using CalorieTracker'* and description is the comment user added (e.g. stop eating go running) and the hashtag is *'#CalorieTracker'*. Tap *'Post'*, the sharing content will be sent to your Facebook timeline.

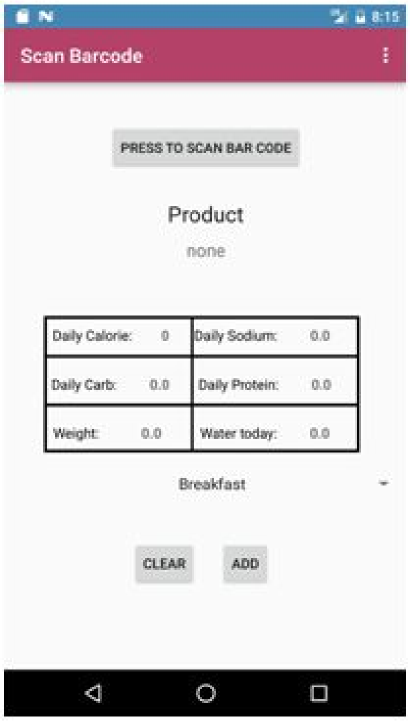


Li Huang
Just now · CalorieTracker ·

Hello World
#CalorieTracker

I am using CalorieTracker
stop eating go running
DEVELOPERS.FACEBOOK.COM

Barcode Scanning Page

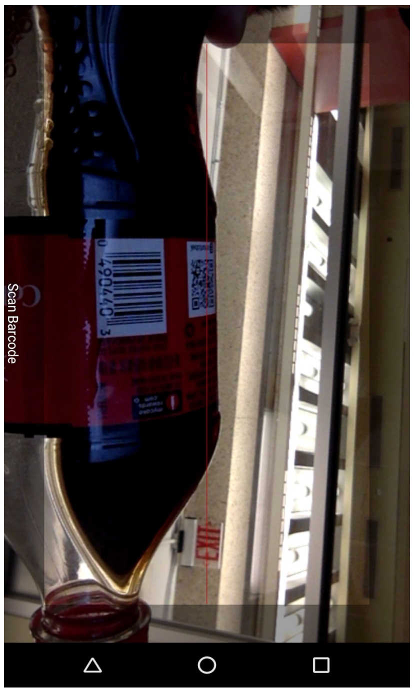


Daily Calorie:	0	Daily Sodium:	0.0
Daily Carb:	0.0	Daily Protein:	0.0
Weight:	0.0	Water today:	0.0

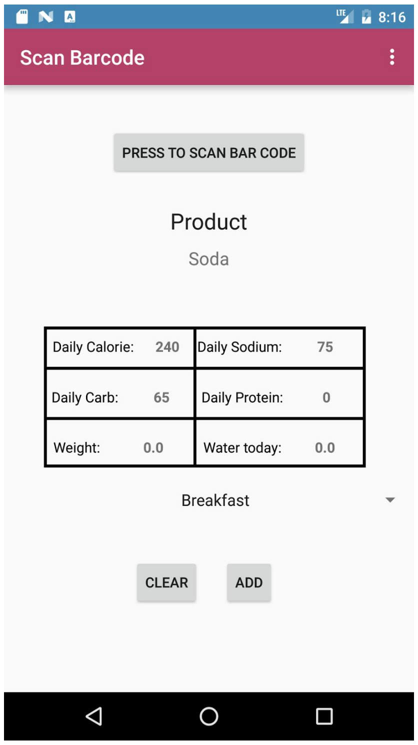
This screen features barcode scanning, which allows user to scan barcode of a product and display all nutrition information in the table shown in the middle of the screen.

The screen consists of a start scanning button, scanned product title, nutrition information table, and clear/add buttons. Users can start scanning by click “PRESS TO SCAN BAR CODE” button. If successfully scanned product with camera, it will return to this page with nutrition information shown in the table. Then user could choose to clear the scanned product or add the product as a meal to the remote database, which will be shown in the daily meal page when user returns the daily page.

To start scanning items, users could click “PRESS TO SCAN BAR CODE” button.



After clicking the button, mobile phone camera will be started. User should match the red line to the center of barcode in order to catch barcode identity. Once extracting the identity number of barcode, it will automatically return the barcode scanning screen.



Product

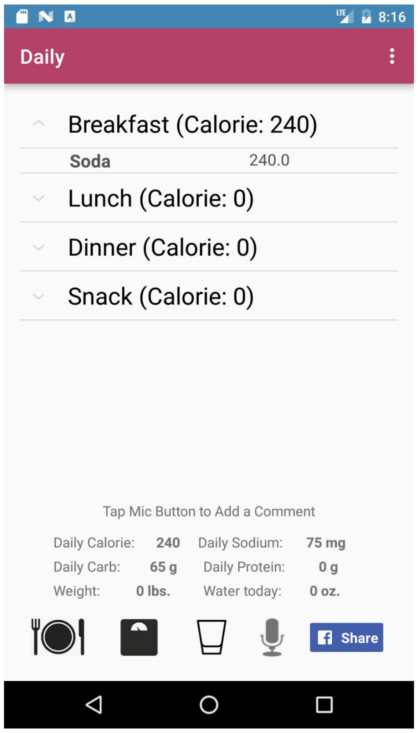
Soda

Daily Calorie:	240	Daily Sodium:	75
Daily Carb:	65	Daily Protein:	0
Weight:	0.0	Water today:	0.0

Breakfast

The screen will display product nutrition information shown in the table as well as the product title. Then user could choose to clear all information or add the product as a meal into the remote database.

If data is cleared, then all information will be erased on this page. If clicking "add" button, we will return the daily page with the newly added meal displayed. Note that, user could choose which type of meal by selecting the spinner before adding to the database.



Breakfast (Calorie: 240)

Soda 240.0

Lunch (Calorie: 0)

Dinner (Calorie: 0)

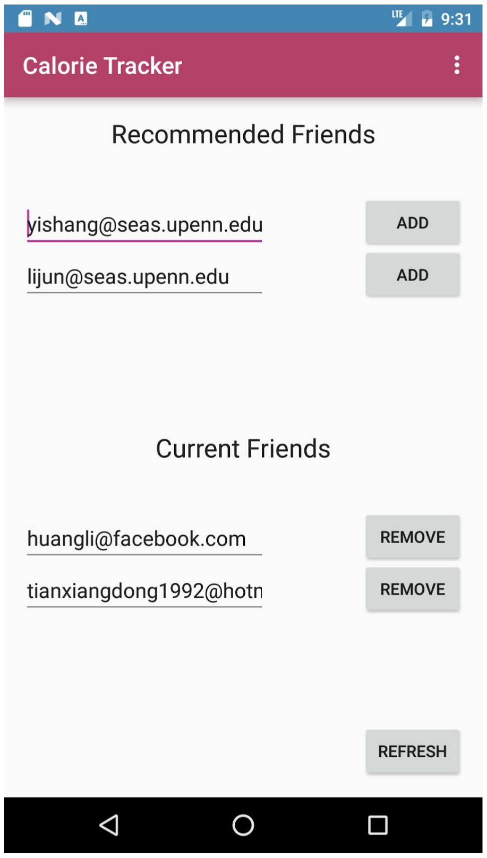
Snack (Calorie: 0)

Tap Mic Button to Add a Comment

Daily Calorie:	240	Daily Sodium:	75 mg
Daily Carb:	65 g	Daily Protein:	0 g
Weight:	0 lbs.	Water today:	0 oz.

This is the updated daily page with new meal "soda" added to breakfast. Users could still change the nutrition information or meal type of it.

Friend and Ranking



Calorie Tracker

Recommended Friends

yishang@seas.upenn.edu ADD

lijun@seas.upenn.edu ADD

Current Friends

huangli@facebook.com REMOVE

tianxiangdong1992@hotn REMOVE

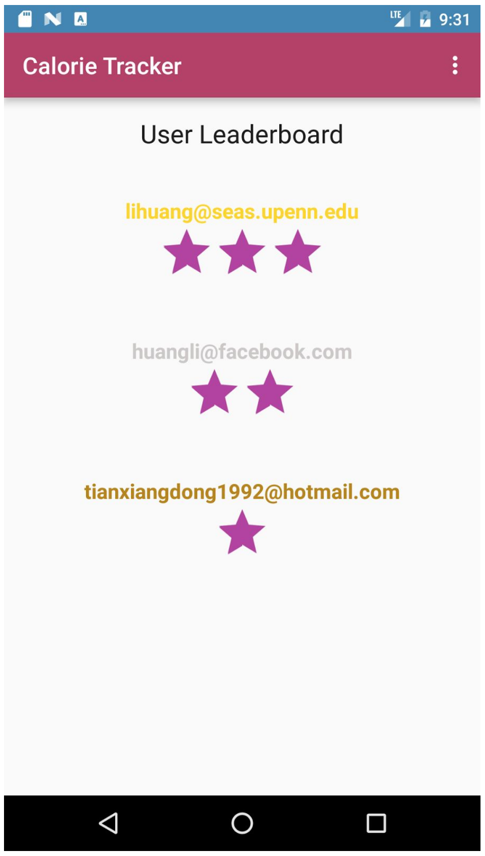
REFRESH

current

User can add new friends in current system via Friend view. They can also delete

friends by clicking "remove" button if they want.

Clicking "refresh" button can refresh the current view, and



Calorie Tracker

User Leaderboard

lihuang@seas.upenn.edu ★★★

huangli@facebook.com ★★

tianxiangdong1992@hotmail.com ★

User can also see the ranking of all his friends in the ranking view, sorted by their difference to target goal values.

First place ranking will get 3 stars in gold, second place get 2 stars in silver, and third place get 1 star in bronze.

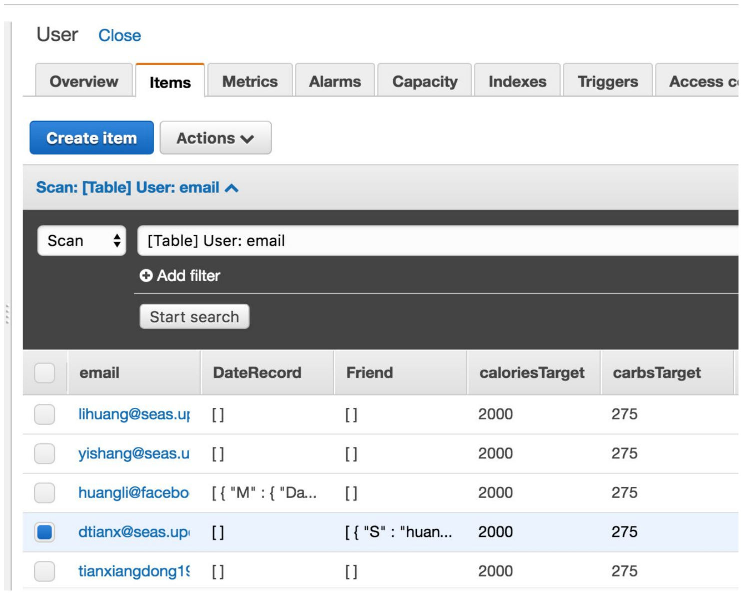
Calorie Tracker
Technical Manual

CIS 573 Team 4

Anqi Chen
Li Huang
Tianxiang Dong
Yi Shang
Lijun Mao

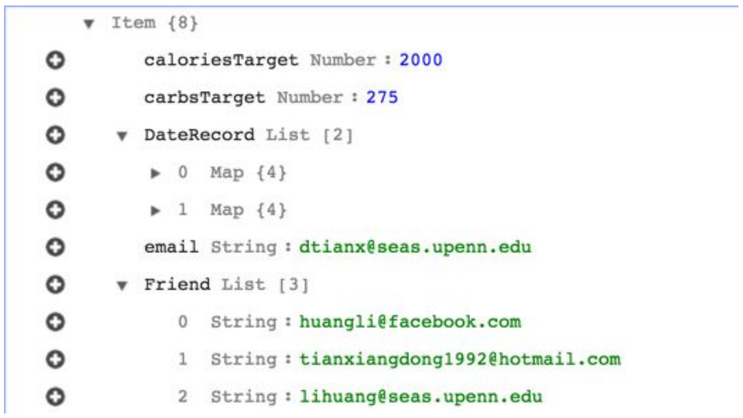
DynamoDB.java (External Database: DynamoDB, Object: User.java)

DynamoDB.java handles data retrieval from external DynamoDB on AWS. All data from user are on DynamoDB, including their personal information and meal records. This class is basically used by all other activities, mostly by DailyActivity and PreviousActivity. The BarcodeActivity inputs the data retrieved from Nutritionix Database into Dynamodb through this class, as well as FriendActivity that enables interaction between different users via this helper class. LoginActivity checks the login data through this class. DynamoDB table is defined in User.java class.



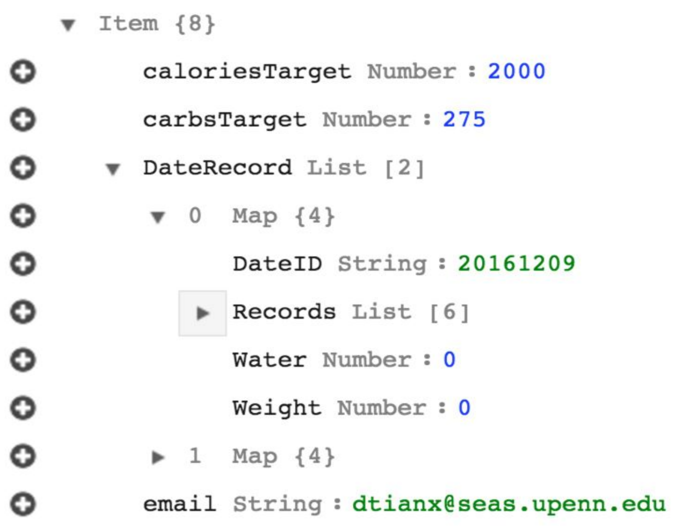
	email	DateRecord	Friend	caloriesTarget	carbsTarget
	lihuang@seas.u	[ ]	[ ]	2000	275
	yishang@seas.u	[ ]	[ ]	2000	275
	huangli@facebo	[{ "M" : { "Da...	[ ]	2000	275
✓	dtianx@seas.up	[ ]	[{ "S" : "huan...	2000	275
	tianxiangdong1	[ ]	[ ]	2000	275

As shown in the screenshot, the dynamoDB table looks like it. There are a list of user objects shown in the database.



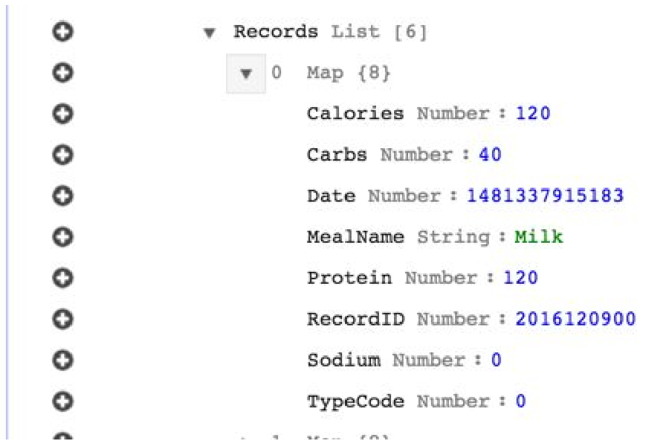
```
▼ Item {8}
    caloriesTarget Number : 2000
    carbsTarget Number : 275
  ▼ DateRecord List [2]
     ▶ 0  Map {4}
     ▶ 1  Map {4}
    email String : dtianx@seas.upenn.edu
  ▼ Friend List [3]
       0  String : huangli@facebook.com
       1  String : tianxiangdong1992@hotmail.com
       2  String : lihuang@seas.upenn.edu
```

The Object of one User includes user’s basic information like the primary key email, password, daily targets, as well as the friends list and the Date Record list.



```
▼ Item {8}
    caloriesTarget Number : 2000
    carbsTarget Number : 275
  ▼ DateRecord List [2]
     ▼ 0  Map {4}
            DateID String : 20161209
          ▶ Records List [6]
            Water Number : 0
            Weight Number : 0
     ▶ 1  Map {4}
    email String : dtianx@seas.upenn.edu
```

The Date Record list includes information from all single date, as well as some basic information about the date such as date ID, daily input water and daily weight of that user. The specific meal information are stored as Record object in Records list.

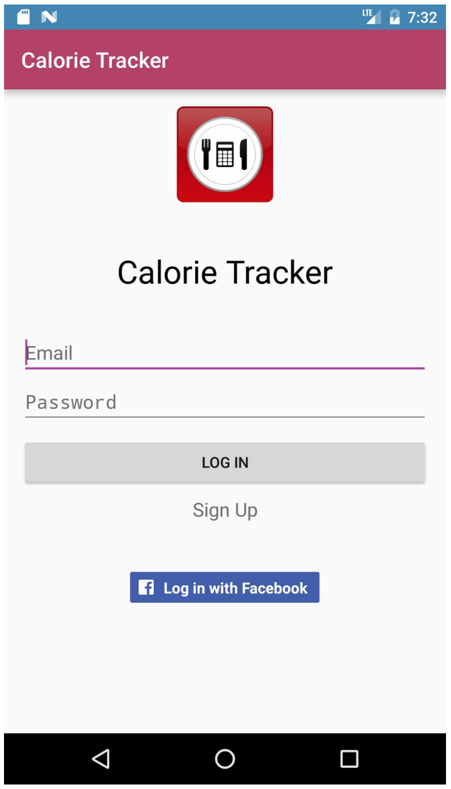


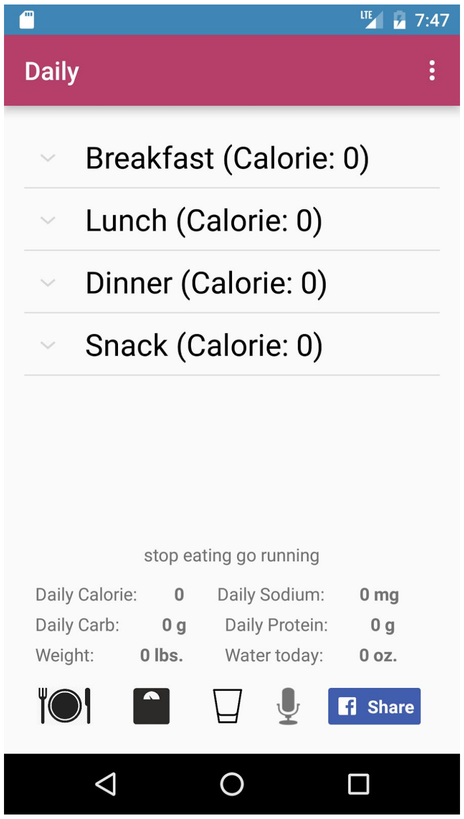
```
▼ Records List [6]
  ▼ 0  Map {8}
        Calories Number : 120
        Carbs Number : 40
        Date Number : 1481337915183
        MealName String : Milk
        Protein Number : 120
        RecordID Number : 2016120900
        Sodium Number : 0
        TypeCode Number : 0
```

The Record object stores detailed information about a single meal, which basically will be shown in either daily activity or previous activity depending on current date and the date of this record.

DatabaseHandler.java

Local database SQLite handler. Basically from previous version. It's used in Calendar Activity to set up the calendar. Most of its methods are deprecated and replaced by external database methods in DynamoDB.java. This class is preserved for potential usage in future version. Or special needs that will target to local version app.

 7:32 Calorie Tracker Calorie Tracker Email Password LOG IN Sign Up Log in with Facebook	**Login View** Activity Class: LoginActivity.java XML Layout: activity_login.xml Class Relations: ● (Extends) CalorieTrackerActivity.java ● (Uses) DynamoDB.java This is the login page where user first opens up the app. CalorieTrackerActivity.java is the parent class for most classes in this app. DynamoDB.java is the database handler class, also used for most classes.



Daily Calorie: 0 Daily Sodium: 0 mg
Daily Carb: 0 g Daily Protein: 0 g
Weight: 0 lbs. Water today: 0 oz.

Daily View

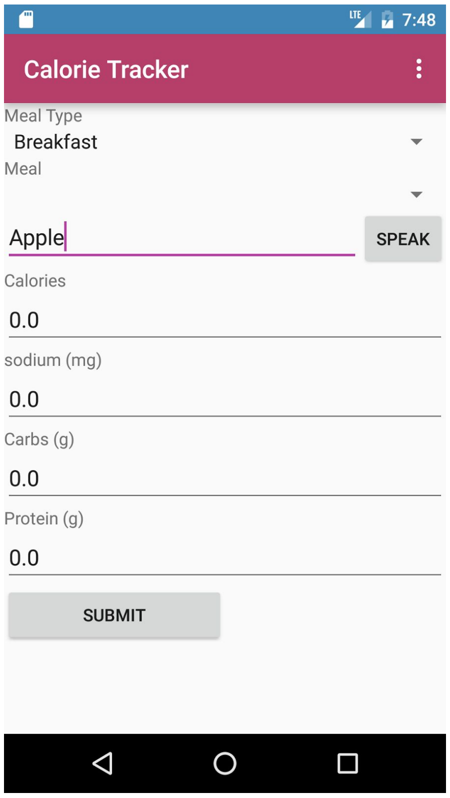
Activity Class: DailyActivity.java
XML Layout: activity_daily.xml

Class Relations:

- (Extends) CalorieTrackerActivity.java
- (Uses) Meal.java
- (Uses) DynamoDB.java
- (Uses) BottomMenu.java
- (Uses) User.java

Daily view lists the diet information for today.

CalorieTrackerActivity is the basic class that most activity extends. It shows a menu on the top right corner and directs the user to different views.
A meal object contains meal name, meal type, date, calories, protein, carbs and sodium for a particular meal. BottomMenu class sets button handlers for water button and weight botton.



Meal Type: Breakfast
Meal
Apple
Calories: 0.0
sodium (mg): 0.0
Carbs (g): 0.0
Protein (g): 0.0

Input Page

Activity Class: InputActivity.java
Xml Layout: activity_input.xml

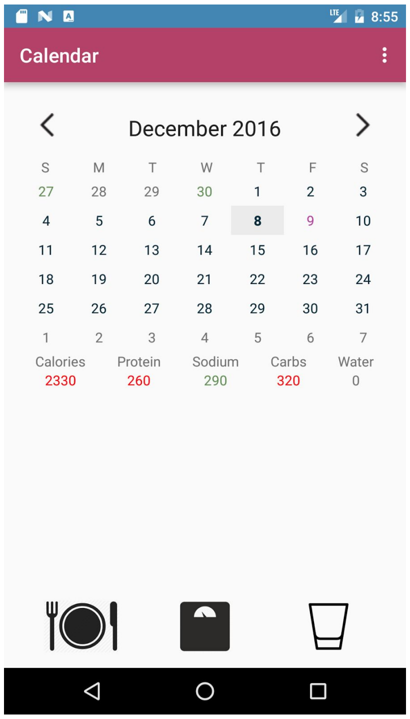
Class Relations:

- (Extends) CalorieTrackerActivity.java
- (Uses) Meal.java
- (Uses) DynamoDB.java

This class enables user to add a meal to a particular day. Meals could be chosen from a frequent meals list, or recognized by the 'SPEAK' button.

For the speech recognizer parts(input page and daily view page), some older versions of AVDs do not support speech input. If AVD not supporting speech input, ActivityNotFoundException is handled and a Toast text will pop up.

This not-supporting issue could be solved by downloading a Google app (to access Google Speech API) in AVD or physical cell phone.



8:55

Calendar

December 2016

S	M	T	W	T	F	S
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Calories	Protein	Sodium	Carbs	Water
2330	260	290	320	0

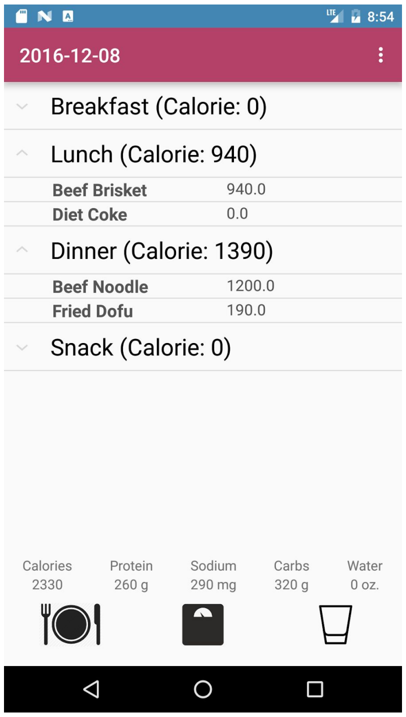
Calendar View

Activity Class: CalendarActivity.java
XML Layout: activity_calendar.xml

Class Relations:

- (Extends) CalorieTrackerActivity.java
- (Uses) Meal.java
- (Uses) DynamoDB.java
- (Uses) BottomMenu.java
- (Uses) User.java

This class displays dates of the current month.



8:54

2016-12-08

Breakfast (Calorie: 0)

Lunch (Calorie: 940)

Beef Brisket	940.0
Diet Coke	0.0

Dinner (Calorie: 1390)

Beef Noodle	1200.0
Fried Dofu	190.0

Snack (Calorie: 0)

Calories	Protein	Sodium	Carbs	Water
2330	260 g	290 mg	320 g	0 oz.

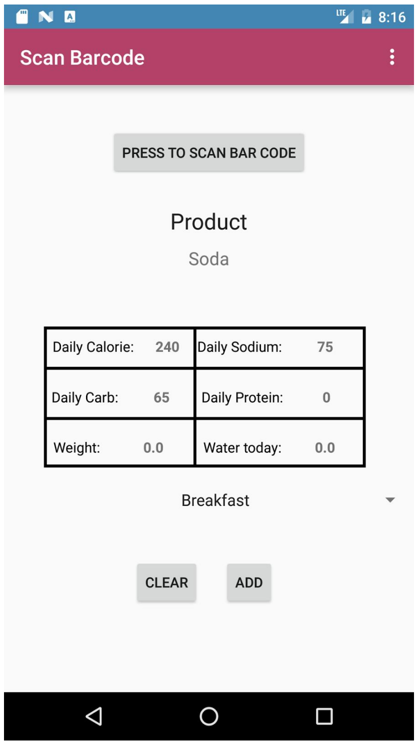
Previous View

Activity Class: PreviousActivity.java
Xml Layout: activity_previous.xml

Class Relations:

- (Extends) CalorieTrackerActivity.java
- (Uses) Meal.java
- (Uses) DynamoDB.java
- (Uses) BottomMenu.java
- (Uses) User.java

This view lists the diet information for a previous day.



Product

Soda

Daily Calorie:	240	Daily Sodium:	75
Daily Carb:	65	Daily Protein:	0
Weight:	0.0	Water today:	0.0

Breakfast

Barcode Scanning

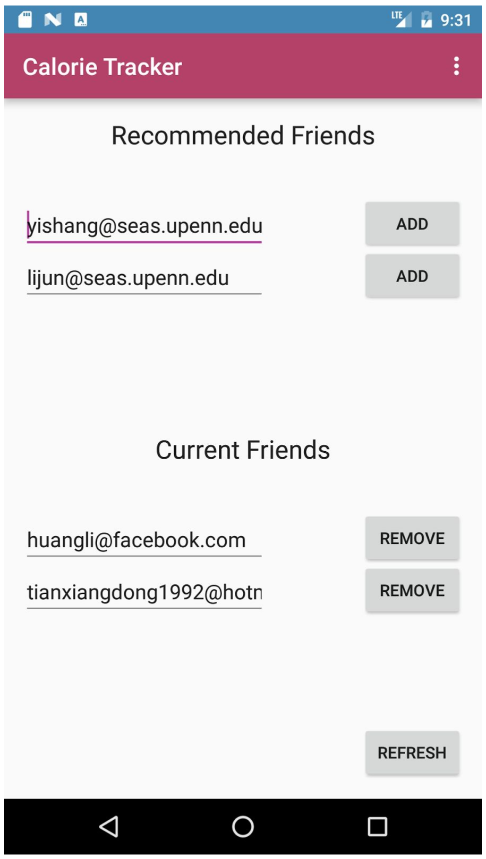
Activity Class: BarcodeActivity.java
XML Layout: activity_barcode.xml

Class Relations:
- (Extends) CalorieTrackerActivity.java
- (Uses) DataBaseHandler.java
- (Uses) Meal.java

This class allows the user to scan barcode of product, showing nutrition information of the scanned product and add the product as a meal of specific type to the remote Dynamo database which would be displayed on Daily page.

BarcodeActivity.java

BarcodeActivity.java is an entrance for barcode scanning using mobile phone camera as well as displaying the scanned product nutrition information. This class extends CalorieTrackerActivity, and uses Meal and DataBaseHandler in order to interact with database. In order to scan the barcode, I use ZXing library to configure the camera and extract barcode identity with its image processing functionality. Then I use the extracted barcode identity to send request to Nutritionix API and receive JSON response. Note that I define an asynchronous task in BarcodeActivity such that sending request to Nutritionix will be done in background thread, and user can still operate freely in user interface without disturbance. After receiving JSON response, we can parse it to extract desired information and fill them into the table. If user would like to add the product as a meal to database, we would create a Meal object and use Database handler to add the meal to database. The newly added meal would be shown in daily page with scanned information, and users are allowed to edit the meal.



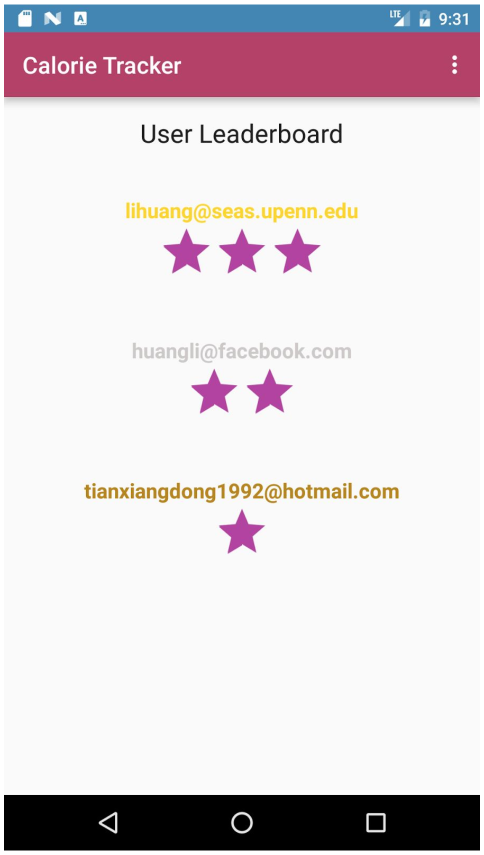
Friend Page

Activity Class: FriendActivity.java
Xml Layout: activity_friend.xml

Class Relations:

- (Extends) CalorieTrackerActivity.java
- (Uses) DynamoDB.java

This view allows user to add and remove friends.



Ranking Page

Activity Class: RankingActivity.java
Xml Layout: activity_ranking.xml

Class Relations:

- (Extends) CalorieTrackerActivity.java
- (Uses) DynamoDB.java

This class shows 3 stars ranking between all friends of current user

Credentials

The external system involved are *AWS DynamoDB, Cognito, IAM*, the credentials used to get access to our app is stored in *Amazon Cognito*. Since it might expire some time, so future users might have to set up their own AWS environment to get this app working normally.