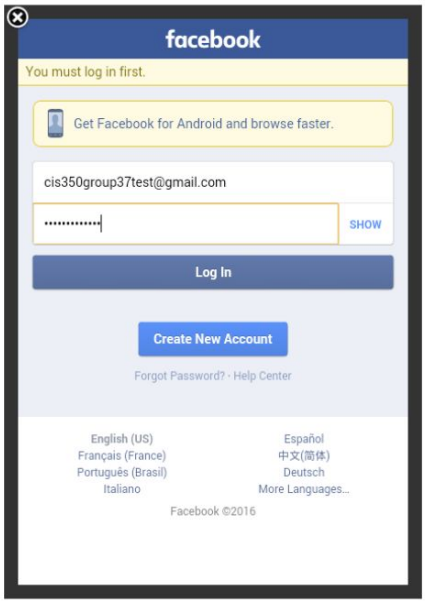
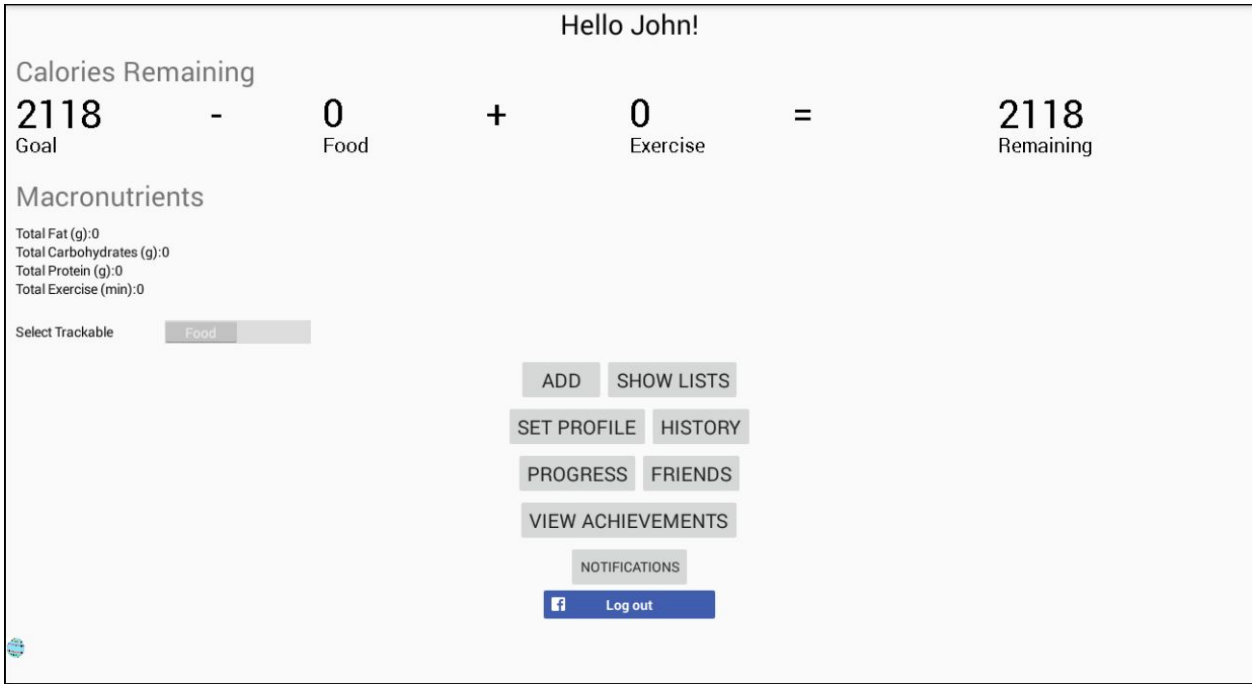
User Manual

Login Page

Opening the app takes the user to the login page. From here, the user can log in with his/her Facebook account. If the Facebook app is installed, the user will be taken directly to the main screen with the credentials of whoever is currently logged into the Facebook app. Otherwise, the login page pops up in which the user can provide his/her credentials to be taken to the main page.



Main Screen



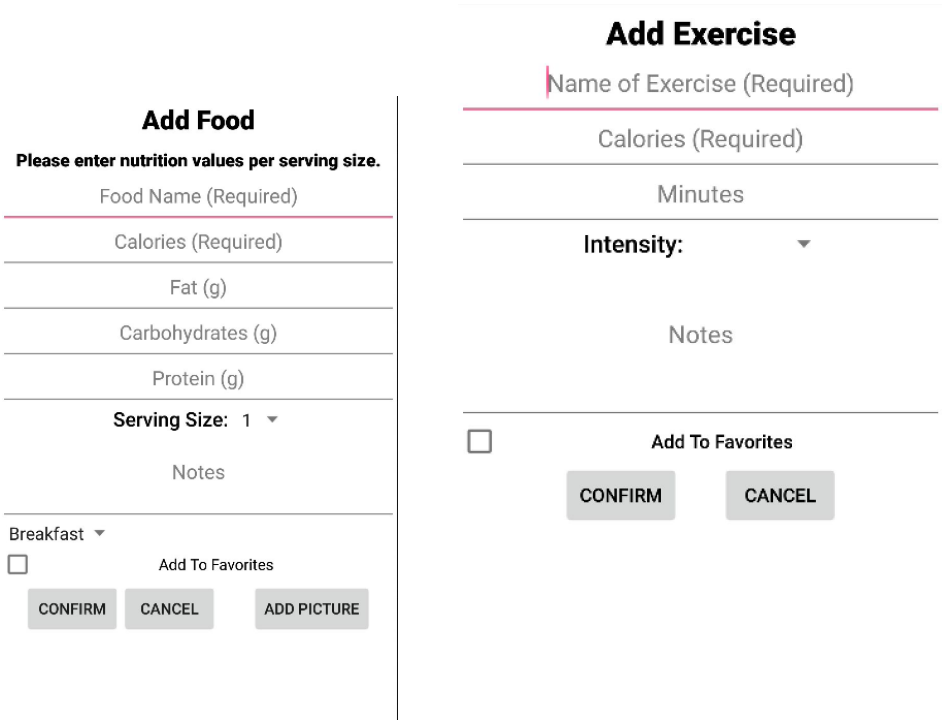
On the main screen, the user will see a welcome message that says “Hi ____!”. Underneath, the user will see today’s total caloric values: The goal for today MINUS the total calories consumed from food PLUS the total calories burned from exercise EQUALS remaining calories for today. Beneath the calorie equation are texts that show the daily consumption of macronutrients (fat, carbohydrate, protein), as well as the total number of minutes spent exercising.

Underneath are the app’s navigation buttons. “Add” leads to the “Add Trackable” page. “Show Lists” navigates to the “Show Lists” page. “Set Profile” leads to the Profile page, where users can update their profile. “History” leads to the “History” page. “Progress” leads to the Progress page. “Friends” leads the user to the Friends page. “View Achievements” brings the user to the Achievements page. Finally, at the bottom, the user can log out of Facebook by the “Log out” button, and can go to the Notifications page to set personalized notifications using the “Notifications” button.

Add Trackable

The “Add Trackable” page allows the user to input nutritional and essential values for food and exercise, as well as add any notes. The page consists of editable text fields that allow the user to input name of food/exercise, calorie content, and add notes. The exercise page allows for adding minutes and a dropdown menu for intensity of the workout. The food page allows the user to add macromolecules (i.e. fats, carbohydrates, and proteins), and employs dropdown menus to select serving size and for which meal this food was consumed. The app also encourages the user to snap a picture of a food label and add the nutrients to the app later. There is an area at the bottom where the user can “Add Picture”, which will take the user to his or her image gallery. There, the user can select the image of the food label where it will be placed on the same screen as the input fields for the user to directly reference as he or she is

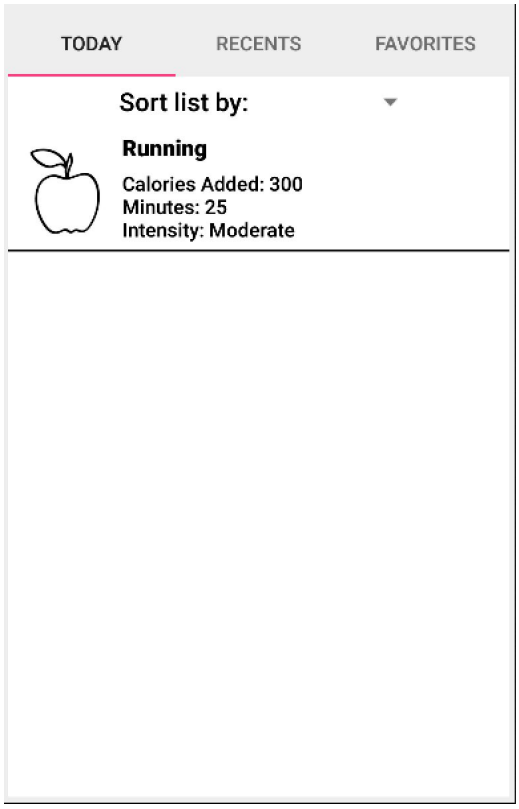
adding the values. For both exercise and food, there is a checkbox allowing the user to add the food/exercise to his or her list of favorites. When the user hits “Enter” then the trackable will be added to the list of foods/exercises consumed on that day, and the total caloric goal will be updated to show the newest input.



“Show Lists”

The “Show Lists” button opens a page with three tabs. For both exercise and for food, the tabs show “Today”, “Recents”, and “Favorites”. The Today tab shows all the food/exercise that was entered that day. The Recent tab shows the ten most recent entries to either list, and the Favorite tab shows all the foods and exercises that were checked as favorites when adding to the app. For all three lists, the user can click directly on one of the items and it will prompt the user to either “Add Food/Exercise” or “Cancel”. If Add Food/Exercise is clicked, it will take the user to the “Add Trackable” page and will automatically input the values for that given trackable into the corresponding editable text fields. Foods/Exercises in the Today and Favorites lists can also be long-clicked, showing a menu to “Edit”, “Delete”, and “Cancel” (which exits the menu). If Edit is clicked, the user will again be taken to “Add Trackable” where the same effect occurs as adding a pre-loaded food; only this time, when “Enter” is clicked, the trackable in that corresponding list will be updated, rather than adding another item to that list. The “Delete” button simply deletes that trackable from either the Today or Favorite list. If it was deleted from the Today list, the total caloric consumption and goal on the main page will be updated. Each list can also be sorted to make finding a Trackable easier. Each food list can be sorted by

Name, Calorie input (low to high), and by the type of meal it is. For exercise, lists can be sorted by Name and Calorie.

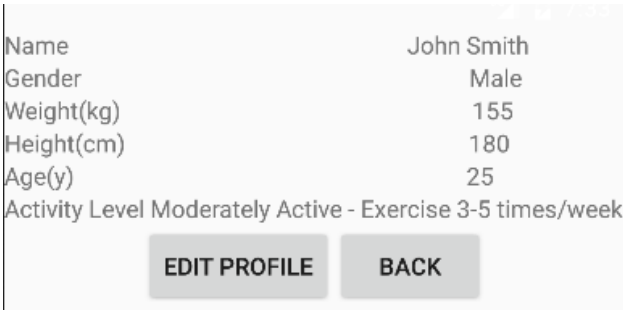


Profile Page

The Profile page can be accessed from the main screen using the highlighted “Set Profile” button below. The Profile Page contains the user’s name, gender, weight, height, age, and activity level information used to calculate the proper daily caloric intake. The user can also edit this information from the profile page.

Once the user is on the profile page, he will see the below information and the Edit Profile and Back buttons. Clicking the Back button will return the user to the main screen. The Edit Profile

button will bring the user to a form to input new profile information, shown below. The weight must be in kilograms, and the height must be in centimeters.

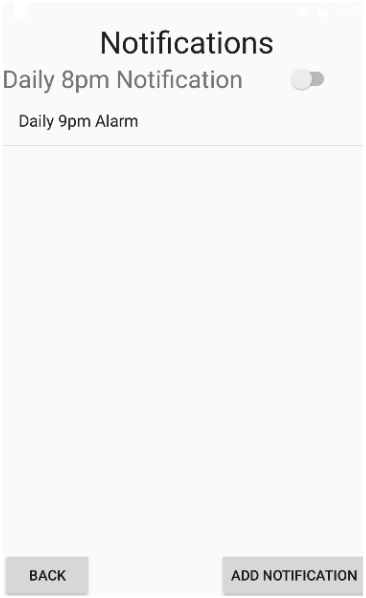


Notifications Page

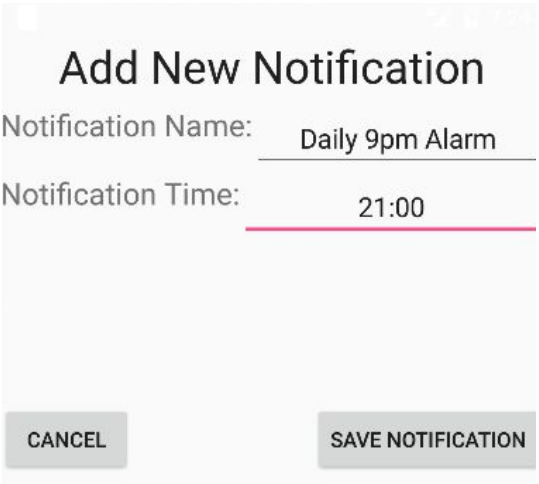
The user can access the notifications page from the “Notifications” button on the main screen highlighted below. The Notifications Page is used to set daily alarms that will send a notification to your android device at the designated time.

This is the Notification Page. The page contains a pre-programmed daily 8pm notification alarm that the user can control using the switch. Below the pre-programmed alarm, the user will see a list of all alarms that are currently active. In this case there is a Daily 9pm alarm. The user can also remove alarms by clicking and holding on the alarm they wish to remove from the list of

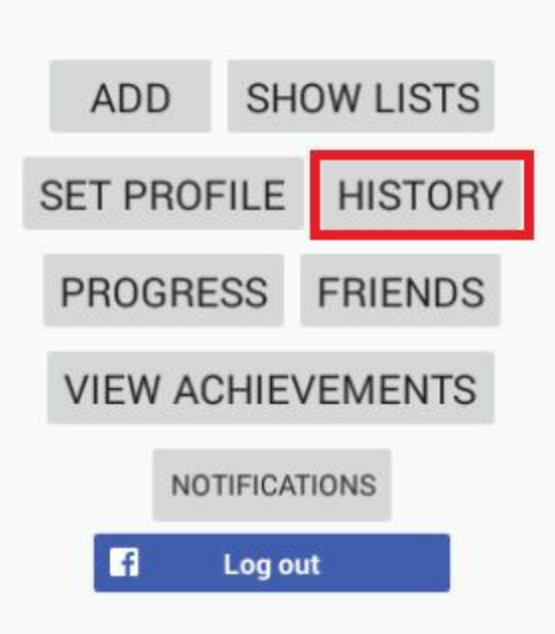
user created alarms. The user can add new alarms using the “Add Notification” button at the bottom right of the screen.



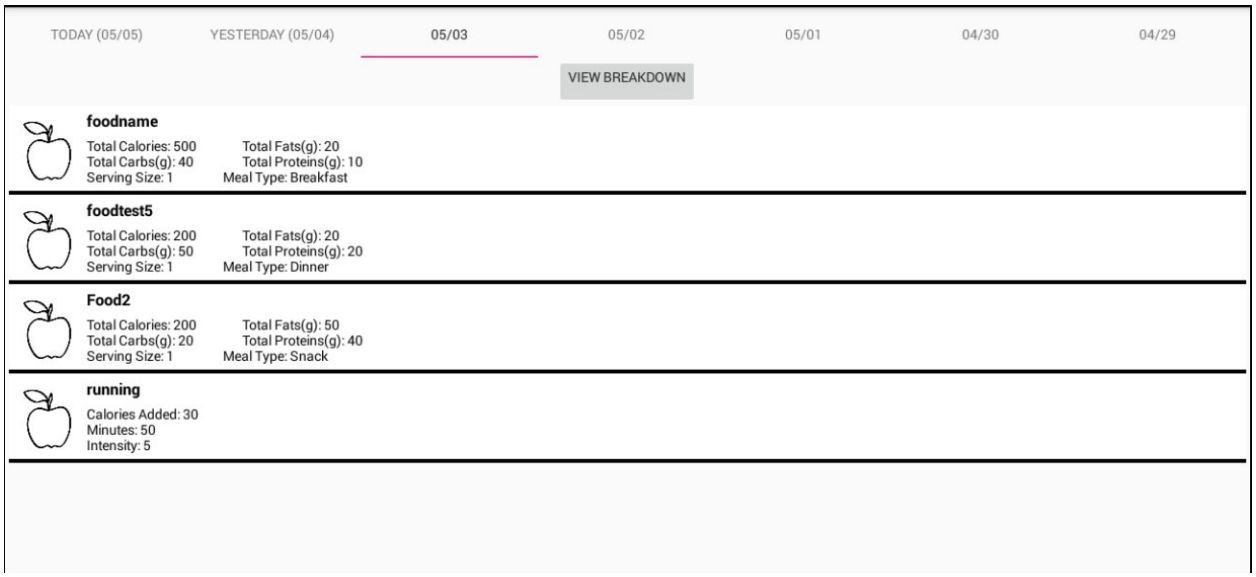
This is the Add Notification Page that will appear when the button is selected. The Add Notification requires a name, that will show up on the list of alarms on the main notification page, and a notification time, the daily time at which a notification with the name message will be sent. The Notification Name should be descriptive so that the user knows what the alarm is for. The Notification Time must also be of the form HH:MM. Once the user finishes inputting the information they should click the save notification page. The cancel button return to and not update the main notifications page.



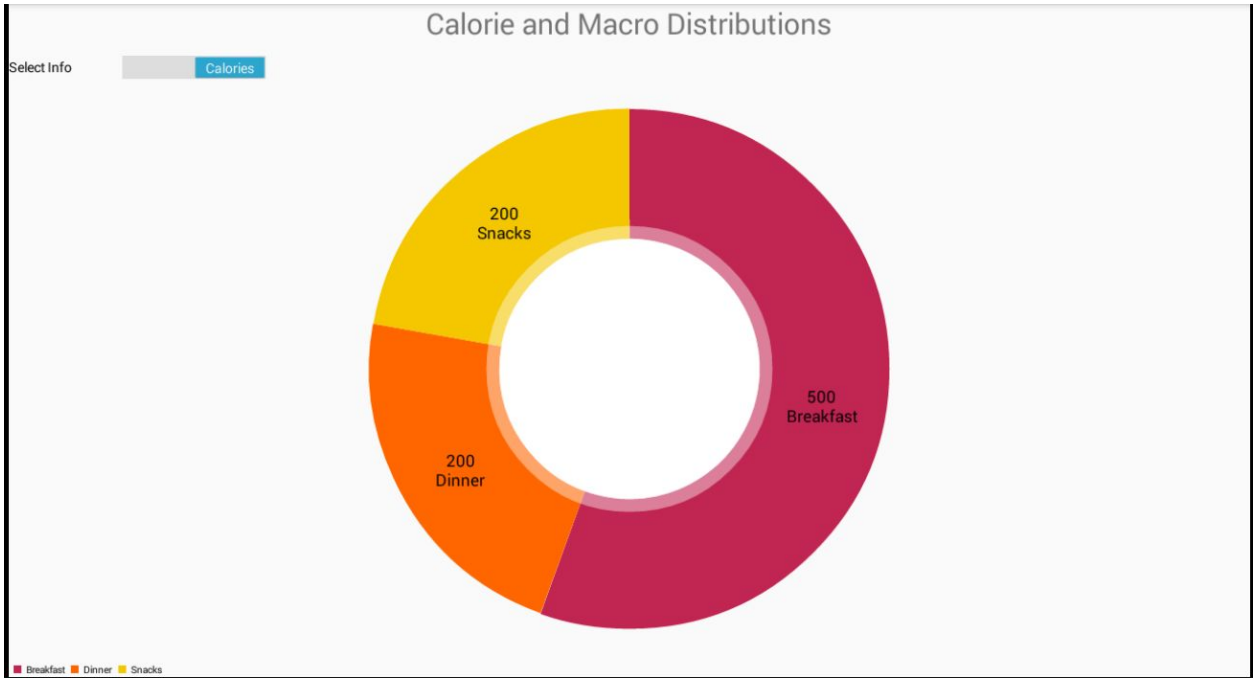
History

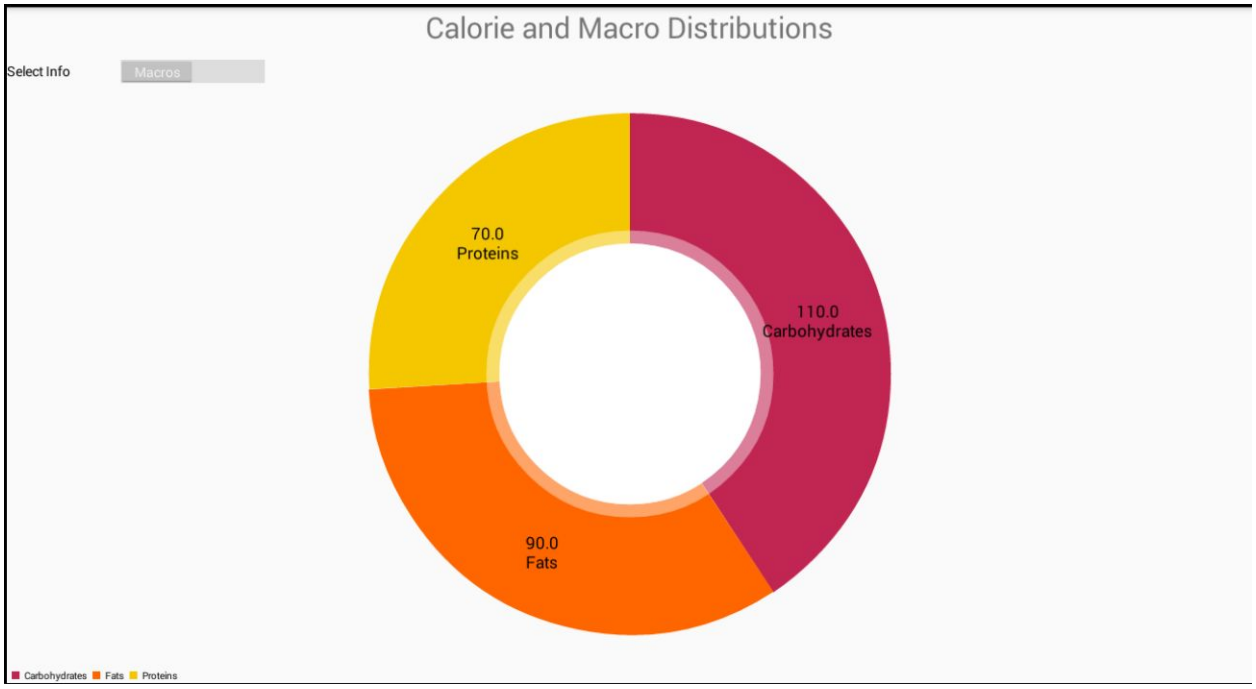


The History page can be accessed by the history button on the main screen. The history page's purpose is to show the user's caloric intake, food items, exercise items in the last 7 days.



Here, we can see the page. The tabs show the different days. Clicking on the day will show the food items and the exercise items for that day, along with their details. Clicking on the list will give the user the option to see the note for that food/exercise item as well as the option to add the food/exercise to today's list, as shown in the lists above. However, unlike the today/recent/favorites list, there is no way to edit the list since it reflects the user's history.

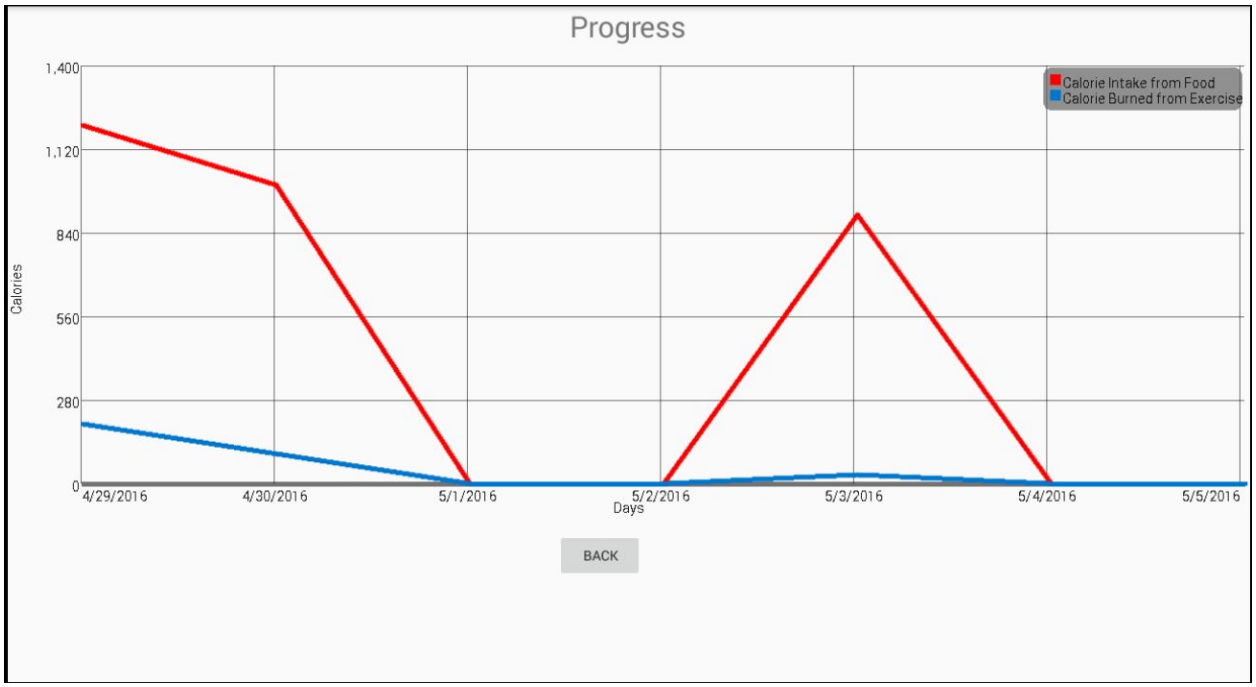




Clicking on the “View breakdown” button leads to a pie chart breakdown of the caloric (above) or macronutrient (below) intakes for that day. If there are no recorded meals for the day, the screen will not show any chart (not pictured).

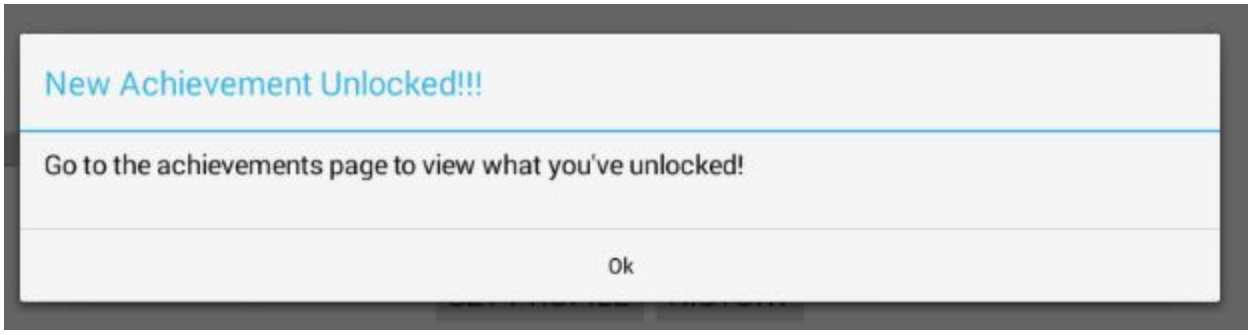
Progress

The Progress page can be accessed by clicking on the progress button. The purpose of the progress page is to graphically display the user's progress in their calorie tracking in the last week. It shows the caloric intake from food and the calories burned from exercise in the last 7 days.

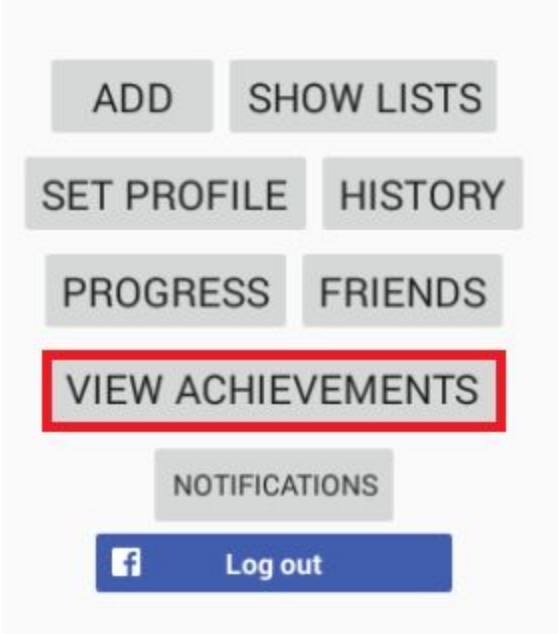


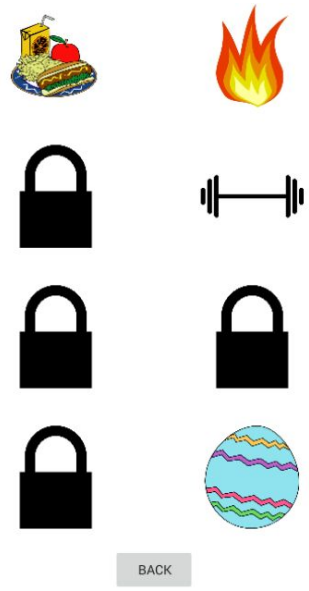
As shown above, the red line displays the caloric intake from food. The blue line shows the calories burned from exercise. The x-axis shows the days, in which the farthest date on the right represents today. The Y-axis shows the calories. Pressing "Back" takes the user to the main screen.

Achievements

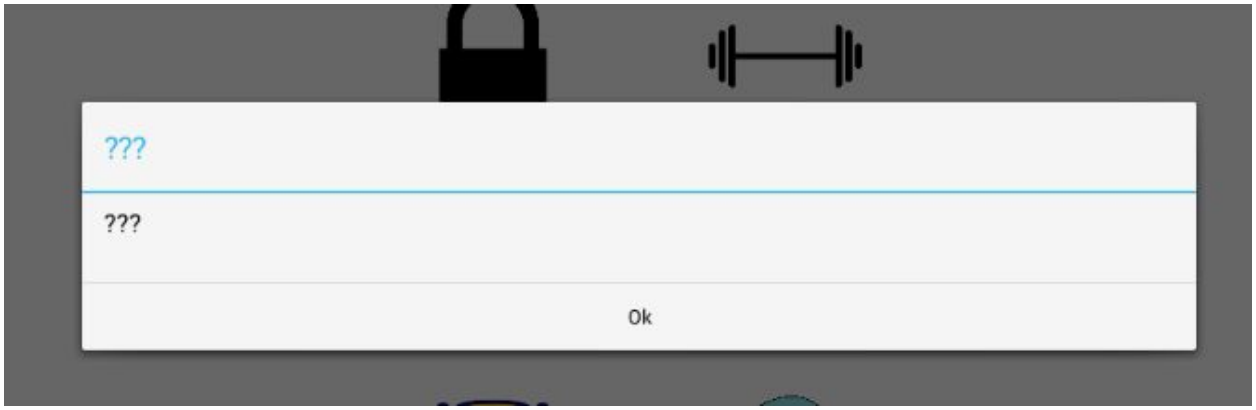


During the usage of this app, the user can accumulate and unlock different achievements. Above is an example of a notification that pops up if the user does something to unlock an achievement. Clicking the “View Achievements” button will take the user to the achievements page.

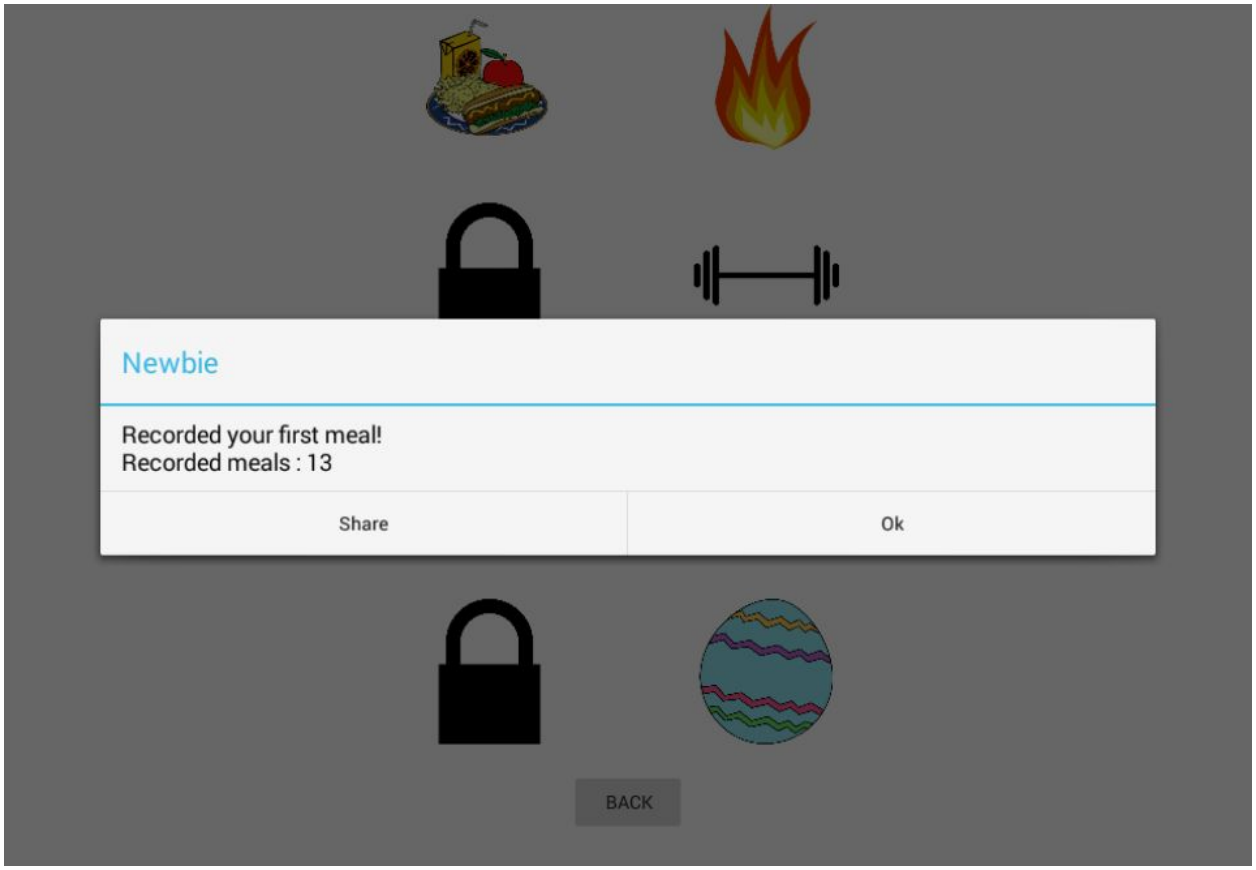




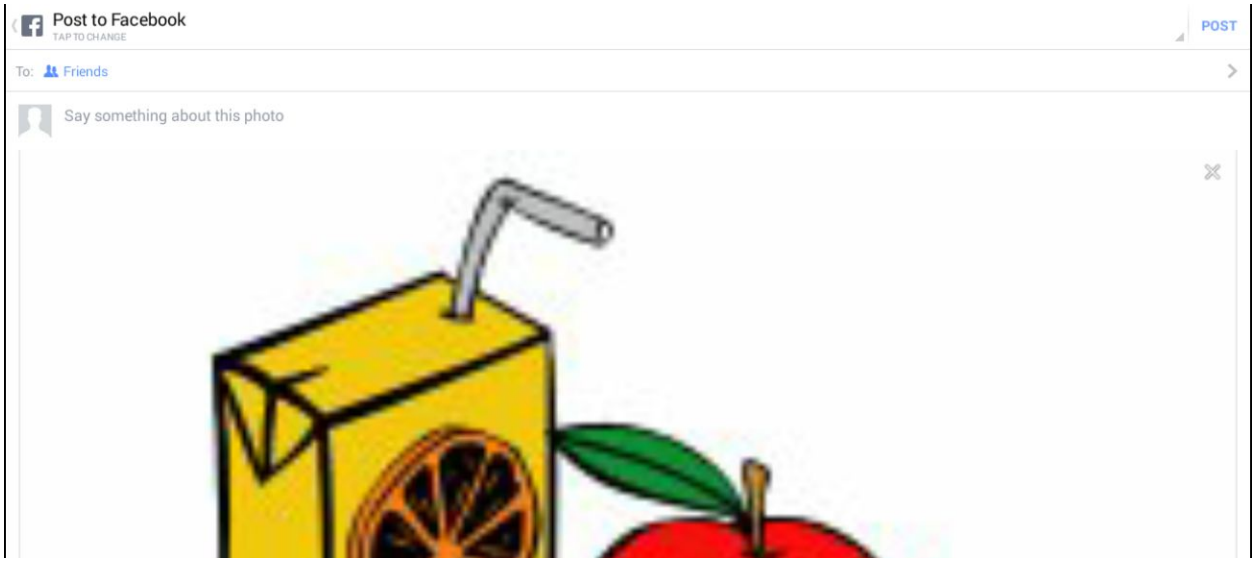
The achievements page shows the current achievements the user has unlocked as well as the ones that he/she has not unlocked yet. Clicking on the lock will only show ??? as shown below:



However, if the achievement is unlocked, the user will be able to see what the achievement was for as well as being able to see the numbers for the achievement as shown below for the first meal achievement:

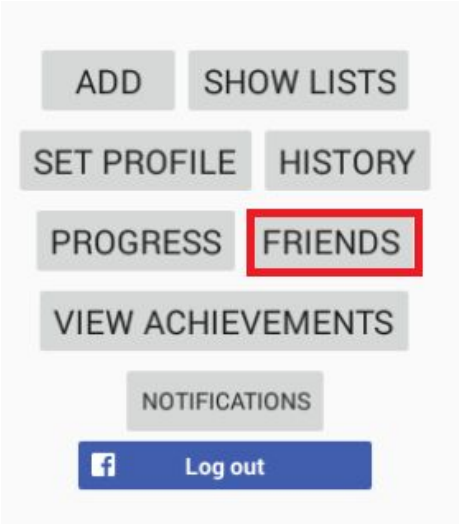


The user can choose to share this achievement on his/her facebook wall by clicking the “share” button. Doing so will open up the Facebook app (if installed) and prompt the user to say something about the photo. If the Facebook app is not installed, the user will not be able to share the achievement.

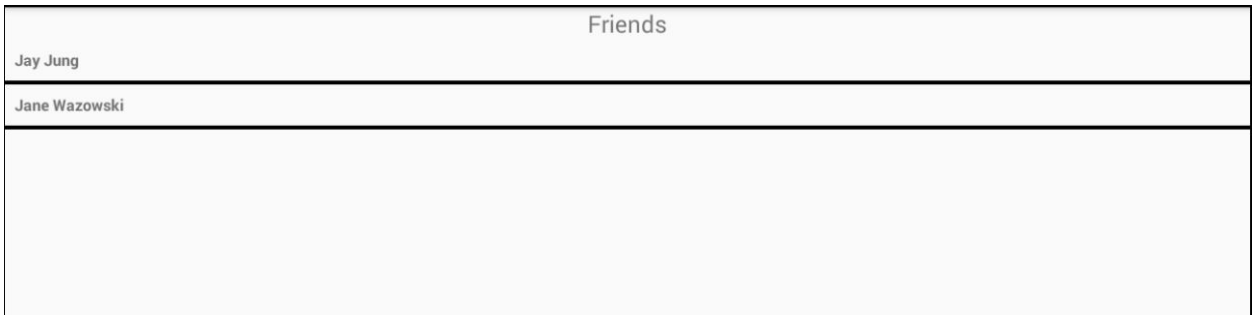


After clicking post, the post will be shared on the user's wall and the user will be taken back to the achievements page.

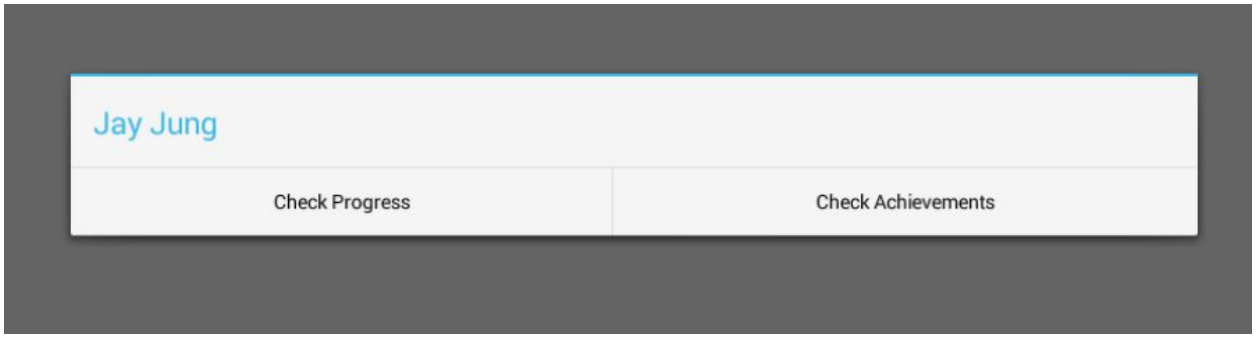
Friends



The friends page can be accessed by clicking on the “friends” button on the main screen, shown above.



The friends page shows the list of the user’s Facebook friends that are currently using the app. Clicking on the friend’s name on the list brings up a pop-up as shown below:



Clicking on the “check progress” or “check achievements” page will allow the user to see the friend’s progress and currently unlocked achievements.

Tech Manual

All screens are referenced within the user manual

Login Screen

- Classes Used:
    - LoginActivity.java - The activity for the login screen. It handles the login page that handles the Facebook login and goes into the Main screen. It also transmits the user's profile information for the Main Activity to use.
    - activity_login.xml - Layout file for the login screen

Main Screen

- Classes Used:
    - MainActivity.java - The activity for the main screen. It handles the buttons pressed and what occurs when the user returns to this screen. Also handles properly logging in through facebook and uploading the database.
    - activity_main.xml - The layout file for the main screen
    - CalorieView.java - The view class that sets up the equation at the top of the main screen.

Add Trackables

- Classes Used:
    - AddTrackableActivity.java - gets user input from TextFields and creates new Trackable and inserts that into the corresponding list. Also updates the database upon entry.
    - activity_food.xml - Layout for adding a food
    - activity_exercise.xml - Layout for adding an exercise
    - Trackable.java - Abstract class that serves as the base element for Food and Exercise.
    - Food.java - "is-a" Trackable
    - Exercise.java - "is-a" Trackable

Show Lists

- Classes used:
    - TabLayoutActivity.java - Overarching activity class that handles the tabs.
    - PagerAdapter.java - Handles switching from one tab to the other
    - TrackableListFragment.java - Handles user interaction with lists within each tab.
    - TrackableList.java - Singleton class containing the locally stored ArrayLists of Trackables that can be sorted, added to, deleted from, and edited.
    - FoodDialog.java - A Dialog class that pops up when a trackable item is clicked on one of the lists to show the trackable's note and the option to add the food.
    - activity_list.xml - handles each individual list layout
    - activity_tab.xml - handles layout for tabs in lists.
    - row_layout.xml - handles layout of individual elements of list and labeling

Profile Page

- Classes Used:
    - ProfileActivity.java - Activity for the Profile Page. Handles displaying the stored user information and button clicks.
    - EditProfileActivity.java - Activity for the Edit Profile Page. Handles button clicks and calculating the required daily caloric intake.
    - activity_profile.xml - Handles the display of user information
    - activtiy_edit_profile.xml - Handles display of the user profile information form.

Notifications Page

- Classes Used:
    - NotificationsActivity.java - Activity for Notifications page. Stores and loads saved alarms.
    - NotificationReceiver.java - Called when an alarm goes off, handles display and name of notification.
    - NewNotificationActivity.java - Activity for new notifications page. Stores the new notification on save.
    - activity_notifications.xml - Handles display of all notifications and pre-programmed alarm.
    - activity_add_notifications.xml - Handles display of the new notification form

History Page

- Classes Used:
    - HistoryActivity.java - Overarching history activity that sets up the list and the adapter for the lists and tabs.
    - HistoryAdapter.java - The adapter decides the list based on the tab chosen and displays the list
    - HistoryListFragment.java - The list that the adapter sets the arguments for. The list shows the information of each of the trackable items.
    - MealChart.java - The activity page to show the pie chart for the breakdown of the caloric/macro breakdowns of the day chosen.
    - activity_history.xml - The layout page for the tabs and the lists
    - activity_meals.xml - The layout page for the pie chart

Progress Page

- Classes Used:
    - ProgressActivity.java - The activity that shows the graph for the progress page. Uses the graphview API to show the data in a graph form.
    - activity_progress.xml - The layout page for the progress page.

Achievements Page

- Classes Used:
    - AchievementsActivity.java - The activity page to show achievements. The achievements are kept in two int arrays of size 8. One array (*achievementFlags*) keeps track of whether the user has unlocked the array or not (0 for locked, 1 for unlocked). Another array (*achievementProgress)* keeps track of the progress of each achievement.
        - There are 8 total achievements:
            - 1. Adding your first food
            - 2. Collecting a deficit of 500 calories
            - 3. Collecting a surplus of 500 calories (both calculated at the end of the day)
            - 4. Logging in 100 hours of exercise
            - 5. Adding 5 foods to the favorites list
            - 6. Adding 5 exercises to the favorites list
            - 7. Sharing an achievement on Facebook
            - 8. Finding the Easter Egg
    - AchievementDialog.java - The dialog that pops up when the achievement picture is clicked on. Gives the option for the user to see what the achievement is for and to share it on Facebook.
    - UnlockDialog.java - The dialog class that pops up when the user unlocks an achievement.
    - activity_progress.xml - The layout page for the achievements page.

Friends Page

- Classes Used:
    - FriendsList.java - The activity to display the list of friends. Pulls the list of friends of the current user from Facebook and puts them into a simple list using an adapter.
    - FriendDialog.java - The dialog that shows up when the user clicks on the friend's name on the list. Gives the user the option to check the friend's progress or achievements.
    - activity_friends.xml - the layout page to show the list of the user's friends
    - row_friend.xml - the layout page to show what to display for each row

Other classes not mentioned above:

Database:

We used the SQLite Database that is in all android phones. The API to use within Java for the database is detailed here:
http://developer.android.com/training/basics/data-storage/databases.html

The database uses one table for which the schema looks like this:

UserInfo
String fbid - stores the fb id of the user
int goal - stores the user's saved goal for each day based on profile
int weight - contains the user's weight - not used in the app currently but was intended to be used
Blob favfood - stores the ArrayList of favorite foods
Blob recfood - stores the ArrayList of recent foods
Blob favex - stores the ArrayList of favorite exercises
Blob recex - stores the ArrayList of recent exercises
Blob d1 - stores the ArrayList of foods in day 1(today)
Blob d2 - stores the ArrayList of foods in day 2(yesterday)
Blob d3 - stores the ArrayList of foods in day 3
Blob d4 - stores the ArrayList of foods in day 4
Blob d5 - stores the ArrayList of foods in day 5
Blob d6 - stores the ArrayList of foods in day 6
Blob d7 -stores the ArrayList of foods in day 7
Blob d1e - stores the ArrayList of exercises in day 1(today)
Blob d2e - stores the ArrayList of exercises in day 2(yesterday)
Blob d3e - stores the ArrayList of exercises in day 3
Blob d4e - stores the ArrayList of exercises in day 4
Blob d5e - stores the ArrayList of exercises in day 5
Blob d6e - stores the ArrayList of exercises in day 6
Blob d7e - stores the ArrayList of exercises in day 7
Blob aflag - stores the int[] of achievement flags - indicators whether the user has unlocked them
Blob aprog - stores the int[] of achievement progress

Relevant Classes

- DatabaseContract.java - uses the "BaseColumns" innerclass to declare the columns for the database table mentioned above
- DatabaseHelper.java - SQLiteOpenHelper class - keeps track of the persistent SQLite database within the android. Has the "onCreate" method to create the table if accessed for the first time. Also keeps track of the "DATABASE_VERSION" variable, that when changed, the database automatically updates itself using "onUpgrade" and "onDowngrade" methods.
- Serializer.java - The class that has static methods to serialize lists or arrays into byte[]'s so that they can be stored into the SQLite or deserialize byte[]'s back into their original forms so that the app can use it.
- ResetReceiver.java - The class that is responsible for receiving the notification to reset all of the current day's lists, store them into the database, and update the table.

Facebook Login Information

Currently, because the application is in test mode, not any user can log in to the app.
The login credentials for the test are:

User: cis350group37test@gmail.com (John Smith)
Password: cis350group37

User: group37test@gmail.com (Jane Wazowski)
Password: jungshi12