Weight Management App

Group 4

Richard Karl

Alexander Joiseus

Andrew Barbosa

Katherin Arreaza

Demetrius Dukes

Abstract

ScaleFit is a Weight Management mobile application which incorporates both measurements (such as weight, height, and abdominal circumference) and an android phone. The measurements that were listed above would be provided by the current user and manually inputted into the android phone. It is used for a variety of people including those wanting to lose weight and have an efficient way of keeping track of their weight loss. The measurements would be used in order to calculate one’s body mass index which is commonly used to give an estimate of one’s health risk due to their weight. The body mass index (BMI) gauges your weight in relation to height. However, this measurement is rather flawed, particularly for those who are very muscular. These individuals can easily be misclassified as obese (since muscle weighs more than fat) when in fact they're just carrying a lot of muscle and very little fat. The BMI measurement also does not give you any indication of where the fat is located in your body, and this appears to be more important than the amount of fat when it comes to measuring heart risks.

Background

ScaleFit is a prototype medical application that takes one’s measurements and provides the user with update to date tracking of their weight along with a current BMI (body mass index) for their recently added weight. Although not added, the vision for Scalefit was to have a calorie tracker alongside the weight tracker to ensure that the user was able to record and be able to lose weight to the most efficient way possible when using this app to help with the management of their weight. Suggestions would have also been added to ensure daily tips could be seen and to help the user stay on task with losing the desired weight.

Although many times when weight is looked at the easiest and perhaps best way to measure your body composition is to simply measure your waist size. Studies have shown that your waist size is a far more accurate predictor of your heart risks than your BMI, as abdominal fat in particular is strongly associated with heart disease. Your waist size is also a powerful indicator of insulin sensitivity, as studies clearly show that measuring your waist size is one of the most powerful ways to predict your risk for diabetes.

Determining your waist size is easy. With a tape measure, figure the distance around the smallest area of your abdomen below your rib cage and above your belly button. If you're not sure if you have a healthy waist circumference, a general guide is:

* For men, between 37 and 40 inches is overweight and more than 40 inches is obese
* For women, 31.5 to 34.6 inches is overweight and more than 34.6 inches is obese

Methods

Below are basic method definitions for our application. Certain methods are derived from super classes and add additional functionality, such as onCreate, onStart, onSaveInstaceState, and a few others. The Reading class has two private classes which are represented by indenting further than previous classes to differentiate between the public classes and Reading’s private classes.

**aboutActivity**

onCreate: Sets up view for activity

onCreateOptionsMenu: adds items to the action bar

onOptionsItemSelected: handles the action bar item clicks

**CustomPagerAdapter**

Class CustomPagerAdapter: sets pictures in the custom pager to be used

getCount: retrieves number of items in resources array (where pictures are held)

isViewFromObject: returns the layout view of an object

inatantiateItem: takes picture depending on position and displays it

destroyItem: destroys the item after usage

**detailsinfragment**

onCreateView: Inflates fragment layout

onStart: Overrides super method and check to see if there are any arguments

updateArticleView: updates current view

onSaveInstanceState: saves information in a bundle if activity is restarted

**EULAActivity**

onCreate: creates EULA to appear whenever activity is called

**LoginActivity**

LoginActivity: takes email and passwords and checks with database

populateAutoComplete: initiates loader manager at null.

attemptLogin: attempts to login with credentials entered compared to database

isEmailValid: checks for @

isPasswordValid: checks if password is longer than 4 characters

showProgress: shows an animation while checking credentials in background if isEmailValid and isPasswordValid hold true

onCreateLoader: retrieves email from user’s profile contact to be checked with email given

onLoadFinished: adds email from sever once data rows are retrieved from database

onLoaderReset: resets the cursor for the Loader

addEmailsToAutoComplete: Create adapter to tell the AutoCompleteTextView what to show in its dropdown list

UserLoginTask: takes email and password and puts into strings from database

doInBackground: background checking, simulating network access

onPostExecute: goes to main activity if successful, displays error if not

setUser: pulls information from database

**LoginForRegistrationActivity**

Same as Activity above, however this is selected after registration and if login after registration is successful then goes to user information activity

**MainActivity**

onCreate: Sets up activity view

onCreateOptionsMenu: Inflates options menu

onOptionsItemSelected: goes to selected Activity along with a brief message of that action at bottom of screen

**MainMenuAdapter**

Sets up hamburger menu (items clicked through hamburger menu not registered to starting of that activity which is why no description is given)

**MedicalEULA**

MedicalEula: Creates popup with medical EULA. If accepted not seen again (only first time on app use) if cancelled app closes.

**ProfileActivity**

onCreate: sets up activity view and finds activity profile for display (not finished)

**RegisterActivity**

attemptLogin: attempts to login with credentials entered compared to database

isEmailValid: checks for @

isPasswordValid: checks if password is longer than 4 characters

showProgress: shows an animation while checking credentials in background if isEmailValid and isPasswordValid hold true

onCreateLoader: retrieves email from user’s profile contact to be checked with email given

onLoadFinished: adds email from sever once data rows are retrieved from database

onLoaderReset: resets the cursor for the Loader

addEmailsToAutoComplete: Create adapter to tell the AutoCompleteTextView what to show in its dropdown list

doInBackground: background checking, simulating network access and sends strings of email and password provided for registration

onPostExecute: goes to login for registration activity if successful, displays error if not

onCancelled: cancels show progress

**ScaleFitEula**

ScaleFitEula: Creates popup with ScaleFit EULA different from medical EULA (appears on weight screen not welcome screen). If accepted not seen again (only first time on app use) if cancelled app closes.

**Settings**

Would have provided different types of profile settings and potentially app changing settings (not finished)

**splashScreenActivity**

onCreate:Displays a small animation when the app is loading up to be used

**team4**

hard coded information for fragments of team 4

**Team4Activity**

Oncreate: Displays fragments and gives information of each member when clicked on

**UserInformationActivity**

Oncreate: Sets up activity view and receives basic user information to be set to datebase

**userObject**

guidelines to be used for future information to take from user and variables to store user information.

**weightScreenActivtiy**

onCreate: sets up activity view with track weight, current weight, bmi, confirmation buttons to send information to database, and graph of weight progress (hard coded graph)

onCreateOptionsMenu: Inflates options menu

onOptionsItemSelected: goes to selected Activity along with a brief message of that action at bottom of screen

WelcomeActivity

onCreate: sets up activity view with custom pager, sign up button (goes to registration activity), login button (goes to login activity), and skip text (skip to main activity)

Discussion

A few improvements would be to add support for multiple users, cleaner, design for the app, making the app more efficient, adding support for healthcare providers to be able to monitor user’s logs and ensure patients are taking their healthcare providers advice between visits, and adding suggestions to keep the users on track as well as providing the users with helpful advice. Challenges that were faced include being able to secure information from hacking on the database, finishing desired result of app due to lack of time. Things that would have been added were: Suggestions/daily tips, working graph to show ongoing process of the weight loss, more information regarding the medical effects of being obese, personalization of profile (profile picture, settings, etc.) and calorie tracker to help with better eating habits that could help with weight loss.

Conclusions

Scalefit gives people the chance to monitor their health and the health of their love ones. The key to having a successful medical app is informing the user on the symptoms and causes that can occur and cause bigger health issues due to a poor weight management, so they can better their health. By explaining to the user what their results mean in medical terms and having two emergency plans depending on how high or low their BMI value is and how different their value is compared to normal ranges could save lives and prevent further weight related issues.

References

<http://developer.android.com/training/index.html>

<http://developer.android.com/reference/packages.html>

American Heart Association. (2014). Losing weight. Retrieved

from http://www.heart.

org/HEARTORG/HealthyLiving/WeightManagement/LosingW

eight/Losing-Weight\_UCM\_307904\_Article.jsp#.Vs5tiiorKSM

Bray, G. A. (2016). Obesity in adults: Prevalence, screening and

evaluation. Retrieved from www.uptodate.com