NutriMe Design Document

Nutrition Tracker

Leon Fattakhov

Table of Contents

1 Introduction

1.1 Purpose

The purpose of this document is to outline the technical design of the Android nutrition tracker application, called NutriMe.

1.2 Description

NutriMe is designed for use on an Android device running version 4.4 or later. NutriMe is an application that tracks your nutrition intake based on the nutrition facts of the food items you eat. It connects to an online database in order to retrieve the information about each product. Then the application displays it to the user allowing them to select and consume the food item. Afterwards the food item is added to a local database that stores the information for later use. This allows users to reference data offline reducing the overall mobile data usage.

1.3 Glossary

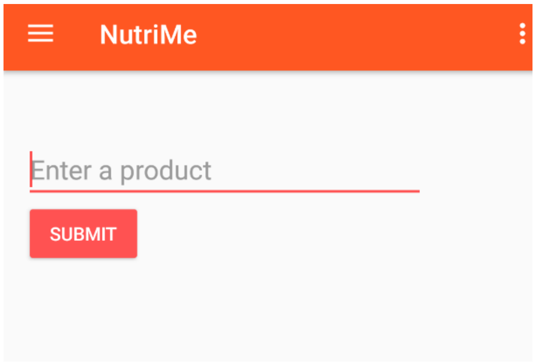
Activity	An activity is a single, focused thing that the user can do usually by interacting with a graphical user interface.
Asynchronous Task	A task that is conducted in the background allowing the main thread to run untouched
JSON object	JSON stands for JavaScript Object Notation it is a lightweight data transfer format. Which is structured with unordered name value pairs.
SQL	SQL stands for Structured Query Language it is a special-purpose domain-specific language used in programming and is designed for managing data held in a relational database management system.
URL	URL Stands for Uniform Resource Locator it is a protocol for specifying addresses on the internet.

2 Design Overview

2.1 Graphical User Interface

2.1.1 Main Activity

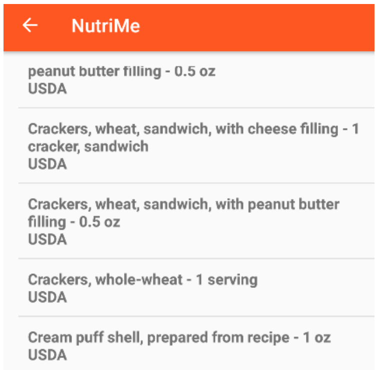
When the NutriMe nutrition tracker application is opened, the user is greeted with a screen containing an auto-completing text entry field. In this field the user will type the food item that they wish to consume. In addition to the entry field, there will also be a submit button. After the button has been pressed the application will redirect to the Display Activity (section 2.1.2). On the left side of the screen a navigation drawer will slide out when the user swipes from the



left screen bezel to the right. In the navigation drawer there are buttons for each graph of the individual nutrition facts.

2.1.2 Display Activity

After the button has been clicked to submit the user's food item the application will take the input and send a GET request to an online database which will retrieve a JSON object with the possible food items that match the search criteria. All the possibilities are then displayed to the user in a list format with the name of the product and the brand name. When the user selects a product in the list, they will then be redirected to an activity with the Nutrition Facts (section 2.1.3) display of the products.



peanut butter filling - 0.5 oz
USDA

Crackers, wheat, sandwich, with cheese filling - 1 cracker, sandwich
USDA

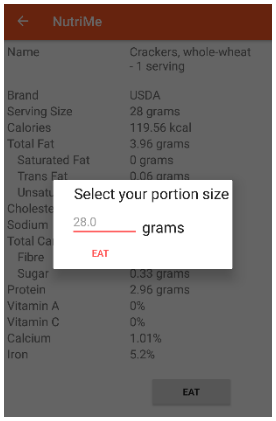
Crackers, wheat, sandwich, with peanut butter filling - 0.5 oz
USDA

Crackers, whole-wheat - 1 serving
USDA

Cream puff shell, prepared from recipe - 1 oz
USDA

2.1.3 Nutrition Facts Activity

After the user selects the food item they would like to consume they are taken to a screen that displays the nutrition facts of the chosen product. This allows the user to cross reference that they have found the correct product that they wish to consume. After verifying that the online food product matches the one they currently possess, they may click a button that will bring up a dialog permitting the user to enter the serving size they wish to consume of the product. This will then be added to the local database for later reference. More information regarding the storage of user data is available in section 3.



Name	Crackers, whole-wheat - 1 serving
Brand	USDA
Serving Size	28 grams
Calories	119.56 kcal
Total Fat	3.96 grams
Saturated Fat	0 grams
Trans Fat	0.06 grams
Unsatu	
Choleste	
Sodium	
Total Ca	
Fibre	
Sugar	0.33 grams
Protein	2.96 grams
Vitamin A	0%
Vitamin C	0%
Calcium	1.01%
Iron	5.2%

Select your portion size

28.0 grams

2.1.4 Graphing Activity

After the user has added food to the database, they can view the results in an interactive graph by going to the graphing tab in the navigation drawer (Part of Main Activity section 2.1.1). Once the nutrition fact graph, they wish to view then they may click the label and be redirected to an Activity with a graph of the selected nutrition facts.

2.2 Class Overview

2.2.1 Backend Class

Methods	
double ObjectToDouble	Converts an Object to a double
JSONObject GetJSON	Downloads a JSON from the URL that it is given
@Override JSONObject doInBackground	Calls GetJSON asynchronously in the background
String toStringWithNulls	Converts value to string and properly handles nulls
double Ratio	Calculates the ratio between the serving size and the user consumption amount
Display	Gathers all possible food items from a JSON Array and returns an Array list of possibilities

2.2.2 Brand Product Object

Fields	
HashMap<String, Object> nutrition facts	Stores all nutrition facts information in a Map
HashMap<String, String> info	Stores all information related to the product in a Map

Methods	
Scale (double ratio)	Returns a BrandProduct object that has been scaled by the ratio
String Get_id()	returns the item's identification string
Get_name()	Returns the name of the product
String prepareForStorage()	Serializes the BrandProduct object into a JSON allowing it to be stored
Map<String, Object> Get_nutritoinFacts()	Get the nutrition facts in a Map
Map<String, Object> getInfo()	Gets the information of the product in a Map
HashMap<String, Object> getEntry()	Returns a HashMap of the Brandproducts nutrition information prepared for graphing.

3 Storage of Data

3.1 Introduction

The users' data is stored locally on the user's device in a Structured Query Language database (SQL). It will consist of two tables one being a table that stores the individual product information and another that stores ratios of the products in the first table.

3.1.1 Table One: Product information table

This table stores the product information of the items that the user eats. It contains two columns one being the identification of the item as the primary key and another column that stores a JSON string of a BrandProduct object that is serialized allowing it to be stored and later recreated.

Product Information	
Identification Number	JSON String

3.1.2 Table Two: Reference to product

This table stores ratios relating to the Product information table as when the user eats a product they may eat different quantities of the product than the recommended serving size and thus creating a double that is then stored in the ratio column. The table also has an identification number allowing for the item to be found in the Product information table. The third column in the table acts as the primary key which stores the date in milliseconds allowing for use later when graphing.

Reference to products		
Identification Number	Ratio	Date