Design Proposal

Project: Food Neturation Checker

AUTHOR ON GITHUB: PRINCE-M-SINGH

Contents

Description

Food Neturation Checker, Check Food Neturation and eat better.

Everyone are now very health consciences in busy life. Examine the real nutritional value of your favourite food, and discover healthier similar products for you and your family.

You like choose good products but you are unable to decipher the label? No time to examine for when you go shopping? With the Food Neturation Checker, by scanning the product barcode, you get in one click an product's nutritional score for what you hold in your hands. And above all, you get a list of similar products with a better nutritional score.

Whence comes this nutritional score? This is a system of ratings from A to E to afford to simply compare the nutritional quality of products. It has been defined by Professor Serge Hercberg in the work of the Research Team on Nutritional Epidemiology (EREN) from Université Paris 13 / Avicenne Hospital. These colors grade are set by calculating a nutritional quality score that reflects a part of the energy, saturated fat, sugars, sodium (high levels are considered unhealthy), and secondly the proportion of fruits, vegetables and nuts, fiber and protein (high levels are considered good for health).

Where do the data used for the calculation? Open Food Facts Project, a non-profit citizen project, created by thousands of volunteers around the world, to list the ingredients, allergens, nutritional composition and all the information on food labels. Currently, 347387 products are registered in their database.

So what are you waiting to judge the nutritional quality of food products that fill your closets?

Intended User

People who buy food product, who want to take care their health to eat well, who want more transparency about food product, who want a fast solution to compare food products. Currently a lot of people taking packed food.

Mainly for people who live in US, European Countries. In future for India. OpenFoodFacts database contains less amount of data but day to day it increases.

Features

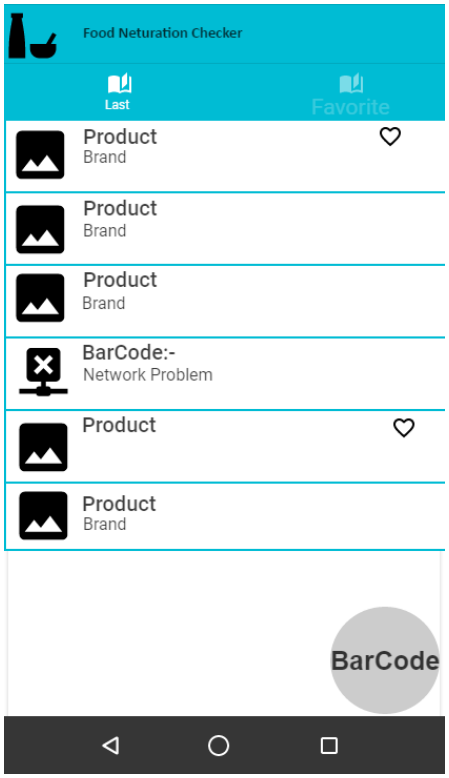
List the main features of your app:

- Scan a barcode of a food product
- Obtain details about this product
- Obtain the nutrition grade of this product
- Obtain the list of product in the same category with better nutrition grade
- Explore the history of scan products and browse products
- Mark a product as favorite
- Explore the list of favorite products
- A widget with the history of scan products and browse products

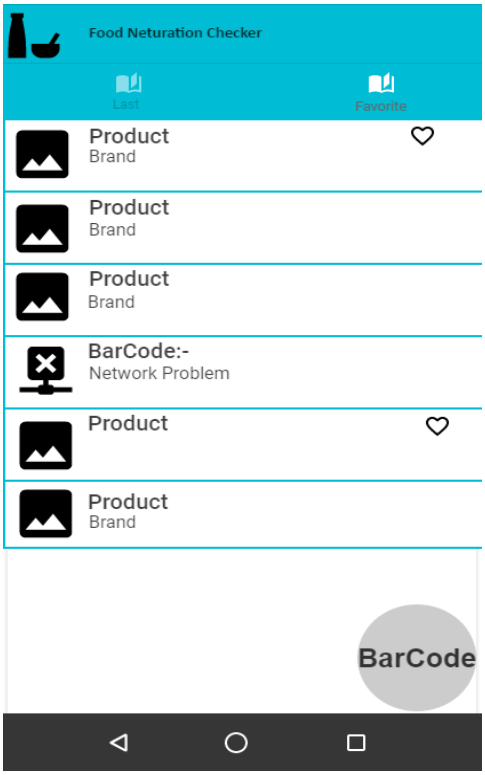
User Interface Mocks

Main screen

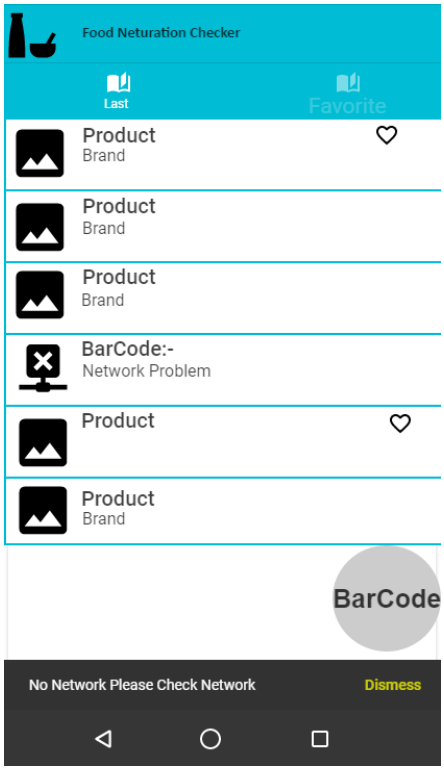
MainScreen1



MainScreen2



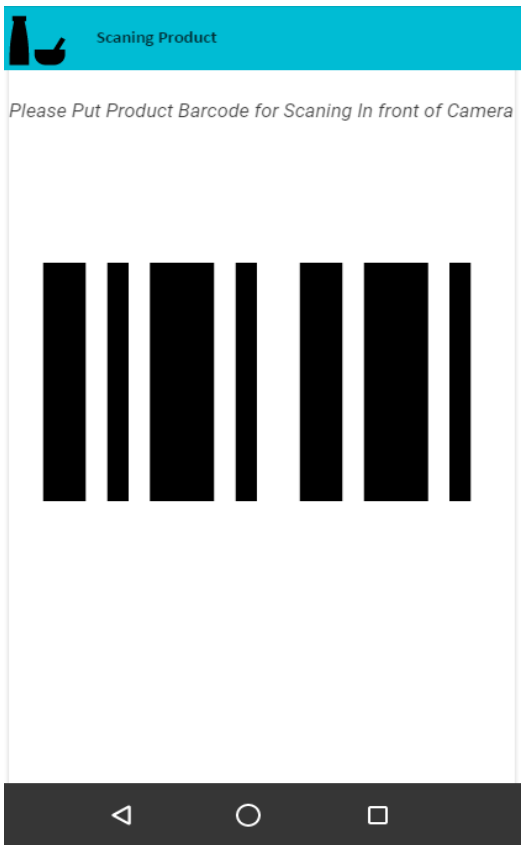
MainScreen 3



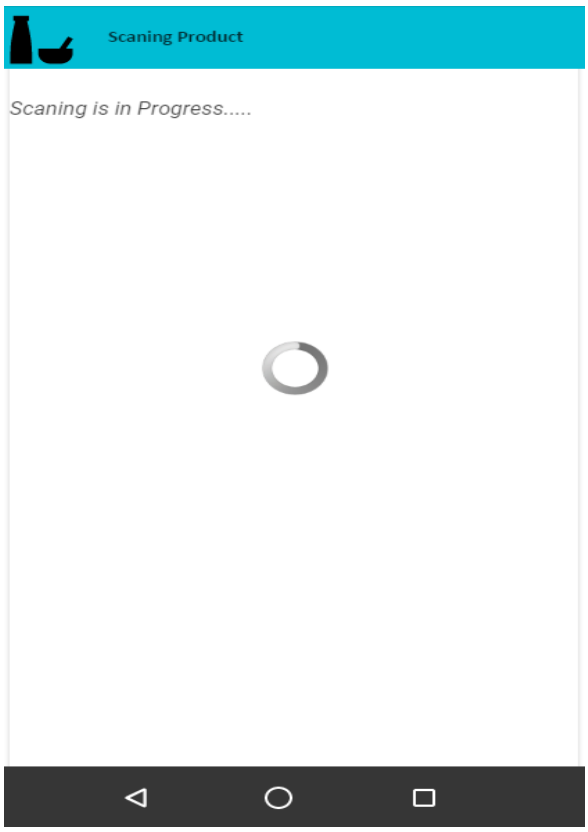
MainScreen 4

Scan screen

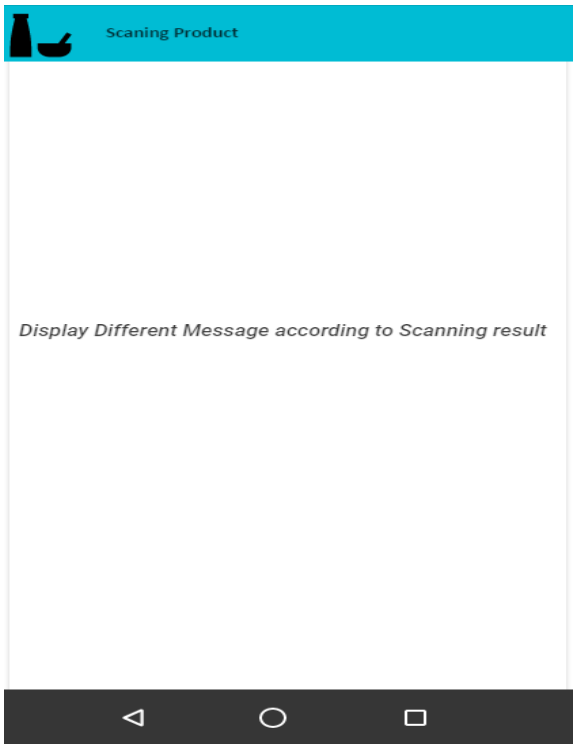
ScanScreen1



ScanScreen2



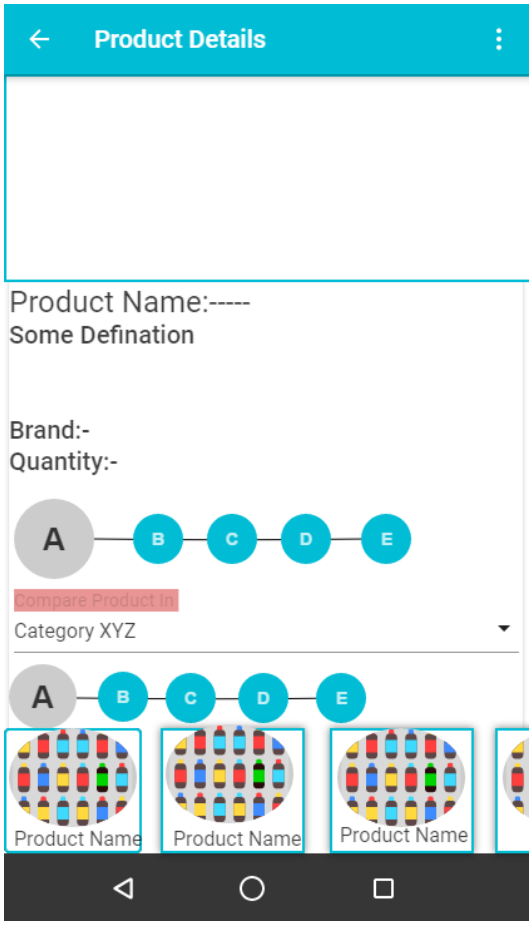
ScanScreen3



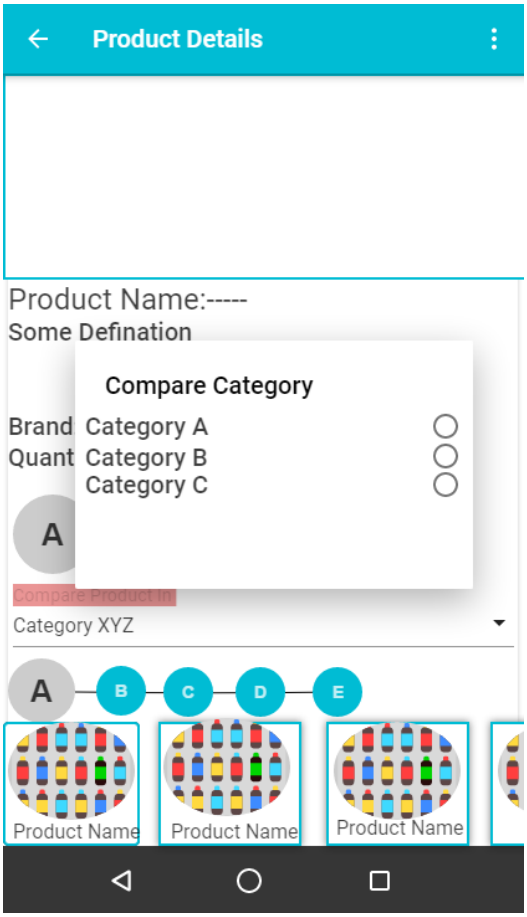
ScanScreen4

Product Detail screen

ProductDetails1



ProductDetails2



Key Considerations

How will your app handle data persistence?

The data is retrieved via an Open Food Fact API. They are stored in a content provider. There are two levels of data for a product:

- The full level, used to the details screen of the product, and for some information displayed on the main screen. Downloaded once on demand during the first display of the detail screen of product The following times, never updated if this is already in the database
- The minimum level used in the comparison lists of similar products Downloaded on demand when displaying a comparison list Then when a new display of comparison list, fully remove and re-download

Despite the redundancy of information, these two levels of data will be stored in two different tables to facilitate the different data processing (persistent or volatile). They will share the unique identifier of the product.

When not connected to the network, the user can browse the data present in the local database.
It can also scan products, associated data will be downloaded the next network access.
If some data are not displayed, the user will be if the cause is the lack of data in the overall base or no network connection.

Describe any corner cases in the UX?

The user starts the application.
Main Screen:
The screen contains two tabs: historyList and favorites.
Each tab contains a list (empty at the first use).
When a list is empty, it contains an image and a text explaining how to populate the list.

A floating share button is present in the lower right, click on it leads to the scan screen.

History tab:
The history tab that is displayed by default when displaying the main screen.
The history tab can contain four types of very similar elements in its list:

- A scanned item (pending network connection)
- A scanned item, not found in the open food fact database
- A scanned item, found in database
- A previously displayed item during browsing on detailed products screens

Only the last two elements are clickable and allow access to the detailed display of a product.

A long press on one of the list items to delete it from the list, and the local database.

When deleting a sketchbar appears explaining the action in progress, with a cancel button.
The back button to exit the application.

Favorites tab:
The Favorites tab contains two type of item:
・A scanned item, found in database

・A previously displayed item during browsing on detailed products screens

Both are clickable and allow access to the detailed display of a product.
A long press on one of the elements of the list to remove it from the list of favorites.
When deleting favorites, toast appears explaining the action in progress, with a cancel button.
The back button to return to the history list.

Scan Page:
The Scan page appears as a shooting photo screen, accompanied by an explanatory text inviting to place the goal in front of a bar code of a food product.
The back button to return to the previously displayed list.
When the barcode is detected, a circle of progress is shown.
Three possibilities depending on the result:

- No network. An explanatory message is displayed. Back to the history tab, the list displays the scanned item pending network connection
- Unrecognized. An explanatory message is displayed. Back to the history tab, the list displays the scanned item not found in the open food fact database
- Recognized. A brief explanatory message appears. The product details screen then appears.

Detail Screen:
The detailed screen shows first pictures and information about the food product.
A button allows the user to mark or unmark a product as favorite.
Then comes the comparison section. A drop-down list of categories of the food product is proposed:

- the amount of products in a category is displayed next
- Only categories containing more than one product are displayed

The default selected category is the one that best describes the product. When a new category is selected, the five scrollable lists are updated. Then there are five horizontal scrollable lists, one for each of level of nutrition grade, each with a title and the number of items in the list. If one of the lists do not contain products, it is not displayed.
If there is no network, an entire area is present to make clear that no network deprives the user of important content. The scrollable lists items, as cardview, are clickable and lead to more detailed screens of other food products. The back button to return to the history list or to previous product consulted.

No network

The lack of network is indicated by the appearance / disappearance of a gray headband and a text at the top of each screen :

- On the main screen, it is located under the toolbar tabs. It remains displayed even when the toolbar is hidden.
- On the scan screen, it is placed on top.
- On the detailed screen it is set on top, and appears permanently, including during scroll.

Elements included in the item of list of the main screen:

A scanned item (pending network connection) An icon of barcode, the barcode number, an explanatory text on the network waiting

A scanned item not found in the open food fact base A lack of data icon, the barcode number, a text helping to solve a potential problem (not barcode of food, not present in the open food facts database...)

A scanned item and found base / A previously displayed item during browsing on the product page A thumbnail image of the product, its title / brand / size, nutritional rating, favorites? (Information whether the product is favorite or not, is not displayed in the Favorites tab)

Elements included in the list of the detailed screen:
a product photo product name the brand the format nutritional rating [...]
Interstitial Ads
Interstitial advertising will appear sometimes :

- after the product scan and before the display of the detailed display
- between two detailed screen displays.

Display for Tablet
The main intended target for this application is not about tablets and big screens. A special display will be developed without making any real changes in functionality.

Widget
The widget provide the same list than History tab on the Main screen. Click on an item and the associated Detail screen is launched.

Describe any libraries you'll be using and share your reasoning for including them

Image Load:
Picasso - A powerful image downloading and caching library for Android.
OR
Glide - Picasso image loading alternative endorsed by Google
OR

Android Universal Image Loader - Popular alternative for image loading that can replace Picasso or Glide.

Network Communication:
Retrofit - A type-safe REST client for Android and Java which intelligently maps an API into a client interface using annotations.
OR
Android Async HTTP - Asynchronous networking client for loading remote content such as JSON.
OR
Volley - Google's HTTP library that makes networking for Android apps easier and most importantly, faster.
OR
OkHttp - Square's underlying networking library with support for asynchronous requests.
OR
Fast Android Networking - Fast Android Networking is a powerful library for doing any type of networking in Android applications which is made on top of OkHttp Networking Layer.

Bar Code Reader:
ZXing - Barcode or QR scanner
OR
Mobile Vision: Barcode Scanner API detects barcodes in real time in any orientation. You can also detect and parse several barcodes in different formats at the same time
OR
barcodescanner- Newer alternative

Other Library:
ButterKnife - Using Java annotations, makes Android development better by simplifying common tasks.
Parceler - Android Parcelable made easy through code generation
IcePick - Android Instance State made easy
LeakCanary - Catch memory leaks in your apps
Espresso - Powerful DSL for Android integration testing
Robolectric - Efficient unit testing for Android

Describe how you will implement Google Play Services

com.google.android.gms.vision.barcode : To detect barcode and recognize it

com.google.android.gms.ads : to display interstitial ads

Next Steps: Required Tasks

Task 1: Setup

- Configure libraries

- Choose an Architecture and implement skeleton Android Architecture Blueprints
- Create a debug and build variant
- Configure tools

Task 2: Database

- Database
- Content Provider
- Data models

Task 3: Network

- Data parsing
- Data syncing service

Task 4: Barcode scanner

- Define what we want to detect
- Create specific activity
- Handle error and success and adapt user information

Task 5: User interface

- String and image resources
- UI for main activity
- UI for detail activities
- Gestures, transition

Task 6: Pull data

- CursorLoader
- Ask for data

Task 7: Handle Error Cases

- Empty views
- Error message

Task 8: Implement Google Play Services

- Google Barcode
- Google AdMob

Task 9: Improve design

- Landscape mode
- Large screen
- Tablets
- Transitionnal screen

Task 10: Generate and Deploy

- Generate app flavor keys
- Create Google Play Account
- Create APK
- Create Google Play image and text
- Push the APK on the store