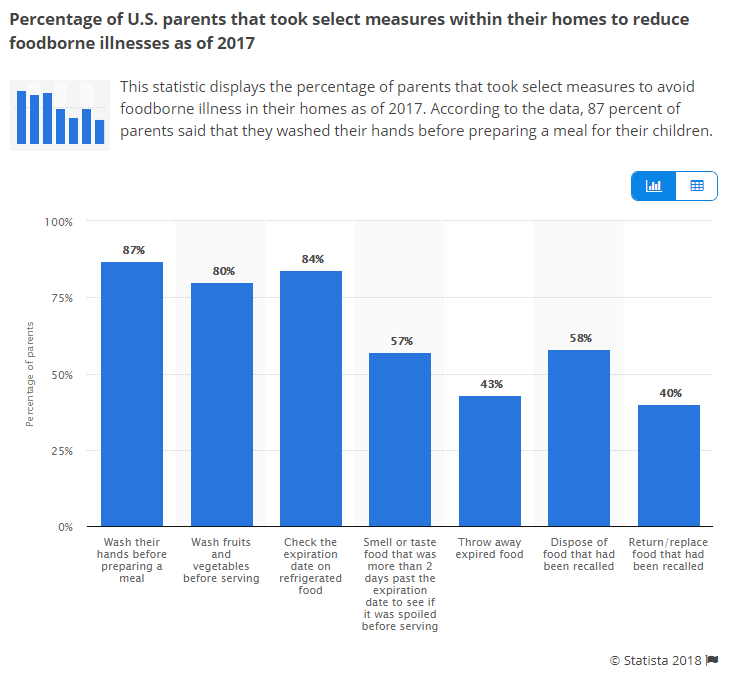
Expiry Tracker

Research



https://www.statista.com/statistics/877291/parent-food-safety-measures-us/

More than 4-in-5 consumers check dates on their food before consuming, with almost half throwing away expiring food

<https://www.cnbc.com/2015/04/22/americas-165-billion-food-waste-problem.html> (news link)

2. Don't put too much stock in the expiration date.

There is a great deal of confusion about expiration dates, and contrary to popular belief, most dates aren't statements about food safety.

There are two kinds of dates that commonly appear on food products. A "sell by" date is intended to be a message from the food manufacturer to the retailer so the store knows how long to display an item. It indicates that the product will still have significant shelf life once it reaches a consumer's home. "Best by" dates refer to quality, not safety, and signifies best flavor or peak freshness. A product will still be edible for several days afterward.

Read MoreA chef going where fast-casual boom has failed: healthy fare

Unfortunately, due to the lack of federal regulation about date labeling and confusion among consumers, many retailers and consumers throw food out on or before the date on the package, no matter what the date was intended to mean. "This contributes to enormous food losses at home and in the store," Berkenkamp said. "The best bet for consumers is to use your own judgment about whether the food in your fridge is still good. And if you think you can't use it up soon enough, pop it in your freezer rather than throwing it out," she said.

Github: <https://www.github.com/trod-123>

Description

Tired of throwing away food because you discovered they’re expired? Did you know that some foods are perfectly okay to use even beyond their printed date? What if there were a way to track your food and get notifications on when they’re about to expire?

*Expiry Tracker* allows you to enter expiration dates for your food products and reminds you when your food is about to expire. This app is designed to be interactive, so can enter your food with your date assistant. *Expiry Tracker* also provides additional information about how long your food can last even beyond the food’s printed expiration date.

Audience

Food-conscious consumers who

1. Care about the quality of the food they eat
2. Want to be more aware of whether their food is still good to eat
3. Wish to save money from throwing away food that’s still safe to use

This app is also tailored to store owners who need to

1. Keep track of the “sell by” dates for their products
2. Be notified in advance when their products will expire
3. Wish to prevent expired products from getting into the hands of consumers

Features

* Saves product and date information locally and in the cloud. Access your food information offline and online across your phones and tablets *(currently, only Android)*
* Barcode scanner to simplify putting food into the database. Pulls your food info quickly so you don’t need to enter it yourself! Also fetches images of your food online so you don’t need to take pictures
* Interact with your date assistant who can help store your food info for you
* Voice recognition to speed up food input, instead of typing it in
* Notifications to remind you when your food is about to expire. Customize how often and when you want to be notified before your food expires
* Home screen widget to display a list of foods that are expiring soon

Key Considerations

How will your app handle data persistence?

This app uses Room database to locally cache user data. For online storage, it leverages Firebase Realtime Database, which users can access through Firebase Authentication.

Food and product data received through barcode scanning and item search is gathered from the [UPC Item Database](https://devs.upcitemdb.com/) as Json objects, which are then parsed into Food objects which are stored in users’ own databases.

Ancillary data on how long food is still good for beyond the expiry date is gathered from [Eat By Date](https://www.eatbydate.com/) and returned to the user in the form of search results on a webpage. These pages can be bookmarked by the user and linked with their items.

Describe any edge or corner cases in the UX

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

The implementation of this app will rely on several libraries. All of the libraries used will be the latest stable versions

Android libraries

* Android Support Library 27.1.1
* Android Architecture Components
  + [Lifecycle](https://developer.android.com/topic/libraries/architecture/adding-components#room) 1.1.1
    - For LiveData and ViewModel support with Room
    - android.arch.lifecycle:extensions:1.1.1
    - android.arch.lifecycle:compiler:1.1.1
  + [Room Database](https://developer.android.com/topic/libraries/architecture/adding-components#room) 1.1.1
    - For storing users’ data locally
    - android.arch.persistence.room:runtime:1.1.1
    - android.arch.persistence.room:compiler:1.1.1
  + [Paging](https://developer.android.com/topic/libraries/architecture/adding-components#room) 1.0.0
    - To be used with Recycler View to display users’ data in a list format, loading and displaying items efficiently on demand
    - android.arch.paging:runtime:1.0.0

Google Play Services / Firebase libraries

* [Firebase Realtime Database](https://firebase.google.com/docs/database/) 16.0.1
  + For providing online storage for users’ food items across their own devices.
  + Complements Firebase Authentication so users can only access their own content, and not anyone else’s
  + com.google.firebase:firebase-database:16.0.1
* [Firebase Storage](https://firebase.google.com/docs/storage/) 16.0.1
  + For users to save their pictures of their food online in a way so only users can access their own pictures.
  + Complements Firebase Authentication so users can only access their own content and not anyone else’s
  + com.google.firebase:firebase-storage:16.0.1
* [Firebase Authentication](https://firebase.google.com/docs/auth/) 16.0.3
  + For users to sign-in to their accounts to save and access their data across devices
  + com.google.firebase:firebase-auth:16.0.3
* [Firebase x Google AdMob](https://firebase.google.com/docs/admob/admob-firebase) 15.0.1
  + For providing ads to the app’s free version only
  + com.google.firebase:firebase-ads:15.0.1
* [Firebase ML Kit](https://firebase.google.com/docs/ml-kit/) 17.0.0
  + For allowing users to take pictures of their products, either barcode or label, and parsing that data to simplify search and input
  + com.google.firebase:firebase-ml-vision:17.0.0

3rd party utility libraries

* [Retrofit](https://github.com/square/retrofit) 2.4.0
  + HTTP client that simplifies gathering and parsing API results from online into JSON. Will be used for the APIs listed below
  + com.squareup.retrofit2:retrofit:2.4.0
* [Moshi](https://github.com/square/moshi) 1.6.0
  + For simplifying parsing JSON into Java objects
  + com.squareup.moshi:moshi:1.6.0
* [Glide](https://github.com/bumptech/glide) 4.8.0
  + For simplifying image caching and loading. The app will display images of the food items that users enter in the app
  + com.github.bumptech.glide:glide:4.8.0
  + com.github.bumptech.glide:compiler:4.8.0
* [Timber](https://github.com/JakeWharton/timber) 4.7.1
  + For simplifying logging and debugging
  + com.jakewharton.timber:timber:4.7.1
* [ButterKnife](https://github.com/JakeWharton/butterknife) 8.8.1
  + For reducing boilerplate code for view binding
  + com.jakewharton:butterknife:8.8.1
  + com.jakewharton:butterknife-compiler:8.8.1
* [LeakCanary](https://github.com/square/leakcanary) 1.6.1
  + For detecting memory leaks across fragments and activities
  + debugImplementation 'com.squareup.leakcanary:leakcanary-android:1.6.1'
  + releaseImplementation 'com.squareup.leakcanary:leakcanary-android-no-op:1.6.1'
  + debugImplementation 'com.squareup.leakcanary:leakcanary-support-fragment:1.6.1'

APIs

Additionally, this app will rely on the following APIs:

* UPC Item Database: For grabbing data associated with barcodes and food

**Required Tasks**

Implementation

This app will be built on Java using the latest stable versions of Android Studio and the Gradle Build platform. It will target P, SDK 28, and support older versions back to K, SDK 19, to target at least 90% of devices. Gradle will be used to manage all app dependencies. See below for a list of the versions of software that will be used.

Software versions

* Android Studio 3.1.4
* Android Gradle plugin 3.1.4
* Gradle 4.10
* Java 1.7
* Minimum Android SDK version 19
* Target Android SDK version 28

The implementation will not use hardcoded values, except those defined in the res/values folder. This will hold true for dimensional values, strings, integers, Booleans, and so forth

This app will be localized through supporting RTL layouts for RTL languages, specifying which strings need translations, and providing flexible layouts to accommodate nuances of different languages

Additionally, this app will support Android’s accessibility features, through content descriptions and navigation through D-pad

This app will utilize responsive design to provide flexible layouts to accommodate the available space on the different screen sizes and densities of most devices used, including phones and tablets

Errors from server and user input, data parsing, and UI loading will be handled gracefully to prevent the app from crashing for users

A home screen widget will provide a list that updates daily, showing the foods that will expire the soonest

The user interface will follow Material Design guidelines as outlined in the official documentation [here](https://material.io/design/), also including fluid animations, transitions, and shared elements to provide delightful and engaging user experiences.

Free and Paid versions of the app will be available, with the free version containing non-obstructive banner ads. The Paid version will contain no ads.

This app will build and deploy a Release build configuration, and the source code will provide the associated keystore and passwords with the signing configuration used. The keystore will be referred to by a relative path.