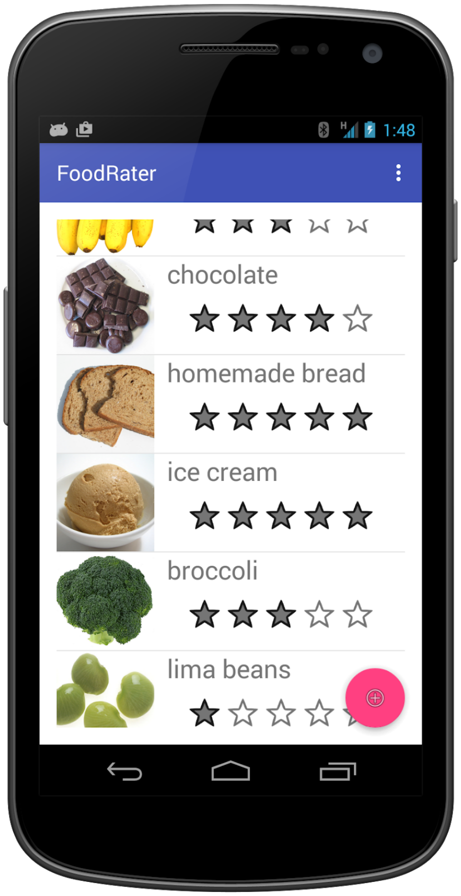
Lab 4: Food Rater (Adapters)

Goal

The goal for this lab is to learn how to work with RecyclerViews & RecyclerAdapters. We will also get experience with RatingBar widgets.

Part 1: ListView

In the first part of this lab, you will create the following app:



Requirements

1. In the codealong, we used a ListView first and then changed it later to a RecyclerView, so you'd have an example of each. You may choose to do the same here (as I did), or you can go straight to using the RecyclerView. Either way, **you only have to demo the final project (the RecyclerView) but** you need to make sure you meet all the requirements for parts 1 and 2.
2. When the app is started, a list of foods of your choosing appears, each with a picture and a rating of 0 stars.
3. The user can scroll through the list of foods to view them.
4. The user can rate each food so many stars using the ratings bar. The rating stays with the food when a new one is added (so the rating of each food needs to be stored).[1] This means that if we scroll down do the food goes off the screen, when we scroll back, its rating doesn't suddenly change!
5. The user can add a new random food with its corresponding picture to the top of the list by clicking the floating action button on the lower right. The initial rating is 0.0 (no stars selected). A snackbar appears, displaying the name of the food added.
6. (Optional) Dis-allow duplicate foods, so you only get at most one "chicken", etc.

Hints/guidelines:

1. Your activity will have only a single ListView that takes up the entire main content. I suggest starting with a BasicActivity so you get the FloatingActionBar.
2. Write a Food model class to hold the name, resource ID, and rating of each food.
3. Write a FoodAdapter class that extends BaseAdapter. It will manage the list of Food objects. The most interesting method is getView(), since it needs to set the text, image, and rating.
4. I am giving you a set of images for download here for 8 foods, to save you from having to hunt online for foods. Since the resource IDs must be single words and I wanted more descriptive names in a couple places, I used a map of food names and resource IDs:

   ```
   sDefaultNamesAndIds = new HashMap<>();
   sDefaultNamesAndIds.put("banana", R.drawable.banana);
   sDefaultNamesAndIds.put("broccoli", R.drawable.broccoli);
   sDefaultNamesAndIds.put("homemade bread", R.drawable.bread);
   sDefaultNamesAndIds.put("chicken", R.drawable.chicken);
   sDefaultNamesAndIds.put("chocolate", R.drawable.chocolate);
   sDefaultNamesAndIds.put("ice cream", R.drawable.icecream);
   sDefaultNamesAndIds.put("lima beans", R.drawable.limabeans);
   sDefaultNamesAndIds.put("steak", R.drawable.steak);
   ```

   You can do likewise or choose another solution if you prefer.

5. The ratings bar is just a RatingBar widget: http://developer.android.com/reference/android/widget/RatingBar.html. It has an inner

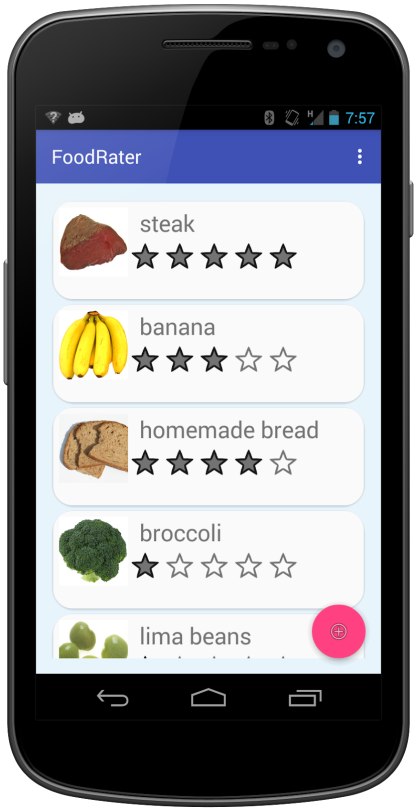
[1] This doesn't mean that every ice cream in the list has to have the same number of stars. Also, it does need to be stored in the model object, but not persisted through rotation or activity destruction.

class you need to use to listen for changes from the user. I suggest reading the javadoc for it, as the callback has a helpful `fromUser` parameter.

6. Each row in the adapter was specified using a food_view.xml file. It just has a 92x92dp ImageView, a 24sp TextView, and a RatingBar, scaled down to 75% size (android:scaleX and :scaleY both set to 0.75 and android:transformPivotX="0dp") since otherwise it was a bit too big for my phone. You can modify sizes, margins, etc if needed to make it look nice on your device.
7. Put the method to add a food in the FoodAdapter.

Part 2: RecyclerView

In the second part of this lab, you will modify your app to the following:



Additional requirements:

1. The whole view is now a RecyclerView instead of a ListView.
2. Each food row view is now a CardView as shown above, with noticeably-rounded corners. Add a light-colored, semi-transparent background color of your choice to the activity so the cards stand out visually.
3. When the user adds a new random food, it will *animate in* to position 0 and appear at the top of the screen.
4. Long pressing anywhere on a row will delete that food from the array and view, using a delete animation on that position.

5. Partial credit: must show food nicely (2), add (2), remove (2), ratings can change (2), card with color (2), scroll to top / animations (2).

Additional Hints/guidelines:

1. You can make a CardView just by wrapping your item view's layout xml in a cardview widget tag. The links below show how to do this, plus how to get rounded corners.
2. Don't forget to add dependencies for RecyclerView and CardView, as discussed at the bottom of this Android training page: https://developer.android.com/guide/topics/ui/layout/cardview.html
3. The bulk of the work on this part is to break up the work done by getView into the ViewHolder, createViewHolder, and bindViewHolder methods.

Additional resources

http://stackoverflow.com/questions/2874537/how-to-make-a-smaller-ratingbar
Training (working demo): http://developer.android.com/training/material/lists-cards.html
Vogel tutorial (notifyItemInserted/Removed/Edited for animations):
http://www.vogella.com/tutorials/AndroidRecyclerView/article.html
Better onClick() in VH constructor: https://gist.github.com/grantland/cd70814fe4ac369e3e92
Another CardView example:
http://code.tutsplus.com/tutorials/getting-started-with-recyclerview-and-cardview-on-android--cms-23465

Image Credits

http://www.picserver.org/pictures/broccoli01-lg.jpg
https://pixabay.com/static/uploads/photo/2013/01/08/01/49/boiled-beef-74374_640.jpg
https://c1.staticflickr.com/5/4124/5034578144_9d82641e7c.jpg
https://upload.wikimedia.org/wikipedia/commons/thumb/3/37/NCI_lima_beans.jpg/1024px-NCI_lima_beans.jpg
https://upload.wikimedia.org/wikipedia/commons/d/df/Chocolate.png
https://encrypted-tbn2.gstatic.com/images?q=tbn:ANd9GcQdX9I-X33ENQBLWHb9q7md2to4IILGW7Lb-Cib6CCmiEvX6nyI
https://upload.wikimedia.org/wikipedia/commons/9/97/Stale_bread.jpg
https://upload.wikimedia.org/wikipedia/commons/thumb/6/69/Banana.png/836px-Banana.png