

Overview:

My proposed organizing system is a library for customizable Android code. As an Android developer I use many libraries to develop applications (specifically utility applications). However I've been frustrated by the fact that I couldn't find any organizing systems that had organized existing android libraries on the web and had an intuitive user interface. To overcome this I've had to use different websites such as Google Search, Google Forums, Stack Overflow and recommendations from friends to figure out what libraries are best to use.

Domain: Java code running on Android SDK.

Scope: An organizing system for customizable Android libraries.

Scale: All free Android code libraries and tutorials that are available for free on the web.

Tags: # Hyponymy/Hyperonymy (5.4.1.1), # Hierarchical Classification, # Structure (5.5.3) for intuitive user interface.

5 dimensions of an organizing system

1. What is being organizing:

My proposed organizing system will organize any customizable Android library and tutorial that is available for free on the Internet. These resources are mainly in Github. Github resources show the list of available folders and usually a short tutorial of what elements of the code are customizable. However in many instances the resources may be a website solely dedicated to a particular code (for instance push notification) or a site which has 5-10 Android specific libraries made by a developer and available for download. These websites have their unique structure but can be extremely valuable resources for finding free code to use.

Many of these resources have a limited useful life timeline as they may only be compatible with a certain version of Android SDK for instance a code library may be compatible with Android 4.2 and when a new update for Android operating systems comes out the library may no longer be compatible. The useful life-time of these libraries is also dependent on the creator as many library developers update their code hence increasing the life time and usefulness of their code. These resources are all digital and unique.

2. Why is it being organized:

The reason for organizing this is to make it easier for developers to find ready to use libraries and utilize them for their existing projects. Current methods for finding these libraries are through Google or sites such as stackoverflow.com that are not solely dedicated for Android libraries and can be very inefficient at times. (see Figure 6)

Reasons for using such a library can be both personal and social. From a personal perspective someone may find that using such a library is very useful and they feel indebted to give back thus they vote and recommend good library resources to be added to the organizing system. From a social perspective, many Android developer have the sense that they are all part of a community, as an Android developer myself I have been to community gathering dedicated for Android Developer such as Android TO (the largest Canadian Conference for Android Developers) and we as a community believe we have to help each other out and help this space grow. Thus it feels as an obligation to help other people to write code for the Android platform.

3. How is it being organized:

To make this library as useful as possible, building the appropriate semantic hierarchy, which users can find the most relevant library with minimal number of clicks is key. For the first version of my library I created 5 library categories (Database, Navigation, Widgets, Image & List View) that based on my experience are usually needed for any Android application. Thus what TDO refers to “carving nature at its joints” is the approach I took to create the highest-level groups. To create subgroups I used Hyponymy/Hyperonymy to differentiate how they are different from each other and why they are a subset of their high level groups. These are shown in Figure 3.

Additionally the structure of the website layout needs to be intuitive as such I used different font sizes (as shown in Figure 2 and 4) to demonstrate the difference between group, subgroups and code libraries (main resources of our collection). I also placed the title on top to make it clear for the users where they are and what the main topic of each page is. For a rating system I used a star method that is similar to rating hotels (e.g. 5 star hotel is the best and 1 star the worst). This is to make it easier for users to understand which library is rated higher and will most likely be relevant to the task at hand. Overall the structure is build to reflect the hierarchy of the system (contrast Figures 1 & 2) to make the site as intuitive as possible.

This system is organized in both a top-down and bottom-up manner. Bottom-up, because the content of the site is user generated and organized within each category and Top-down because the creator/moderator will ultimately be responsible for modifying the hierarchy, main groups and subgroups for all the libraries to fall under. All resources are organized in the same manner and to the same degree.

4. When is it being organized:

The libraries will be updated real-time based on input from the users. There are two types of user inputs:

1. A new library added to the system (including name, description and URL) and
2. Rating the libraries based on how well designed they are.

Additionally, The creators/moderators of the entire collection will look at the library once every month to ensure that the high level categories are relevant and do a good job of grouping libraries such that no category contains too many or two few libraries.

5. Who (or what) is organizing it:

This organizing system will take links to libraries from users and will organize the libraries when they are added to the collection. Under each category there will be links to various external libraries and the users will be able to vote on which site they found most useful, this allows the organizing system to optimize its results and show results that other users find most relevant and useful.

Overall three groups of people impact the site's organization:

1. The creators/moderators: who create the hierarchy and main level library code groups
2. The users: who add or recommend code libraries to the system and also vote existing libraries up or down within the system.
3. They system: takes the input from the users and based on votes (or number of clicks) re arranges the libraries within each category such that the most popular ones appear on top and least used (and low rated) ones appear on the bottom.

Supporting Material:

Figure 1. Hierarchical Classifications

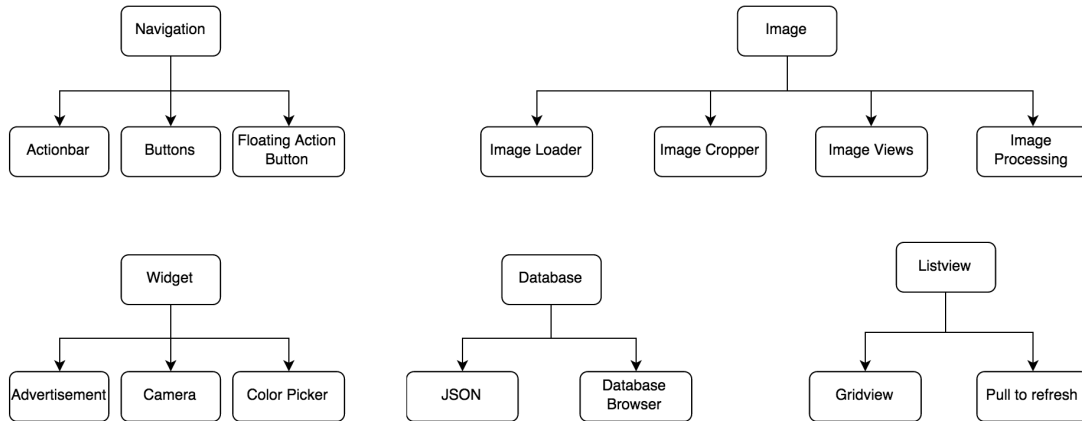


Figure 2. Website Structural Design based on Hierarchical classifications

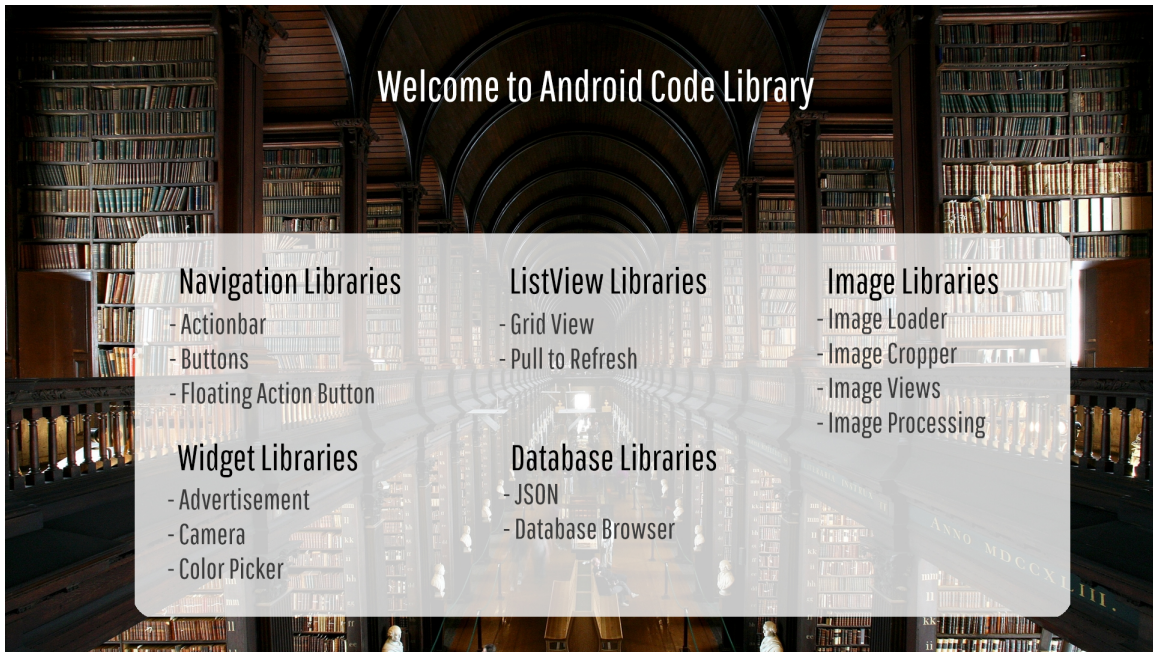


Figure 3. Hyponym and Hypernums for the hierarchy

Hyponym	Hypernym	{adjective+} hypernym {distinguishing clause+}
Navigation	Actionbar	{High level} Navigation Libraries {Placed on top of the android screen allowing quick access to the main actions of an app}
Navigation	Buttons	{Action oriented} Navigation libraries {that provide buttons with various characteristics and toggle a specific action when pressed}
Navigation	Floating Action button	{floating} Navigation Libraries {that place an icon on top of the other element of an app, clicking on these buttons override functions of layers that are behind}
Image	Image Loader	{Loader} Image Libraries {that allows for fast and efficiently image loading}
Image	Image Cropper	{Modifying} Image Libraries {that crop a specific section of an image}
Image	Image Views	{Customizable} Image Libraries {that provide a set of custom shaped android imageview components, and a framework to define more shapes.}
Image	Image Processing	{Processor} Image Libraries {that provide frameworks for faster image processing or image transformations}
List view	GridView	{multicolumn} List view libraries {that are customizable and provide unique layouts such as Pinterest's layout}
List view	Pull to refresh	{easy to use} List view libraries {that allow users to pull down a list to refresh the list}
Widgets	Advertisement	{money generating} widgets {that allow placements of advertisements within mobile app}
Widgets	Camera	{Camera} widget {allowing users to capture pictures with their phones camera and use/modify them in the app}
Widgets	Color picker	{Graphical} Widget Libraries {that allows users to pick a color from a color canvas}
Database	JSON	{data-interchangeable format} Database libraries {that are based on Javascript Object notation and are relatively easy for people to read and write}
Database	Database browsers	{visual} Database libraries {that provide tool to create, design, and edit database files}

Figure 4. Structure for a sample library sub category

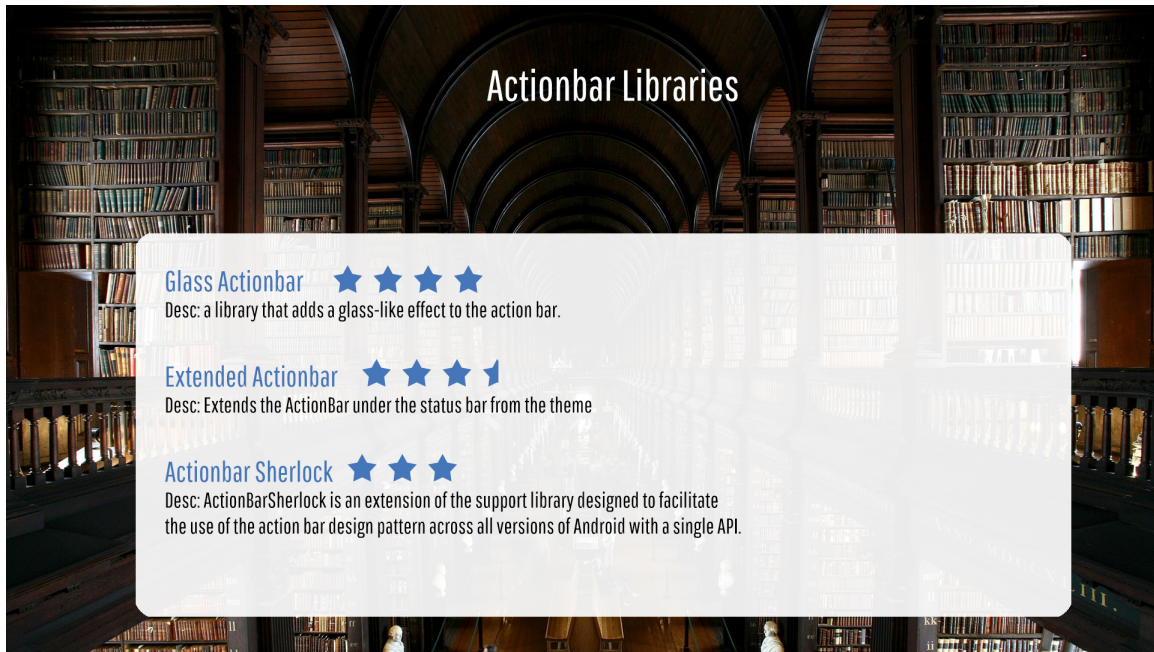


Figure 5. Link to actual libraries for each subcategory

ActionBar		
Library Name	Description	Github/Tutorial URL
GlassActionBar	a library that adds a glass-like effect to the action bar.	https://github.com/ManuelPeinado/GlassActionBar
Extended ActionBar	An example of how to extend the ActionBar under the status bar from the theme	https://github.com/Takhion/android-extendedactionbar
Actionbard Sherlock	ActionBarSherlock is an extension of the support library designed to facilitate the use of the action bar design pattern across all versions of Android with a single API.	http://actionbarsherlock.com/

Buttons		
Library Name	Description	Github/Tutorial URL
Flat Buttons	a beautiful flat button library for Android	https://github.com/hoang8f/android-flat-button
Progress Button	A custom progress indicator with a tiny footprint.	https://github.com/f2prateek/progressbutton
Process Button	Android Buttons With Built-in Progress Meters.	https://github.com/dmytrodanlyk/android-process-button

Floating Action button		
Library Name	Description	Github/Tutorial URL
Android Floating Action Menu	Floating Action Menu for Android. Inspired by the Google Plus floating menu.	https://github.com/sephiroth74/android-floating-action-menu
Floating Action Button	Floating Action Button for Android based on Material Design specification.	https://github.com/futuresimple/android-floating-action-button

Image Loader		
Library Name	Description	Github/Tutorial URL
Novoda Image Loader	Image download happens off the UI thread and the images are cached with a two-level in-memory/SD card cache.	https://android-arsenal.com/details/1/212
Cube SDK	Cube SDK is a light android development framework that allows images to load efficiently and easily.	https://github.com/etao-open-source/cube-sdk
Universal Image Loader for Android	This project aims to provide a reusable instrument for asynchronous image loading, caching and displaying.	https://github.com/nostra13/Android-Universal-Image-Loader

Image Cropper		
Library Name	Description	Github/Tutorial URL
Android CropImage	CropImage Activity from Gallery.apk packaged as a reusable Android library	https://github.com/lvillani/android-cropimage
android-crop	An Android library project to provide a simple image cropping Activity, based on code from AOSP.	https://github.com/jdamcd/android-crop

Image Views		
Library Name	Description	Github/Tutorial URL
Expandable Panel	It implements the needed logic for integrating the expandable logic into your own Android application. Use listeners to implement a response for opening/shrinking events, and combine any of custom attributes to customize a behaviour.	https://github.com/JorgeCastilloPrz/ExpandablePanel
Android Gif Drawable	Bundled GIFLib via JNI is used to render frames. This way should be more efficient than WebView or Movie classes. Animation starts automatically and run only if View with attached GifDrawable is visible	https://github.com/koral-/android-gif-drawable
Rounded ImageView	A fast ImageView (and Drawable) that supports rounded corners (and ovals or circles) based on the original.	https://github.com/vinc3m1/RoundedImageView

Database browsers		
Library Name	Description	Github/Tutorial URL
Android DbInspector	Provides a simple way to view the contents of the in-app database for debugging purposes. No need to pull the database from a rooted phone.	https://github.com/infinum/android_dbinspector
SQLite Browser	High quality, visual, open source tool to create, design, and edit database files compatible with SQLite.	http://sqlitebrowser.org/

Image Processing		
Library Name	Description	Github/Tutorial URL
Android GPU Image	Goal is to have something as similar to GPUImage as possible. Vertex and fragment shaders are exactly the same.	https://github.com/CyberAgent/android-gpuimage
Android Image Filtering	Image Filtering and FX for the Android platform.	https://code.google.com/p/android-image-filtering

Grid View		
Library Name	Description	Github/Tutorial URL
Android Staggered Grid	An Android staggered grid view which supports multiple columns with rows of varying sizes.	https://github.com/etsy/AndroidStaggeredGrid
Dynamic Grid	Drag and drop GridView for Android.	https://github.com/askerov/DynamicGrid
Free Flow	A layout engine for Android that decouples layouts from the View containers that manage scrolling and view recycling.	https://github.com/Comcast/FreeFlow

Pull to refresh		
Library Name	Description	Github/Tutorial URL
ActionBar-PullToRefresh	ActionBar-PullToRefresh provides an easy way to add a modern version of the pull-to-refresh interaction to your application.	https://github.com/chrisbanes/ActionBar-PullToRefresh
android-PullRefreshLayout	This component like SwipeRefreshLayout, it is more beautiful than SwipeRefreshLayout.	https://github.com/baoyongzhang/android-PullRefreshLayout
PullToRefresh-ListView	A generic, customizable, open source Android ListView implementation that has 'Pull to Refresh' functionality. This ListView can be used as a replacement of the normal Android android.widget.ListView class.	https://github.com/erikwt/PullToRefresh-ListView

Advertisement		
Library Name	Description	Github/Tutorial URL
Admob	A leading global mobile advertising network that helps app developers monetize and promote their mobile and tablet apps with ads.	http://www.google.com/ads/admob
InMobi	Performance based mobile ad network.	http://www.inmobi.com/

Camera		
Library Name	Description	Github/Tutorial URL
Landscape Video Camera	Highly flexible Android Camera which offers granular control over the video quality and file size, while restricting recordings to be landscape only.	https://github.com/jmolmobile/LandscapeVideoCamera
Android Webcam	This project is an Android library to provide an API to access an external USB webcam that supports UVC. The library provides an in-process service (WebcamManager) so that multiple parts of an application can share the same webcam.	https://github.com/openxc/android-webcam

Color picker		
Library Name	Description	Github/Tutorial URL
Android Color Picker	Color picker dialog library for Android users to select custom colors.	https://code.google.com/p/android-color-picker
Color Picker collection	Color Picker from Stock Calendar	https://github.com/gabrielemarotti/colorpickercollection
Color Picker View	A simple yet beautiful color picker component for Android.	https://code.google.com/p/color-picker-view

JSON		
Library Name	Description	Github/Tutorial URL
GSON	Gson is a Java library that can be used to convert Java Objects into their JSON representation. It can also be used to convert a JSON string to an equivalent Java object.	http://code.google.com/p/google-gson
Jackson	High-performance JSON processor.	http://jackson.codehaus.org/
RapidJson	RapidJSON is a JSON parser and generator for C++. It was inspired by RapidXml.	https://github.com/miloyip/rapidjson

Figure 6. The main problem we are trying to solve by implementing this code library

The Problem: The Beginner-Professional Chasm

- The Beginner
 - Learning to code is really hard
 - Platforms like Code Academy and code.org are growing quickly but are focused on beginners
- The Professional
 - Understanding of core concepts and ecosystem of libraries
 - Know where to look for information, can accurately judge quality, can read documentation

Segment	Beginners	Intermediate	Professionals
Profile	Needs to understand the coding fundamentals	Ready to build something but lacks ecosystem understanding	Has worked on more than two commercial applications
Available resources	Code Academy Khan Academy Code.org Code Avengers ...		Github Cocoa Controls (iOS) Android Arsenal Stack Overflow Google forums