PriorityList - Documentation

Table Of Contents:

Description of the App

PriorityList is a semi-classic TO-DO List with a twist - it's tailored to suit Your Priority. Unlike conventional TO-DO Lists, where having favorites of favorites is not feasible, PriorityList addresses this limitation by allowing you to assign a higher priority. Moreover, the app offers customization options, empowering you to personalize your experience.

Device and Software Information

Software Information:

Android Version:	API 33: Android 13.0 (Tiramisu)
Java Version:	Java 11
Android Gradle Plugin Version:	7.4.2
Gradle Version:	7.5

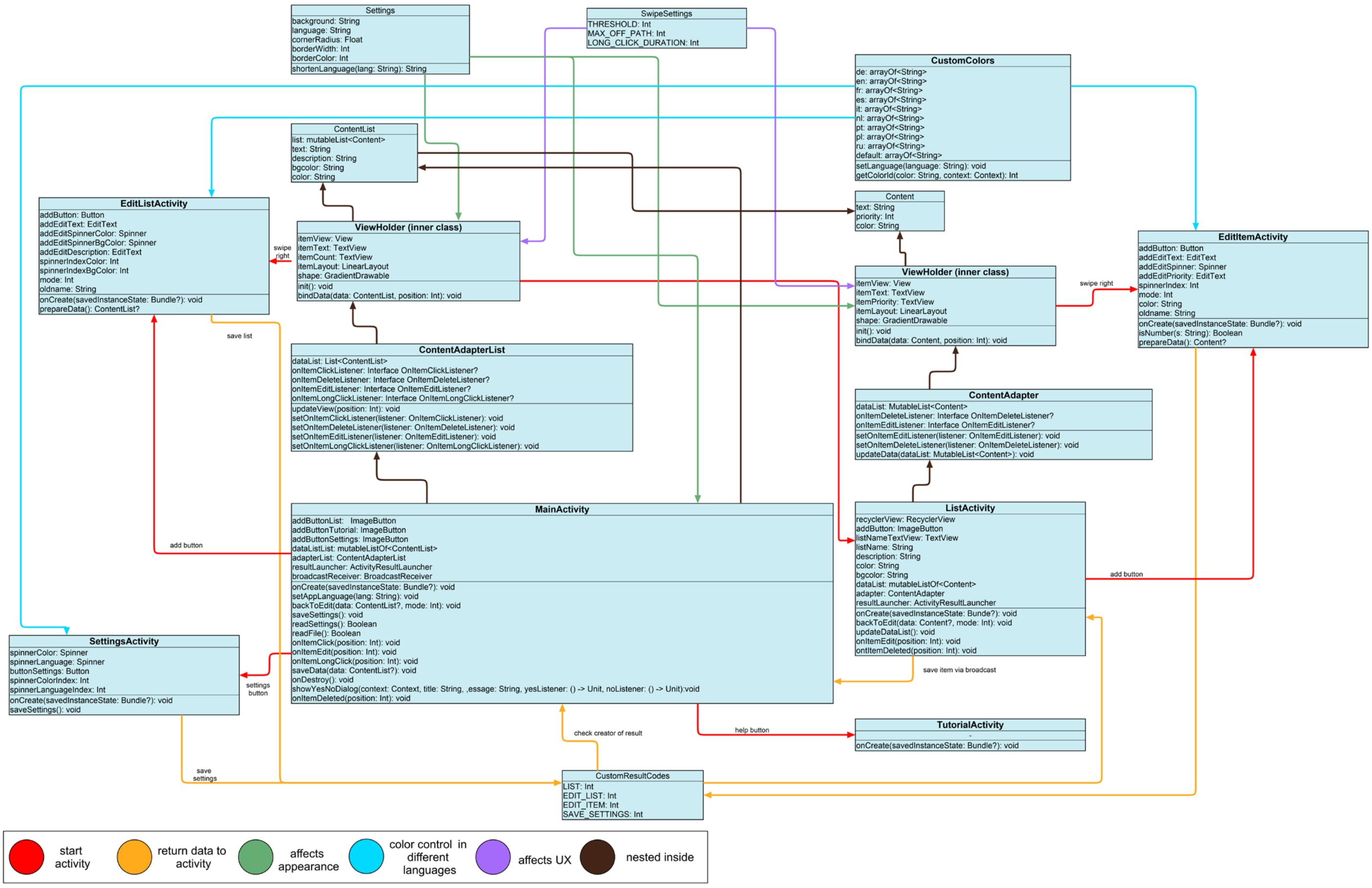
Device Information (Tested):

Category:	Phone
Name:	Pixel 4
API:	34

App Information:

App Version:	1.0.0
Languages Supported:	English, German, French, Spanish, Italian, Dutch, Portuguese, polish and Russian
Creators:	Daniel Hagemann (inf3459@hs-worms.de), Artem Gaus (inf3916@hs-worms.de)

Functions


Settings
background: String
language: String
cornerRadius: Float
borderWidth: Int
borderColor: Int
shortenLanguage(lang: String): String
SwipeSettings
THRESHOLD: Int
MAX_OFF_PATH: Int
LONG_CLICK_DURATION: Int
CustomColors
de: arrayOf<String>
en: arrayOf<String>
fr: arrayOf<String>
es: arrayOf<String>
it: arrayOf<String>
nl: arrayOf<String>
pt: arrayOf<String>
pl: arrayOf<String>
ru: arrayOf<String>
default: arrayOf<String>
setLanguage(language: String): void
getColorId(color: String, context: Context): Int
ContentList
list: mutableList<Content>
text: String
description: String
bgcolor: String
color: String
Content
text: String
priority: Int
color: String
EditListActivity
addButton: Button
addEditText: EditText
addEditSpinnerColor: Spinner
addEditSpinnerBgColor: Spinner
addEditDescription: EditText
spinnerIndexColor: Int
spinnerIndexBgColor: Int
mode: Int
oldname: String
onCreate(savedInstanceState: Bundle?): void
prepareData(): ContentList?
swipe right
ViewHolder (inner class)
itemView: View
itemText: TextView
itemCount: TextView
itemLayout: LinearLayout
shape: GradientDrawable
init(): void
bindData(data: ContentList, position: Int): void
ViewHolder (inner class)
itemView: View
itemText: TextView
itemPriority: TextView
itemLayout: LinearLayout
shape: GradientDrawable
init(): void
bindData(data: Content, position: Int): void
swipe right
EditItemActivity
addButton: Button
addEditText: EditText
addEditSpinner: Spinner
addEditPriority: EditText
spinnerIndex: Int
mode: Int
color: String
oldname: String
onCreate(savedInstanceState: Bundle?): void
isNumber(s: String): Boolean
prepareData(): Content?
save list
ContentAdapterList
dataList: List<ContentList>
onItemClickListener: Interface OnItemClickListener?
onItemDeleteListener: Interface OnItemDeleteListener?
onItemEditListener: Interface OnItemEditListener?
onItemLongClickListener: Interface OnItemLongClickListener?
updateView(position: Int): void
setOnItemClickListener(listener: OnItemClickListener): void
setOnItemDeleteListener(listener: OnItemDeleteListener): void
setOnItemEditListener(listener: OnItemEditListener): void
setOnItemLongClickListener(listener: OnItemLongClickListener): void
ContentAdapter
dataList: MutableList<Content>
onItemDeleteListener: Interface OnItemDeleteListener?
onItemEditListener: Interface OnItemEditListener?
setOnItemEditListener(listener: OnItemEditListener): void
setOnItemDeleteListener(listener: OnItemDeleteListener): void
updateData(dataList: MutableList<Content>): void
add button
MainActivity
addButtonList: ImageButton
addButtonTutorial: ImageButton
addButtonSettings: ImageButton
dataListList: mutableListOf<ContentList>
adapterList: ContentAdapterList
resultLauncher: ActivityResultLauncher
broadcastReceiver: BroadcastReceiver
onCreate(savedInstanceState: Bundle?): void
setAppLanguage(lang: String): void
backToEdit(data: ContentList?, mode: Int): void
saveSettings(): void
readSettings(): Boolean
readFile(): Boolean
onItemClick(position: Int): void
onItemEdit(position: Int): void
onItemLongClick(position: Int): void
saveData(data: ContentList?): void
onDestroy(): void
showYesNoDialog(context: Context, title: String, ,essage: String, yesListener: () -> Unit, noListener: () -> Unit):void
onItemDeleted(position: Int): void
ListActivity
recyclerView: RecyclerView
addButton: ImageButton
listNameTextView: TextView
listName: String
description: String
color: String
bgcolor: String
dataList: mutableListOf<Content>
adapter: ContentAdapter
resultLauncher: ActivityResultLauncher
onCreate(savedInstanceState: Bunde?): void
backToEdit(data: Content?, mode: Int): void
updateDataList(): void
onItemEdit(position: Int): void
ontItemDeleted(position: Int): void
add button
save item via broadcast
SettingsActivity
spinnerColor: Spinner
spinnerLanguage: Spinner
buttonSettings: Button
spinnerColorIndex: Int
spinnerLanguageIndex: Int
onCreate(savedInstanceState: Bundle?): void
saveSettings(): void
settings button
check creator of result
help button
TutorialActivity
-
onCreate(savedInstanceState: Bundle?): void
save settings
CustomResultCodes
LIST: Int
EDIT_LIST: Int
EDIT_ITEM: Int
SAVE_SETTINGS: Int
start activity
return data to activity
affects appearance
color control in different languages
affects UX
nested inside

For more details of code inside of the functions, check comments in code

CLASSES:

Class Content

Usage	Storing Item data from a list
Subclass of	Parcelable (and its traditional implementation)
Attributes	text: String priority: Int color: String
Functions	Only functions from Parcelable
URL to this	https://medium.com/@shahnimesh1992/parcelable-in-kotlin-be42b9f55db3

Class ContentList

Usage	Storing List data from main mutable list
Subclass of	Parcelable (and its traditional implementation)
Attributes	list: MutableList<Content> text: String description: String bgcolor: String color: String
Functions	Only functions from Parcelable
URL to this	https://medium.com/@shahnimesh1992/parcelable-in-kotlin-be42b9f55db3

class ContentAdapter

Usage	Manages every ViewHolder of RecyclerView and binds Data to them
Subclass of	RecyclerView.Adapter
Attributes	dataList: MutableList<Content> onItemDeleteListener: Interface OnItemDeleteListener? usage of Interface inside Adapter possible by calling onItemDeleteListener onItemEditListener: Interface OnItemEditListener? usage of Interface inside Adapter possible by calling onItemEditListener
Functions	override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder Binds ViewHolder to layout overwritten function from Adapter override fun onBindViewHolder(holder: ViewHolder, position: Int): Void bind ViewHolder to out dataList (Content) (Array index) overwritten function from Adapter override fun getItemCount(): Int

<table>
<tr>
<td></td>
<td>length of Adapter / RecyclerView = length of List (of Content, ContentList.list)
overwritten function from Adapter
fun setOnItemEditListener(listener: OnItemEditListener): Void
connect listener to edit listener (used in ListActivity)
fun setOnItemDeleteListener(listener: OnItemDeleteListener): Void
connect listener to delete listener (used in ListActivity)
fun updateData(dataList: MutableList<Content>): Void
updates whole adapter with new list data</td>
</tr>
<tr>
<td>Inner class ViewHolder</td>
<td>
<table>
<tr><td>Usage</td><td>manages ViewHolder (item | Content)</td></tr>
<tr><td>Subclass of</td><td>RecyclerView.ViewHolder</td></tr>
<tr><td>Attributes</td><td>itemView: View
itemText: TextView
itemPriority: TextView
itemLayout: LinearLayout
shape: GradientDrawable</td></tr>
<tr><td>Functions</td><td>Init
= constructor
init shape and color of ViewHolder
fun bindData(data: Content, position: Int): Void
bind ViewHolder to data (Content)
set OnTouchListener for tracking user actions (checks swiping)</td></tr>
</table>
</td>
</tr>
<tr>
<td>interfaces</td>
<td>
OnItemEditListener
<table>
<tr><td>functions</td><td>fun onItemEdit(position: Int): Void
function from List Activity
run ListActivity.onItemEdit()</td></tr>
</table>
OnItemDeleteListener
<table>
<tr><td>functions</td><td>fun onItemDeleted(position: Int): Void
function from List Activity
run ListActivity.onItemDeleted()</td></tr>
</table>
</td>
</tr>
</table>

class ContentAdapterList

<table>
<tr><td>Usage</td><td>Manages every ViewHolder of RecyclerView and binds Data to them (from list overview)</td></tr>
<tr><td>Subclass of</td><td>RecyclerView.Adapter</td></tr>
<tr><td>Attributes</td><td>dataList: MutableList<ContentList>
onItemDeleteListener: Interface OnItemDeleteListener? usage of Interface inside Adapter possible by calling onItemDeleteListener
onItemEditListener: Interface OnItemEditListener? usage of Interface inside Adapter possible by calling onItemEditListener
onItemClickListener: Interface OnItemClickListener? usage of Interface inside Adapter possible by calling onItemClickListener
onItemLongClickListener: Interface OnItemLongClickListener? usage of Interface inside Adapter possible by calling onItemLongClickListener</td></tr>
<tr><td>Functions</td><td>override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder
Binds ViewHolder to layout
overwritten function from Adapter
override fun onBindViewHolder(holder: ViewHolder, position: Int): Void
bind ViewHolder to out dataList (Content) (Array index)
overwritten function from Adapter
override fun getItemCount(): Int
length of Adapter
overwritten function from Adapter

fun setOnItemClickListener(listener: OnItemClickListener): Void
connect listener to click listener (used in MainActivity)
fun setOnItemDeleteListener(listener: OnItemDeleteListener): Void
connect listener to delete listener (used in MainActivity)
fun setOnItemEditListener(listener: OnItemEditListener): Void
connect listener to edit listener (used in MainActivity)
fun setOnItemLongClickListener(listener: OnItemLongClickListener)
connect listener to long click listener (used in MainActivity)
fun updateView(position: Int): Void
updates adapter at position if item (in ContentList.list) changed via broadcast</td></tr>
<tr><td>inner class ViewHolder</td><td>
<table>
<tr><td>Usage</td><td>manages ViewHolder (item | Content)</td></tr>
<tr><td>Subclass of</td><td>RecyclerView.ViewHolder</td></tr>
<tr><td>Attributes</td><td>itemView: View
itemText: TextView
itemCount: TextView
itemLayout: LinearLayout
shape: GradientDrawable</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td><table><tr><td>Functions</td><td>init
= constructor
init shape and color of ViewHolder
fun bindData(data: ContentList, position: Int): Void
bind ViewHolder to data (ContentList)
set OnTouchListener for tracking user actions (checks swiping)</td></tr></table></td></tr>
<tr><td>interfaces</td><td>OnItemDeleteListener<table><tr><td>functions</td><td>fun onItemDeleted(position: Int): Void
function from MainActivity
run MainActivity.onItemDeleted()</td></tr></table>
OnItemClickListener<table><tr><td>functions</td><td>fun onItemClick(position: Int): Void
function from MainActivity
run MainActivity.onItemClick()</td></tr></table>
OnItemEditListener<table><tr><td>functions</td><td>fun onItemEdit(position: Int): Void
function from MainActivity
run MainActivity.onItemEdit()</td></tr></table>
OnItemLongClickListener<table><tr><td>functions</td><td>fun onItemLongClick(position: Int): Void
function from MainActivity
run MainActivity.onItemLongClick()</td></tr></table></td></tr>
</table>

class EditItemActivity

Usage	activity for editing a list item
Subclass of	AppCompatActivity
Attributes	addButton: Button addEditText: EditText addEditSpinner: Spinner addEditPriority: EditText spinnerIndex: Int mode: Int color: String oldname: String
Functions	override fun onCreate(savedInstanceState: Bundle?): Void = constructor of activity init screen, set colors, add functionality, bind data to buttons and spinner fun isNumber(s: String?): Boolean check if string is number

	instrumented test tested private fun prepareData(): Content? prepare final data for return in ListActivity return null on failure

class EditListactivity

Usage	activity for editing a list
Subclass of	AppCompatActivity
Attributes	addButton: Button addEditText: EditText addEditSpinnerColor: Spinner addEditSpinnerBgColor: Spinner addEditDescription: EditText spinnerIndexColor: Int spinnerIndexBgColor: Int mode: Int oldname: String
Functions	override fun onCreate(savedInstanceState: Bundle?): Void = constructor of activity init screen, set colors, add functionality, bind data to buttons and spinner private fun prepareData(): ContentList? prepare final data for return in MainActivity return null on failure

class ListActivity

Usage	activity for showing all items in a selected list
Subclass of	AppCompatActivity
Used interfaces	ContentAdapter.OnItemDeleteListener, ContentAdapter.OnItemEditListener
Attributes	recyclerView: RecyclerView addButton: ImageButton listNameTextView: TextView listName: String description: String color: String bgcolor: String dataList: MutableList<Content> adapter: ContentAdapter resultLauncher: ActivityResultLauncher onActivityResult is deprecated -- new way of handling result value get data from EditItemActivity
Functions	override fun onCreate(savedInstanceState: Bundle?): Void = constructor of activity init screen, set colors, add functionality set adapter listener setBroadcast if back button is pressed private fun backToEdit(data: Content?, mode: Int): Void switch to EditItemActivity with Content data private fun updateDataList(): Void sort list by priority and send sorted list via broadcast to MainActivity for saving without leaving current activity override fun onItemEdit(position: Int): Void called from interface -- event listener -- ContentAdapter start EditItemActivity with data override fun onItemDeleted(position: Int): Void called from interface -- event listener -- ContentAdapter removes item (Content) from list - without asking

class MainActivity

Usage	main activity -> shows a list of selectable lists
Subclass of	AppCompatActivity
Used interfaces	ContentAdapterList.OnItemClickListener, ContentAdapterList.OnItemDeleteListener, ContentAdapterList.OnItemEditListener, ContentAdapterList.OnItemLongClickListener
Attributes	addButtonList: ImageButton addButtonTutorial: ImageButton addButtonSettings: ImageButton recyclerViewList: RecyclerView dataListList: MutableList<ContentList> adapterList: ContentAdapterList resultLauncher: ActivityResultLauncher onActivityResult is deprecated -- new way of handling result value get Data from EditListactivity and SettingsActivity and ListActivity broadcastReceiver: object BroadcastReceiver similar to resultLauncher / onActivityResult, receive data without changing activity get data from ListActivity, if item is added (prohibits usage of global variable)
Functions	override fun onCreate(savedInstanceState: Bundle?): Void = constructor of activity init screen, set colors, add functionality set adapter listener setBroadcast to listening read settings / data set language private fun setAppLanguage(languageCode: String): Void changes language package uses updateConfiguration() which is deprecated private fun backToEdit(data: ContentList?, mode: Int): Void switch to EditListActivity with some data private fun saveSettings(): Void saves settings from global Settings in local file called “settings.json” in Json-format private fun readSettings(): Boolean read settings from “settings.json” in global Settings return false on error | true on success private fun readFile(): Boolean read data from “data.json” in local dataListList return false on error | true on success override fun onItemClick(position: Int): Void called from interface -- event listener -- ContentAdapterList start ListActivity with data

	override fun onItemLongClick(position: Int): Void called from interface -- event listener -- ContentAdapterList show description of list override fun onItemEdit(position: Int): Void called from interface -- event listener -- ContentAdapterList edit existing list fun saveData(data: ContentList?): Void saves list data from local dataListList in local file called “data.json” in Json-format override fun onDestroy(): Void safe unregister broadcast close App private fun showYesNoDialog(context: Context, title: String, message: String, yesListener: () ->Unit, noListener: () -> Unit): Void customizable dialog box using AlertDialog override fun onItemDeleted(position: Int): Void called from interface -- event listener -- ContentAdapterList asking to delete list and deleting

class SettingsActivity

Usage	activity for changing Settings (main background color and language)
Subclass of	AppCompatActivity
Attributes	spinnerColor: Spinner spinnerLanguage: Spinner buttonSettings: Button spinnerColorIndex: Int spinnerLanguageIndex: Int
Functions	override fun onCreate(savedInstanceState: Bundle?): Void = constructor of activity init screen, add functionality private fun saveSettings(): Void save settings return settings to MainActivity

class TutorialActivity

Usage	activity to view a help screen no functionality only viewing
Subclass of	AppCompatActivity
Functions	override fun onCreate(savedInstanceState: Bundle?): Void = constructor of activity init screen

GLOBAL OBJECTS

object CustomColors

Usage	control every color by using a global color system
Attributes	de: Array<String> en: Array<String> es: Array<String> fr: Array<String> it: Array<String> nl: Array<String> pt: Array<String> pl: Array<String> ru: Array<String> tr: Array<String> default: Array<String> Country code as Arrays because every language has its own word for the same color
Functions	fun setLanguage(language: String): Void set default to variable of short language code (de, en, ...) by passing equal string fun getColorId(color: String, context: Context): Int uses default color ids in "./values" (equal to English names) instrumented test tested returns color id | 0 on error

object CustomResultCodes

Usage	Result Codes from Activity were not suitable, so it’s better to ONLY use my own result codes. custom result codes because default codes don’t fit in meaning
Attributes	LIST: Int EDIT_LIST: Int EDIT_ITEM: Int SAVE_SETTINGS: Int

object Settings

Usage	global settings of ViewHolder appearance, language, background color of main screen
Attributes	background: String language: String cornerRadius: Float borderWidth: Int borderColor: Int
Functions	fun shortenLanguage(lang: String): String converts long languages to its shorten form Unit Test tested return “en” on error

object SwipeSettings

Usage	Used to manage SwipeSettings of ViewHolder better
Attributes	THRESHOLD: Int MAX_OFF_PATH: Int LONG_CLICK_DURATION: Long

Design

An attempt was made to adhere to the design rules of Dieter Rams.

"Less is more" came to the fore.

The user should not be overwhelmed by too many buttons. He should act intuitively, clean, and smooth.

Tests

Unit Tests

testShortLanguage():

Tests function shortLanguage(lang: String) in object Settings.

There are no more Unit Test possible.

Because you need a running instance to test functions in an activity. And then you need an Instrumented Test.

Instrumented Tests

testIsNumber()

Tests function isNumber(s: String) in class EditItemActivity.

testGetColorId()

Tests function getColorId(color: String) in object CustomColors.

There are no more Instrumented Tests because the data could be overwritten or deleted.

So, it is safer to not test these and keep the state as it is.

Possible Updates

- more languages
- more customizable
- notifications