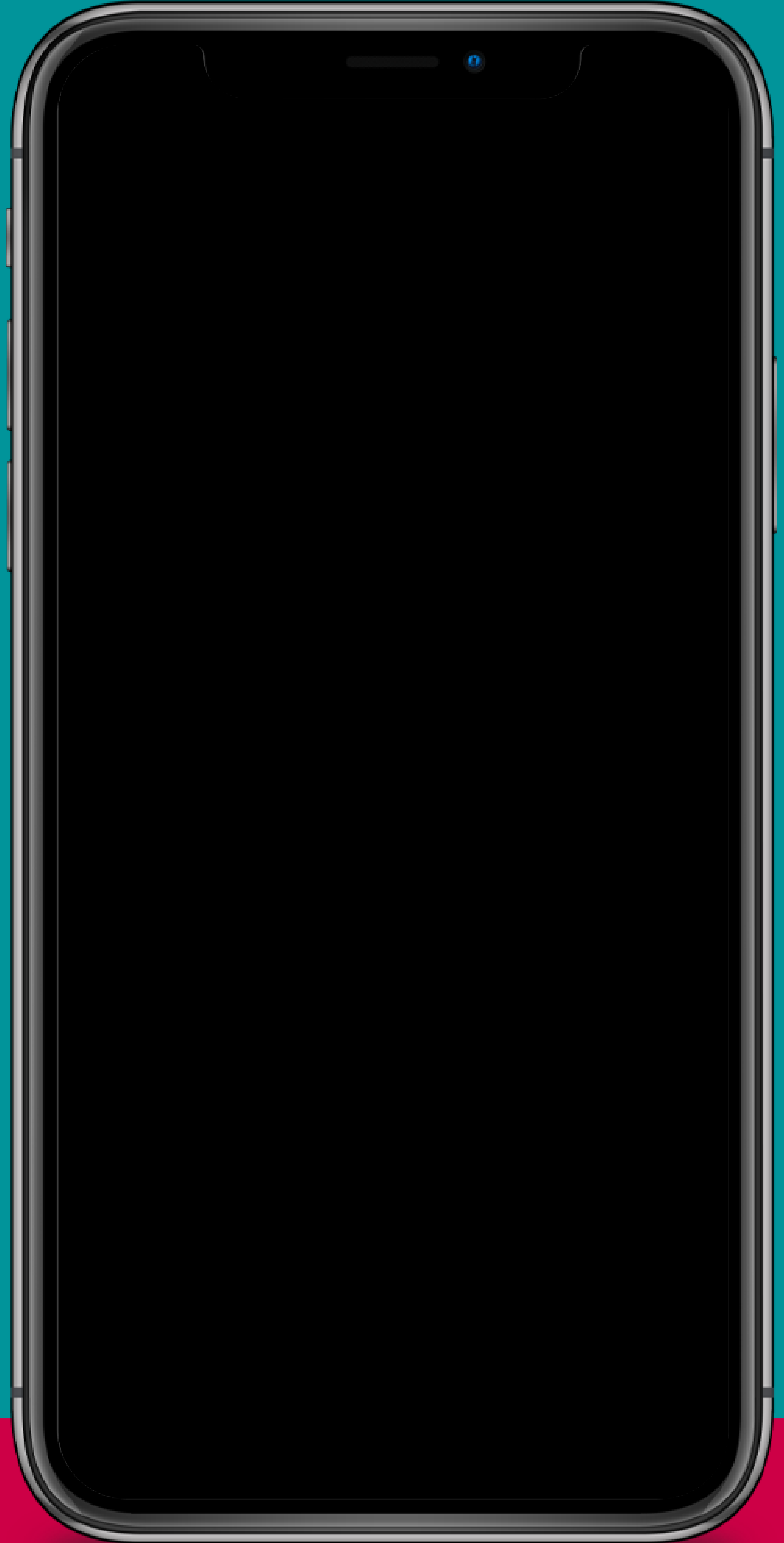
YALOVA UNIVERSITY

Android Programming with Huawei Mobile Services

Berk Ozyurt
Mehmet Yozgatli
Cengiz Toru



W8: ANDROID

- Lists
  - RecyclerView
  - ViewPager
- Data Persistence - Data Store
  - Shared Preferences
  - Data Store
  - SQLite
  - Room DB

Lists

https://developer.android.com/guide/topics/ui/layout/recyclerview

https://developer.android.com/reference/androidx/recyclerview/widget/ListAdapter

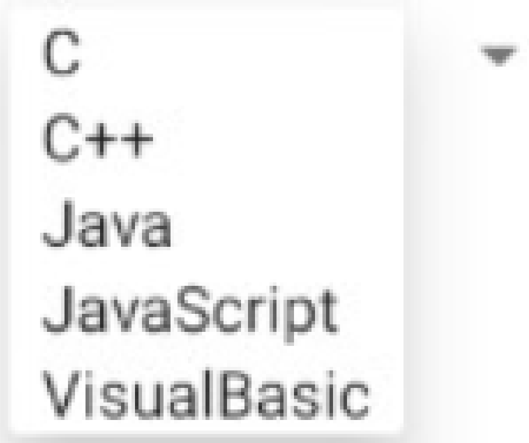
https://www.raywenderlich.com/21954410-speed-up-your-android-recyclerview-using-diffutil

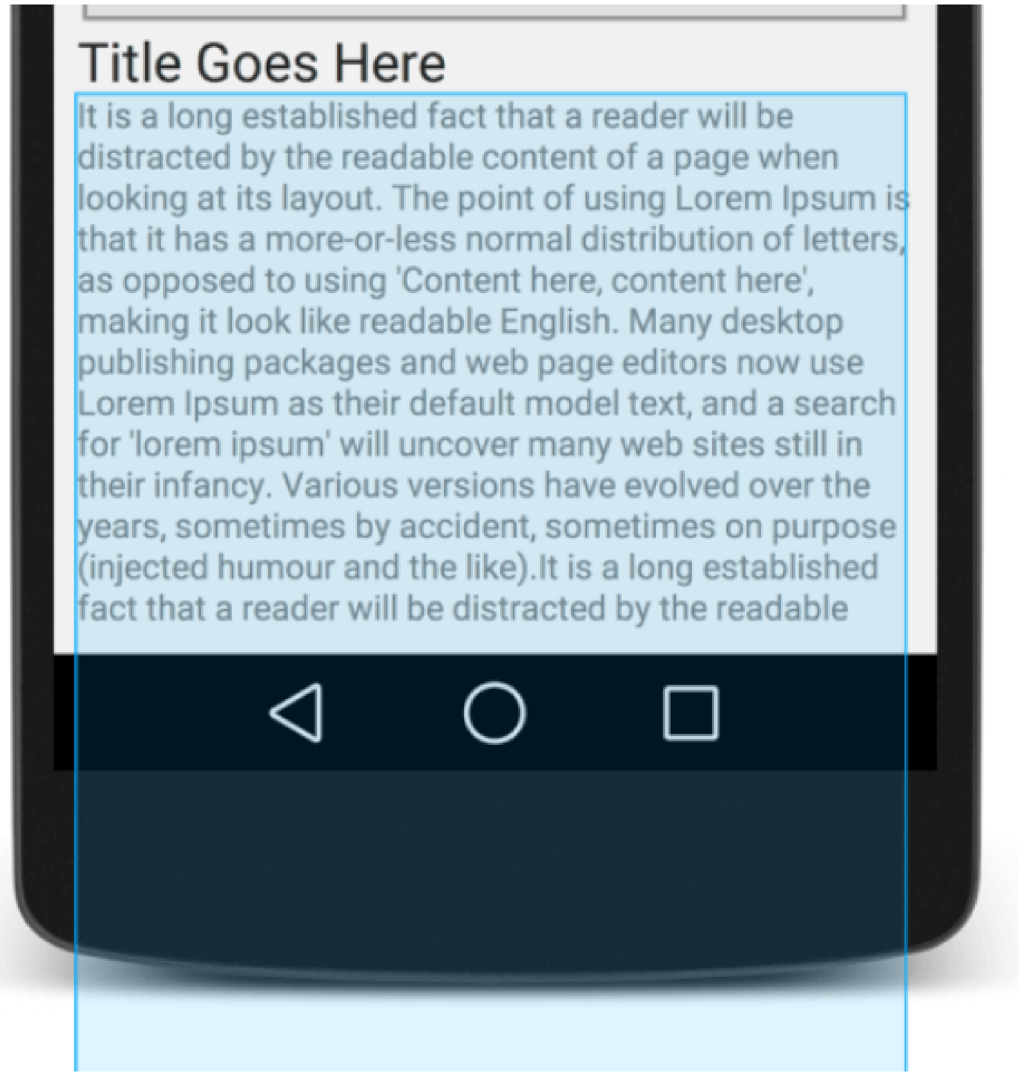
https://www.raywenderlich.com/1560485-android-recyclerview-tutorial-with-kotlin#toc-anchor-003

https://medium.com/geekculture/everything-you-should-know-to-create-a-recyclerview-3defdb660a2f

Listing Items into UI

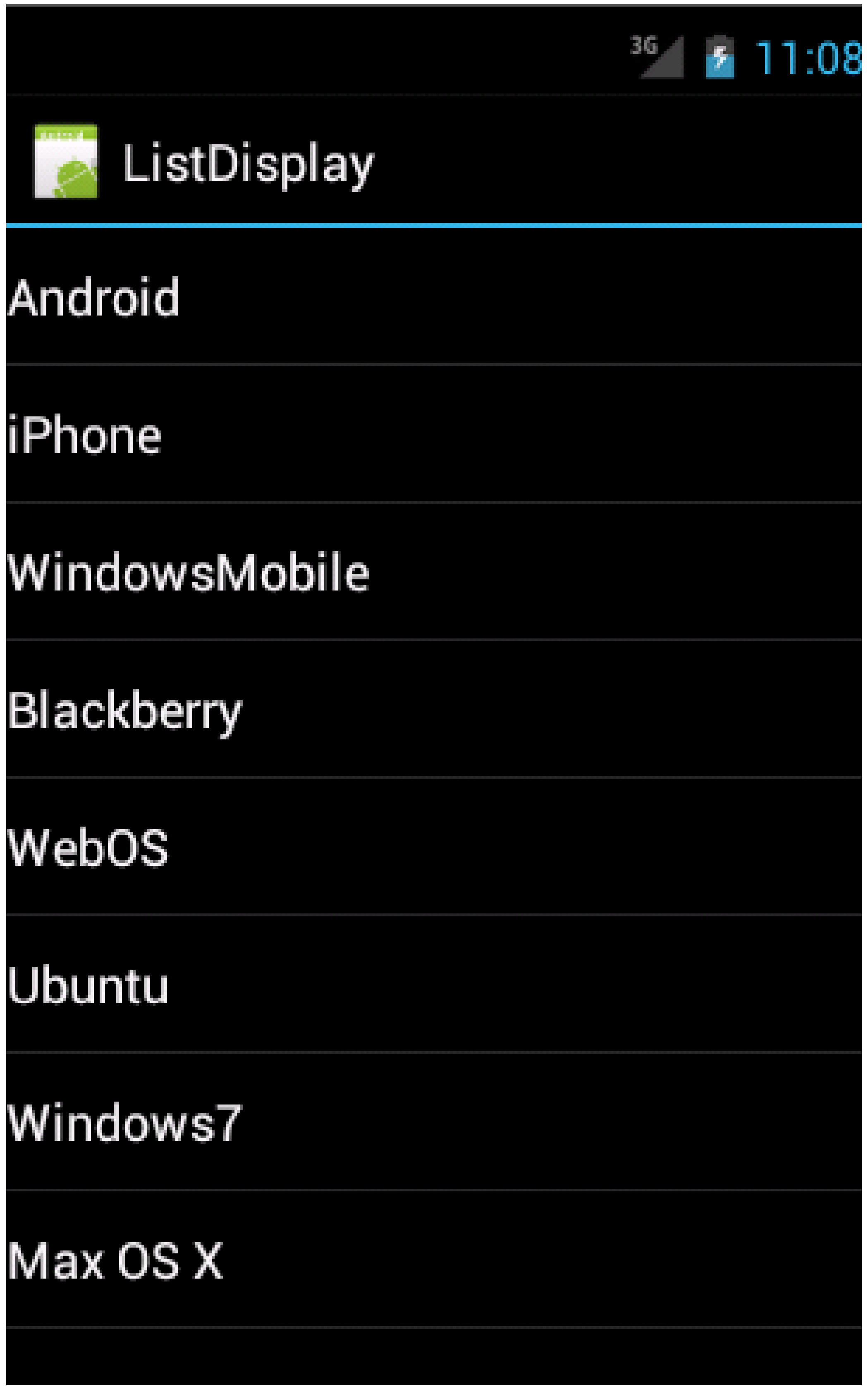
- ✓ Spinner - Drop Down List
- ✓ Scroll View
- ✓ List View
- ✓ RecyclerView





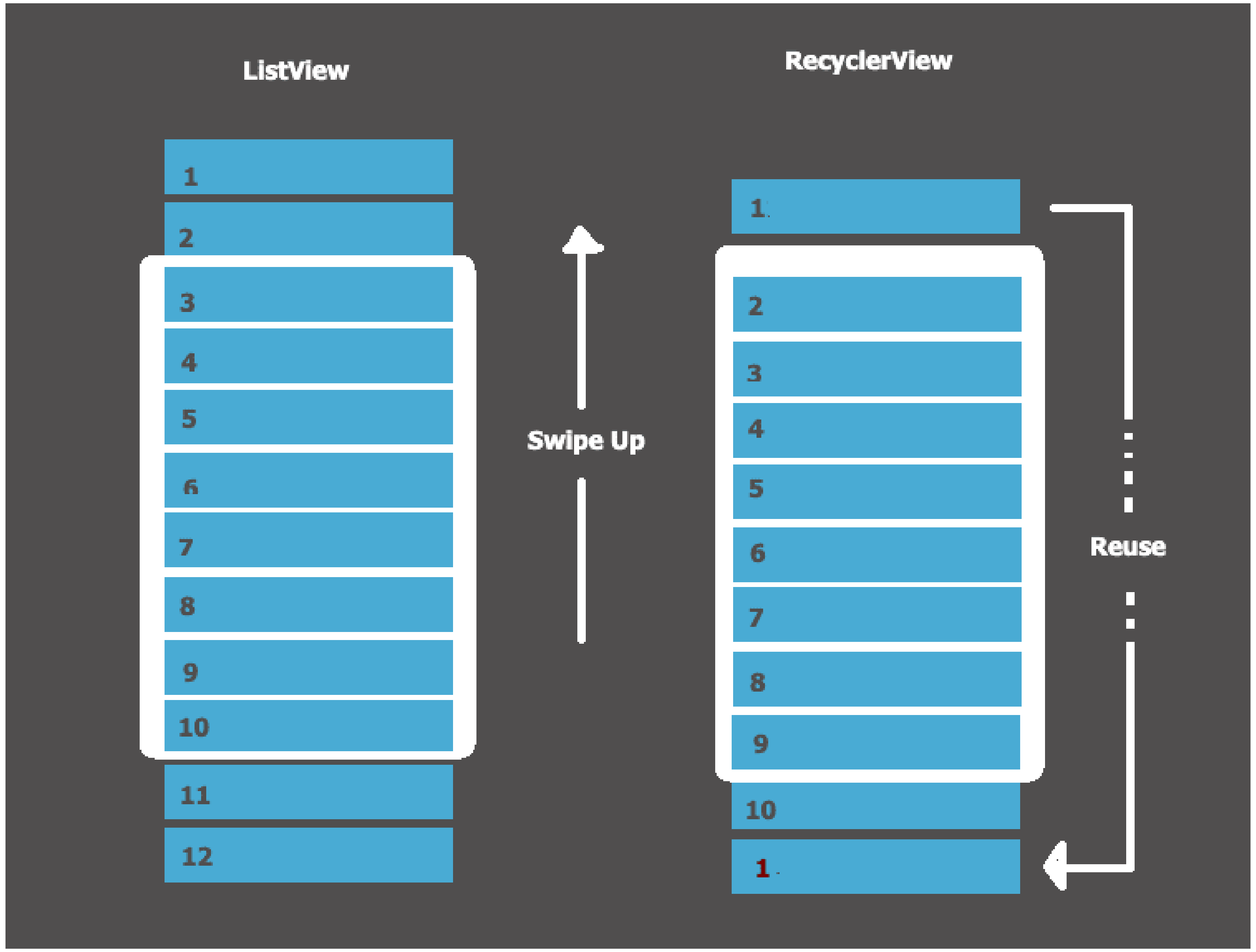
Title Goes Here

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).It is a long established fact that a reader will be distracted by the readable



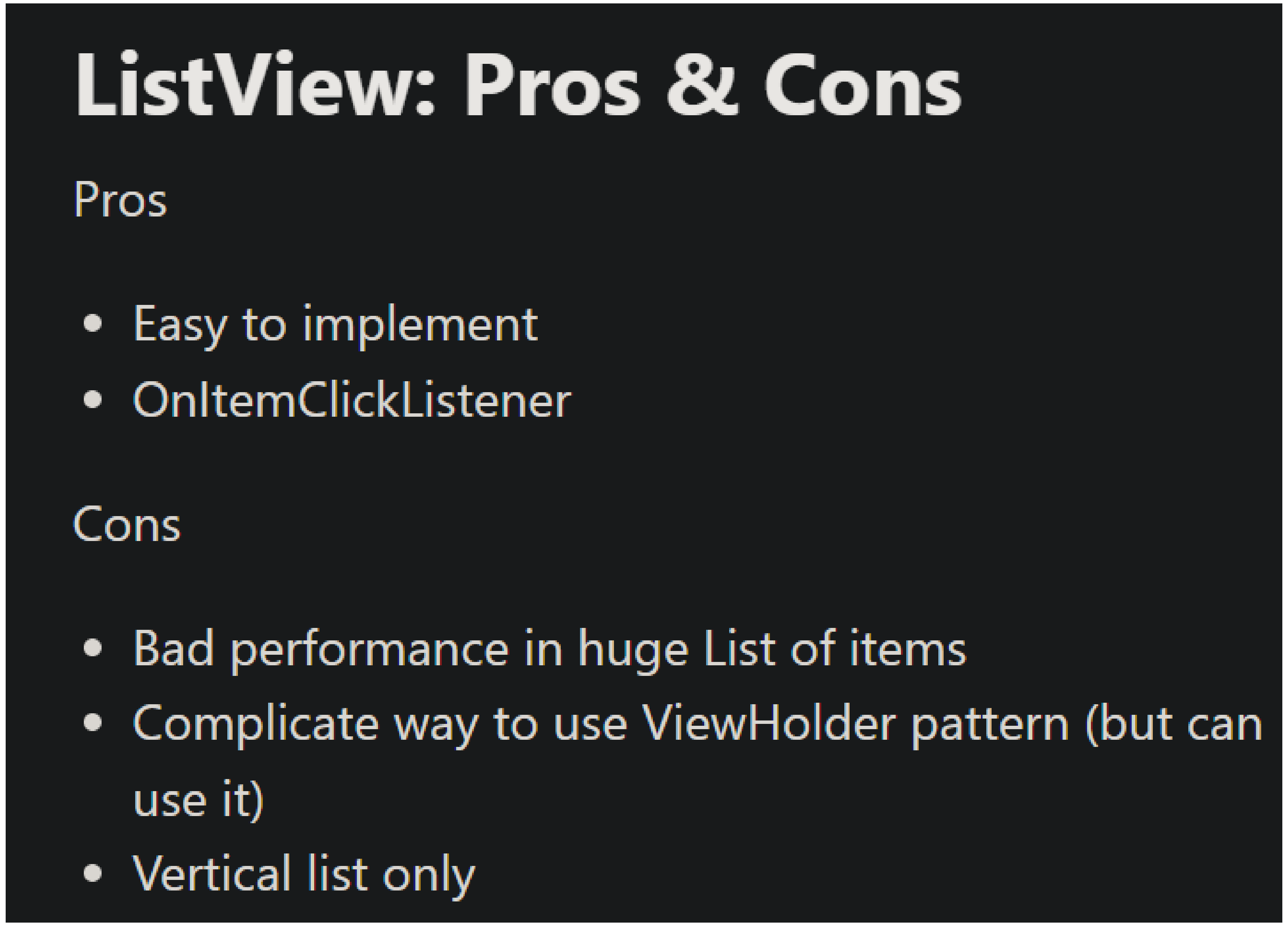
RecyclerView

We use the recyclerview to display large sets of data in UI. You supply the data and define how each item looks, and the RecyclerView library dynamically creates the elements when they're needed.



- ✓ Recycles individual elements
- ✓ Improves performance, reducing power consumption and app's responsiveness

RecyclerView vs ListView



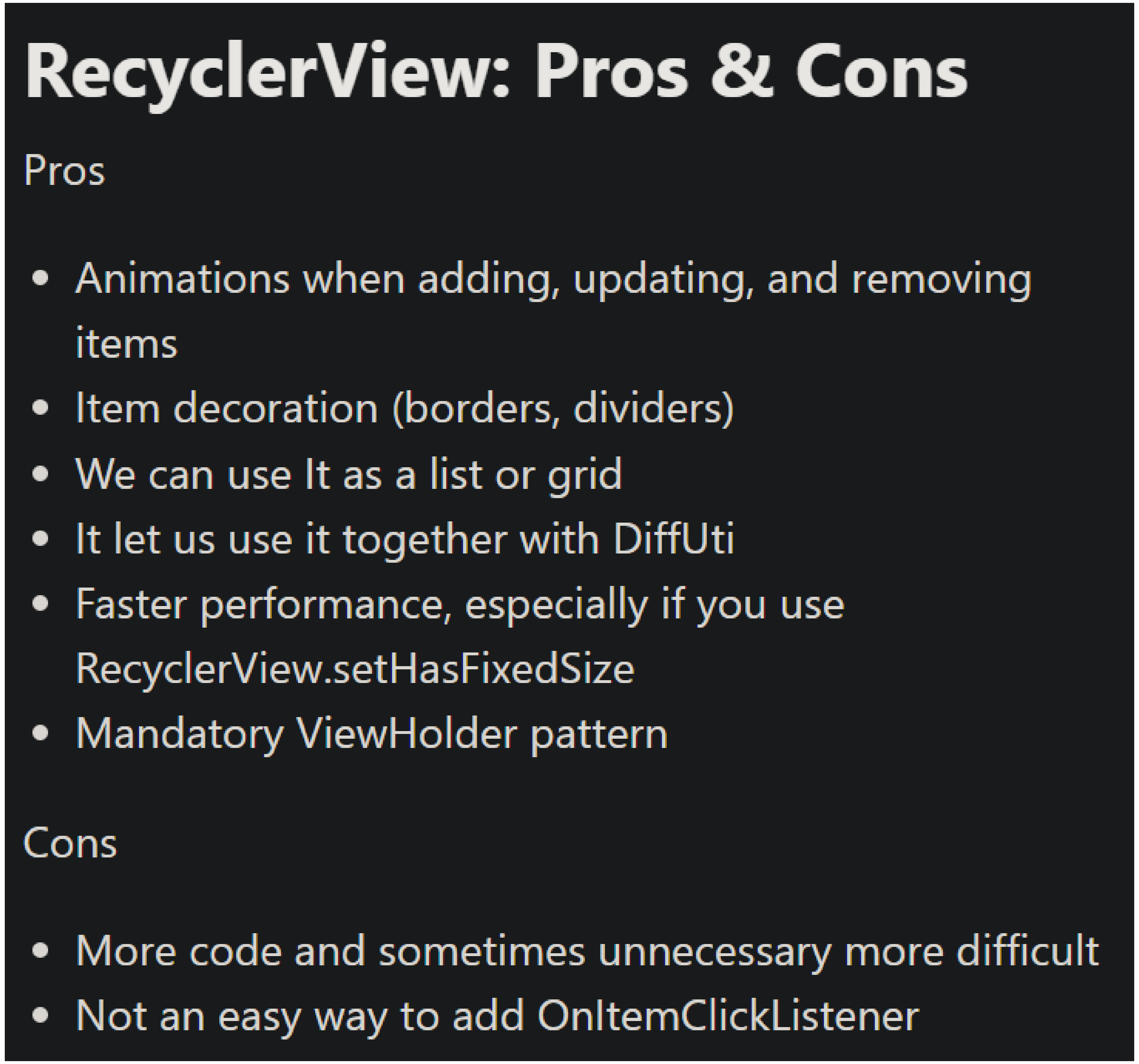
ListView: Pros & Cons

Pros

- Easy to implement
- OnItemClickListener

Cons

- Bad performance in huge List of items
- Complicate way to use ViewHolder pattern (but can use it)
- Vertical list only



RecyclerView: Pros & Cons

Pros

- Animations when adding, updating, and removing items
- Item decoration (borders, dividers)
- We can use It as a list or grid
- It let us use it together with DiffUti
- Faster performance, especially if you use RecyclerView.setHasFixedSize
- Mandatory ViewHolder pattern

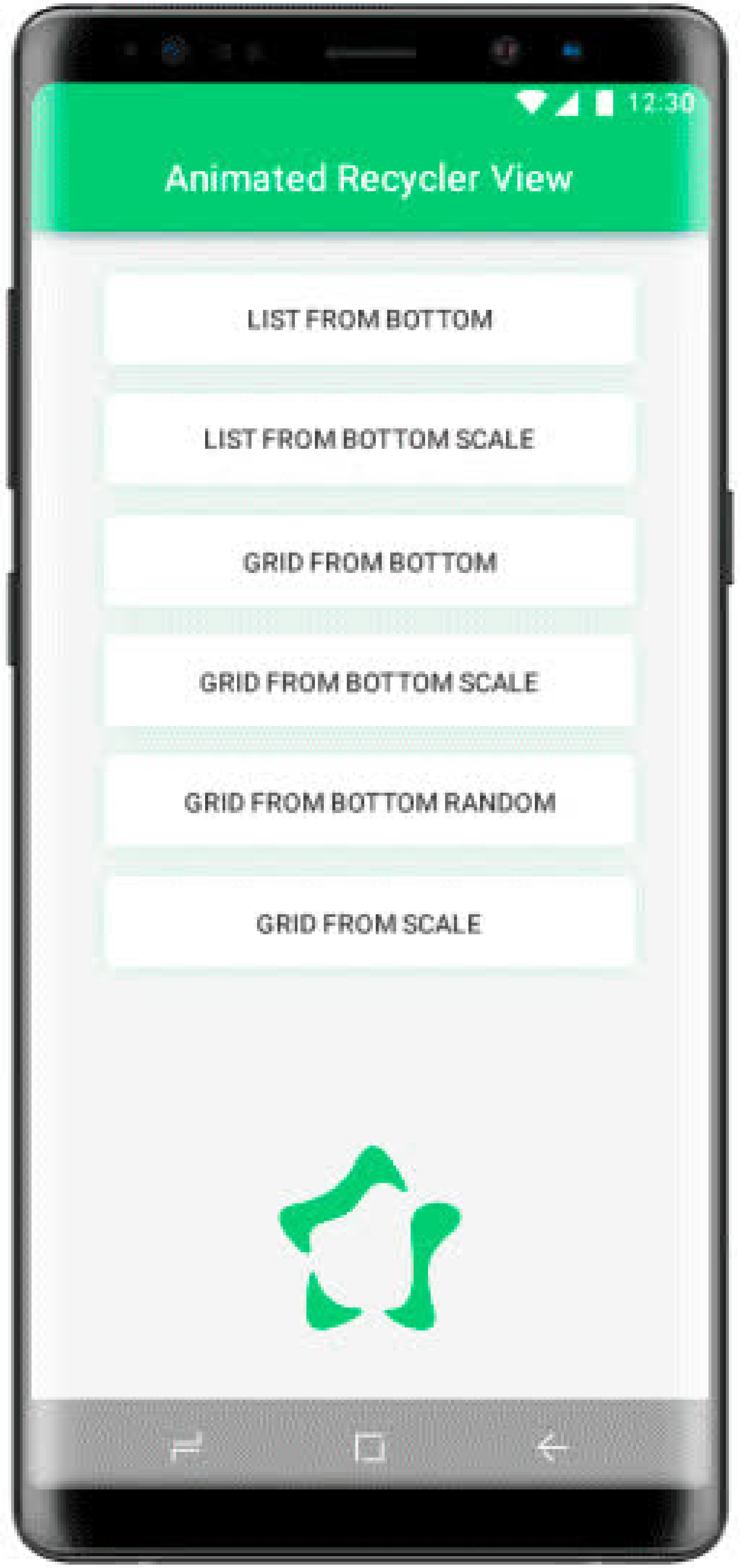
Cons

- More code and sometimes unnecessary more difficult
- Not an easy way to add OnItemClickListener

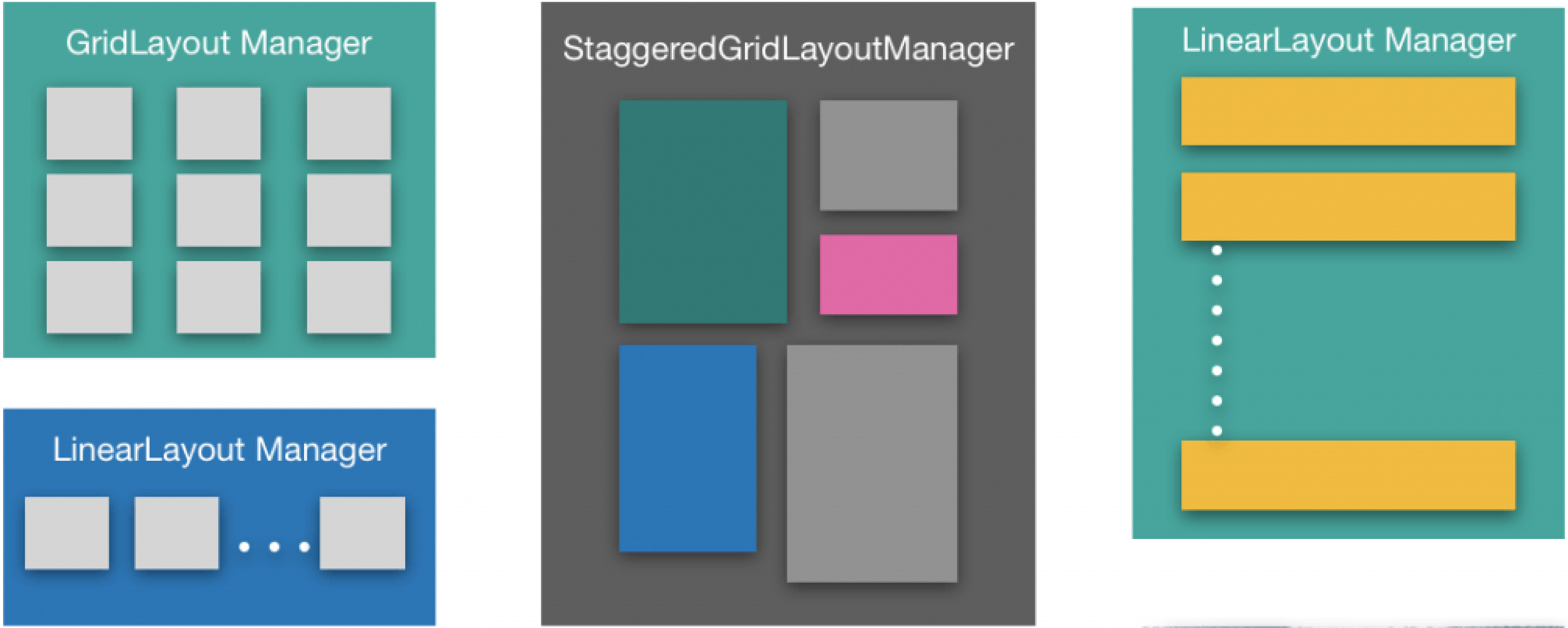
RecyclerView Steps

- ✓ Decide to how list will be look like; list, grid
- ✓ Create item data class
- ✓ Create item layout / design
- ✓ Create adapter class
- ✓ Decide to Layout Manager

  LinearLayoutManager, GridLayoutManager, StaggeredGridLayoutManager
- ✓ Set adapter to Recycler View
- ✓ Handle Clicks

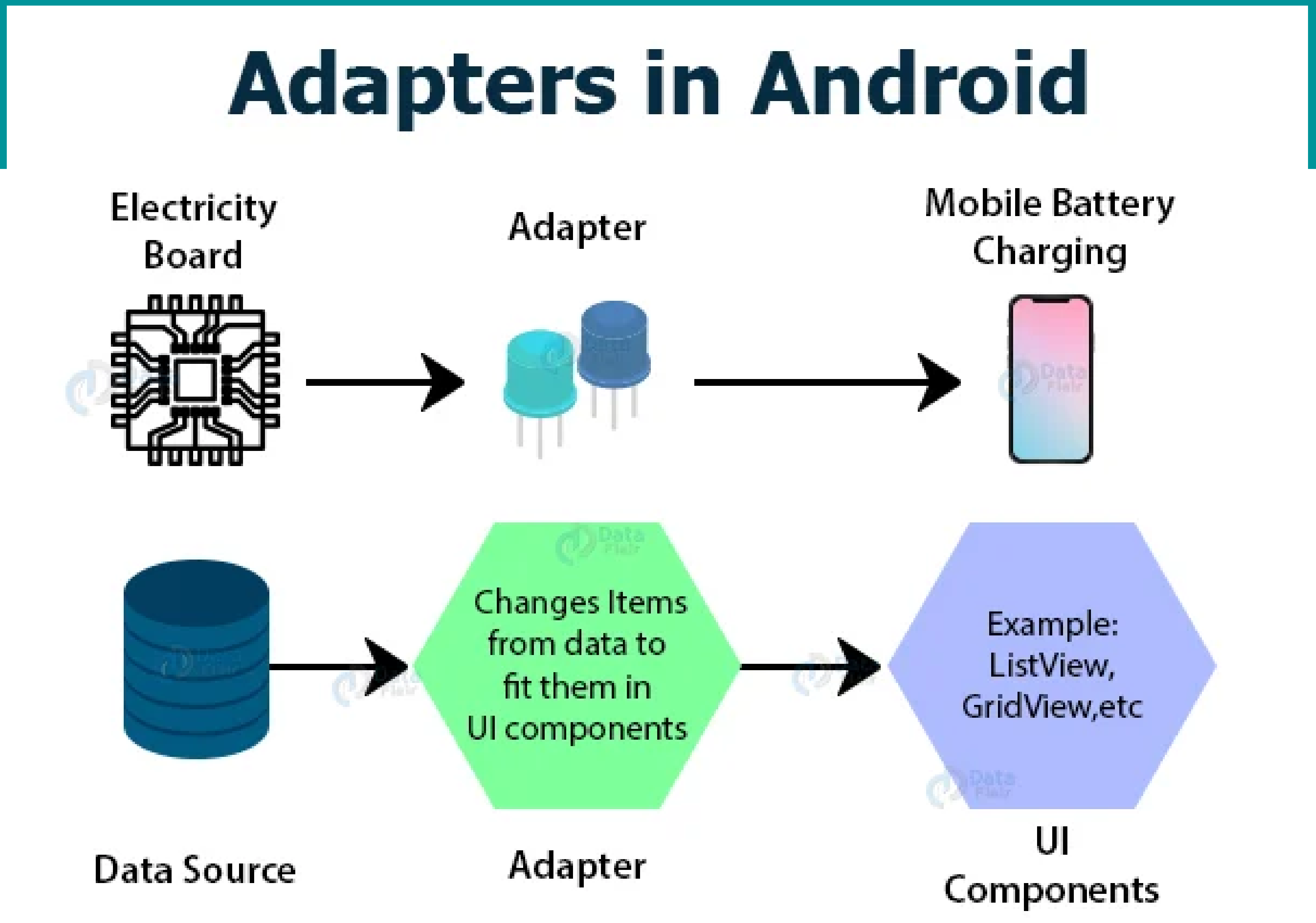


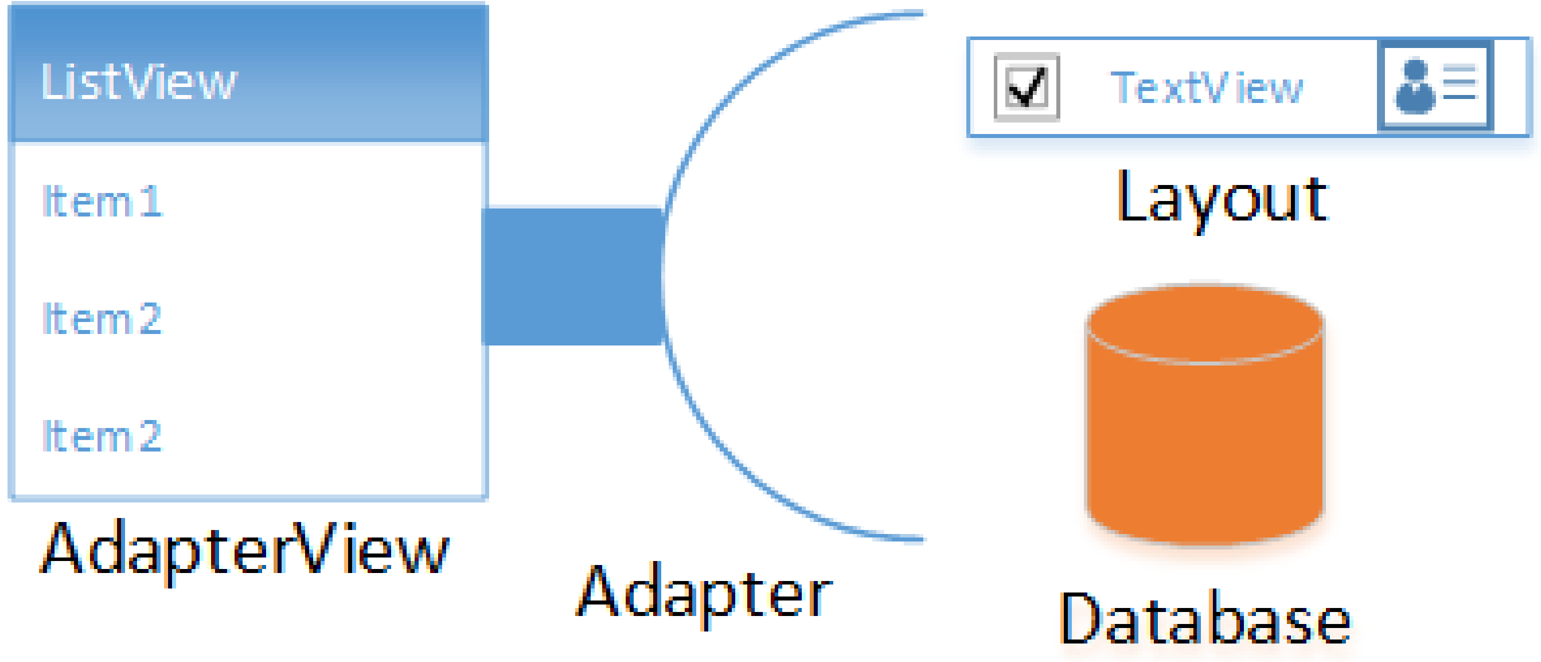
Layout Managers

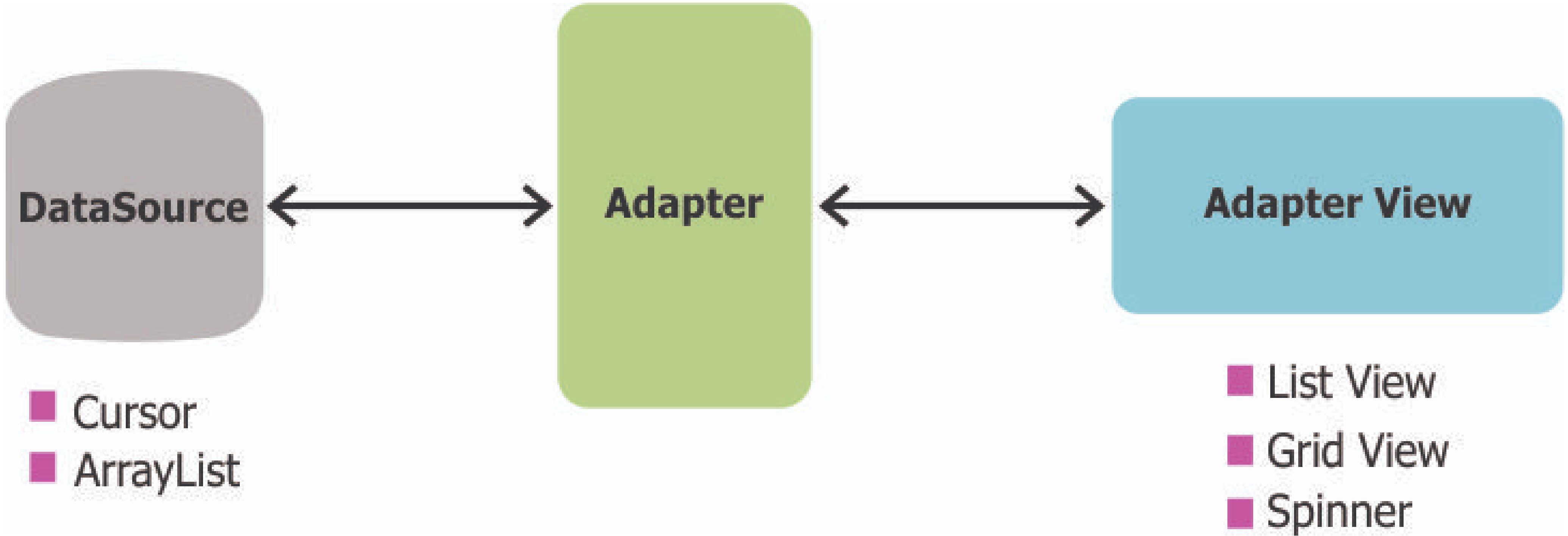


Adapters

We use adapters to populate the data into views. It also converts a data object into an individual list item View to be inserted and displayed to the user.



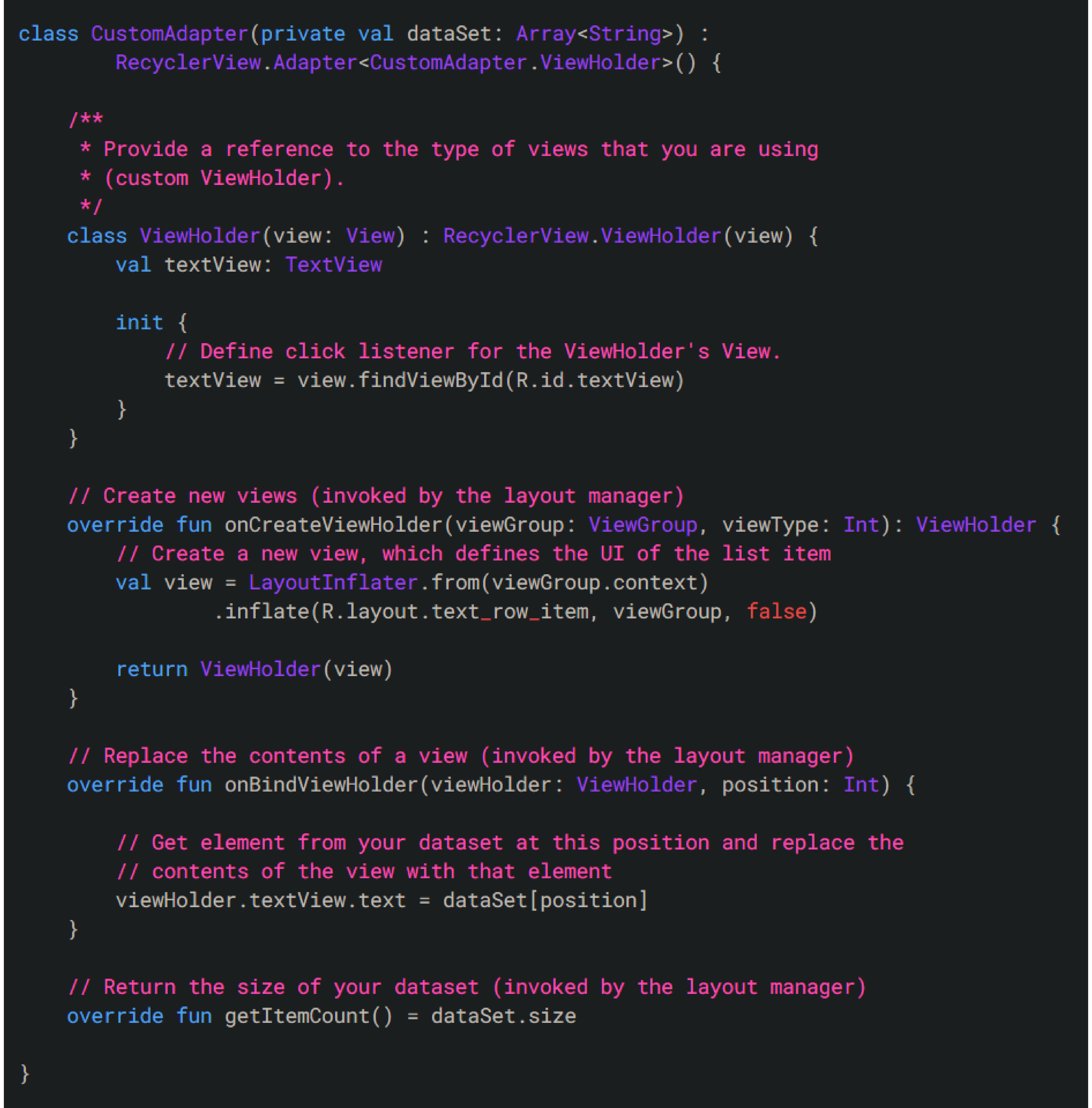




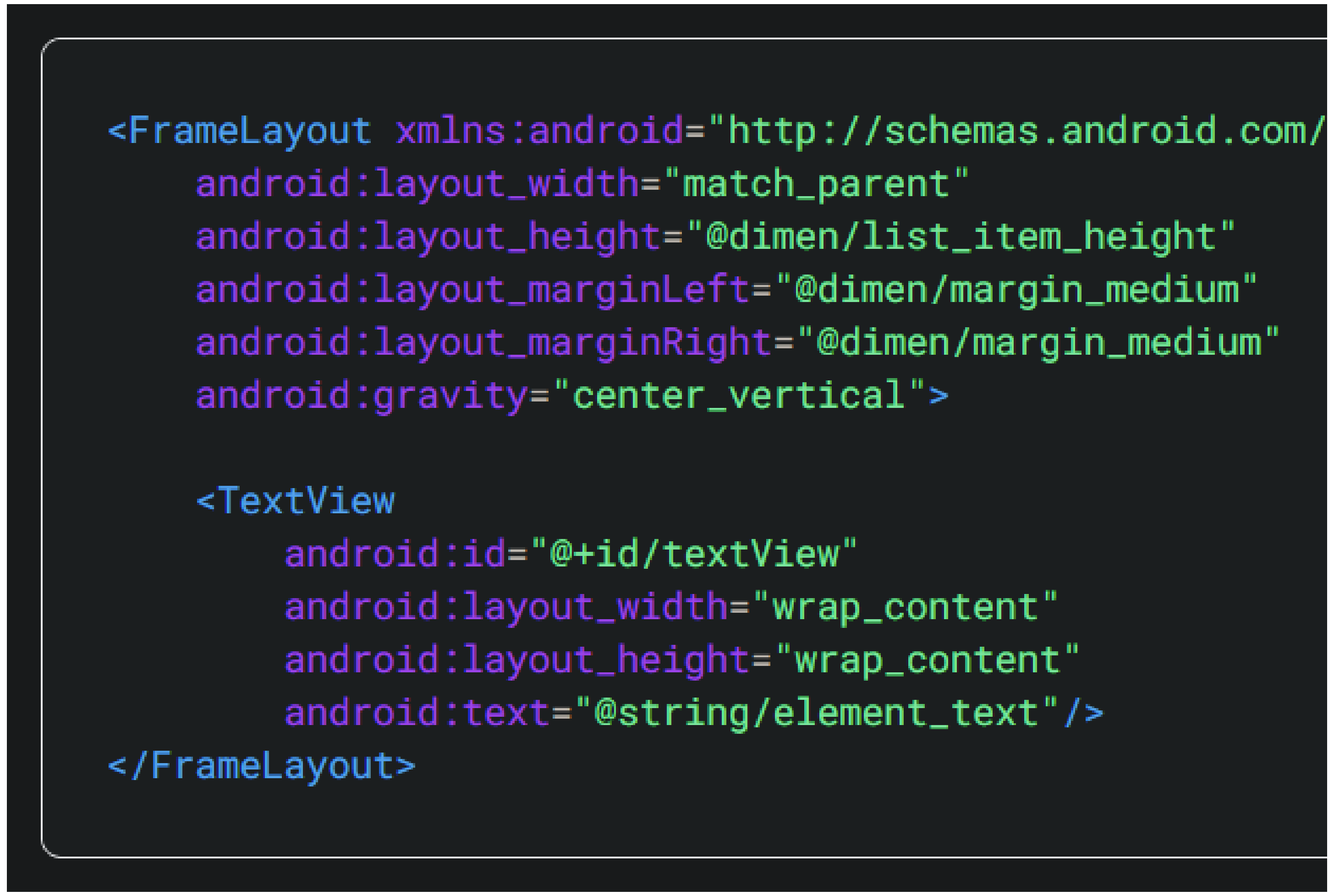
RecyclerView Adapter

- ✓ View Holder Class; a wrapper around a view, holds an object of an item in the list, provides the functionality of the list item, .
- ✓ onCreateViewHolder(); RecyclerView calls this method whenever it needs to create a new ViewHolder and its associated View
- ✓ onBindViewHolder(); RecyclerView calls this method to associate a ViewHolder with data. The method fetches the appropriate data and uses the data to fill in the view holder's layout.
- ✓ getItemCount(); RecyclerView calls this method to get the size of the data set
- ✓ DiffCallback object; Callback for calculating the diff between two non-null items in a list

RecyclerView Adapter Example



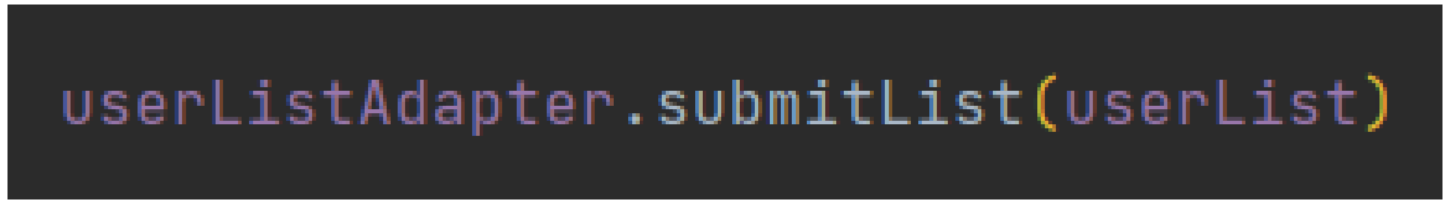
```kotlin
class CustomAdapter(private val dataSet: Array<String>) :
        RecyclerView.Adapter<CustomAdapter.ViewHolder>() {

    /**
     * Provide a reference to the type of views that you are using
     * (custom ViewHolder).
     */
    class ViewHolder(view: View) : RecyclerView.ViewHolder(view) {
        val textView: TextView

        init {
            // Define click listener for the ViewHolder's View.
            textView = view.findViewById(R.id.textView)
        }
    }

    // Create new views (invoked by the layout manager)
    override fun onCreateViewHolder(viewGroup: ViewGroup, viewType: Int): ViewHolder {
        // Create a new view, which defines the UI of the list item
        val view = LayoutInflater.from(viewGroup.context)
                .inflate(R.layout.text_row_item, viewGroup, false)

        return ViewHolder(view)
    }

    // Replace the contents of a view (invoked by the layout manager)
    override fun onBindViewHolder(viewHolder: ViewHolder, position: Int) {

        // Get element from your dataset at this position and replace the
        // contents of the view with that element
        viewHolder.textView.text = dataSet[position]
    }

    // Return the size of your dataset (invoked by the layout manager)
    override fun getItemCount() = dataSet.size

}
```



```xml
<FrameLayout xmlns:android="http://schemas.android.com/
    android:layout_width="match_parent"
    android:layout_height="@dimen/list_item_height"
    android:layout_marginLeft="@dimen/margin_medium"
    android:layout_marginRight="@dimen/margin_medium"
    android:gravity="center_vertical">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/element_text"/>
</FrameLayout>
```

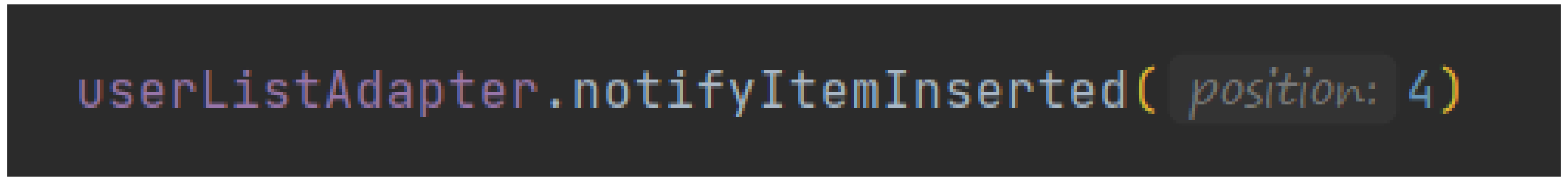
RecyclerView ListAdapter Example

- ✓ set list



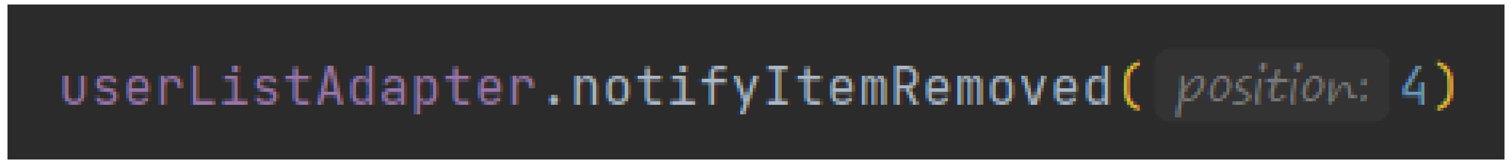
```
userListAdapter.submitList(userList)
```

- ✓ add item



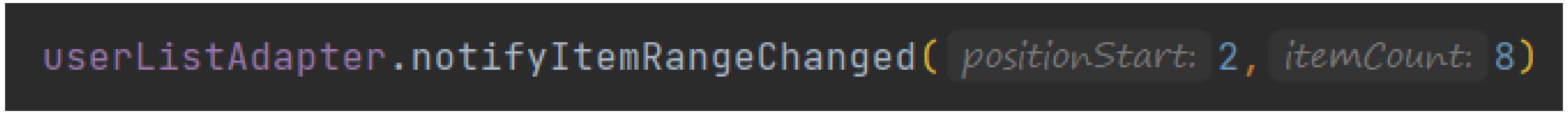
```
userListAdapter.notifyItemInserted( position: 4)
```

- ✓ remove item



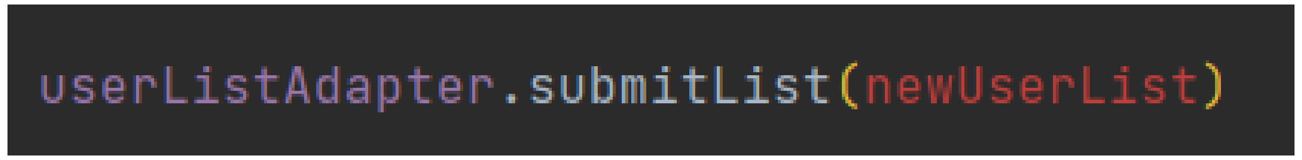
```
userListAdapter.notifyItemRemoved( position: 4)
```

- ✓ update a range of items

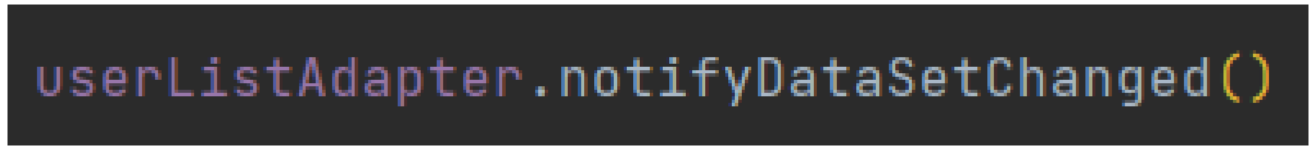


```
userListAdapter.notifyItemRangeChanged( positionStart: 2, itemCount: 8)
```

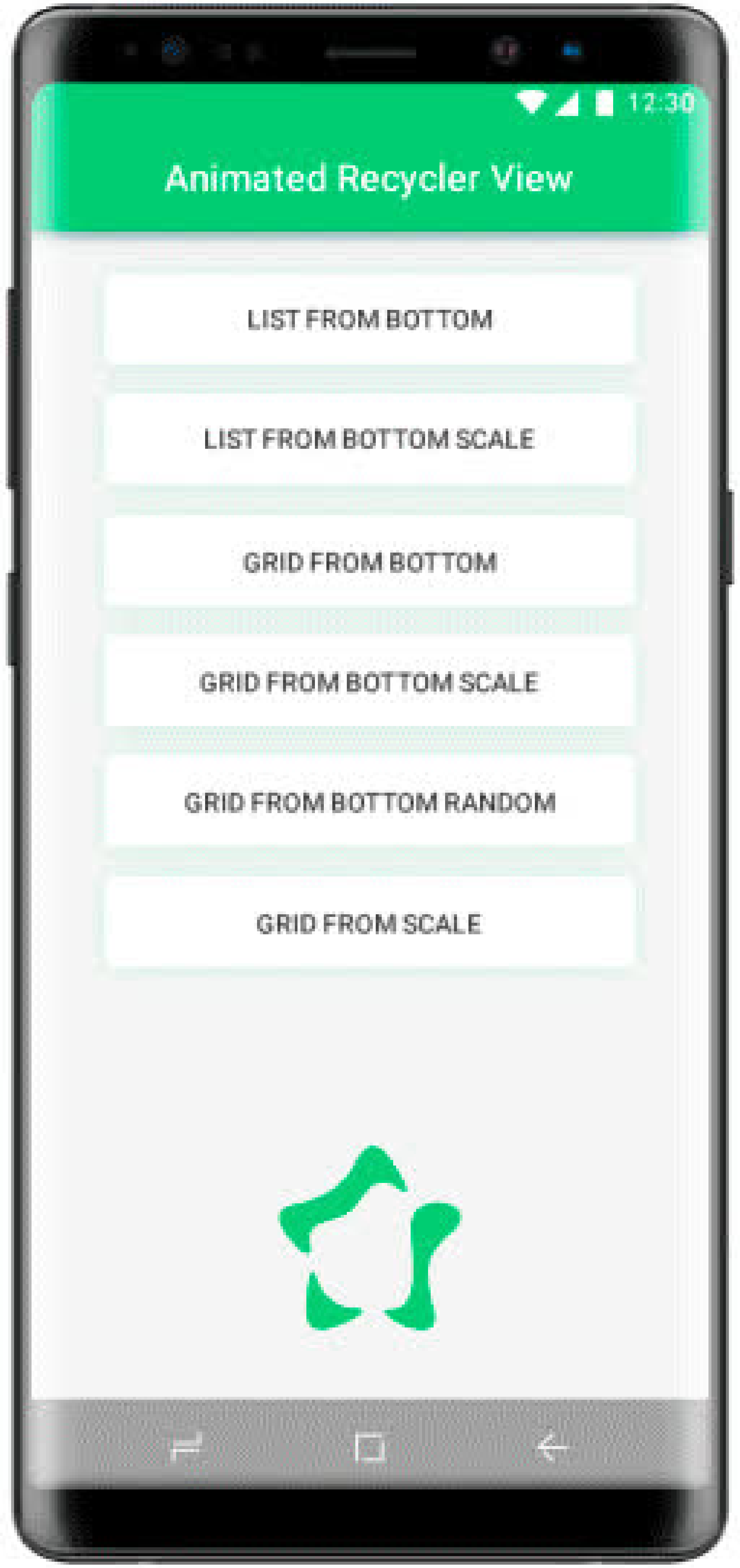
- ✓ update all list



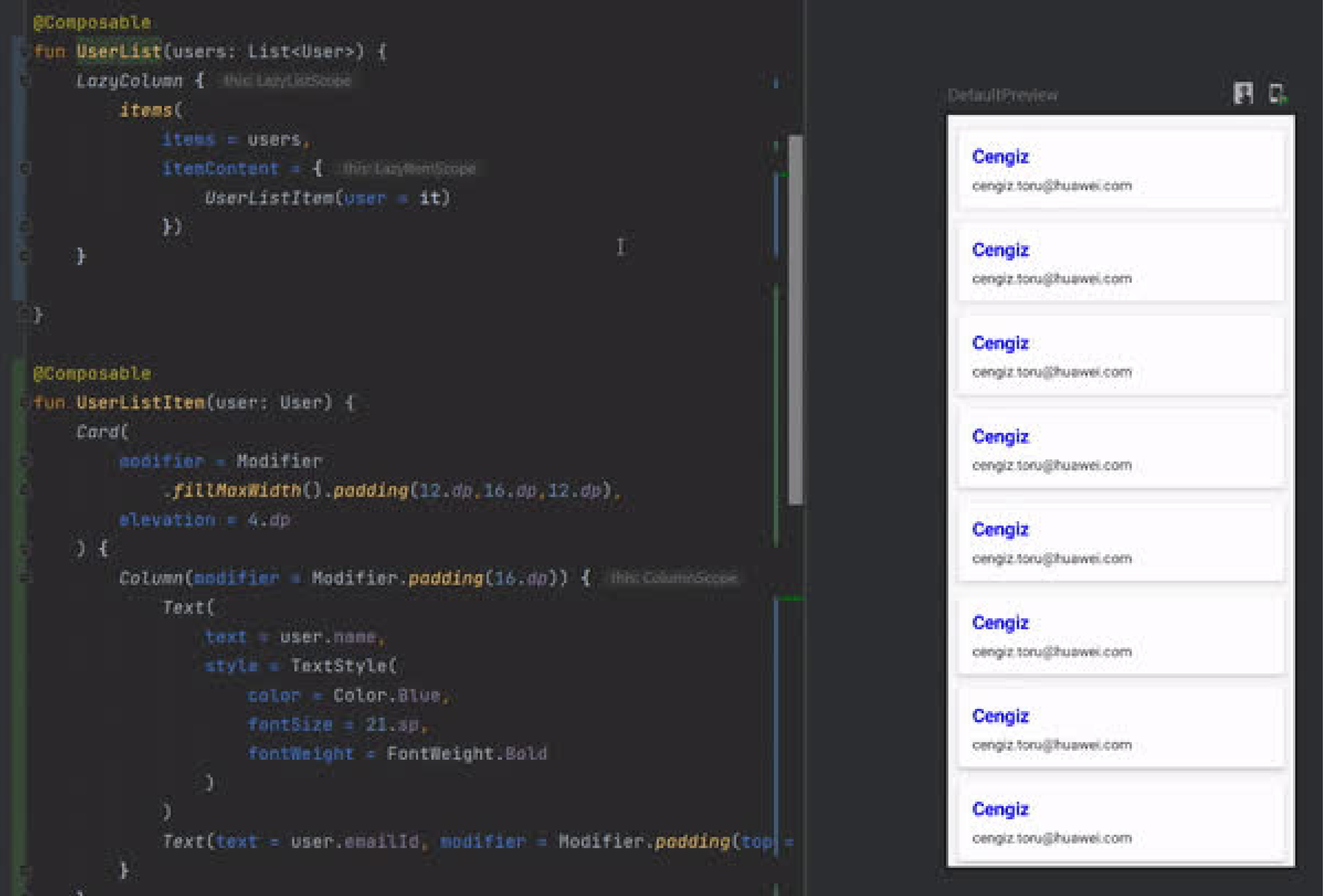
```
userListAdapter.submitList(newUserList)
```



```
userListAdapter.notifyDataSetChanged()
```



RecyclerView in Jetpack Compose



```
@Composable
fun UserList(users: List<User>) {
    LazyColumn {  this: LazyListScope
        items(
            items = users,
            itemContent = {  this: LazyItemScope
                UserListItem(user = it)
            })
    }

}

@Composable
fun UserListItem(user: User) {
    Card(
        modifier = Modifier
            .fillMaxWidth().padding(12.dp,16.dp,12.dp),
        elevation = 4.dp
    ) {
        Column(modifier = Modifier.padding(16.dp)) {  this: ColumnScope
            Text(
                text = user.name,
                style = TextStyle(
                    color = Color.Blue,
                    fontSize = 21.sp,
                    fontWeight = FontWeight.Bold
                )
            )
            Text(text = user.emailId, modifier = Modifier.padding(top =
        }
```

Cengiz
cengiz.toru@huawei.com

Cengiz
cengiz.toru@huawei.com

Cengiz
cengiz.toru@huawei.com

Cengiz
cengiz.toru@huawei.com

Cengiz
cengiz.toru@huawei.com

Cengiz
cengiz.toru@huawei.com

Cengiz
cengiz.toru@huawei.com

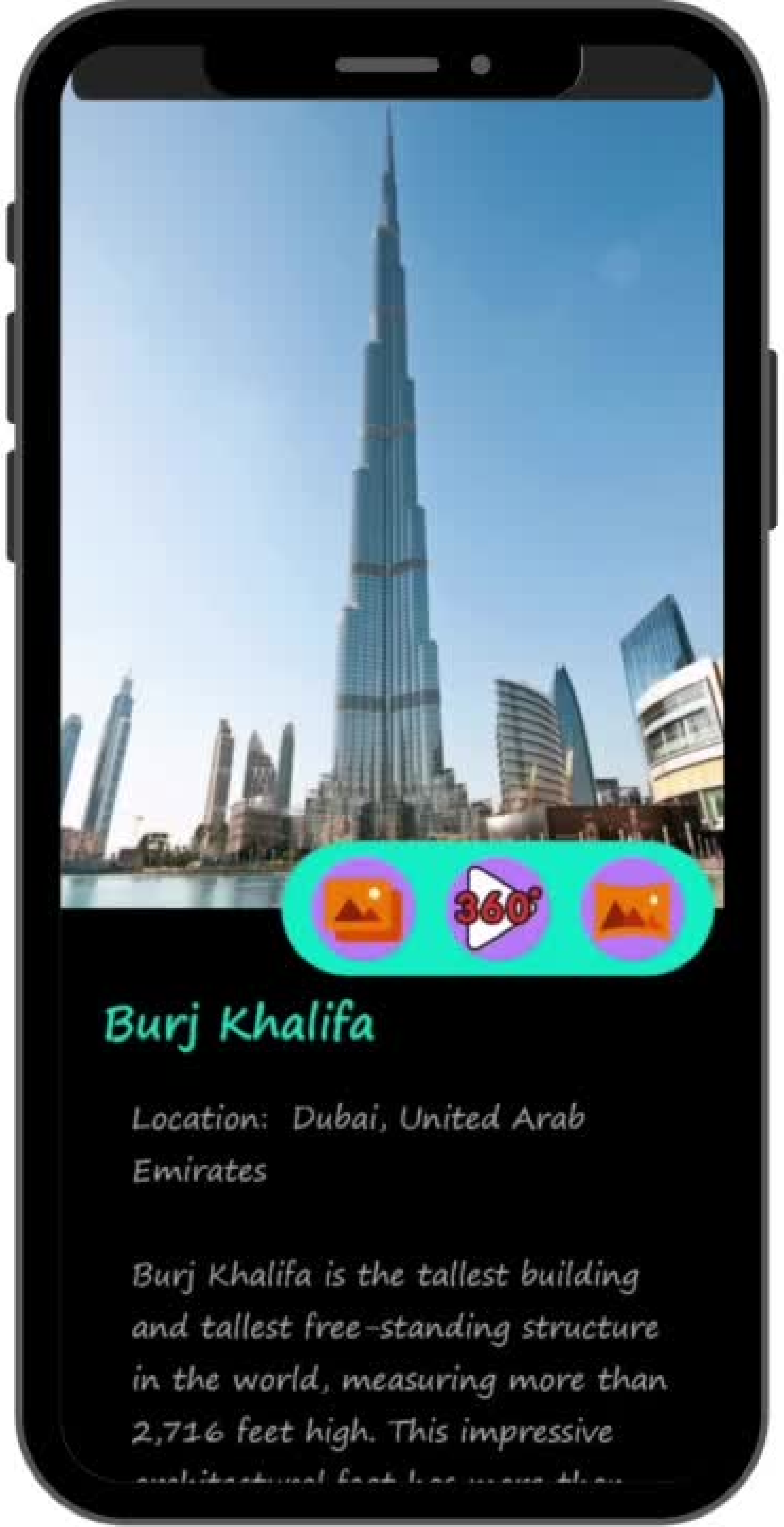
Cengiz
cengiz.toru@huawei.com

ViewPager

- ✓ Swipe views allow you to navigate between sibling screens, such as tabs, with a horizontal finger gesture, or swipe. This navigation pattern is also referred to as horizontal paging

- ✓ ViewPager in Android allows the user to flip left and right through pages of data.

- ✓ ViewPager objects have built-in swipe gestures to transition through pages, and they display screen slide animations by default, so you don't need to create your own animation.



Burj Khalifa

Location: Dubai, United Arab Emirates

Burj Khalifa is the tallest building and tallest free-standing structure in the world, measuring more than 2,716 feet high. This impressive

Data Persistence into Local

Huawei • 2022

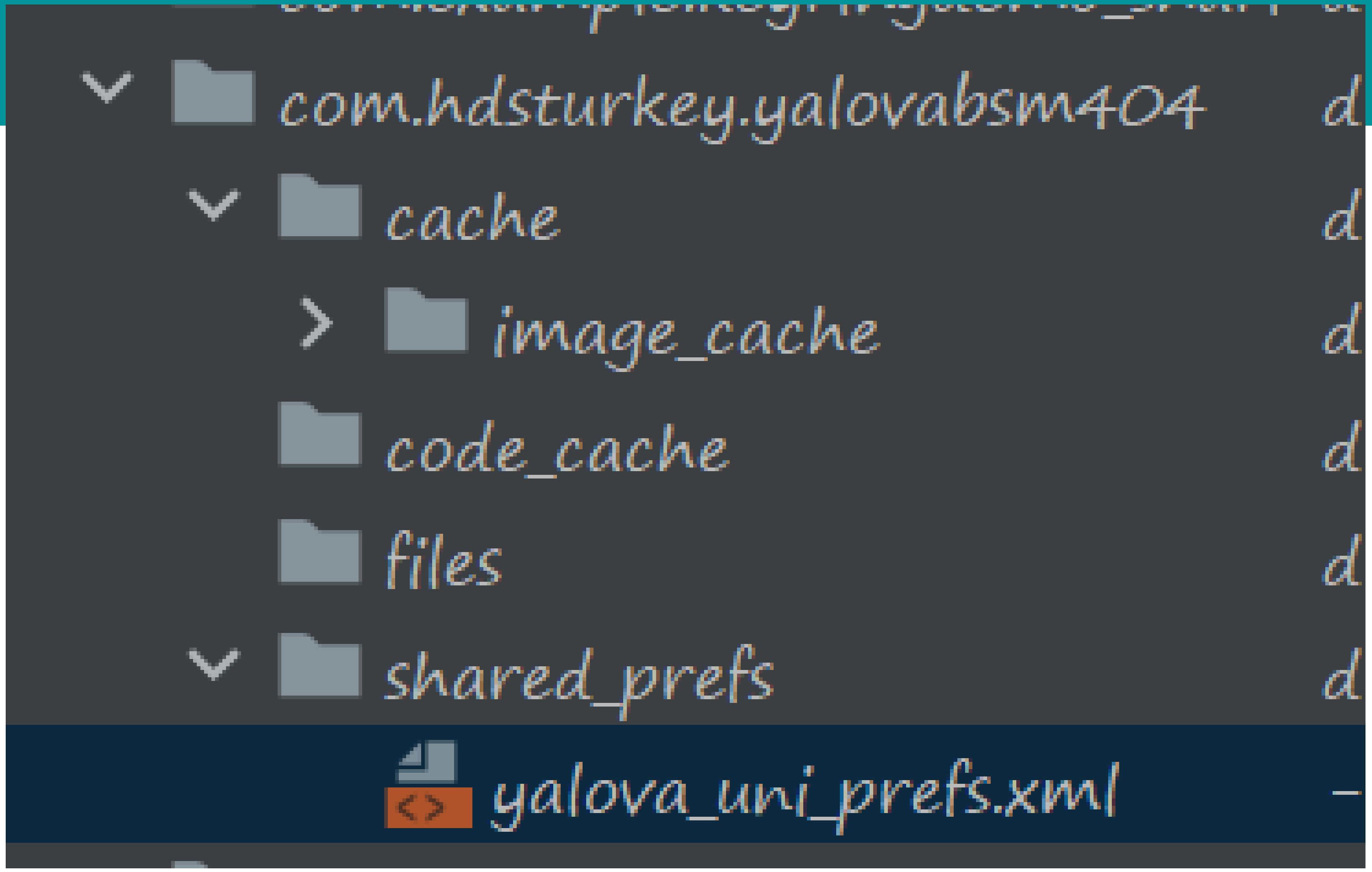
Key - Value Pairs

https://developer.android.com/reference/android/content/SharedPreferences

https://www.geeksforgeeks.org/shared-preferences-in-android-with-examples/#:~:text=Shared%20Preferences%20is%20the%20way,app%20on%20the%20device%20storage.

Shared Preferences

- ✓ Shared Preferences is the way in which one can store and retrieve small amounts of primitive data as key/value pairs to a file on the device storage such as String, int, float, Boolean that make up your preferences in an XML file inside the app on the device storage.



```
com.hdsturkey.yalovabsm404
  cache
    image_cache
  code_cache
  files
  shared_prefs
    yalova_uni_prefs.xml
```

Some Alternatives

- ✓ Encrypted Shared Preferences
  https://developer.android.com/topic/security/data
- ✓ Data Store
  https://developer.android.com/topic/libraries/architecture/datastore

Relational Database

https://developer.android.com/training/data-storage/sqlite

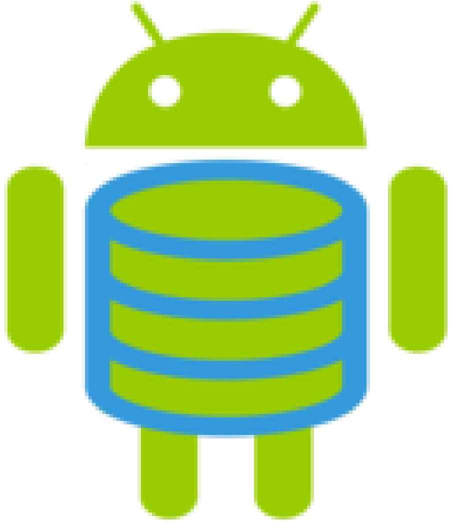
https://developer.android.com/training/data-storage/room

https://medium.com/mindorks/using-room-database-android-jetpack-675a89a0e942

https://betterprogramming.pub/a-detailed-guide-on-room-database-with-kotlin-rx-mvvm-ea982e9c5abe

ROOM

- ✓ Room is a Database Object Mapping library that makes it easy to access database on Android applications.
- ✓ Rather than hiding the details of SQLite, Room tries to embrace them by providing convenient APIs to query the database and also verify such queries at compile time. This allows you to access the full power of SQLite while having the type safety provided by Java SQL query builders. The Room persistence library provides an abstraction layer over SQLite to allow for more robust database access while harnessing the full power of SQLite



✓ Database Class ✓ Data Entities ✓ DAO (Data Access Object)

ROOM

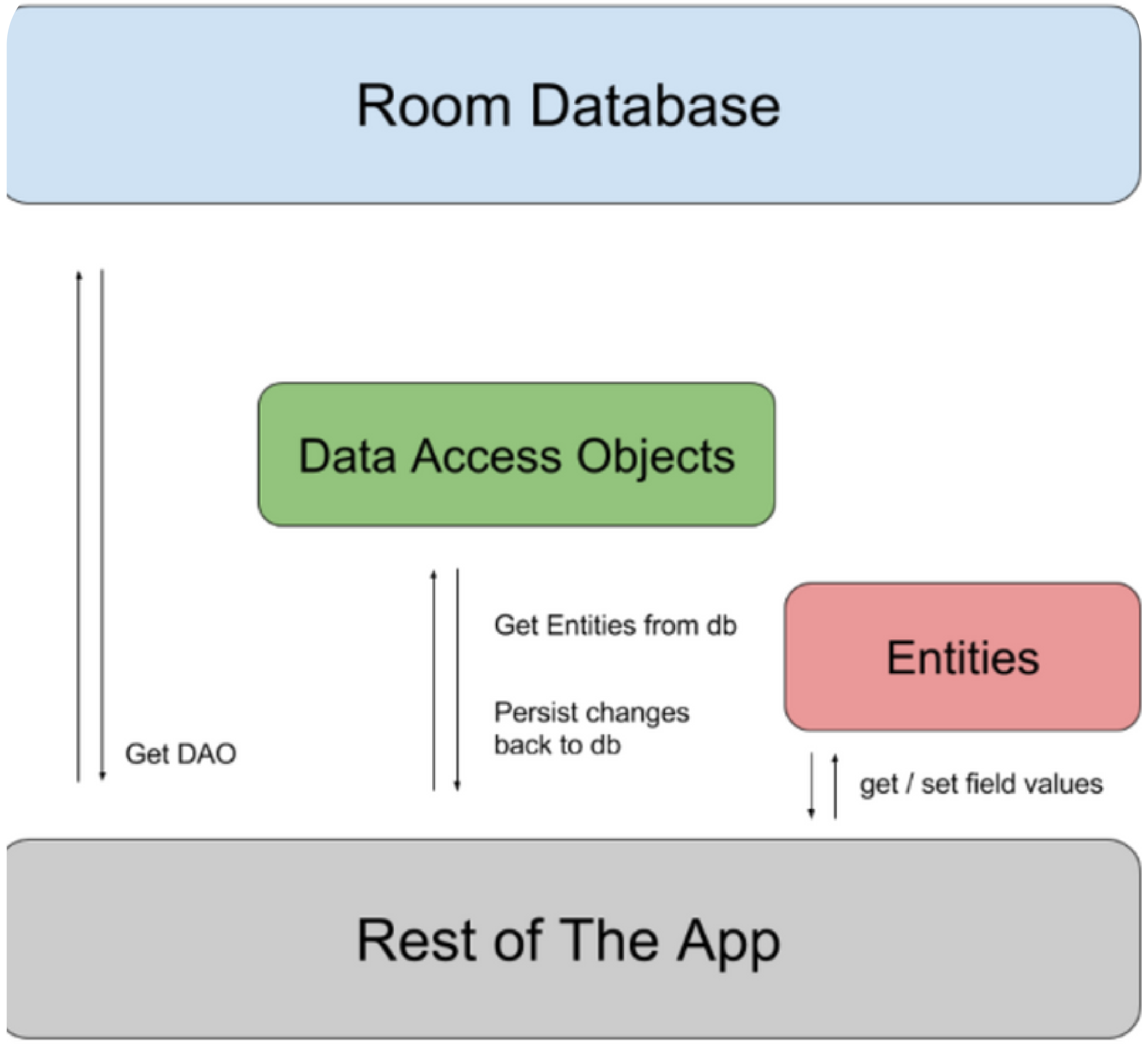
SQLite

Major Components of Room

✓ Database Class; holds the database and serves as the main access point for the underlying connection to your app's persisted data

✓ Data Entities; represent tables in your app's database.

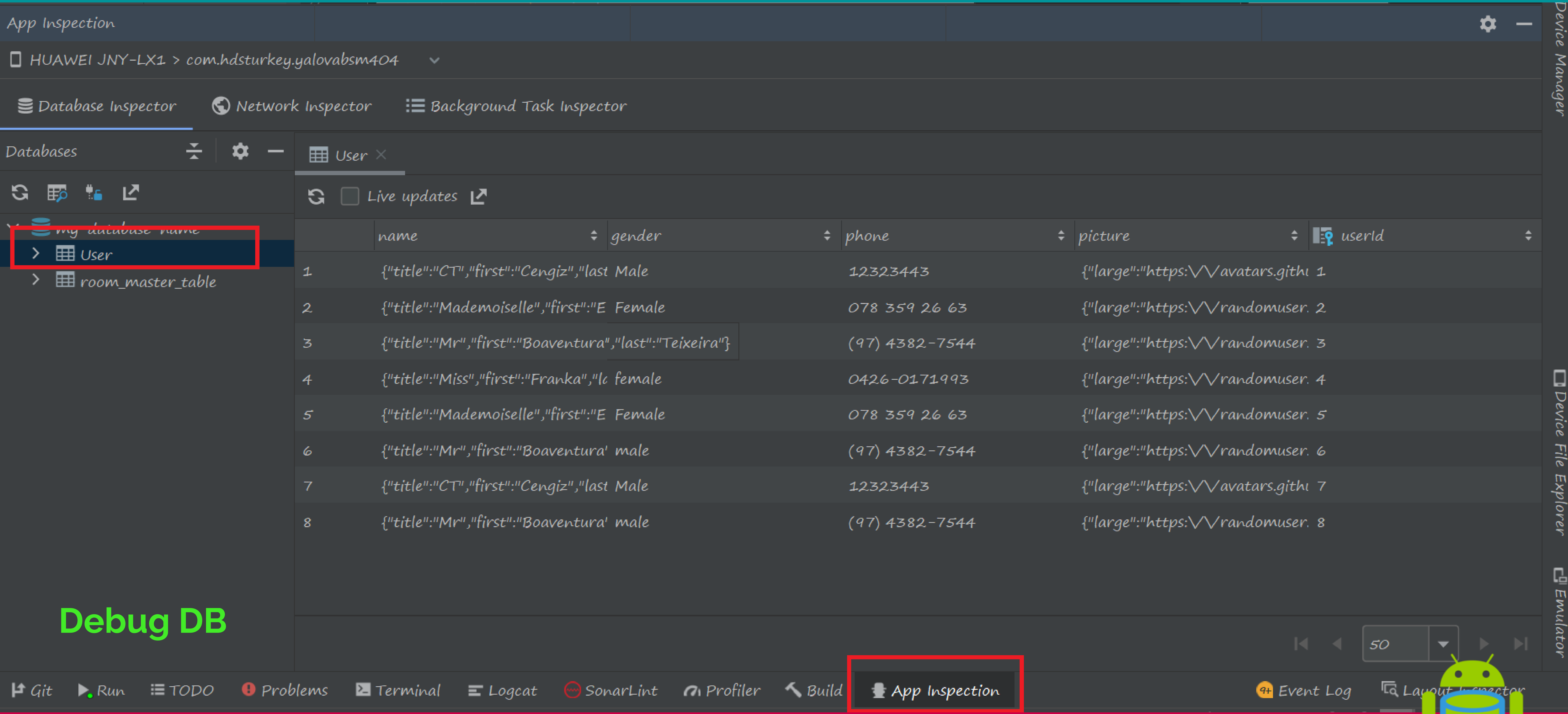
✓ DAO (Data Access Object); provide methods that your app can use to query, update, insert, and delete data in the database.



ıre 1. Diagram of Room library architecture.

ROOM

SQLite



App Inspection

HUAWEI JNY-LX1 > com.hdsturkey.yalovabsm404

User

	name	gender	phone	picture	userId
1	{"title":"CT","first":"Cengiz","last	Male	12323443	{"large":"https:\/\/avatars.githu	1
2	{"title":"Mademoiselle","first":"E	Female	078 359 26 63	{"large":"https:\/\/randomuser.	2
3	{"title":"Mr","first":"Boaventura","last":"Teixeira"}		(97) 4382-7544	{"large":"https:\/\/randomuser.	3
4	{"title":"Miss","first":"Franka","la	female	0426-0171993	{"large":"https:\/\/randomuser.	4
5	{"title":"Mademoiselle","first":"E	Female	078 359 26 63	{"large":"https:\/\/randomuser.	5
6	{"title":"Mr","first":"Boaventura'	male	(97) 4382-7544	{"large":"https:\/\/randomuser.	6
7	{"title":"CT","first":"Cengiz","last	Male	12323443	{"large":"https:\/\/avatars.githu	7
8	{"title":"Mr","first":"Boaventura'	male	(97) 4382-7544	{"large":"https:\/\/randomuser.	8

Debug DB

https://developer.android.com/studio/inspect/database

HUAWEI

LET'S TALK

CONTACT INFORMATION

Berk Ozyurt
berk.ozyurt1@huawei.com

Mehmet Yozgatli
mehmet.yozgatli1@huawei.com

Cengiz Toru
cengiz.toru@huawei.com

Telegram Channel
will be created

Official Website
https://developer.huawei.com