

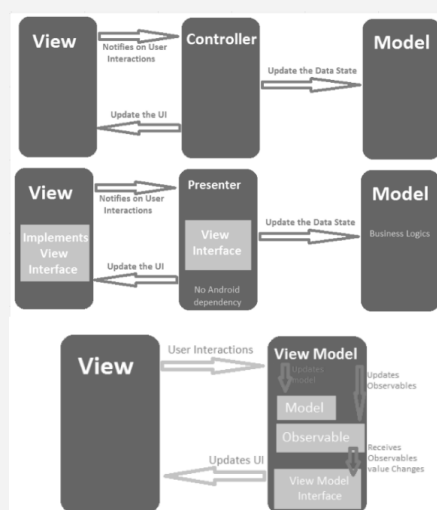
Mobile App Development Lifecycle

Inception > Design(UX->UI) > Development > Stabilization > Distribution

IDEs for Mobile App Development

Android Studio (Java/Kotlin), XCode (Objective C/Swift, can only run on MacOS), Microsoft XNA (C#, for Windows Phone apps), cocos2d(multi-language and platform, for games), Unity (C#, cross-platform), React Native (JS, cross-platform), Flutter (Dart language), Apache Cordova (HTML5, CSS3 and JS, cross-platform), MiniProgram (for making lightweight apps that don't require installation and occupy little memory/space)

Design Patterns

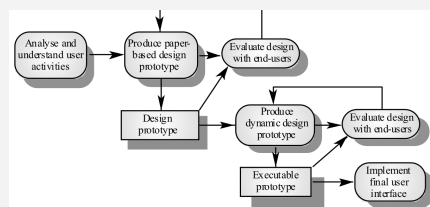


PATTERN	Dependency on Android APIs	XML Complexity	Unit Testability	Modular & SRP
MVC Controller	High	Low	Difficult	No
MVP Presenter	Low	Low	Good	Yes
MVVM ViewModel	Low or No dependency	Medium or High	Great	Yes

UI Design

Principles: User familiarity, consistency, minimal surprise, recoverability, user guidance, user diversity

UI Design Process



Broadcasts

Broadcasts are messages sent whenever an event of interests occurs from the Android System or from apps. Apps can register to receive certain broadcasts. They must be defined programmatically in the code in addition to being declared in the manifest with intent filters. `<intent-filter>` `<action android:name="AC TIO N"/>` `<intent-filter>` BroadcastReceivers have an `onReceive()` function to check `intent.action`.

Server Connection

Must include the following permissions:

```

<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
  
```

WebView is a View that can be display web pages in your app.

Java vs Kotlin

Functions	Variables
<code>public fun sum(a: Int, b: Int): Int { return a + b }</code>	<code>//val is immutable. var is mutable// val a: Int = 1 val b: String = "tim" //Types of Int, String are optional and can be excluded//</code>

Nullable variable	Arrays
<code>var str: S2? = "hku"</code>	<code>val num = arrayOf(1, 2, 3, 4) //implicit type declaration</code>
<code>var str: S2? = "hku"</code>	<code>f<Int>(1, 2, 3, 4) //explicit type declaration</code>
<code>var str: S2? = "hku"</code>	<code>for (i in 0..num.size-1) { print(num[i]) }</code>
<code>var str: S2? = "hku"</code>	<code>val I = S2?.length //if S2 is null, I is set o null</code>

By thecontinentalreaper (thcontinentalreaper)

cheatography.com/thecontinentalreaper/

Not published yet.
Last updated 15th December, 2022.
Page 1 of 3.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>

Java vs Kotlin (cont)

Classes	UI Elements
class	val edit_text1 = findViewById<TextView>(-bel ow/ toL eft Of/ -
Person(val name: String, val age: Int? = null)	<Vi ew> (R.i d.e di tText1) as EditText
//Declares class and constructor in one line!	val button1 = findViewById<Vi ew> (R.i d.b ut ton1) as Button
val person = Person("Sam", 20) //No new keyword	val listener = View.OnClickListener { edit_text1.setText(edit_text1.getText() + "Hello!") } button1.setOnClickListener(listener)

Android Layout (cont)

RelativeLayout: android d:l ayo ut_ -

TableLayout: <Ta ble Lay out >< T -

ConstraintLayout: Constrain a component to be in a position relative to another element. For example, app:la you t_c ons - tra int Lef t_t oRi ght Of= " - @+i d/e lem ent " /> would constrain the left side of the target to the right side of the other element. Remember, it is **app:layout**, NOT **android:layout**. Use sp for font size, as it scales with user font preferences, and dp for others as it changes based on different screen density.

Concurrency (cont)

Remember to use **myUIElement.post{ Runnable }** to force the Runnable object to join a queue so as to not break the rule. Handlers can also be used to update the UI thread (handle r.post{ Runnable }). Remember, **Threads cannot update UI, only Handlers**. Before running post() using Handler, call handler.removeCallbacks(runnable) to remove any pending posts of the runnable in the queue so as to avoid repeated task. **handler.postDelayed(runnable, time)** delays the runnable from starting until after the specified time..

Android Layout

Inflate the layout file in onCreate():

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
```

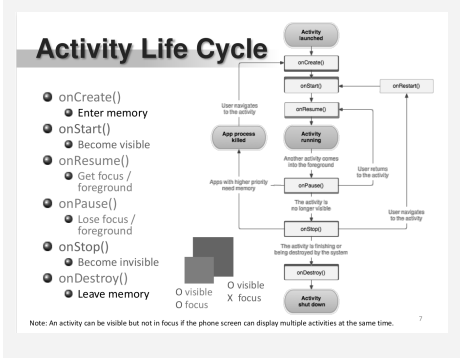
FrameLayout: Display a single item at a time. All elements are positioned based on top left of the screen, and multiple elements will overlap.

LinearLayout: Organizes elements along a single line, either horizontal or vertical (defined in XML property android:o - riente nta tio n="h ori zon tal ")

Concurrency

A process is an instance of a program that is being executed or processed. They don't share resources. Switching between process is expensive. Threads are segments of processes and share memory. **Thread{...}.start()** **Main Thread** is the UI thread which renders everything onscreen. Two rules of Android UI: **Do not block the UI thread. Do not access the UI toolkit from outside the UI thread.** **Runnable** is a class that can be run inside a Thread with just 1 method: run().

Activity LifeCycle



Intents and Filters

Intents are messages sent between Activity, Service and Broad Receivers. **Explicit intents** are used within the application for tasks such as switching between activities. They specify which component should be called. Kotlin: `intent = Intent (this, FooActivity::class.java) startActivity(intent);` **Implicit intents** only specify the action to be performed, and are sent to the Android system which chooses which component should be used. Kotlin: `intent = Intent (Intent.ACTION_VIEW, Uri.parse(URLEncoder.encode("<ID of region in main layout >", <class name of fragment>()), <number tag to represent fragment>));` `commitAllowingStaleData();`

Activity needs **Intent Filter** to receive implicit intents. `<intent-filter> <action android:name = " hku.cs.m.y.intentfilter" /> <category android:name = " android.intent.category.DEFAULT" /> </intent-filter>` **Passing extra data:** Sender: `myIntent.putExtra("ID", 6963)` Recipient: `val bundle: Bundle? = intent.extras?.let { bundle.apply { val inputString: String? = getString("ID") // id = 6963}}`

Fragment

Must implement: **onCreate()** - called when creating the fragment, should initialize essential components you want to retain when the fragment is paused or stopped, then resumed. **onCreateView** - called when the fragment draws its user interface for the first time. Optional: **onPause()** - called when the user is leaving the fragment.

Kotlin implementation

Fragment (cont)

```
val manager: FragmentManager =
    supportFragmentManager
val ft: FragmentTransaction =
    manager.beginTransaction()
if (fragment != null) ft.replace(
    ID of region in main
    layout >, <class name of
    fragment>(), <number tag to
    represent fragment>); ft.commit
    AllowingStaleData();
```

Service

StartService(): runs indefinitely even if caller app dies. Simple, single task. No return result. Cannot be called back or modified after it is sent out.

BindService(): Can be bound to multiple components. Terminates if all callers die. Can be modified after being sent out.

A service can be both started and bound simultaneously.

Activating service: `val intent = Intent (this, HelloService::class.java); startService (Intent);`

Return value of onStartCommand():
START_NOT_STICKY: Do not recreate after kill. Caller can restart unfinished jobs
START_STICKY: Recreate, but do not redeliver last intent. Continuous work but stateless, e.g. media players.
START_REDELIVER_INTENT: Recreate, and redeliver last intent. Actively performing a job, e.g. file download

Storage

SharedPreferences: Private primitive data in key-value pairs (persistent storage)

Bundles: Private primitive data in key-value pairs (temp storage for activity-fragment transfer)

Internal storage: Private data on device memory

External storage: Public data on internal or shared external storage, e.g., SD card

SQLite database: Structured data (table) in a private database



By thecontinentalreaper
(thcontinentalreaper)

cheatography.com/thecontinentalreaper/

Not published yet.

Last updated 15th December, 2022.

Page 3 of 3.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>