

Native Apps

- * Mobile applications that is tailored and dedicated to a specified platform like iOS or Android.
- * Build for specific operating systems, provide higher user engagement than hybrid apps.
- * Written in Java or Kotlin programming languages.

* Benefits of Native Apps:-

1. Best performance
2. More secure
3. More Interactive and Intuitive
4. Allow Developers to access full set of features of devices
5. Tends to have fewer bugs during development.

1. Best performance:

- * Created and optimized for a specific platform.
- * Very fast and responsive as built for specific operating system.
- * Compiled using platform's core programming language and API.
- * More efficient
- * Content and visual elements of app are stored on user's device, results in quick load times.

2. More secure

- * Guarantees your users reliable data protection

3. More Interactive and Intuitive

- * Runs smoothly
- * Superior user experience
- * App feel like an integrated part of device.
- * Flow of app is more natural.

classfellow

4. Allow Developers to access full features of devices
- * App can directly access hardware of device such as GPS, camera, mic, etc.
 - * Faster execution, better user experience
 - * Push notifications is an advantage
 - *

5. Tends to have fewer bugs during development
- * Fewer dependencies for bugs to occur.
 - * Do not rely on cross platform, no bridge communication, hence fewer bugs

Difference b/w Native and Hybrid Apps

Native Apps

- * Require installation

- * High maintenance.

- * Large budget.

- * Multiple codebases.

- * Best user experience

- * Languages used are Java, Kotlin, Swift

- * Developed for a particular platform.

- * Updation can be done from play store or app store

Hybrid Apps

- Do not require installation

- : Low maintenance.

- : Cost effective.

- Single codebase.

- Not good user experience

- Languages used are HTML, CSS, JS

- Can work on various platforms.

- Updations are centralized

classfellow

Storing Data

- Shared Preferences → Primitive data in key-value pair (private)
- Internal Storage → Private data on device memory
- External Storage → Public data on device or external storage
- SQLite Database → Structured data in private database
- Content Providers → Store privately and make available publicly.

Storing Data beyond Android

- Network Connection → On the web with your own server
- Cloud Backup → Backup app and user data in cloud.
- Firebase Realtime Database → Store and sync data with nosql cloud database across clients in realtime.

Android File System

Internal
Storage

External
Storage

⇒ Internal Storage

Refers to built-in storage memory of an Android device which cannot be removed or replaced. It is typically used to store system files, apps and user data. When an Android app is installed on a Android device, it is stored in the internal storage space. App's data including settings and user data is also stored in internal storage.

Internal storage is divided into partitions, one partition is used for the os while another for apps and their data.

Internal Storage is more secure than external

storage. Data in the internal storage is protected by Android system and apps are only able to access their own data.

⇒ Applications of Internal Storage:-

- ⇒ 1. Used to store apps and their data.
- 2. System files and configuration data is stored in internal storage.
- 3. Used for caching data that can speed up app performance.
- 4. More secure.

⇒ External Storage

Refers to removable storage options that can be added or removed from an Android device, such as microSD card. It is used for storing media files like photos, videos, audios, etc.

Generally slower than internal storage which can impact the performance that use it.

It provides users with the ability to expand their device's storage capacity, to store large amount of files or documents.

Not secure than internal storage, more prone to attacks and vulnerable usage, also as it can be removed from the devices, more chances of damage or lost.

⇒ Applications:-

- 1. Used for storing media files like photos, videos, etc.
- 2. Can be removed easily and accessed by other devices.
- 3. Provides users with the ability to expand their device's storage capacity.

=> Difference b/w Internal and External Storage

Internal Storage

External Storage

- | | |
|---|--|
| 1. Built-in memory that can not be removed from the device. | Removable memory that can be added or removed from the device. |
| 2. Less storage space. | More storage space. |
| 3. Used to store system files, apps and user data. | Used to store media, documents etc. |
| 4. Can't be accessed by other devices. | Can be accessed by other devices. |
| 5. Data is more secure. | Less secure. |

=> Firebase Database

- * Cloud hosted NoSQL database that allows developers to store and sync data in realtime b/w multiple clients.
- * Uses a publish-subscribe model, that means change to database are instantly propagated to all the connected clients.
- * Provides offline support, clients can read and write data even when not connected to internet.
- * Powerful security model with the help of which developers can control who can read or write data.
- * Security rules are written in JSON file.
- * Integrates seamlessly with Firebase services.

Features:-

- * Realtime data synchronization
- * Offline support
- * Powerful security model
- * Integration with other Firebase services.

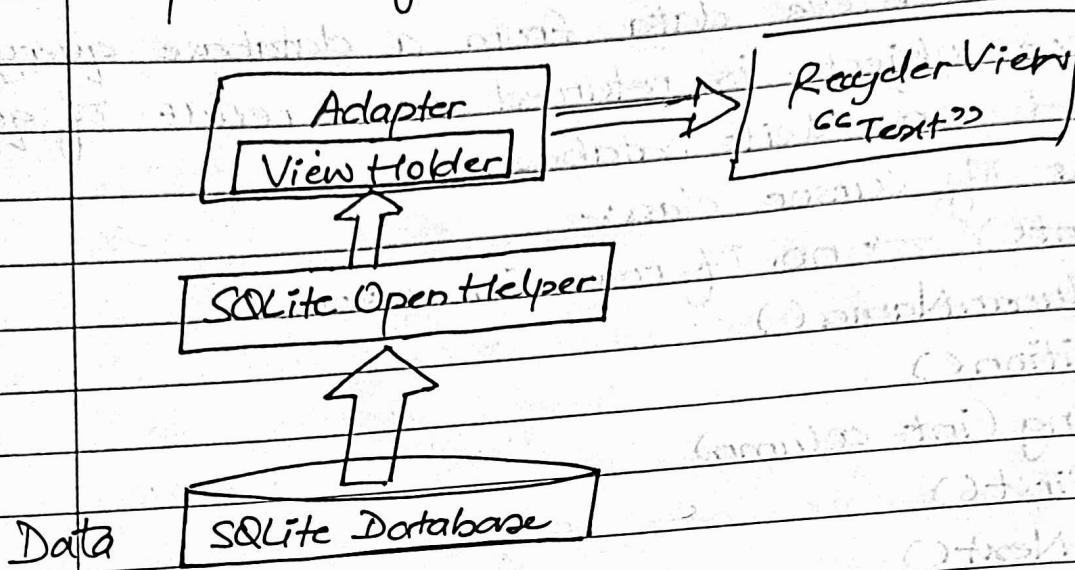
classfellow

⇒ SQLite Database

- * Software library that provides a relational database management system designed for embedded and small-scale applications.
- * Lightweight, self-contained database engine that can be used to store and manage data in Android Apps.
- * Can be used to store a wide variety of data types, including text, numbers and binary data.
- * Features of SQLite:
 1. Easy to use
 2. Highly optimized database engine that can handle large datasets
 3. Efficient and fast
 4. Provides a layer of security to protect sensitive data.
 5. Highly compatible with Android platform.
- * Applications:-
- * Used to store and manage data locally on Android devices
- * Used to cache frequently accessed data, improving app performance.
- * Used to store user preferences and settings for an app.
- * Used to implement content providers to manage access to data from different apps.

classfellow

Components of SQLite



⇒ SQLite Open Helper

- * Helper class in Android that provides an abstraction layer for working with SQLite databases.
- * Methods provided by SQLiteOpenHelper
 - onCreate() → called when DB is created for first time.
 - onUpgrade() → called when DB is upgraded.
 - onDowngrade() → called when DB is downgraded.
 - onReadableDatabase() → when object can read data from DB
 - onWritableDatabase() → when object can read and write data from DB.

* Uses:-

- Executes requests to DB
- Manages DB
- Separates data and interaction from app
- Keeps complex apps manageable

classfellow

=> Cursors

- * Used to retrieve data from a database query.
- * A cursor object is returned as a result of query executed on SQLite Database.
- * Methods of cursor class:-
 - getCount() → no. of rows in cursor
 - getColumnNames()
 - getPosition()
 - getString(int column)
 - moveToFirst()
 - moveToNext()
 - close()

=> Content Values

- * Used to insert and update data in DB.
- * Create a key-value pair to store data in ContentValues object.
- ** Key is name of column, value is the value (data) for the field.
- * Methods:-
 - get() → to retrieve data
 - put() → to insert or update

=> Database operation

- query()
- insert()
- update()
- delete()

classfellow

⇒ Android Graphics

Android SDK provides a comprehensive set of 2D graphics APIs that can be used to create visually appealing and interactive user interface.

Android 2D graphics framework provides a variety of classes and interfaces that allow developers to draw shapes, text and images on the screen.

These classes include:-

1. Canvas → Provides a drawing surface
2. Paint → to define colour, style and properties of graphics
3. Bitmap → Represents a bitmap image
4. Drawable → Used to draw complex shapes
5. Path → Represents geometric path.

⇒ Animations

Animation systems allows developers to create visually appealing, and engaging user interfaces that provide a great user experience.

1. View Animations

* Simplest Animation

* Allows to animate the properties of view such as position, size or transparency.

* Easy to use

* Can be applied using XML or Java code.

2. Property Animation

* More advanced Animation

* Allow to animate the properties of any object not just views

* Use a ValueAnimator object to animate values of property over time.

classfellow

- * Used to create complex animations.
- 3. Drawable Animation
- * Allows to animate the properties of drawable such as shape, colour, size
- * Can be used to create simple animation such as pulsing buttons or rotating icons.

⇒ Canvas

- * 2D drawing surface that allows to draw graphics, shapes, text on screen.
- * Two ways to draw 2D graphics
 - Draw animation into a View object
 - Draw animation directly to Canvas
- * Methods of Canvas Class:-
 1. drawRect()
 2. drawCircle()
 3. drawLine()
 4. drawText()
 5. drawBitmap()
- * Canvas can also be used to add transitions to drawings such as rotating, scaling

⇒ Methods of Animation

1. startAnimation()
2. clearAnimation()

classfellow

⇒ Sprucing Up Mobile Apps

- * Refers to enhancing the app's design, functionality, and user experience.
- * This can include adding new features, updating app's design, fixing bugs, making app user-friendly.
- * Techniques for sprucing up mobile app:-
 1. App should have intuitive and user friendly interface.
 2. Adding Animations and Transitions can make the app visually appealing.
 3. Custom Themes, colors or fonts can make app more appealing to users.
 4. Clean and organized layout can make the app easy to use and navigate.
 5. Optimized for different screens.
 6. Use of high-quality images and graphics.
 7. Regular updates and bug fixes.

⇒ Audio Recorder in Android

- * MediaRecorder class is used for recording audio or video
- * For recording audio, device microphone is used along with MediaRecorder class
- * For recording video, device camera is used along with MediaRecorder class.

classfellow

⇒ Methods of MediaRecorder Class:-

1. `setAudioSource()` → to specify the audio source to be recorded
2. `setAudioEncoder()` → to specify audio encoder
3. `setOutputFormat()` → to specify output format of audio.
4. `setOutputFile()` → to specify path of recorded files that are to be stored.
5. `stop()` → to stop the recording process
6. `start()` → to start the recording process
7. `release()` → to release the resources associated.

⇒ Device Location

1. Android's Location Manager API

2. Fused Location Provider: Google Play Services Location API

Fused is more efficient as it optimizes the battery usage.

classfellow

⇒ Video Player

1. Add Video View to XML file:-

<Video View

 android: id = " @+id/videoView " "

 android: layout_width = " match-parent " "

 android: layout_height = " match-parent " />

2. Java code in MainActivity

 VideoView videoView = findViewById(R.id.videoView);

 String videoPath = " path/to/video.mp4 " ;

 Uri uri = Uri.parse(videoPath);

 videoView.

 videoView.setVideoURI(uri);

 videoView.start();

URI → Uniform Resource Identifier

uri is a class that represents a URI and provides methods to manipulate it.

It is used to represent various types of resources such as files, videos, audio, images, etc.

classfellow

- => Debugging Database in Android
- * Helps developers identify and fix issues related to data storage and retrieval.
- * Ways to debug database :-
 1. Logcat → Helps you see all the operations your application is performing. Helps to filter out specific messages from the database and by analyzing these messages, issues with the database can be resolved.
 2. SQLite Browser → Open source tool that helps you view and edit SQLite Database. Helps to open your application's database file and inspect the content.
 3. Android Studio → Built-in support for debugging SQLite Database. You can use Database Inspector to view contents of application's database and inspect the schema of the tables.
 4. Unit Testing → An effective way to test your database operations and ensure they are working as expected.
- => Steps involved in the journey of an Android App.
 1. Build the APK
 2. Deploy the Application
 3. Run the Application.

⇒ Black Box Testing,

- * Testing method in which tester does not have any knowledge of internal workings of the system.
- * Tester only interacts with system's input and observe the outputs.
- * Primary goal is to verify that the system meets the functional and non-functional requirements of the system.
- * Testing is done from a user's perspective.

⇒ White Box testing

- * Tester has full knowledge of internal working of the system.
- * Tester has access to system's source code, and internal components.
- * Primary goal is to check completeness and optimisation of system's internal logic.
- * Testing is done from a developer's perspective.

⇒ Difference b/w Black Box + White Box Testing

Black Box

- * Tester has no access to internal workings of system.
- * Based on external requirements.
- * Focuses on inputs and outputs only.
- * Tests are done from user's perspective.
- * Don't require knowledge of programming or codes.

White Box

- Tester has access to internal working of system.
- Based on internal requirement.
- Focuses on internal logic and code structure.
- Tests are done from developer perspective.
- Requires knowledge of programming or codes.

classfellow

Black Box

- * Can be used for both functional and non-functional testing.
- * Testing is done without looking at the code.
- * Testing is done using pre-built test cases.

White Box

Primarily for functional testing.

Testing is done by looking at the code.

Testing is done using custom test cases.

⇒ Automation Tools

- * Software programs to automate the testing of Android apps.
- * Helps testers and developers to perform various types of testing.

1. Appium → Open-source test automation tool that allows automate testing for native, hybrid and mobile web apps.

It supports different programming languages like Java, Ruby, Python, JS

2. Robotium → Open-source test automation tool that allows automate testing for Android apps. It provides APIs for interacting with UI elements and support both black box and white box testing.

3. UI Automator → Testing framework developed by Google for testing user interface of Android apps. It provides APIs to interact with UI elements.

classfellow

4. Espresso → Testing framework by Google for testing Android Apps.

It provides simple APIs for testing UI and can be used for unit and functional testing.

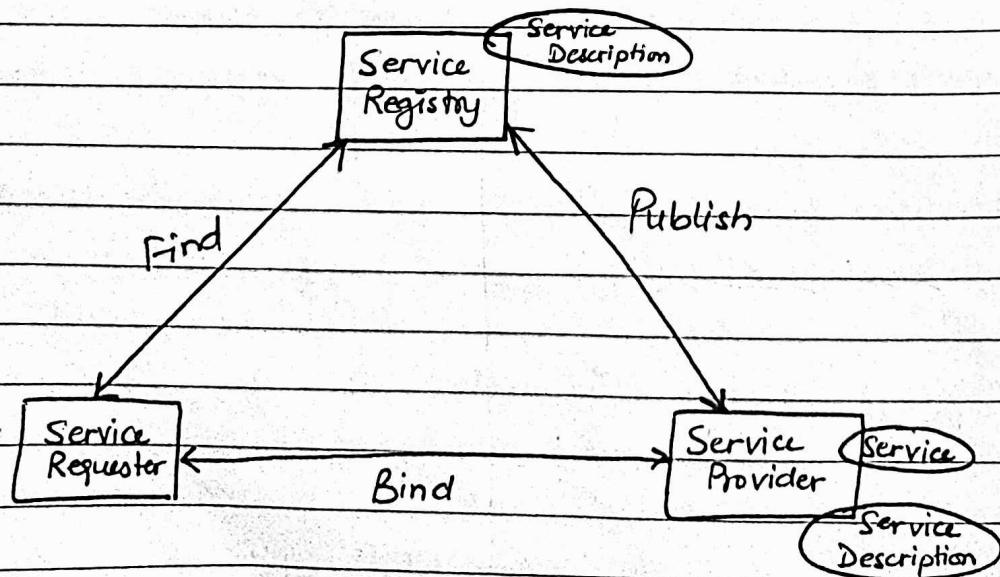
5. Calabash → Open-source tool for automate testing of Android Apps, Mobile Apps.

Uses the cucumber testing framework and provide support for both iOS and Android.

⇒ Web Services

- * Software system designed to allow communication b/w two different applications over the Internet.
- * Enables exchange of data b/w two different platforms regardless of their OS, hardware or software.
- * Uses standard protocols such as SOAP, XML, WSDL, UDDI
- * Can be accessed using HTTP and HTTPS
↓
Secure

⇒ Web Service Model



classfellow

- Service Provider

- * Owner of the service
- * Platform that hosts access to the service.

- Service Requester

- * Business that requires certain functions to be satisfied
- * Application looking for and invoking an interaction with a service

- Service Registry

- * Where service providers publish their service description

- Operations

1. Publish → Service description needs to be published in order for service requester to find them.
2. Find → Service requestor queries the service registry for the service required.
3. Bind → Service requestor initiates an interaction with the service at runtime.

classfellow

⇒ SOAP → Simple Object Access Protocol

- * Messaging protocol that allows communication b/w applications over Internet.
- * Uses an XML based msg format to transfer data b/w applications
- * Platform independent
- * SOAP Msg consists of an envelope that defines msg structure
- * Header that provides additional information about msg
- * A body, that contains actual data to be transferred
- * SOAP envelope also contains set of rules for processing the msg.

Why SOAP?

- * Allows communication b/w applications written in different languages.
- * Makes it easier to integrate and exchange data b/w different systems.
- * Data transferred is secure and protected.
- * Provides security features such as msg authentication, digital signature, etc.
- * Built-in mechanism for handling errors and faults
- * Msg delivered are reliable and consistent.

classfellow

⇒ WSDL → Web Service Description Language

* XML based language used to describe interface of web service.

* Provides a machine readable description of the web service

* WSDL document contains 4 sections :-

1. Types → Defines data types used in web service

2. Message → Defines input and output msg for each operation in web service

3. Operations → Defines methods or functions

4. Service → Defines end points of web service

⇒ UDDI → Universal Description, Discovery and Integration

* XML based standard for creating and publishing web service directories.

* Provides a way for business to publish their web service

* Provides a way for clients to discover and access web services.

* UDDI defines a set of APIs and protocols for creating and querying web service directories, including -

1. Publishing → Business can publish their web services specifying service name, location and other metadata

2. Inquiry → Clients can query UDDI registry to discover web services.

3. Binding → Provides a way for clients to bind to a web service and obtain info needed to communicate with the service.