

Flutter Assignment 1

Question 1 - What platforms are available for developing mobile apps?

There are Basically 3 Different Types Available to develop Mobile Applications .

1 . Mobile OS Native

This is a Development where the Manufacturer of the Operating System be it Android or iOS has opened up Apis for all the different components and these apps can use full potential of the mobile device being it Graphics intensive tasks or Handling or scaling requests . These codes run directly on the Device level . For different operating systems we need to write different code bases.

Examples : JAVA , KOTLIN , SWIFT , SWIFT UI , OBJC

2 . Hybrid

The hybrid type of development is where the application is basically a browser which loads and creates an ui via apis . Here the application acting as browser can call internal system level apis and functions which will then in turn handle system level processes like VIBRATION or Implicit Intents . Or we can say a javascript code calling an Api or plugin to use Vibration Motor .

Examples : IONIC , CORDOVA , PHONEGAP

3 . Cross Platform

Cross Platform technologies are where we write a single codebase for multiple operating systems or even Web so that we don't have to do the native code twice if we are planning to launch an app on multiple platforms. These apps run on a top level of native devices where every component is similar in both screens or platforms and converts its code to native everytime the apps are compiled . Cross platform apps are in huge boom but lack the performance of native code.

Examples : Flutter , React Native , Xamrin

Question 2 - Native(Java/Kotlin/Swift/Objective-c) vs Cross Platforms(React-native, Flutter, Cordova, ionic)

Native	Cross Platform
High Performance	Basically a browser running an app (hybrid) Near to native but not for huge scale apps (Flutter , RN)
Can access device's full capabilities , functions like (Hardware & system events)	Needs to call callers from native platform to work with hardware and features
Multiple codebase	Single Codebase
Multiple languages to be learnt for different os Ex : java , Swift	Single language can be used Ex : Dart , Js
Not Cost Effective	Cost Effective
Internal Root level Permissions 100%	No internal API Access or Root Permissions

Question 3 - Core differences among - React-native, Flutter, Cordova, ionic etc

Cross Platform	Hybrid
Code converts to native and runs on an ENGINE / BRIDGE	Everything Runs over a Browser
Higher Performance	Low performance
Better Graphics Rendering	Renders Graphics Until browser / Webview fallbacks
Hot Reloading is Perfect	Live Reloading is a bit on Lower side
Use of languages like Dart , React	Use of Languages like JS , HTML
UI Framework	Web Framework

Question 4 - Why react native and why not? (Pros and Cons)

Pros :

- Faster Build Times
- Hot Reloading
- Single Codebase for Multiple Platforms
- Simplified Components
- Need to learn Only one Programming Language
- Cost Effective
- Bigger Community

Cons :

- Complete code is interpreted by on Device JS bridge
- Performance Degradation
- Less Fluid Animations
- Documentation is weaker
- Bad Backward Compatibility
- Depends on External Plugins to Call Device level Features
- Components needs to be organised according to Operating System

Question 5 - Why flutter and why not? (Pros and Cons)

Pros :

- Great Hot Reloading
- Ui Components Look exactly the same on all Operating Systems
- Great graphics Engine developed on top C++
- Increasing Community Support
- Documentation is really good
- Native code to Write is less
- Easy to Implement Animations
- Lot of Inbuilt Components
- Lot of Libraries
- Good backward compatibility
- Single language to learn (Dart)
- Great open Source Community
- Great Implementation of Development Architectures like Flutter MVC MVVM

Cons :

- Being a cross platform technology it converts complete code base to native and then its rendered on top of application layer
- Performance is hampered
- Not Great for Huge scale applications
- Device level Implementation is hard to access
- Less Libraries than RN
- Still some parts in Beta
- New Language to learn (Dart)

Question 6 - Execution process of flutter code.

When Flutter code is compiled the code is basically converted into native codebase and then this code is then converted into byte code . Definitely flutter requires some endpoints to be written in native just in case permissions or let's take an example to block users taking screenshots from android here we need to implement the code in native code like for a window and add a flag similar in IOS and Android . So Flutter uses components which in turn can be said as 1 - Material Components , 2 - Cupertino Components that are rendered over an C/C++ level Engine of that application on different platforms . Here the android and ios kernels then both have the only work to handle the C++ engine than to handle the complete app . So with this the performance is a bit on the downside for cross platform . The Ui Framework Flutter renders over a Canvas Called SKIA CANVAS . The important part is threads of Business Logic and the Graphics UI thread both are highly efficient and makes things possible to achieve near to native performance . Widget in flutters have different life cycle .to access kernel level events of the system flutter needs to use different plugins which are basically written in NATIVE code and just giving up functions to the flutter code. Cross-platform frameworks typically work by creating an abstraction layer over the underlying native Android and iOS UI libraries, attempting to smooth out the inconsistencies of each platform representation.