



Mobile App Development Course: Flutter and React Native for iOS and Android



Mobile App Development Course: Flutter and React Native for iOS and Android

Introduction:

Mobile app development with Flutter and React Native for iOS and Android is a five-day course for web, back-end and desktop development teams moving into mobile, ending with a publish-ready case app and its release checklist. Many organisations depend on mobile channels yet staff them with developers who know web or server code but not app lifecycles, offline behaviour, device permissions or store release. Nominees already write production code in at least one language and work hands on in daily labs on emulators and physical devices. CoreConcept Training Center delivers this course on mobile app development.

Course Objectives:

- Select between native and cross-platform toolkits for a given app using a decision matrix that weighs performance, team skills and device access
- Structure a mobile app into UI, domain and data layers with unidirectional data flow, state holders and repositories
- Build screens, responsive layouts and stack and tab navigation in Flutter or React Native
- Manage app state, call REST APIs and cache data on the device so the app keeps working offline
- Integrate camera, location and push notifications, implement sign-in with OAuth2 and PKCE and keep tokens in platform secure storage
- Test, profile and prepare a case app for store submission with code signing, store listing assets and a release checklist

Target Audience:

- Web development teams responsible for browser front ends who are taking on mobile app work
 - Back-end and API development teams that now build the mobile clients consuming their services
 - Desktop application developers responsible for moving line-of-business tools onto phones and tablets
 - Junior mobile developers who maintain existing apps and need a structured cross-platform workflow
 - Technical leads responsible for choosing the mobile toolkit for a product team
-

Course Outline:

Day 1: Mobile Platform Landscape, App Lifecycle and Toolkit Selection

- Native Versus Cross-Platform Toolkit Selection Decision Matrix
- Flutter Rendering Engine, Dart Language and Widget Tree Basics
- React Native Components, TypeScript and Native View Mapping
- Mobile App Lifecycle States and Background Execution Limits
- Device Debugging with Emulators, Simulators and Hot Reload

Day 2: Layered App Architecture, Screen Building and Navigation

- Layered App Architecture with UI, Domain and Data Layers
- Stateless and Stateful Widgets with Material and Cupertino Sets
- Responsive Layouts for Phone and Tablet Screen Sizes
- Stack and Tab Navigation with Deep Link Routing
- Unidirectional Data Flow with ViewModel State Holders

Day 3: State Management, REST API Integration and Offline Data

- Flutter State Options with ValueNotifier, InheritedWidget and Provider
- React Native Screen State with Hooks and Shared Stores
- Repository Pattern for REST API Calls and JSON Parsing
- Offline-First Local Database Caching and Sync Conflict Handling
- Dependency Injection for Testable Services and Repositories

Day 4: Device Features, Secure Sign-In, Testing, Performance and Accessibility

- Camera, Location and Push Notification Permission Flows
- OAuth2 Authorization Code Flow with PKCE for Sign-In
- Token Storage in iOS Keychain and Android Keystore
- Unit, Widget and On-Device Integration Test Suites
- Frame Rendering Profiling, App Size and Accessibility Labels

Day 5: Lab: Building and Releasing a Publish-Ready Case App

- Case App Brief, Screen Map and Toolkit Choice Justification
 - Lab Build of Case App Screens, Navigation and State
 - Lab Integration of REST Data, Offline Cache and Camera
 - Code Signing, Store Listing Assets and Review Readiness Check
 - Automated Release Lane, Staged Rollout and Release Checklist Completion
-

Skills You Will Gain:

- Mobile Toolkit Evaluation
- Widget and Component Composition
- App State Design
- Offline Data Synchronisation
- Device Capability Integration
- Mobile Authentication Flows
- On-Device Debugging
- App Store Release Preparation

Why Attend This Course:

- Deliver a publish-ready case app with its release checklist to the product owner and the team lead responsible for mobile releases
- Decide on evidence whether a new app should be built natively or with Flutter or React Native
- Avoid store rejections, token leaks from insecure storage and data loss when users lose their connection
- Share the case app code, architecture template and release checklist with colleagues starting their first mobile project

Conclusion:

Back at work, the participant gives the product owner and the mobile team lead a publish-ready case app and a release checklist that can serve as the template for the unit's next app. Team leads use the toolkit decision matrix when approving a new mobile project, and developers reuse the layered architecture, offline cache and secure sign-in patterns. After the first real release, the unit should review store review feedback, crash and performance reports, offline sync incidents and whether the checklist caught each release issue before submission.

Frequently Asked Questions FAQ:

What should participants know before a mobile app development course with Flutter and React Native?

Participants should already write production code in at least one language, such as JavaScript, Java, C# or Python, and understand REST APIs and basic version control. No previous mobile experience is needed; the daily labs run on emulators and physical devices.

How does mobile app development with Flutter and React Native differ from a full-stack web development course?

It concentrates on apps that run on phones: widget and component screens, navigation, app lifecycle, offline data, device permissions, secure on-device storage and store release. Web courses cover browser interfaces, server APIs and databases, which this course treats only as the back end an app calls.

When is native development a better choice than Flutter or React Native for mobile app development?

Native development suits apps that depend on the newest platform features, intensive graphics or deep hardware access. Flutter and React Native suit most business apps, letting one team share most code across iOS and Android while still calling native modules where needed.

What do participants take back from the mobile app development course with Flutter and React Native?

Participants take back a publish-ready case app with layered architecture, offline caching, secure sign-in and device features, plus a release checklist, toolkit decision matrix and test suite that can be reused on their organisation's next mobile project.
