Mobile Architecture

Mobile Architecture Document

> **Project:** {{PROJECT_NAME}} **Version:** {{VERSION}} **Date:** {{DATE}}
> **Author:** {{AUTHOR}} **Status:** Draft | In Review | Approved **Reviewers:** {{REVIEWERS}}

Document History

Version	Date	Author	Changes
0.1	{{DATE}}	{{AUTHOR}}	Initial draft

---

1. Framework & Rationale

Framework	Pros	Cons	Decision
React Native (Expo)	JS codebase reuse, large ecosystem, OTA updates	Bridge overhead, some native gaps	`{{Selected / Rejected}}`
Flutter	High performance, consistent UI, strong typing	Dart language, binary size	`{{Selected / Rejected}}`
Swift (iOS native)	Best iOS performance, latest APIs	iOS only, Swift/ObjC required	`{{Selected / Rejected}}`
Kotlin (Android native)	Best Android performance, Material 3	Android only	`{{Selected / Rejected}}`

Selected: `{{FRAMEWORK}}` **Version:** `{{VERSION}}` **Rationale:**

> TODO: 3-5 sentences explaining the final decision including team skills and project requirements.

Runtime environment:

- Min iOS: `{{iOS 16}}`
- Min Android: `{{Android 10 (API 29)}}`
- Target SDK: `{{Android 34 | iOS 17}}`

---

2. Project Structure

```
{{PROJECT_NAME}}/
├── src/
│   ├── app/                # Expo Router routes (file-based) or React Navigation config
│   ├── screens/            # Full-screen components
│   │   ├── auth/
│   │   ├── home/
│   │   └── settings/
│   ├── components/
│   │   ├── ui/             # Design system primitives
│   │   └── features/       # Feature-specific components
│   ├── navigation/         # Navigator definitions, types
│   ├── hooks/              # Shared custom hooks
│   ├── services/           # API client, native integrations
│   ├── stores/             # State management slices
│   ├── utils/              # Pure helpers
│   ├── constants/          # App-wide constants
│   └── types/              # TypeScript interfaces
├── ios/                    # iOS native code (Xcode project)
├── android/                # Android native code (Gradle project)
├── assets/                 # Images, fonts, icons
├── app.config.ts           # Expo config or equivalent
└── package.json
```

TODO: Update to match actual project structure.

---

3. Navigation Architecture

```
graph TD
    Root["Root Navigator"] --> Auth["Auth Stack\n(Unauthenticated)"]
```

```
    Root --> App["App Navigator\n(Authenticated)"]

    Auth --> Login["Login Screen"]
    Auth --> Register["Register Screen"]
    Auth --> ForgotPassword["Forgot Password"]

    App --> BottomTabs["Bottom Tab Navigator"]
    App --> Modal["Modal Stack"]

    BottomTabs --> Home["Home Tab\n(Stack)"]
    BottomTabs --> Explore["Explore Tab\n(Stack)"]
    BottomTabs --> Profile["Profile Tab\n(Stack)"]
    BottomTabs --> Settings["Settings Tab\n(Stack)"]

    Home --> HomeScreen["Home Screen"]
    Home --> DetailScreen["Detail Screen"]

    Modal --> ImageViewer["Image Viewer"]
    Modal --> ShareSheet["Share Sheet"]
```

Navigation library: `{{React Navigation v7 | Expo Router v4}}`

Deep link handling:

- URL scheme: `{{appname://}}`
- Universal links: `{{https://app.domain.com}}`
- See `push-notification-design.md` for notification deep links

Auth flow: Root navigator listens to auth state — no manual navigation required on login/logout.

---

4. Platform-Specific Considerations

Concern	iOS	Android	Solution
Back gesture	Swipe from edge	Back button + gesture	`hardwareBackPress` handler
Status bar	Overlaps content	Separate space	`SafeAreaView` + `StatusBar`
Permissions model	Request at use time	Request at use time + Manifest	`react-native-permissions`
Push notifications	APNs	FCM	Abstraction layer
Keyboard behavior	Push up content	May or may not push	`KeyboardAvoidingView`

Concern	iOS	Android	Solution
Font rendering	System fonts crisp	Sub-pixel differences	Custom font loading
Haptics	`UIFeedbackGenerator`	`Vibrator` API	`expo-haptics`
Secure storage	Keychain	Keystore	`expo-secure-store`

TODO: Add platform differences discovered during development.

---

5. Build Variants & Flavors

Variant	Bundle ID	API URL	Debug	Analytics	Push Env
Dev	`{{com.company.app.dev}}`	`http://localhost:4000`	Yes	Off	Development
Staging	`{{com.company.app.staging}}`	`https://api-staging.domain.com`	No	Off	Development
Production	`{{com.company.app}}`	`https://api.domain.com`	No	Yes	Production

Environment variable handling: `{{expo-constants | react-native-config}}`

TODO: Link to `.env.example` for each variant.

---

6. Code Sharing Strategy

```
shared/           # 80% of codebase — platform-agnostic logic
  ├── hooks/      # Custom hooks
  ├── services/   # API, storage abstractions
  ├── stores/     # State management
  └── utils/      # Pure utilities

platform/
  ├── ios/        # iOS-specific implementations
  └── android/    # Android-specific implementations

components/
  ├── Button.tsx          # Shared component
  ├── Button.ios.tsx      # iOS-specific override (if needed)
```

```
    └── Button.android.tsx   # Android-specific override (if needed)
```

Platform file resolution: Bundler automatically resolves `.ios.tsx` / `.android.tsx` before `.tsx`.

---

7. Native Module Integration

Module	Purpose	Source	Platform
`expo-camera`	QR scanning, photo capture	Expo SDK	Both
`expo-location`	Geolocation	Expo SDK	Both
`expo-notifications`	Push notifications	Expo SDK	Both
`expo-secure-store`	Keychain/Keystore	Expo SDK	Both
`expo-biometrics`	Face ID / fingerprint	Expo SDK	Both
`{{custom-native-module}}`	`{{PURPOSE}}`	Custom	`{{iOS/Android/Both}}`

New native module process:

1. Evaluate if Expo SDK covers the need
2. Check community modules (well-maintained, typed)
3. Write custom module as last resort
4. Native module must have TypeScript wrapper with full types

---

8. Performance Optimization Strategy

Strategy	Implementation	Status
JS thread optimization	Move heavy computation to worklets (Reanimated)	`{{Done/Planned}}`
Image caching	`expo-image` with disk cache	`{{Done/Planned}}`
List performance	`FlashList` instead of `FlatList`	`{{Done/Planned}}`
Bundle size	Hermes engine enabled, tree shaking	`{{Done/Planned}}`
JS startup time	Lazy loading of non-critical screens	`{{Done/Planned}}`
Memory management	Subscription cleanup in `useEffect`	`{{Done/Planned}}`
Render optimization	`React.memo`, `useCallback` on list items	`{{Done/Planned}}`

Performance targets:

- App cold start to interactive: < 2 seconds
- Screen transition: < 300ms
- List scroll: 60fps (120fps on ProMotion)
- Bundle size: < 20MB (iOS), < 15MB (Android)

---

9. Crash Reporting & Analytics Integration

Service	Purpose	Library
`{{Sentry}}`	Crash reporting, error tracking	`@sentry/react-native`
`{{Firebase Analytics}}`	User analytics, funnel tracking	`@react-native-firebase/analytics`
`{{Datadog}}`	APM, performance monitoring	`@datadog/mobile-react-native`

Privacy rules:

- No PII in analytics events
- User ID: hashed, not raw
- Comply with GDPR — analytics requires consent
- Crash reports: scrub sensitive data from breadcrumbs

Event naming convention: `{screen}_{action}` — e.g., `checkout_payment_started`

---

10. CI/CD for Mobile

```
flowchart LR
    PR["Pull Request"] --> UnitTests["Unit Tests\n(Jest)"]
    UnitTests --> E2ETests["E2E Tests\n(Detox / Maestro)"]
    E2ETests --> Build["Build\n(EAS Build / Fastlane)"]
    Build --> Distribute["Distribute\n(TestFlight / Firebase App Distrib)"]
    Distribute --> QA["QA Approval"]
    QA --> Store["Store Submission\n(EAS Submit / Fastlane deliver)"]
```

Stage	Tool	Trigger
Build	`{{EAS Build / Fastlane}}`	Push to `main` or `release/*`
Test distribution	`{{TestFlight / Firebase App Distribution}}`	Every staging build

Stage	Tool	Trigger
E2E tests	`{{Detox / Maestro}}`	PR checks
App Store submit	`{{EAS Submit / Fastlane deliver}}`	Manual trigger (release manager)
Code signing	`{{Expo managed / Fastlane match}}`	Automated

TODO: Link to CI configuration files in `.github/workflows/` or Fastfile.

---

11. Architecture Diagram

```
graph TB
    subgraph "Mobile App"
        UI["Screens & Components"]
        Nav["Navigation Layer"]
        State["State Management"]
        Services["Service Layer"]
        Native["Native Modules"]
    end

    subgraph "External"
        API["REST API"]
        Auth["Auth Service"]
        Push["Push Service\n(APNs / FCM)"]
        Analytics["Analytics\n(Firebase / Sentry)"]
    end

    UI --> Nav
    UI --> State
    State --> Services
    Services --> API
    Services --> Auth
    Native --> Push
    Services --> Analytics
```

Approval

Role	Name	Date	Signature

Author			
Mobile Lead			
Tech Lead			
Product Owner			

Revision #3
Created 2026-02-24 15:16:27 UTC by John
Updated 2026-05-25 07:33:04 UTC by John