

1DreamUnited Mobile Development Guide

Platform Overview

The 1DreamUnited mobile application is built using React Native with Expo, providing native iOS and Android experiences with shared codebase efficiency.

Development Environment Setup

Prerequisites

- **Node.js** 18.0.0 or higher
- **npm** or **yarn** package manager
- **Git** for version control

iOS Development (macOS only)

- **Xcode** 14.0 or higher
- **iOS Simulator** (included with Xcode)
- **CocoaPods** (installed automatically)

Android Development

- **Android Studio** with Android SDK
- **Android Virtual Device (AVD)** or physical device
- **Java Development Kit (JDK)** 11 or higher

Expo Development Tools

```
# Install Expo CLI globally
npm install -g @expo/cli

# Install EAS CLI for building and deployment
npm install -g eas-cli

# Login to Expo account (required for builds)
expo login
eas login
```

Getting Started

1. Project Setup

```
# Navigate to mobile project
cd mobile/react_native/

# Install dependencies
npm install

# Start development server
npm start
```

2. Running on Devices

iOS Simulator

```
npm run ios  
# or  
expo start --ios
```

Android Emulator/Device

```
npm run android  
# or  
expo start --android
```

Web Browser (for testing)

```
npm run web  
# or  
expo start --web
```

Build Process

Development Builds

```
# Create development build for testing  
expo prebuild --clean  
  
# Build for specific platform  
expo prebuild --platform ios  
expo prebuild --platform android
```

Production Builds with EAS

```
# Configure EAS (first time only)  
eas build:configure  
  
# Build for both platforms  
npm run build:all  
# or  
eas build --platform all  
  
# Platform-specific builds  
npm run build:ios  
npm run build:android
```

Local Development Builds

```
# Generate native code for local development  
npm run prebuild  
  
# Clean and regenerate  
npm run prebuild:clean
```

Project Structure

```
mobile/react_native/
├── src/
│   ├── components/           # Reusable UI components
│   │   ├── AIEcosystemCard.tsx
│   │   ├── BandMemberCard.tsx
│   │   ├── DRMVideoPlayer.tsx
│   │   └── ...
│   ├── screens/             # Screen components
│   │   ├── HomeScreen.tsx
│   │   ├── AIEcosystemScreen.tsx
│   │   └── ...
│   ├── sections/            # Page sections
│   ├── contexts/            # React contexts
│   ├── hooks/               # Custom React hooks
│   ├── services/            # API and business logic
│   ├── styles/              # Global styles
│   ├── utils/               # Utility functions
│   ├── assets/              # Images, fonts, etc.
│   ├── App.tsx              # Main app component
│   ├── app.json             # Expo configuration
│   └── package.json         # Dependencies and scripts
```

Key Features Implementation

AI Ecosystem Integration

```
// Example: Using AI ecosystem hook
import { useAIEcosystem } from '../hooks/useAIEcosystem';

const AIScreen = () => {
  const { recommendations, loading } = useAIEcosystem();

  return (
    <View>
      {recommendations.map(item => (
        <AIEcosystemCard key={item.id} data={item} />
      ))}
    </View>
  );
};
```

DRM Video Streaming

```
// Example: DRM video player usage
import { DRMVideoPlayer } from '../components/DRMVideoPlayer';

const VideoScreen = () => {
  return (
    <DRMVideoPlayer
      source={{ uri: 'https://dr33mtv.com/stream/video.m3u8' }}
      drm={{
        type: 'fairplay', // or 'widevine' for Android
        licenseServer: 'https://license.dr33mtv.com'
      }}
    />
  );
};
```

Multi-Language Support

```
// Example: Language context usage
import { useLanguage } from '../contexts/LanguageContext';

const Component = () => {
  const { t, currentLanguage, setLanguage } = useLanguage();

  return (
    <Text>{t('welcome_message')}</Text>
  );
};
```

Testing

Unit Testing

```
# Run all tests
npm test

# Run tests in watch mode
npm run test:watch

# Generate coverage report
npm run test:coverage
```

E2E Testing with Detox (Optional)

```
# Install Detox CLI
npm install -g detox-cli

# Build for testing
detox build --configuration ios.sim.debug

# Run E2E tests
detox test --configuration ios.sim.debug
```

Configuration

Expo Configuration (app.json)

```
{
  "expo": {
    "name": "1DreamUnited",
    "slug": "1dreamunited-mobile",
    "version": "1.0.0",
    "orientation": "portrait",
    "icon": "./assets/icon.png",
    "userInterfaceStyle": "dark",
    "splash": {
      "image": "./assets/splash.png",
      "resizeMode": "contain",
      "backgroundColor": "#000000"
    },
    "ios": {
      "supportsTablet": true,
      "bundleIdentifier": "com.1dreamunited.mobile"
    },
    "android": {
      "package": "com.onedreamunited.mobile",
      "adaptiveIcon": {
        "foregroundImage": "./assets/adaptive-icon.png",
        "backgroundColor": "#000000"
      }
    }
  }
}
```

EAS Build Configuration (eas.json)

```
{
  "cli": {
    "version": ">= 3.0.0"
  },
  "build": {
    "development": {
      "developmentClient": true,
      "distribution": "internal"
    },
    "preview": {
      "distribution": "internal"
    },
    "production": {}
  },
  "submit": {
    "production": {}
  }
}
```

Deployment

App Store (iOS)

```
# Build for App Store
eas build --platform ios --profile production

# Submit to App Store
eas submit --platform ios
```

Google Play Store (Android)

```
# Build for Play Store
eas build --platform android --profile production

# Submit to Play Store
eas submit --platform android
```

Internal Distribution

```
# Build for internal testing
eas build --platform all --profile preview

# Share build with testers
# EAS will provide shareable links
```

Troubleshooting

Common Issues

Metro bundler issues

```
# Clear Metro cache
npx expo start --clear

# Reset npm cache
npm start -- --reset-cache
```

iOS build issues

```
# Clean iOS build
cd ios && xcodebuild clean

# Update CocoaPods
cd ios && pod install --repo-update
```

Android build issues

```
# Clean Android build
cd android && ./gradlew clean

# Reset Android project
expo prebuild --platform android --clean
```

Performance Optimization

- Use `React.memo` for expensive components
- Implement lazy loading for screens
- Optimize images and assets
- Use Flipper for debugging performance

Memory Management

- Avoid memory leaks in `useEffect` hooks
- Properly cleanup event listeners
- Use `FlatList` for large lists
- Optimize image loading and caching

Analytics Integration

Expo Analytics

```
import * as Analytics from 'expo-analytics';

// Track screen views
Analytics.track('Screen View', {
  screen: 'Home',
  user_id: userId
});

// Track user actions
Analytics.track('Music Play', {
  song_id: songId,
  artist: artistName
});
```

Security Best Practices

API Security

- Use HTTPS for all API calls
- Implement proper authentication tokens
- Validate all user inputs
- Use secure storage for sensitive data

DRM Implementation

- Implement proper license validation
- Use encrypted storage for DRM keys
- Handle license renewal automatically

- Implement offline playback securely

Platform-Specific Considerations

iOS Specific

- Follow Apple Human Interface Guidelines
- Implement proper App Store review guidelines
- Handle iOS-specific permissions properly
- Test on various iOS device sizes

Android Specific

- Follow Material Design guidelines
- Handle Android permissions system
- Test on various Android versions
- Optimize for different screen densities

Continuous Integration

GitHub Actions Example

```
name: Mobile CI/CD
on: [push, pull_request]

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-node@v3
        with:
          node-version: '18'
      - run: npm install
      - run: npm run lint
      - run: npm run test
      - run: eas build --platform all --non-interactive
```

Additional Resources

- [Expo Documentation](https://docs.expo.dev/) (https://docs.expo.dev/)
- [React Native Documentation](https://reactnative.dev/docs/getting-started) (https://reactnative.dev/docs/getting-started)
- [EAS Build Documentation](https://docs.expo.dev/build/introduction/) (https://docs.expo.dev/build/introduction/)
- [React Navigation](https://reactnavigation.org/) (https://reactnavigation.org/)
- [Expo AV for Video/Audio](https://docs.expo.dev/versions/latest/sdk/av/) (https://docs.expo.dev/versions/latest/sdk/av/)

For technical support or questions, refer to the main project documentation or contact the development team.