Mobile Video Converter Product Requirements Document (PRD)

Date	Version	Description	Author
Sep 16, 2025	2.0	Mobile Android Version	John, PM

Goals and Background Context

Goals

- **Simplicity**: Provide a user experience where a video can be converted with a minimum number of taps on mobile.
- **Efficiency**: Create a high-quality, web-optimized MP4 that is significantly smaller than the source file using mobile device processing power.
- **Satisfaction**: Achieve high user satisfaction by being a reliable, easy-to-use mobile tool for its single purpose.
- **Portability**: Deliver the application as a single `.apk` file that can be installed on Android devices without requiring Google Play Store distribution.
- **Offline Operation**: Ensure the app works completely offline without requiring internet connectivity for video processing.

Background Context

Many mobile users need a simple tool to convert video files for web use, social media sharing, or storage optimization without relying on cloud services or complex mobile apps. This project aims to solve that problem by creating a focused Android mobile application that does one thing well: it converts any video file into a web-optimized MP4 using the device's own processing power. The project will leverage React Native or Flutter framework to achieve cross-platform compatibility with native performance.

Requirements

Functional

1. **FR1**: The user must be able to select a video file from their device's gallery, camera roll, or file system.
2. **FR2**: The application must convert the selected video into a web-optimized MP4 file using only device processing power (no cloud processing).
3. **FR3**: The application must provide a real-time visual indicator of the conversion progress with estimated time remaining.
4. **FR4**: Upon successful conversion, the file is saved to the device's gallery or a designated app folder with user confirmation.
5. **FR5**: The application must contain an "About" section accessible through settings, displaying information about the developer and app version.

6. **FR6**: The user must be able to cancel a conversion that is in progress. The application should safely terminate the process and clean up temporary files.
7. **FR7**: The app must handle device rotation and maintain conversion state during orientation changes.
8. **FR8**: The app must manage device resources efficiently to prevent overheating and battery drain during conversion.

Non-Functional

1. **NFR1**: The user interface must be clean, simple, and follow Material Design guidelines for Android.
2. **NFR2**: The application must be built using React Native or Flutter framework for cross-platform compatibility.
3. **NFR3**: The final application must be distributed as a single `.apk` file that can be sideloaded without Google Play Store.
4. **NFR4**: The video conversion process must produce optimal quality while being mindful of mobile device processing limitations.
5. **NFR5**: The app must work entirely offline without requiring internet connectivity.
6. **NFR6**: The app must request only necessary permissions (storage, camera access if needed).
7. **NFR7**: The app must be compatible with Android 8.0 (API level 26) and above.
8. **NFR8**: The app must handle low storage scenarios gracefully with appropriate user warnings.

User Interface Design Goals

Overall UX Vision

The application should be minimalist, touch-friendly, and intuitive for mobile users. The interface should be optimized for one-handed use where possible. The entire user journey should be smooth and require minimal interactions.

Key Interaction Paradigms

- **File Selection**: Users can select videos from gallery/camera roll through native Android file picker or capture new video directly.
- **Conversion**: A single, prominent button initiates conversion with clear visual feedback.
- **Mobile-First**: All interactions designed for touch interface with appropriate touch targets (minimum 44dp).

Core Screens and Views

1. **Main Screen**: Single primary screen with file selection, conversion controls, and progress display
2. **Settings Screen**: Simple settings including output quality options and about information
3. **Results Screen**: Post-conversion summary with options to share, save, or convert another file

Accessibility

- **Target**: Android Accessibility Guidelines. Support for TalkBack, large text, and high contrast modes.
- **Touch Targets**: All interactive elements minimum 48dp as per Material Design guidelines.

Branding

- The application will have a clean, modern Material Design aesthetic suitable for Android ecosystem.

Target Device and Platforms

- **Platform**: Android 8.0+ (API level 26 and above)
- **Architecture**: ARM64 and ARM32 support
- **Screen Sizes**: Phone and tablet layouts (responsive design)

Technical Assumptions

Repository Structure

- **Monorepo**: A single repository for the self-contained mobile application.

Service Architecture

- **Monolith**: Single mobile application with all processing done locally on device.
- **Offline-First**: No server dependencies or cloud processing.

Testing Requirements

- **Unit + Integration + Device Testing**: Include unit tests, integration tests, and real device testing across various Android versions and hardware configurations.

Additional Technical Assumptions and Requests

- The application **must** be built using **React Native** or **Flutter** framework for efficient mobile development.
- Video processing **must** utilize device's native capabilities (hardware acceleration where available).
- **FFmpeg** or similar mobile-optimized video processing library integration required.
- App must handle various video formats commonly found on mobile devices.

Epic List

1. **Epic 1: Core Mobile App Foundation & Video Processing**: Establish the mobile application foundation, implement touch-optimized UI, and build core video conversion functionality.
2. **Epic 2: Mobile-Specific Features & Optimization**: Add mobile-specific features like device resource management, orientation handling, and performance optimization.
3. **Epic 3: App Polish & Distribution Preparation**: Implement settings, about screen, and prepare APK for distribution.

Epic 1: Core Mobile App Foundation & Video Processing

Epic Goal: Deliver a functional mobile video converter with intuitive touch interface and reliable offline video processing capabilities.

Story 1.1: Mobile App Setup and Main Screen

As a mobile user,
I want to open the app and see a simple, touch-friendly interface,
so that I can easily start converting videos on my phone.

Acceptance Criteria

1. A new React Native/Flutter project is created and configured for Android.
2. App launches with a clean, Material Design-compliant main screen.
3. Screen is responsive and works in both portrait and landscape orientations.
4. App requests necessary storage permissions on first launch.

Story 1.2: Implement Mobile File Selection

As a mobile user,
I want to select videos from my gallery or capture new ones,
so that I can choose any video on my device for conversion.

Acceptance Criteria

1. Main screen displays a prominent "Select Video" button with gallery icon.
2. Tapping the button opens native Android file picker with video filter.
3. Option to access recent videos from gallery with thumbnail preview.
4. Selected video displays thumbnail, filename, and file size.
5. "Record New Video" option opens camera for direct video capture.

Story 1.3: Implement Mobile-Optimized Conversion Process

As a mobile user,
I want the conversion to work efficiently on my device without overheating,
so that I can convert videos reliably using my phone's processing power.

Acceptance Criteria

1. "Convert" button becomes active after video selection.
2. Conversion process shows progress bar, percentage, and estimated time remaining.
3. Device thermal management prevents overheating during processing.
4. Battery usage is optimized during conversion process.
5. Conversion can be paused/resumed if user switches apps or receives calls.
6. "Cancel" button stops conversion and cleans up temporary files.
7. Converted file is saved to device gallery/downloads with appropriate naming.

Epic 2: Mobile-Specific Features & Optimization

Epic Goal: Enhance the app with mobile-specific capabilities and ensure optimal performance across different Android devices.

Story 2.1: Device Resource Management

As a mobile user,
I want the app to work smoothly without draining my battery or slowing down my device,
so that I can use other apps while conversion happens in background.

Acceptance Criteria

1. App monitors device temperature and throttles processing if needed.
2. Battery usage is displayed and conversion can be scheduled for charging time.
3. Low storage warnings appear before starting conversion.
4. Background processing continues when app is minimized (with notification).
5. App gracefully handles interruptions (calls, low battery, etc.).

Story 2.2: Mobile UX Enhancements

As a mobile user,
I want the app to work seamlessly with my device's features,
so that I have the best possible mobile experience.

Acceptance Criteria

1. Share functionality to directly share converted videos to other apps.
2. Orientation changes maintain conversion state and progress.
3. Pull-to-refresh gesture on main screen.
4. Haptic feedback for button interactions.
5. Dark mode support following system theme.
6. Swipe gestures for navigation between screens.

Epic 3: App Polish & Distribution Preparation

Epic Goal: Add final features and prepare the application for APK distribution outside of Google Play Store.

Story 3.1: Settings and About Screen

As a mobile user,
I want to access app settings and information,
so that I can customize the app behavior and learn about the developer.

Acceptance Criteria

1. Settings screen accessible via hamburger menu or settings icon.
2. Options for output quality (High, Medium, Low) selection.
3. Toggle for auto-save to gallery vs. manual save location.
4. About section with app version, developer info, and open source licenses.
5. Help section with simple usage instructions.

Story 3.2: APK Packaging and Distribution Preparation

As a developer,
I want to package the app for easy distribution outside Google Play Store,
so that users can install it directly on their Android devices.

Acceptance Criteria

1. Build process configured to generate signed APK file.
2. APK works on target Android versions (8.0+) without requiring Play Store.
3. App handles installation from unknown sources appropriately.
4. All necessary permissions are properly declared in manifest.
5. APK size is optimized for easy sharing and download.
6. Testing completed on various Android devices and versions.

Story 3.3: Performance Testing and Optimization

As a user,
I want the app to work reliably across different Android devices,
so that I can convert videos regardless of my phone's specifications.

Acceptance Criteria

1. App tested on low-end, mid-range, and high-end Android devices.
2. Conversion times are reasonable across device performance spectrum.
3. Memory usage optimized to prevent crashes on devices with limited RAM.
4. App provides fallback options for unsupported video formats.
5. Error handling for corrupted or problematic video files.

Checklist Results Report

- **Result**: **PASS**
- **Summary**: The mobile PRD addresses the unique challenges and opportunities of mobile video conversion. Requirements are adapted for touch interfaces, device resource management, and offline operation. The epic structure progresses logically from core functionality to mobile-specific optimizations.

Next Steps

This PRD will be handed off to the Mobile UX Expert and the Mobile App Architect for detailed technical and design specifications.

Mobile UX Expert Prompt

> Based on the attached PRD for the 'Mobile Video Converter', please create a detailed Mobile UI/UX Specification. Focus on touch-optimized interfaces, Material Design compliance, and mobile-first user flows. Consider various Android screen sizes and one-handed usage patterns.

Mobile App Architect Prompt

> Based on the attached PRD, please create a Mobile Architecture document for the 'Mobile Video Converter'. Address React Native/Flutter implementation, offline video processing with FFmpeg integration, Android-specific optimizations, and APK distribution strategy.