Feature Specification: Mobile Video Converter Android App

Feature Branch: `001-mobile-video-converter`
Created: September 17, 2025
Status: Draft
Input: User description: "Mobile Video Converter Android App - A comprehensive mobile application for offline video conversion to web-optimized MP4 format with touch-optimized UI, device resource management, and APK distribution capabilities"

Execution Flow (main)

```
1. Parse user description from Input
   → Parsed: Mobile video converter app for Android with offline processing
2. Extract key concepts from description
   → Actors: Mobile users
   → Actions: Select videos, convert to MP4, manage device resources
   → Data: Video files, conversion settings, app preferences
   → Constraints: Offline operation, APK distribution, touch interface
3. For each unclear aspect:
   → All requirements clarified from PRD
4. Fill User Scenarios & Testing section
   → Clear user flow: Select video → Convert → Save/Share
5. Generate Functional Requirements
   → All requirements are testable and derived from PRD
6. Identify Key Entities
   → Video files, conversion jobs, app settings, device resources
7. Run Review Checklist
   → No ambiguities, no implementation details
8. Return: SUCCESS (spec ready for planning)
```

⚡ Quick Guidelines

- ☑ Focus on WHAT users need and WHY
- ❌ Avoid HOW to implement (no tech stack, APIs, code structure)
- 👥 Written for business stakeholders, not developers

User Scenarios & Testing

Primary User Story

A mobile user wants to convert a video file on their Android device to a smaller, web-optimized MP4 format for sharing on social media or saving storage space. They open the mobile video converter app,

select a video from their gallery or record a new one, initiate the conversion process, and receive the optimized video file saved to their device - all without requiring internet connectivity.

Acceptance Scenarios

1. **Given** a user has the app installed and launched, **When** they tap "Select Video", **Then** the native Android file picker opens showing only video files
2. **Given** a user has selected a video file, **When** they tap "Convert", **Then** a progress indicator shows conversion status with estimated time remaining
3. **Given** a conversion is in progress, **When** the user rotates their device, **Then** the conversion continues and progress is maintained
4. **Given** a conversion is in progress, **When** the user taps "Cancel", **Then** the process stops safely and temporary files are cleaned up
5. **Given** a conversion completes successfully, **When** the user views the result, **Then** they can share the video or save it to their gallery
6. **Given** the device has low storage, **When** the user tries to convert a large video, **Then** the app warns about insufficient space before starting
7. **Given** the device is getting hot during conversion, **When** thermal limits are reached, **Then** the app throttles processing to prevent overheating

Edge Cases

- What happens when the app is interrupted by a phone call during conversion?
- How does the system handle corrupted or unsupported video files?
- What occurs if the device runs out of battery during conversion?
- How does the app behave when storage becomes full during the conversion process?
- What happens if the user force-closes the app while conversion is running?

Requirements

Functional Requirements

- **FR-001**: System MUST allow users to select video files from device gallery, camera roll, or file system
- **FR-002**: System MUST allow users to record new videos directly within the app
- **FR-003**: System MUST convert selected videos to web-optimized MP4 format using only device processing power
- **FR-004**: System MUST display real-time conversion progress with percentage complete and estimated time remaining
- **FR-005**: System MUST allow users to cancel ongoing conversions with safe cleanup of temporary files
- **FR-006**: System MUST save converted videos to device gallery or designated app folder with user confirmation
- **FR-007**: System MUST maintain conversion state and progress during device orientation changes
- **FR-008**: System MUST monitor device temperature and throttle processing to prevent overheating
- **FR-009**: System MUST optimize battery usage during conversion process

- **FR-010**: System MUST continue background processing when app is minimized with appropriate notifications
- **FR-011**: System MUST gracefully handle interruptions from calls, low battery, and other system events
- **FR-012**: System MUST provide share functionality to send converted videos to other apps
- **FR-013**: System MUST support dark mode following system theme preferences
- **FR-014**: System MUST provide haptic feedback for button interactions
- **FR-015**: System MUST include settings screen with output quality options (High, Medium, Low)
- **FR-016**: System MUST include About section with app version and developer information
- **FR-017**: System MUST warn users when device storage is insufficient for conversion
- **FR-018**: System MUST handle various common mobile video formats as input
- **FR-019**: System MUST work entirely offline without internet connectivity requirements
- **FR-020**: System MUST follow Material Design guidelines for Android
- **FR-021**: System MUST be compatible with Android 8.0 (API level 26) and above
- **FR-022**: System MUST request only necessary permissions (storage, camera access)
- **FR-023**: System MUST provide error handling for corrupted or problematic video files
- **FR-024**: System MUST optimize memory usage to prevent crashes on limited RAM devices
- **FR-025**: System MUST support both ARM64 and ARM32 device architectures

Key Entities

- **Video File**: Represents source and converted video files with attributes like filename, file size, format, duration, and file path
- **Conversion Job**: Represents an active or completed conversion process with status, progress percentage, estimated time, quality settings, and device resource usage
- **App Settings**: User preferences including output quality level, auto-save location, theme preference, and notification settings
- **Device Resources**: System monitoring data including temperature, battery level, available storage, and memory usage

---

Review & Acceptance Checklist

Content Quality

- [x] No implementation details (languages, frameworks, APIs)
- [x] Focused on user value and business needs
- [x] Written for non-technical stakeholders
- [x] All mandatory sections completed

Requirement Completeness

- [x] No [NEEDS CLARIFICATION] markers remain
- [x] Requirements are testable and unambiguous
- [x] Success criteria are measurable
- [x] Scope is clearly bounded
- [x] Dependencies and assumptions identified

Execution Status

- [x] User description parsed
- [x] Key concepts extracted
- [x] Ambiguities marked
- [x] User scenarios defined
- [x] Requirements generated
- [x] Entities identified
- [x] Review checklist passed