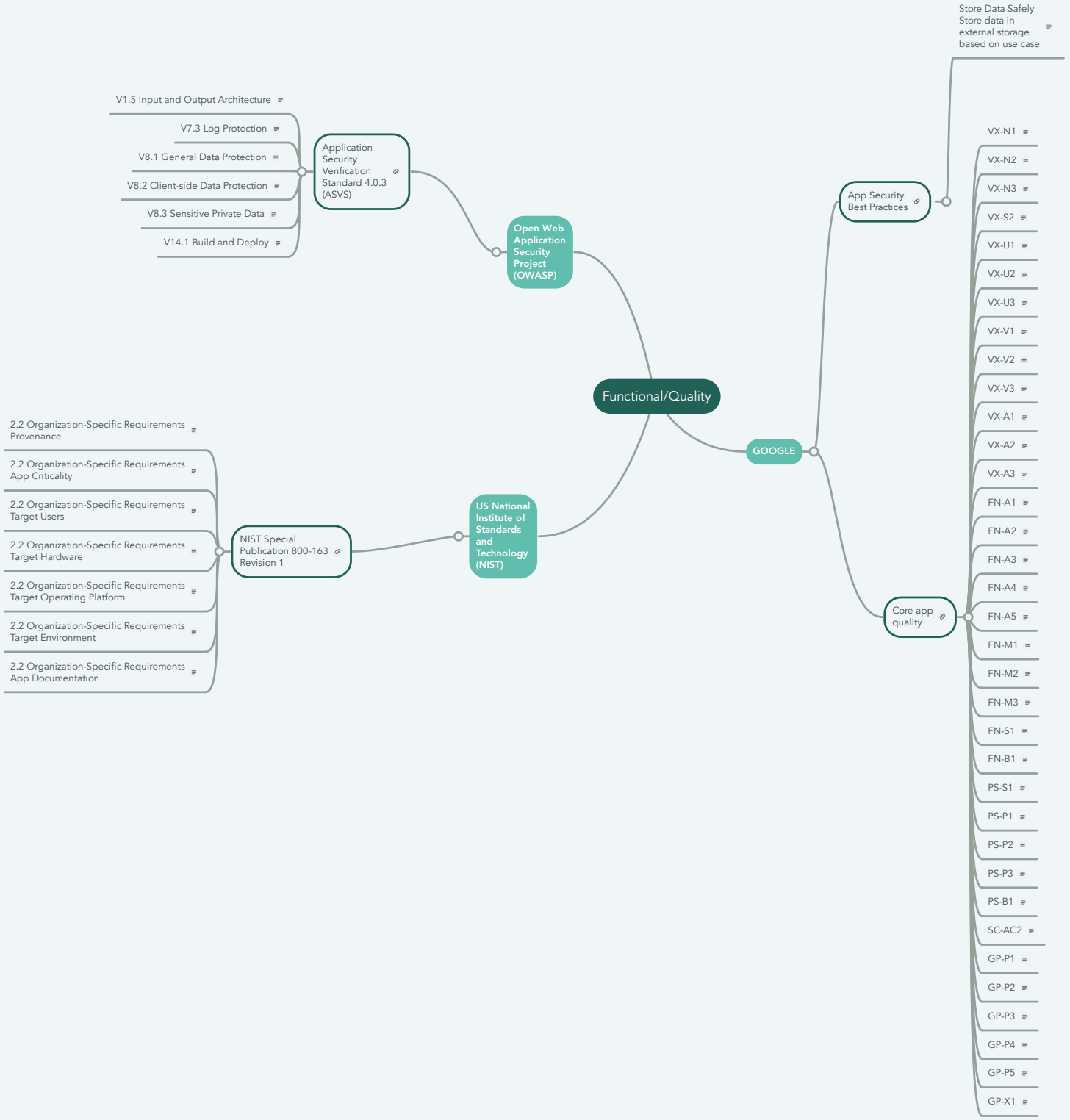
Functional/Quality
GOOGLE
App Security Best Practices
Store Data Safely
Store data in external storage based on use case
Core app quality
VX-N1
VX-N2
VX-N3
VX-S2
VX-U1
VX-U2
VX-U3
VX-V1
VX-V2
VX-V3
VX-A1
VX-A2
VX-A3
FN-A1
FN-A2
FN-A3
FN-A4
FN-A5
FN-M1
FN-M2
FN-M3
FN-S1
FN-B1
PS-S1
PS-P1
PS-P2
PS-P3
PS-B1
SC-AC2
GP-P1
GP-P2
GP-P3
GP-P4
GP-P5
GP-X1
Open Web Application Security Project (OWASP)
Application Security Verification Standard 4.0.3 (ASVS)
V1.5 Input and Output Architecture
V7.3 Log Protection
V8.1 General Data Protection
V8.2 Client-side Data Protection
V8.3 Sensitive Private Data
V14.1 Build and Deploy
US National Institute of Standards and Technology (NIST)
NIST Special Publication 800-163 Revision 1
2.2 Organization-Specific Requirements Provenance
2.2 Organization-Specific Requirements App Criticality
2.2 Organization-Specific Requirements Target Users
2.2 Organization-Specific Requirements Target Hardware
2.2 Organization-Specific Requirements Target Operating Platform
2.2 Organization-Specific Requirements Target Environment
2.2 Organization-Specific Requirements App Documentation

Functional/Quality

1. GOOGLE

1.1. Core app quality

Link: https://developer.android.com/docs/quality-guidelines/core-app-quality

1.1.1. VX-N1

> The app supports standard Back button navigation and does not make use of any custom, on-screen "Back button" prompts.

1.1.2. VX-N2

> The app supports gesture navigation for going back / going to the home screen.

1.1.3. VX-N3

The app correctly preserves and restores user or app state.

The app preserves user or app state when leaving the foreground and prevents accidental data loss due to back-navigation and other state changes.

When returning to the foreground, the app should restore the preserved state and any significant stateful transaction that was pending. Examples include: changes to editable fields, game progress, menus, videos, and other sections of the app or game.

When the app is resumed from the Recents app switcher, the app returns the user to the exact state in which it was last used.

When the app is resumed after the device wakes from the sleep (locked) state, the app returns the user to the exact state in which it was last used.

When the app is relaunched from Home or All Apps, it should do one of the following, depending on how much time has passed since it was last used:
If the app was last used a short time ago (minutes), restore the app state as close as possible to its previous state.
If more time has passed since the app was last used, try to restore the app as close as possible to its previous state; or start it from its home screen or some other default state.

1.1.4. VX-S2

For messaging apps, social apps and conversations:

Use the MessagingStyle notifications for conversations.
Support the direct reply action.
Support conversation shortcuts, and implement best practices for getting the best direct share ranking.
Support bubbles.

1.1.5. VX-U1

The app supports both landscape and portrait orientations (if possible) and folding / unfolding.
Orientations expose largely the same features and actions and preserve functional parity. Minor changes in content or views are acceptable.

1.1.6. VX-U2

The app uses the whole screen in both orientations and does not letterbox to account for orientation changes, including folding and unfolding. Minor letterboxing to compensate for small variations in screen geometry is acceptable.

1.1.7. VX-U3

The app correctly handles rapid transitions between display orientations and device folding / unfolding without rendering problems or losing state.

1.1.8. VX-V1

The app displays graphics, text, images, and other UI elements without noticeable distortion, blurring, or pixelation.

The app should use vector drawables where possible.
The app provides high-quality graphics for all targeted screen sizes and form factors.
No aliasing at the edges of menus, buttons, and other UI elements is visible.

1.1.9. VX-V2

The app displays text and text blocks in an acceptable manner for each of the app's supported languages.

Composition is acceptable in all supported form factors.
No cut-off letters or words are visible.
No improper word wraps within buttons or icons are visible.
There is sufficient spacing between text and surrounding elements.

1.1.10. VX-V3

The app's content, and all web contents referred to by the app, support dark theme.

1.1.11. VX-A1

Touch targets should be at least 48dp in size.

1.1.12. VX-A2

The app’s text and foreground content should maintain a high enough color contrast ratio with its background:

3.0:1 for large text / graphics
4.5:1 for small text (text smaller than 18pt, or if the text is bold and smaller than 14pt)

1.1.13. VX-A3

Describe each UI element, except for TextView, using contentDescription.

1.1.14. FN-A1

Audio resumes when the app returns to the foreground, or indicates to the user that playback is in a paused state.

1.1.15. FN-A2

If audio playback is a core feature, the app should support background playback.

1.1.16. FN-A3

When the user initiates audio playback, the app should do one of the following within one second:

Start playing the audio.
Provide a visual indicator that the audio data is being prepared.

1.1.17. FN-A4

The app should request audio focus when audio starts playing and abandon audio focus when playback stops.

1.1.18. FN-A5

The app should handle other apps’ requests for audio focus. For example, an app might reduce playback volume when another app plays speech.

1.1.19. FN-M1

If the app plays audio in the background, it must create a Notification styled with MediaStyle.

1.1.20. FN-M2

If the app plays video, it should support picture-in-picture playback.

1.1.21. FN-M3

If the app encodes video, it should do so using the HEVC video compression standard.

1.1.22. FN-S1

The app should use the Android Sharesheet when sharing content. It can suggest targets that are unavailable to custom solutions.

1.1.23. FN-B1

The app avoids running unnecessarily long services in the background. To ensure the smooth running of the user's device, the system applies various restrictions on background services. These are not considered good uses of background services:
Maintaining a network connection for notifications
Maintaining a Bluetooth connection
Keeping the GPS powered-on

1.1.24. PS-S1

The app does not crash or block the UI thread causing ANR (Android Not Responding") errors. Utilize Google Play's pre-launch report to identify potential stability issues. After deployment, pay attention to the Android Vitals page in the Google Play developer console.

1.1.25. PS-P1

The app loads quickly or provides onscreen feedback to the user (a progress indicator or similar cue) if the app takes longer than two seconds to load.

1.1.26. PS-P2

Apps should render frames every 16ms to achieve 60 frames per second. Developers can use the Profile HWUI rendering option in testing. If there are issues, tools are available to help diagnose slow rendering.

1.1.27. PS-P3

With StrictMode enabled (see StrictMode Testing, below), no red flashes (performance warnings from StrictMode) are visible when testing the app. Any red flashes indicate bad behaviors regarding storage, network access, or memory leaks.

1.1.28. PS-B1

The app properly supports the power management features that were introduced in Android 6.0 (Doze and App Standby). In the case where core functionality is disrupted by power management, only qualified apps may request an exemption. See Support for other use cases in Doze and App Standby.
During development, developers can test app standby and doze behavior using these ADB commands.

In terms of battery usage, developers can use the Android Studio energy profiler or the Battery Historian tool, combined with planned background work, to diagnose unexpected battery use.

1.1.29. SC-AC2

All intents and broadcasts follow best practices:

Use explicit intents if the destination application is well defined.
Use Intents to defer permissions to a different app that already has the permission.
Share data securely across apps.
Intents that contain a payload are verified before use.
If you need to pass an Intent to another app, so that the receiving app can invoke and expect a callback in the calling app, do not include a nested intent in the extras. Use a PendingIntent.
When setting up your PendingIntents, explicitly set the immutable flag, where applicable.

1.1.30. GP-P1

The app strictly adheres to the terms of the Google Play Developer Content Policy and does not offer inappropriate content, does not use the intellectual property or brand of others, and so on.

1.1.31. GP-P2

The app maturity level is set appropriately, based on the Content Rating Guidelines.

1.1.32. GP-P3

The app’s feature graphic follows the guidelines outlined in this support article. Make sure that:

The app listing includes a high-quality feature graphic.
The feature graphic does not contain device images, screenshots, or small text that will be illegible when scaled down and displayed on the smallest screen size that your app is targeting.
The feature graphic does not resemble an advertisement.

1.1.33. GP-P4

The app’s screenshots and videos do not show or reference non-Android devices.

1.1.34. GP-P5

The app’s screenshots or videos do not represent the content and experience of your app in a misleading way.

1.1.35. GP-X1

Common user-reported bugs in the Reviews tab of the Google Play page are addressed if they are reproducible and occur on many different devices. If a bug occurs on only a few devices, you should still address it if those devices are particularly popular or new.

1.2. App Security Best Practices

Link: https://developer.android.com/topic/security/best-practices

1.2.1. Store Data Safely Store data in external storage based on use case

Use external storage for large, non-sensitive files that are specific to your app, as well as files that your app shares with other apps. The specific APIs that you use depend on whether your app is designed to access app-specific files or access shared files.

Check availability of storage volume
If your app interacts with a removable external storage device, keep in mind that the user might remove the storage device while your app is trying to access it. Include logic to verify that the storage device is available.

Access app-specific files
If a file doesn't contain private or sensitive information but provides value to the user only in your app, store the file in an app-specific directory on external storage.

Access shared files
If your app needs to access or store a file that provides value to other apps, use one of the following APIs depending on your use case:

Media files: To store and access images, audio files, and videos that are shared between apps, use the Media Store API.
Other files: To store and access other types of shared files, including downloaded files, use the Storage Access Framework.

Check validity of data
If your app uses data from external storage, make sure that the contents of the data haven't been corrupted or modified. Your app should also include logic to handle files that are no longer in a stable format.

2. US National Institute of Standards and Technology (NIST)

2.1. NIST Special Publication 800-163 Revision 1

Link: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-163r1.pdf

2.1.1. 2.2 Organization-Specific Requirements Provenance

Identity of the developer, developer’s organization, developer’s reputation, consumer reviews, etc.

2.1.2. 2.2 Organization-Specific Requirements App Criticality

The level of importance of the app relative to the organization’s business.

2.1.3. 2.2 Organization-Specific Requirements Target Users

The app's intended set of users from the organization.

2.1.4. 2.2 Organization-Specific Requirements Target Hardware

The intended hardware platform on which the app will be deployed.

2.1.5. 2.2 Organization-Specific Requirements Target Operating Platform

The operating system, operating system version/Software Development Kit (SDK), and configuration on which the app will be deployed.

2.1.6. 2.2 Organization-Specific Requirements Target Environment

The intended operational environment of the app (e.g., general public use vs. sensitive military environment).

2.1.7. 2.2 Organization-Specific Requirements App Documentation

User Guide
When available, the app's user guide assists testing by specifying the expected functionality and expected behaviors. This is simply a statement from the developer describing what they claim their app does and how it does it

Test Plans
Reviewing the developer's test plans may help focus app vetting by identifying any areas that have not been tested or were tested inadequately. A developer could opt to submit a test oracle in certain situations to demonstrate its internal test effort.

Service-Level Agreement
If an app was developed for an organization by a third-party, a Service-Level Agreement (SLA) may have been included as part of the vendor contract. This contract should require the app to be compatible with the organization's security policy.

3. Open Web Application Security Project (OWASP)

3.1. Application Security Verification Standard 4.0.3 (ASVS)

Link: https://raw.githubusercontent.com/OWASP/ASVS/v4.0.3/4.0/OWASP%20Application%20Security%20Verification%20Standard%204.0.3-en.pdf

3.1.1. V1.5 Input and Output Architecture

1.5.1 Verify that input and output requirements clearly define how to handle and process data based on type, content, and applicable laws, regulations, and other policy compliance.

3.1.2. V7.3 Log Protection

7.3.4 Verify that time sources are synchronized to the correct time and time zone. Strongly consider logging only in UTC if systems are global to assist with post-incident forensic analysis.

3.1.3. V8.1 General Data Protection

8.1.4 Verify the application can detect and alert on abnormal numbers of requests, such as by IP, user, total per hour or day, or whatever makes sense for the application.

3.1.4. V8.2 Client-side Data Protection

8.2.1 Verify the application sets sufficient anti-caching headers so that sensitive data is not cached in modern browsers.

8.2.2 Verify that data stored in browser storage (such as localStorage, sessionStorage, IndexedDB, or cookies) does not contain sensitive data.V8.3 Sensitive Private Data

3.1.5. V8.3 Sensitive Private Data

8.3.5 Verify accessing sensitive data is audited (without logging the sensitive data itself), if the data is collected under relevant data protection directives or where logging of access is required.

3.1.6. V14.1 Build and Deploy

14.1.4 Verify that the application, configuration, and all dependencies can be re-deployed using automated deployment scripts, built from a documented and tested runbook in a reasonable time, or restored from backups in a timely fashion.