

- 4.1 Copycats (a)
- 4.1 Copycats (b)
- 4.1 Copycats (c)
- 4.2 Minimum Functionality 4.2.1
- 4.2 Minimum Functionality 4.2.2
- 4.2 Minimum Functionality 4.2.3
- 4.2 Minimum Functionality 4.2.6
- 4.2 Minimum Functionality 4.2.7 (a)
- 4.2 Minimum Functionality 4.2.7 (b)
- 4.2 Minimum Functionality 4.2.7 (c)
- 4.2 Minimum Functionality 4.2.7 (d)
- 4.2 Minimum Functionality 4.2.7 (e)
- 4.3 Spam (a)
- 4.3 Spam (b)
- 4.4 Extensions 4.4.1
- 4.4 Extensions 4.4.2
- 4.5 Apple Sites and Services 4.5.1
- 4.5 Apple Sites and Services 4.5.2 (i)
- 4.5 Apple Sites and Services 4.5.2 (ii)
- 4.5 Apple Sites and Services 4.5.2 (iii)
- 4.5 Apple Sites and Services 4.5.3
- 4.5 Apple Sites and Services 4.5.4
- 4.5 Apple Sites and Services 4.5.5
- 4.5 Apple Sites and Services 4.5.6
- 4.7 Mini apps, mini games, streaming games, chatbots, plug-ins, and game emulators 4.7.1
- 4.7 Mini apps, mini games, streaming games, chatbots, plug-ins, and game emulators 4.7.2
- 4.7 Mini apps, mini games, streaming games, chatbots, plug-ins, and game emulators 4.7.3
- 4.7 Mini apps, mini games, streaming games, chatbots, plug-ins, and game emulators 4.7.4
- 4.7 Mini apps, mini games, streaming games, chatbots, plug-ins, and game emulators 4.7.5
- 4.8 Login Services
- 4.9 Apple Pay
- 4.10 Monetizing Built-In Capabilities

App Review 1

Apple

Principle 7: Provide clear feedback to developers - 2026

Google

Core App Quality 1

- VX-N1
- VX-N2
- VX-N3
- VX-S1

Principle 7: Provide clear feedback to developers - 2026

1. Google

1.1. Core App Quality

Link: <https://developer.android.com/docs/quality-guidelines/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-S1

Notifications follow Material Design guidelines. In particular:

Notifications are not used for cross-promotion or advertising another product, as this is strictly prohibited by the Play Store.

Notification channels are defined according to best practices, rather than serving all notifications from one channel.

Selecting the correct notification priority.

Multiple notifications are stacked into a single notification group, where possible.

Set timeouts for notifications where appropriate.

Notifications are persistent only if related to ongoing events, such as music playback or a phone call.

2. Apple

2.1. App Review

Link: <https://developer.apple.com/app-store/review/guidelines/#design>

Link: <https://developer.apple.com/app-store/review/guidelines/#design>

2.1.1. 4.1 Copycats (a)

Come up with your own ideas. We know you have them, so make yours come to life. Don't simply copy the latest popular app on the App Store, or make some minor changes to another app's name or UI and pass it off as your own. In addition to risking an intellectual property infringement claim, it makes the App Store harder to navigate and just isn't fair to your fellow developers.

2.1.2. 4.1 Copycats (b)

Submitting apps which impersonate other apps or services is considered a violation of the Developer Code of Conduct and may result in removal from the Apple Developer Program.

2.1.3. 4.1 Copycats (c)

You cannot use another developer's icon, brand, or product name in your app's icon or name, without approval from the developer.

2.1.4. 4.2 Minimum Functionality 4.2.1

Apps using ARKit should provide rich and integrated augmented reality experiences; merely dropping a model into an AR view or replaying animation is not enough.

2.1.5. 4.2 Minimum Functionality 4.2.2

Other than catalogs, apps shouldn't primarily be marketing materials, advertisements, web clippings, content aggregators, or a collection of links.

2.1.6. 4.2 Minimum Functionality 4.2.3

- (i) Your app should work on its own without requiring installation of another app to function.
- (ii) If your app needs to download additional resources in order to function on initial launch, disclose the size of the download and prompt users before doing so.

2.1.7. 4.2 Minimum Functionality 4.2.6

Apps created from a commercialized template or app generation service will be rejected unless they are submitted directly by the provider of the app's content. These services should not submit apps on behalf of their clients and should offer tools that let their clients create customized, innovative apps that provide unique customer experiences. Another acceptable option for template providers is to create a single binary to host all client content in an aggregated or "picker" model, for example as a restaurant finder app with separate customized entries or pages for each client restaurant, or as an event app with separate entries for each client event.

2.1.8. 4.2 Minimum Functionality 4.2.7 (a)

The app must only connect to a user-owned host device that is a personal computer or dedicated game console owned by the user, and both the host device and client must be connected on a local and LAN-based network.

2.1.9. 4.2 Minimum Functionality 4.2.7 (b)

Any software or services appearing in the client are fully executed on the host device, rendered on the screen of the host device, and may not use APIs or platform features beyond what is required to stream the Remote Desktop.

2.1.10. 4.2 Minimum Functionality 4.2.7 (c)

All account creation and management must be initiated from the host device.

2.1.11. 4.2 Minimum Functionality 4.2.7 (d)

The UI appearing on the client does not resemble an iOS or App Store view, does not provide a store-like interface, or include the ability to browse, select, or purchase software not already owned or licensed by the user. For the sake of clarity, transactions taking place within mirrored software do not need to use in-app purchase, provided the transactions are processed on the host device.

2.1.12. 4.2 Minimum Functionality 4.2.7 (e)

Thin clients for cloud-based apps are not appropriate for the App Store.

2.1.13. 4.3 Spam (a)

Don't create multiple Bundle IDs of the same app. If your app has different versions for specific locations, sports teams, universities, etc., consider submitting a single app and provide the variations using in-app purchase.

2.1.14. 4.3 Spam (b)

Also avoid piling on to a category that is already saturated; the App Store has enough fart, burp, flashlight, fortune telling, dating, drinking games, and Kama Sutra apps, etc. already. We will reject these apps unless they provide a unique, high-quality experience. Spamming the store may lead to your removal from the Apple Developer Program.

2.1.15. 4.4 Extensions 4.4.1

Keyboard extensions have some additional rules.

They must:

- Provide keyboard input functionality (e.g. typed characters);
- Follow Sticker guidelines if the keyboard includes images or emoji;
- Provide a method for progressing to the next keyboard;
- Remain functional without full network access and without requiring full access;
- Collect user activity only to enhance the functionality of the user's keyboard extension on the iOS device.

They must not:

- Launch other apps besides Settings; or
- Repurpose keyboard buttons for other behaviors (e.g. holding down the "return" key to launch the camera).

2.1.16. 4.4 Extensions 4.4.2

Safari extensions must run on the current version of Safari on the relevant Apple operating system. They may not interfere with System or Safari UI elements and must never include malicious or misleading content or code. Violating this rule will lead to removal from the Apple Developer Program. Safari extensions should not claim access to more websites than strictly necessary to function.

2.1.17. 4.5 Apple Sites and Services 4.5.1

Apps may use approved Apple RSS feeds such as the iTunes Store RSS feed, but may not scrape any information from Apple sites (e.g. apple.com, the iTunes Store, App Store, App Store Connect, developer portal, etc.) or create rankings using this information.

2.1.18. 4.5 Apple Sites and Services 4.5.2 (i)

MusicKit on iOS lets users play Apple Music and their local music library natively from your apps and games. When a user provides permission to their Apple Music account, your app can create playlists, add songs to their library, and play any of the millions of songs in the Apple Music catalog. Users must initiate the playback of an Apple Music stream and be able to navigate using standard media controls such as "play," "pause," and "skip." Moreover, your app may not require payment or indirectly monetize access to the Apple Music service (e.g. in-app purchase, advertising, requesting user info, etc.). Do not download, upload, or enable sharing of music files sourced from the MusicKit APIs, except as explicitly permitted in MusicKit documentation.

2.1.19. 4.5 Apple Sites and Services 4.5.2 (ii)

Using the MusicKit APIs is not a replacement for securing the licenses you might need for a deeper or more complex music integration. For example, if you want your app to play a specific song at a particular moment, or to create audio or video files that can be shared to social media, you'll need to contact rights-holders directly to get their permission (e.g. synchronization or adaptation rights) and assets. Cover art and other metadata may only be used in connection with music playback or playlists (including screenshots displaying your app's functionality), and should not be used in any marketing or advertising without getting specific authorization from rights-holders. Make sure to follow the Apple Music Identity Guidelines when integrating Apple Music services in your app.

2.1.20. 4.5 Apple Sites and Services 4.5.2 (iii)

Apps that access Apple Music user data, such as playlists and favorites, must clearly disclose this access in the purpose string. Any data collected may not be shared with third parties for any purpose other than supporting or improving the app experience. This data may not be used to identify users or devices, or to target advertising.

2.1.21. 4.5 Apple Sites and Services 4.5.3

Do not use Apple Services to spam, phish, or send unsolicited messages to customers, including Game Center, Push Notifications, etc. Do not attempt to reverse lookup, trace, relate, associate, mine, harvest, or otherwise exploit Player IDs, aliases, or other information obtained through Game Center, or you will be removed from the Apple Developer Program.

2.1.22. 4.5 Apple Sites and Services 4.5.4

Push Notifications must not be required for the app to function, and should not be used to send sensitive personal or confidential information. Push Notifications should not be used for promotions or direct marketing purposes unless customers have explicitly opted in to receive them via consent language displayed in your app's UI, and you provide a method in your app for a user to opt out from receiving such messages. Abuse of these services may result in revocation of your privileges.

2.1.23. 4.5 Apple Sites and Services 4.5.5

Only use Game Center Player IDs in a manner approved by the Game Center terms and do not display them in the app or to any third party.

2.1.24. 4.5 Apple Sites and Services 4.5.6

Apps may use Unicode characters that render as Apple emoji in their app and app metadata. Apple emoji may not be used on other platforms or embedded directly in your app binary.

2.1.25. 4.7 Mini apps, mini games, streaming games, chatbots, plug-ins, and game emulators 4.7.1

Software offered in apps under this rule must:

- follow all privacy guidelines, including but not limited to the rules set forth in Guideline 5.1 concerning collection, use, and sharing of data, and sensitive data (such as health and personal data from kids);
- include a method for filtering objectionable material, a mechanism to report content and timely responses to concerns, and the ability to block abusive users; and
- follow Guideline 3.1 in order to offer digital goods or services to end users.

2.1.26. 4.7 Mini apps, mini games, streaming games, chatbots, plug-ins, and game emulators 4.7.2

Your app may not extend or expose native platform APIs or technologies to the software without prior permission from Apple.

2.1.27. 4.7 Mini apps, mini games, streaming games, chatbots, plug-ins, and game emulators 4.7.3

Your app may not share data or privacy permissions to any individual software offered in your app without explicit user consent in each instance.

2.1.28. 4.7 Mini apps, mini games, streaming games, chatbots, plug-ins, and game emulators 4.7.4

You must provide an index of software and metadata available in your app. It must include universal links that lead to all of the software offered in your app.

2.1.29. 4.7 Mini apps, mini games, streaming games, chatbots, plug-ins, and game emulators 4.7.5

Your app must provide a way for users to identify software that exceeds the app's age rating, and use an age restriction mechanism based on verified or declared age to limit access by underage users.

2.1.30. 4.8 Login Services

Apps that use a third-party or social login service (such as Facebook Login, Google Sign-In, Sign in with Twitter, Sign In with LinkedIn, Login with Amazon, or WeChat Login) to set up or authenticate the user's primary account with the app must also offer as an equivalent option another login service with the following features:

- the login service limits data collection to the user's name and email address;
- the login service allows users to keep their email address private as part of setting up their account; and
- the login service does not collect interactions with your app for advertising purposes without consent.

A user's primary account is the account they establish with your app for the purposes of identifying themselves, signing in, and accessing your features and associated services.

Another login service is not required if:

- Your app exclusively uses your company's own account setup and sign-in systems.
- Your app is an alternative app marketplace, or an app distributed from an alternative app marketplace, that uses a marketplace-specific login for account, download, and commerce features.
- Your app is an education, enterprise, or business app that requires the user to sign in with an existing education or enterprise account.
- Your app uses a government or industry-backed citizen identification system or electronic ID to authenticate users.
- Your app is a client for a specific third-party service and users are required to sign in to their mail, social media, or other third-party account directly to access their content.

2.1.31. 4.9 Apple Pay

Apps using Apple Pay must provide all material purchase information to the user prior to sale of any good or service and must use Apple Pay branding and user interface elements correctly, as described in the Apple Pay Marketing Guidelines and Human Interface Guidelines. Apps using Apple Pay to offer recurring payments must, at a minimum, disclose the following information:

- The length of the renewal term and the fact that it will continue until canceled
- What will be provided during each period
- The actual charges that will be billed to the customer
- How to cancel

2.1.32. 4.10 Monetizing Built-In Capabilities

You may not monetize built-in capabilities provided by the hardware or operating system, such as Push Notifications, the camera, or the gyroscope; or Apple services and technologies, such as Apple Music access, iCloud storage, or Screen Time APIs.