

The 6 rejections that cost Flutter devs a review cycle

Five of these six have nothing to do with what your app does. Reviewers check them on nearly every paid app, and every one can be designed out before you hit Submit. Run this list each time.

1	5.1.1(v) Account deletion, in app	TRIGGER You offer account creation and deletion needs a mailto:, a web form, or support. REVIEWER Creates an account, opens Settings, hunts for Delete Account. ☐ PASS Privileged server function deletes storage objects → rows → auth record. JWT identifies the caller; never accept a user ID in the body.
2	3.1.2 Terms + privacy on the paywall	TRIGGER Legal links live in Settings, or only on your website. REVIEWER Opens the paywall — often signed out — and looks for both links plus the renewal terms. ☐ PASS Both links adjacent to the purchase control, opening real content. Billed amount, period and "renews until cancelled" near the price. Close control visible on the first frame.
3	3.1.1 Restore purchases	TRIGGER A paying user reinstalls or signs in on a second device and can only pay again. REVIEWER Looks for a visible Restore Purchases control. ☐ PASS On the paywall. Call the platform restore API and re-sync entitlement state from the result — never from a local flag a reinstall wipes.
4	4.8 Privacy-preserving login	TRIGGER You offer Google or another third-party login with no private alternative beside it. Email + password alone does <i>not</i> trigger it. REVIEWER Looks at the sign-in screen. ☐ PASS Native Sign in with Apple sheet, not a web redirect. Persist name + email on first authorization — you never get them again. Use a nonce: sheet gets the SHA-256 hash, backend verifies the raw value.
5	2.1 App completeness	TRIGGER Login wall with no credentials, a crash, an empty list, or a spinner that never resolves. The most common rejection of all. REVIEWER Tries to use it. Will not debug it for you. ☐ PASS Demo mode or seeded data so a fresh install shows something real. Credentials tested <i>the day you submit</i>. Readable failure states, never an infinite spinner. Record the full-flow video and attach it unprompted.
6	5.1.1 Purpose strings + privacy label	TRIGGER A missing or lazy usage description, or a privacy label that omits what your SDKs collect. REVIEWER Checks every permission your binary can request, and the label against reality. ☐ PASS Every *UsageDescription names the feature it serves: "...to take a profile picture," not "needs camera access." Audit every third-party SDK — crash reporters collect device IDs, analytics collects usage.

REVIEW NOTES — PASTE AND FILL DEMO ACCOUNT email: demo@yourapp.com password: verified working: [DATE OF SUBMISSION] PAYWALL / PAID FEATURES [Feature] sits behind a subscription. This account is provisioned with an active entitlement — no purchase needed. FULL-FLOW RECORDING [URL to an unlisted video: launch → sign in → core feature → paywall → settings → delete account] NOTES [Anything non-obvious: hardware needed, region restrictions, a flow that looks broken but isn't.]	THE FIVE THAT SAVE THE MOST TIME Record the video before you submit. Nobody asks for it if it's already attached. Removes an entire class of round-trip. Test the demo account the morning you submit. Expired credentials are a 2.1, and they expire quietly. Open your paywall signed out. That's how a reviewer often sees it, and it's where the legal links go missing. Reinstall and try to restore. The only real test of 3.1.1 is a device that has forgotten you. Plan the cycle in. The first submission usually gets rejected. Budget for it instead of being surprised by it.
--	---