

Check your app before customers use it

By Jay Pozo · Reviewed 16 October 2026

Use this checklist with the person building your app. Ask for evidence from the version you plan to launch.

Mark Pass when the evidence meets your agreed requirement. Mark Unknown when evidence is missing. Mark Needs review when a check exposes a concern.

An engineer should perform every item marked Engineer check. These checks support a launch decision; they are not a security certification.

Login and permissions

Try each account role Engineer check

Permissions are rules for what each account can do. Ask for recorded tests using a customer account and an administrator account, including direct requests to restricted actions.

☐ Pass ☐ Unknown ☐ Needs review

Check lost and removed access Engineer check

Ask for test results covering password reset, expired login sessions and removed team members. A session is the access an app remembers after login.

☐ Pass ☐ Unknown ☐ Needs review

Private data: who can read what

Separate customer records Engineer check

Ask for tests with two customer accounts. Show that each account can read and change only its permitted records, including through direct data requests.

☐ Pass ☐ Unknown ☐ Needs review

Check files and shared links Engineer check

Ask for an access map for uploaded files and exports. Test private file links while signed out and while signed in as another customer.

☐ Pass ☐ Unknown ☐ Needs review

Secret keys

Inspect what the browser receives Engineer check

A secret key grants access to a service. Ask for an engineer's scan of browser code and code history for exposed secrets, with findings explained.

☐ Pass ☐ Unknown ☐ Needs review

Prove keys can be replaced Engineer check

Ask for a list of key owners and a replacement procedure. Request evidence that exposed keys were revoked. Record where production keys are stored.

☐ Pass ☐ Unknown ☐ Needs review

Payment paths

Test each payment outcome Engineer check

Ask for payment-provider test results covering success, failure, cancellation and refunds. Confirm amounts and access changes. Separately review the live payment configuration.

☐ Pass ☐ Unknown ☐ Needs review

Test repeated payment messages Engineer check

A webhook is a message a service sends your app. Ask for signature checks that verify its sender. Request repeated-message and delayed-message tests. Confirm access and charges stay correct.

☐ Pass ☐ Unknown ☐ Needs review

Backups and recovery

Restore a recent backup Engineer check

A backup is a saved copy of data. Ask for a dated restore into a separate test environment, including uploaded files. Record missing data and recovery time.

☐ Pass ☐ Unknown ☐ Needs review

Name the recovery owner

Ask who receives failure alerts and who can restore service. Agree how much recent data loss and downtime the business can tolerate.

☐ Pass ☐ Unknown ☐ Needs review

Expected load

Test the expected busy period Engineer check

Ask for a load test that simulates expected simultaneous customers and their main actions. Compare response times and failures with agreed targets. Record service limits.

☐ Pass ☐ Unknown ☐ Needs review

Check running costs and limits

Ask for a cost estimate at expected use and the limits of each paid service. Record alert thresholds and the person responsible for responding.

☐ Pass ☐ Unknown ☐ Needs review

Release and rollback

Rehearse the release Engineer check

A release puts a version in front of customers. Ask for a checklist naming the exact version. Request configuration checks and tests after release.

☐ Pass ☐ Unknown ☐ Needs review

Rehearse returning to a working version Engineer check

A rollback restores an earlier version. Ask for a rehearsal and a plan for database changes, which may need separate recovery. Record who makes the decision.

☐ Pass ☐ Unknown ☐ Needs review

Record each unknown or concern with an owner and a due date. Review the remaining risks with an engineer before deciding to launch.