Erek Mackroy

CS-360 Mobile Architect & Programming

Project Three App Launch Plan

I will publish the app with a short, clear description that focuses on what people can do the first time they open it. The first line will say that it helps small teams track items, update quantities fast, and get a text message when stock is running low. I will highlight the key screens in plain language: an inventory list that is easy to scan, a simple form to add a new item, an edit flow that lets you update quantities in seconds, and alerts that appear as soon as something dips under the set level. I will add three screenshots that match those points in the same order. The app icon will be a simple box with a small tag in the corner and a neat check mark on it. The shapes will be clean, and the colors will match the soft purple used in the app, so it feels consistent on the home screen and in the store.

The app will run on Android 7.0 and up so people with older phones can still use it, and I will target the most recent SDK to stay current with platform rules. I will test on a modern phone image such as Pixel with the latest API and on an older API from the supported range so I can catch layout issues and permission flows that sometimes behave differently.

The app only needs a small set of permissions. It uses the ability to send text messages so it can deliver low stock alerts. I will ask for this permission at the time of need and only after the user taps Allow on the prompt that explains why it is needed. If the user says no, the app will still work for adding and editing items and it will keep a record of alerts in the “Alerts” section, but it will not send a message. I will not request access to phone state, contacts, location, or storage because the app does not need them. Keeping the list short builds trust and makes review smoother.

Before release I will run a simple checklist. I will confirm the login flow opens to the inventory screen every time. I will add five sample items, change a few quantities, delete one, and confirm the list updates without delay. I will drop one item below fifteen and watch for the alert dialog, then raise it above fifteen and confirm the alert pauses for that item. I will test the Alerts and Activity screens to make sure they both capture events in a clear order with the newest entry on top. I will sign out and sign back in to confirm state is correct. I will try both the light and dark system themes and different font sizes to check readability.

My store listing will include a short privacy note that explains that the app does not collect personal data and that messages are sent only to the phone number the user provides. I will link to a simple support page that explains how to turn the message feature on, how to change the alert level, and how to contact me by email if something goes wrong. I will also include a short FAQ about why the message permission is required and how to disable it later in system settings.

For release timing I will start with a small, closed test with classmates or coworkers. I will share the APK through a private channel and ask them to add two items, trigger one alert, and leave one written comment about what felt slow or confusing. I will fix anything that blocks use and then move to an open test track for two weeks. During this time, I will watch crash logs, message send failures, and any feedback about the layout or the alert rules. When the test looks stable, I will ship to production.

For the business side I will begin with a free download and no ads. The goal is to keep the app fast and pleasant to use without distractions. If there is interest later, I can add a small paid upgrade that unlocks extra features such as export to a file or custom alert thresholds per item. If I go that route, I will keep the core experience free so current users can stay productive without paying.

After launch I will keep a simple rhythm. I will bundle small fixes each month, update the screenshots when I add a useful feature, and review new Android changes so I stay in line with policy and platform updates. I will answer support mail within two business days, and I will keep a short public changelog so people can see what changed. This steady approach should make the app dependable for the people who rely on it.