Parul® University
Vadodara, Gujarat

NAAC GRADE A++

Information and Communication Technology

Polish and Publish Application

Study Guide

Mr. Anurag Kewat
Lectuere
CSE, PIT.
Parul University

Contents

INDEX

7 Polish And Publich Application

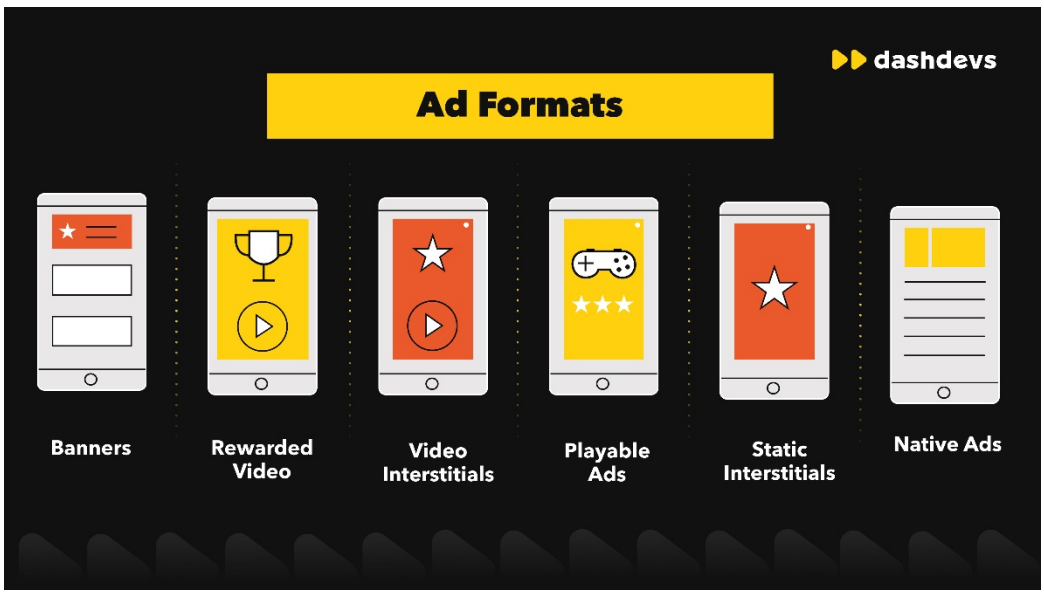
Polishing a mobile application is the final, detail-oriented phase of the development lifecycle where the focus shifts from building functionality to refining quality, usability, performance, and reliability. This stage ensures that the application not only works correctly but also delivers a smooth, consistent, and engaging experience for the end user. While the development phase ensures that the app "works," the polishing phase ensures that the app "feels right," "looks professional," and "performs efficiently" across various devices, environments, and user conditions.

A polished application is crucial because modern users have exceptionally high expectations. They compare every app—whether a small student project or a large enterprise application—with industry-leading platforms like Instagram, YouTube, and WhatsApp. As a result, an unpolished app can immediately lose user trust, reduce install rates, increase uninstall rates, and harm the brand's reputation.

This section discusses in depth the processes, tools, strategies, and industry practices involved in polishing, refining, and preparing a mobile app for public release.

7.1 Different ways to Monetize

- The most common way to monetize mobile apps is by running in-app ads. In fact, 50% of mobile gamers prefer free ad based games.
- This method offers a valuable source of revenue for apps that want to remain free in the app store.
- Providing content and ad formats relevant to your users can create a great user experience.



- **Banner ads:** This relatively simple ad format enables advertisers to place static or animated ads across a banner within your app. Quality graphics and an enticing call to action (CTA) are critical for banner ads.
- **Interstitial ads:** This format displays ads in an expanding, full-screen experience. For this reason,

interstitial ads are more likely to avoid "banner blindness" – where users have become so used to banner ads that they aren't registered by users. Since full screen interstitial can be disruptive, it's essential that these ads are placed at natural pauses, such as in-between levels of a gaming app.

- **Native ads:** These are paid ads that are designed to match the content that the mobile app user is experiencing
- **Video ads:** Video ads are another popular choice because they can be highly engaging and deliver some of the highest CTRs of all formats. The U.S. mobile ad market is expected to reach $247.68 billion by 2026.
- **Rewarded video ads:** You can enhance the appeal of video ads by rewarding users for finishing a video. Unity Ads, a monetization platform, found that 62% of mobile gamers regularly view rewarded video ads and 71% find this advertising preferable to other methods.
- **Playable ads:** These ads are a try-before-you-buy method that exposes users to interactive gameplay. Users gain a limited look at the advertiser's app before they are served a CTA. This is a great way for advertisers to reduce uninstall rates because users who are choosing to install will have a predetermined interest in the gameplay.

7.2 Versioning, Signing

All developers dream of their application's success. The best applications are those that learn continuously from the feedback and fix the issues in the current version of the application and include them in the next versioning.

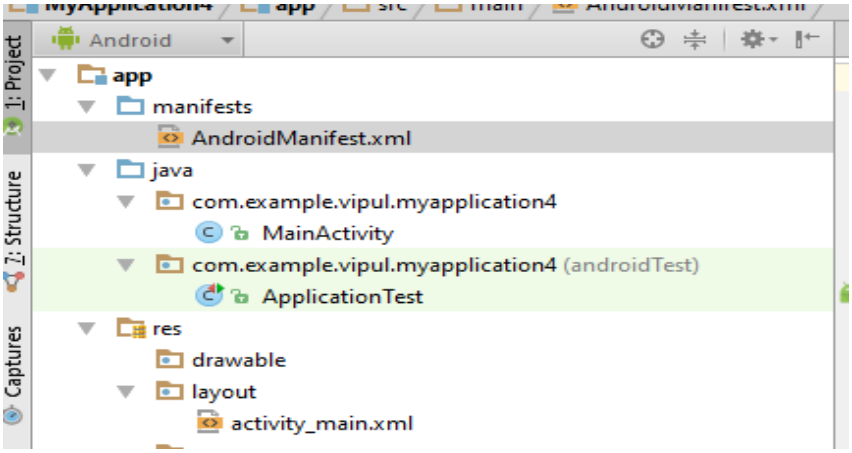
Versioning is important part of application development lifecycle. Here are some reasons that's true:

- Versioning allows users to understand which release of the application they have on their devices and what upgrade versions are available.
- If you have built multiple versions, each version can allow you to ask for a different set of device capabilities for each version.
- If your application works with services, it can use the version information to make sure it uses the correct data schema to work with your service.

Application version can be set by using two attributes. These attributes are:

- **versionCode:** This is an integer value that specifies the version of your code, compared to other versions. This allows the Android OS to check if there is a newer version available from the PlayStore, and then prompt the user to upgrade the application.
- **versionName:** This is a string that represents the release version of the application code. The string value is used to display the application version in the format <major>.<minor>.<point> string. Note that this attribute is not used for any other purposes, and for all practical purposes, versionCode is used by Android.

How to Set an Application Version



Open the file and make the following changes (highlighted)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns_android="http://schemas.android.com/apk/res/android"
    package="com.example.vipul.myapplication4"
    android:versionCode="1"
    android:versionName="1.0">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android_name="android.intent.action.MAIN" />

                <category android_name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

7.3 Packaging of beta version of Mobile Application

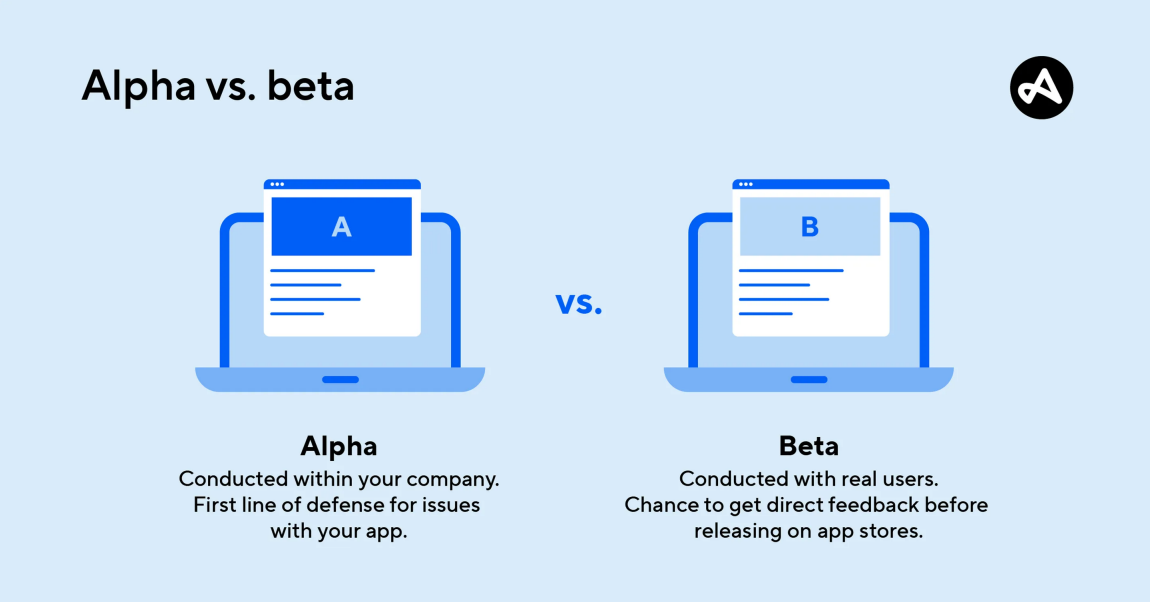
What does beta testing mean?

Mobile app beta testing involves giving a group of target users access to your app to test and evaluate its performance. It's like performers conducting a dress rehearsal before a big show to smooth out any rough edges before launching to a wider audience. The purpose is to ensure that everything runs smoothly when the curtain rises.

Ideally, beta testing should occur when there's still enough time in the development cycle to make meaningful changes based on the feedback, but close enough to the launch to reflect the final product accurately.

What is the difference between alpha and beta testing?

Alpha testing is done in-house while **beta testing is done with end-users**. In alpha testing, employees examine the app's stability and functionality, often combining white-box (testers already familiar with how the app works) and black-box (testing without internal knowledge) approaches. Beta testing, on the other hand, is fully black-box. These users interact with the app in genuine scenarios, providing invaluable insights into their preferences and feelings.



What is the difference between UAT and beta testing?

User acceptance testing (UAT) is a kind of half-step between alpha and beta testing. UAT is more about verification than technical testing. It's the final development step before the app goes live. **UAT confirms that the app performs its intended functions effectively in real-world scenarios.** The test cases are structured and specific.

On the other hand, **beta testing is performed by actual end-users of the app in a real-world environment**. It's less structured than UAT. Beta testers provide feedback on functionality, usability, and overall user experience, which may not necessarily align with specific business requirements but is invaluable for understanding how the software will perform in the hands of everyday users.

WHAT'S TESTED

What is examined in beta testing?
Here are some key areas that are tested:

1. Feature suitability
2. Performance and speed
3. Technical stability
4. Real-world adaptability
5. Hidden bugs
6. Localization nuances
7. Scalability

BENEFITS

Why is beta testing important?
In a nutshell, beta testing is important because it:

- Prioritizes customer needs
- Reduces product failure risk
- Tests post-launch infrastructure
- Identifies overlooked bugs
- Enhances product quality
- Increases customer satisfaction

TYPES

Types of beta testing
It's important to note that while valuable for all apps, beta testing isn't a one-size-fits-all process. It must align with your specific goals and requirements. To accommodate these varying needs, there are two types of beta testing to choose from:

1. **Open beta testing:** No restrictions on the number of users involved. Anyone interested can participate. Think: An open invitation party.
2. **Closed beta testing:** Invite-only testing, where users receive specific invitations to participate. Think: A private dinner party.
3. From there, you can choose from further subsets of beta testing:
4. **Technical beta testing**: Involves tech-savvy users to uncover complex bugs.
5. **Focused beta testing:** Targeted testing for specific features.
6. **Marketing beta testing**: Testing with industry influencers, designed to get media attention and assess marketing channels.

WHEN TO BETA TEST

Optimal timing for maximum beta test effectiveness
For the most accurate and valuable results, beta testing should be done when:

- **Core features are fully functional.** This doesn't mean every feature needs to be final, but the app should be stable enough to perform its primary functions without critical bugs.
- **The user interface and overall usability are in a good state.** While these aspects can still be refined based on beta feedback, they should be developed enough to provide a coherent user experience.

- **Clear goals and metrics are set for what you want to achieve with beta testing.** This could include specific aspects of the app you want feedback on, performance benchmarks, user experience aspects, etc.

- **The necessary resources are in place to support the beta test.** This includes staff to monitor feedback, analyze data, and implement changes based on the feedback received.

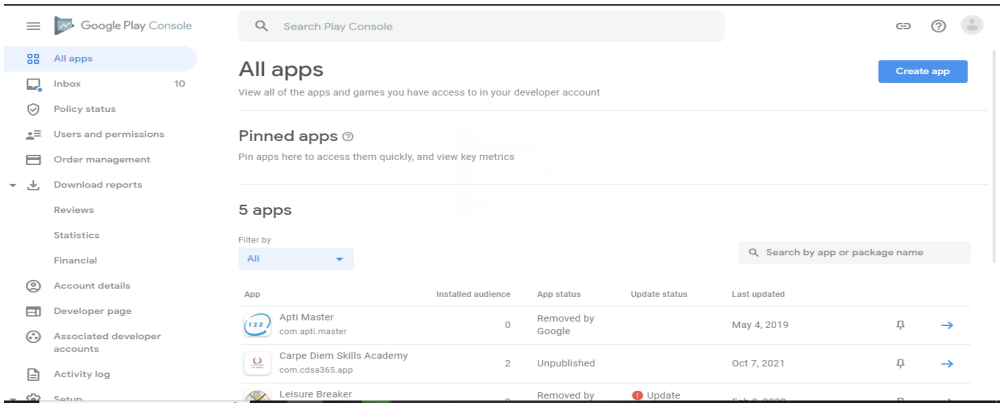
- **All legal and compliance checks are completed**. This is essential to protect both the company and the beta testers.

- **Alpha testing is completed.** Passing initial QA checks is crucial to avoid wasting beta testers' time on issues that could have been easily identified and resolved in-house.

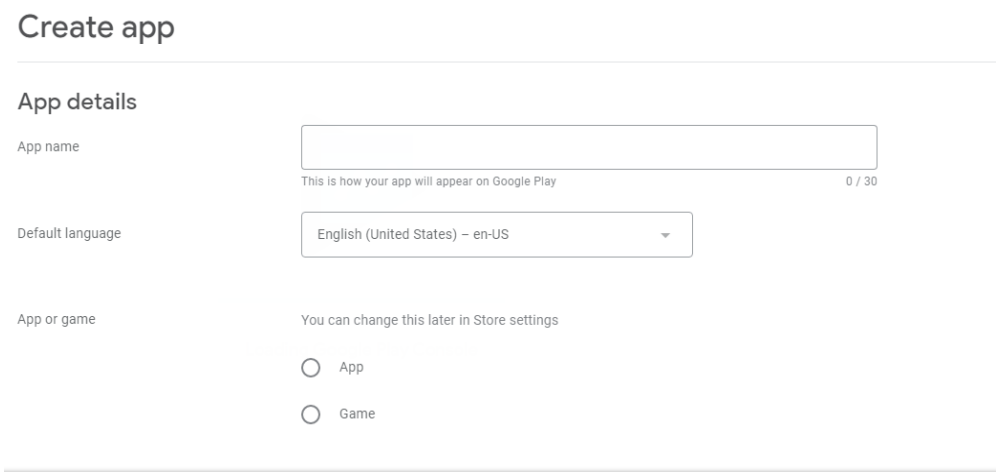
The length of your beta test depends on your objectives. Many tests run for three to five weeks. However, plan for potential delays.

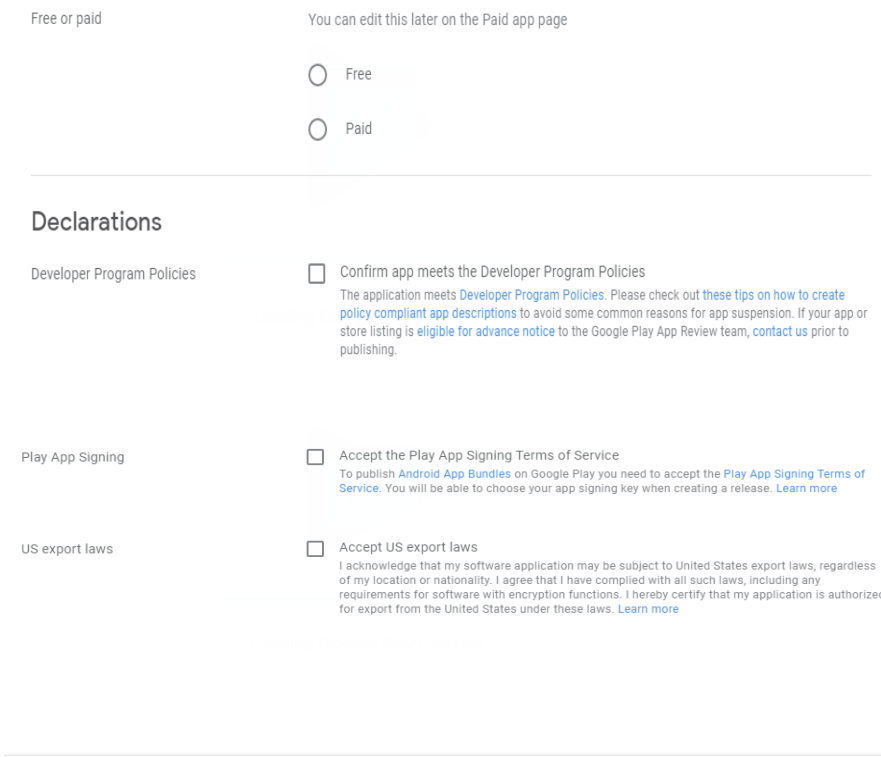
7.4 Distributing Application on Mobile Market Place

1. Click to Create App Button.

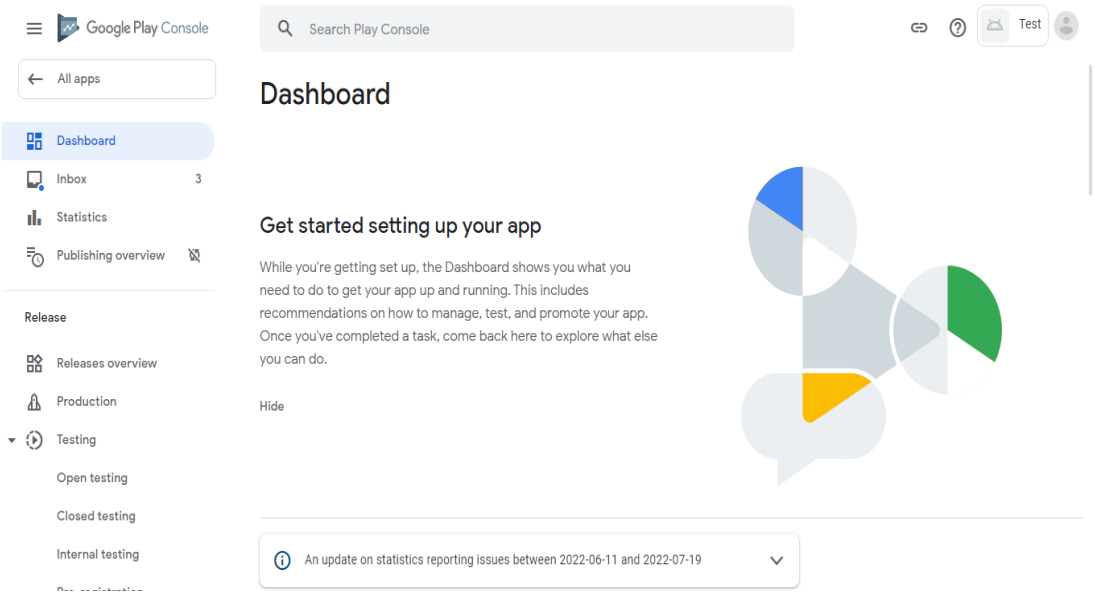


2. Give the required details of the app and check in the checkbox with the declarations.

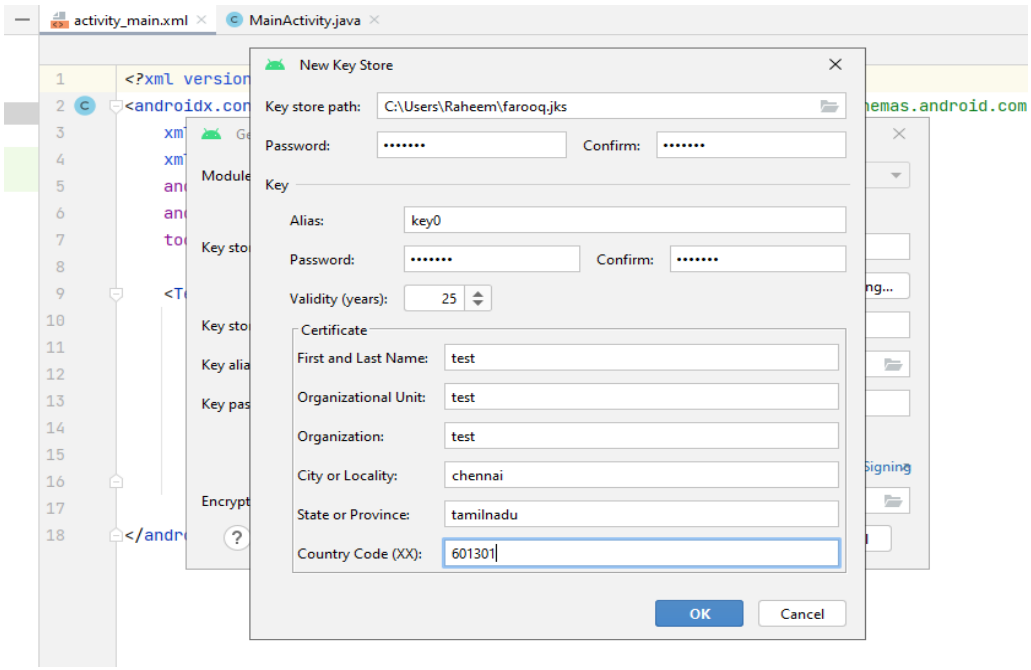




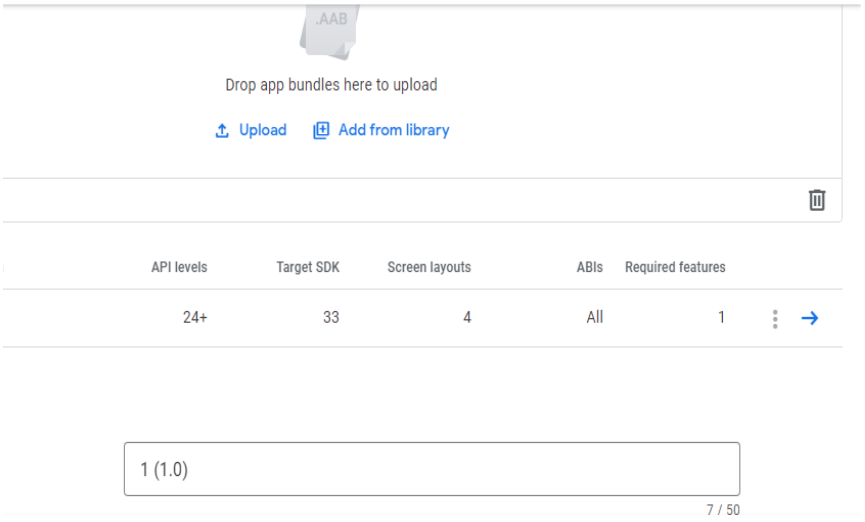
3. Click to create app at the end.
4. It will create a separate dashboard for the app as shown below



5. There are few steps to complete the process of publishing.
   a) Start testing now
      - Select Create and Roll Out a Release
        - It will show a create release button on top click it
        - Upload the signed App Bundles which you have created from the Android Studio.



6. After Uploading of the app bundle



7. Fill the details that are mandatory with the dialogs that appear.
8. Add the tester's email id for the verification.
9. Check for the Dashboard for further approvals.

Parul® University
Vadodara, Gujarat

NAAC GRADE A++

https://paruluniversity.ac.in/