# CS 360 Project Two - App Launch Plan.

# Daniel Gorelkin - Aug 7, 2025

**The Launch Plan**

After the app is built and the installation APK file created, I will start by testing my app directly from a live running device. I will first verify that the functionality is indeed working as expected, running error-free, and all the permissions needed for the app’s functionality are granted, and the desired functionality is achieved. For example, navigating to the app’s permissions section page will provide a confirmation that all of the permissions are functional: See [Screenshot #1](#Screenshot_1). Similarly, I will execute some device internal functionality checks like sending an actual SMS message (See [Screenshot #2](#Screenshot_2)) and test my device’s camera and storage functionality ([Screenshot #3](#Screenshot_3)).

Additionally, I will ask for feedback from my colleagues, asking for their opinion to integrate improvements and polish my work, conduct an initial quality assurance check, and run some internal testing by asking some experts in the field from the Play Developer account center. Not less important, I will make sure that my app is fully compliant with the Google Play policies from the [Developer Policy Center](https://play.google/developer-content-policy/) and run my app through some tests by using the [Firebase Test Lab](https://firebase.google.com/docs/test-lab).

No successful launch can be held without describing my app’s functionality and presenting its main goals. Hence, my description will include the main purpose and the challenges that my app solves, the designated group of users, their age, and obviously some usage images to demonstrate the various screens the app supports.

A screenshot of a calculator

AI-generated content may be incorrect. A screenshot of a chat

AI-generated content may be incorrect. A screenshot of a phone

AI-generated content may be incorrect.

The app will also use some graphic files that will be used as a logo and the background for the activity pages that will be generated by the Asset Studio feature from the IDE. The background for the view will be a vector file that could be easily resized to fit various screen sizes and a tiny .webp app logo that will act as an icon in the device’s menu and in the launch screen. For example:

A blue box with arrows around it

AI-generated content may be incorrect. A logo of a phone

AI-generated content may be incorrect. A logo with a green logo

AI-generated content may be incorrect.

The images will fit the chosen main app’s theme and will use the same colors as the app’s styling features to fit seamlessly within the app and comply with my customer’s requirements.

My app is developed to run on Android 7.0, Nougat OS devices with the API 24 permission level and higher. This means that the app will work properly on most (98%) of the available Android devices on the market, targeted for both small and large screen sizes, with a rotatable layout feature. Hence, the code supports the latest libraries and contains no features that are deprecated in older versions and OSs.

A screenshot of a phone

AI-generated content may be incorrect.

With that said, despite the app stores all its data in a private memory location both for its database and image storage location that is not accessible from using outside the app and are not exposed to other applications, some features include the usage of SMS sending from the device, notifications and camera usage still should be requested and granted by the user to ensure smooth usage and outstanding user experience. Hence, the Manifest asks permissions before allowing users the potentially “dangerous” functionalities, and yet allows limited app usage if such permissions are denied. For example, the app asks the user permission to use the device's camera before taking a picture of an item and adding it to the database (See [Screenshot #1.c](#Screenshot_1_d)). If a user grants their permission to use the camera, the built-in camera function will pop up to take the photo; otherwise, the app will skip this process and will not add the image to the given item. Similarly, the app will ask permission to send SMS messages (Screenshot #1.A), and display notification messages (Screenshot #1.B). Again, denying the SMS permissions will swap the flow to use the internal notification channel if one is available, or remain silent, if the notification permissions are blocked as well. Declaring the required permissions inside the Manifest file will automatically acknowledge Google Play of the requirements and permissions the app will use.

Finally, although the app will be downloadable for free from the Play Store, no one works for free. The app will support special advertising placeholders such as pop-up messages and full-screen scaled graphical images that will act as leads to the advertiser's websites. These features will allow the stakeholder to return the investment quickly and even sell the app in a later stage when more free and paid features will be delivered and added with the next updates. As mentioned, the app will support free and paid features. Hence, free users will have access to the app's basic features, while more advanced features can be unlocked as in-app purchases upon payment.

**Screenshots:**

1. **Permissions 1. A 1. B 1. C**

A screenshot of a phone

AI-generated content may be incorrect.**A screenshot of a chat

AI-generated content may be incorrect. A screenshot of a phone

AI-generated content may be incorrect.** **A screenshot of a phone

AI-generated content may be incorrect.**

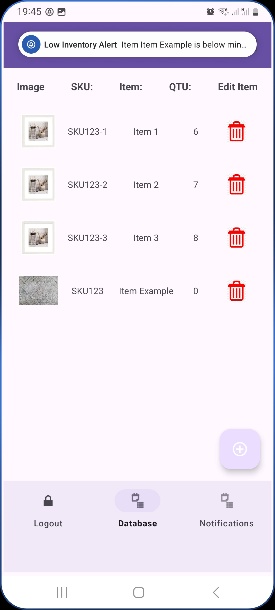
**2 . Acknowledgement flags and usage tips**

**A screenshot of a phone

AI-generated content may be incorrect.**

**3. SMS functionality 4. Camera Functionality 5. Notifications**

**A screenshot of a phone

AI-generated content may be incorrect.** ** A screenshot of a phone

AI-generated content may be incorrect. A screenshot of a phone

AI-generated content may be incorrect.**

**References**

*Console, G. P. (n.d.). Release. Google Play Console.* [*https://play.google.com/console/about/guides/releasewithconfidence/*](https://play.google.com/console/about/guides/releasewithconfidence/)

*Console, G. P. (n.d.-a). Policy status. Google Play Console.* [*https://play.google.com/console/about/policystatus/*](https://play.google.com/console/about/policystatus/)

*Firebase test lab. (n.d.). Firebase.* [*https://firebase.google.com/docs/test-lab*](https://firebase.google.com/docs/test-lab)

*Hughart, N. (2024, September 21). 10 Effective ways to Monetize Mobile Apps | BuildFire. Buildfire.* [*https://buildfire.com/ways-monetize-mobile-app/*](https://buildfire.com/ways-monetize-mobile-app/)

*Mobile | UI Design*. (n.d.). Android Developers. <https://developer.android.com/design/ui/mobile>

Toyo, I. (2015, October 8). *Export your app from Android Studio*. <https://android-coffee.com/export-your-app-from-android-studio/>