**InventoryBuddy**

Comprehensive Launch Plan

InventoryBuddy is a simple mobile app designed to allow secure authentication and authorization to a personal inventory of tracked items and associated quantities. After entering in a username and password, the account is created (if not found), and the user is logged in. They are then alerted about the upcoming SMS permission request that will allow SMS notifications to be received whenever an item is running low (1 left). After accepting or rejecting the permission request, the user can now access their personal inventory. Each item consists of a name and quantity attribute, and the user may edit or delete any item name or quantity at any time. The user may add items by clicking the attractive, “+” floating action button centered at the bottom of the screen.

At this time, there is no app icon; however, an minimalistic icon with a backpack (or similar icon) will appear as the launcher and also in the top-left corner of the screen for LTR (left to right) orientation. The minimum Android version is SDK 33 and aimed at SDK 34 (and the most current Android version). The hardware telephony component is required in the AndroidManifest file, along with the SMS\_SEND permission.

I believe an effective monetization strategy for an inventory app like this is to offer the app with no ads, but include the option to upgrade to the premium edition for a fee so that they can have unlimited items in their inventory. This way users can try out the product, and if it becomes crucial in their life or business, they can pay for more space. Backup cloud storage and regular content updates should be implemented to keep (and gain) users (Hughart, N., n.d.). Data collection may be optionally implemented with the user’s knowledge and consent and can prove quite valuable!

Before bringing to market, testing tools and pre-launch reports should be utilized in conjunction with staged rollouts to identify issues before release. Firebase Crashlytics, TestLab, and Performance Monitoring can be used, along with Google Developer Console to test multiple Android versions and devices (including the latest version) (Google, n.d.). Ensure your app meets guidelines/frameworks provided by authorities like the following:

* IEEE (institute of Electrical and Electronics Engineers)
* ISO (International Organization for Standardization)
* W3C (World Wide Web Consortium), ANSI (American National Standards Institute)
* OWASP (Open Web Application Security Project)

According to Google, launching your store listing should consist of the following steps as well:

1. optimize **graphic assets, screenshots, & video** to accurately represent your app experience
2. provide accurate **categorization** & **content rating** in Google Play
3. capture early consumer interest with a **pre-registration campaign**
4. use an **app campaign** to drive installs & in-app conversions
5. maximize global reach with localized app strings & store listings with **translation services**
6. use **custom store listings** for custom countries or pre-registration campaign
7. run **store listing experiments** to A/B test different versions of your listing page on Google Play

Finally, be sure to review the guidance listed below from Google:

* Developer Program Policies should be understood
* target API level requirements should be defined
* publish with Android App Bundle or use Play Asset Delivery or Play Feature Delivery (150+ MB)
* specify Android versions & device screen sizes the app is designed to work on
* define quality guidelines to test for basic UI design, features, & functions expected by Android users

Finally, after thorough testing, a signed APK can be uploaded to the Google Play Store or Apple App Store. Maintaining the app should include more testing and possible refactoring and reading user comments and feedback to improve and make a secure and robust product. Thanks for reading!

**References**:

Google Play. (n.d.). *Release with confidence*. Google. <https://play.google.com/console/about/guides/releasewithconfidence/>.

Hughart, Nic. (n.d.). App Monetization Guide: 10 Ways to Effectively Monetize Your Mobile App. BuildFire. <https://buildfire.com/ways-monetize-mobile-app/>.