**Option 1: Inventory App Launch Plan**

Rayyan Abdulmunib

Southern New Hampshire University

CS-360

**App Description:**

The inventory mobile app simplifies stock tracking for businesses. With its intuitive design and simplicity, it enables easy addition, modification, and removal of items. Not only that, real-time updates, categorization, and customizable alerts ensure accurate and efficient inventory control. Moreover, its multi-device sync and reporting features empower seamless collaboration and informed decision-making, optimizing stock management for enhanced business efficiency.

**Icon:** A black and white clipboard with check marks

Description automatically generated

**Version:**

Prior to deployment and releasing the app on Google Play store, I would verify that my application's API complies with their specified target API level, which presently stands at any level from 31 to 34 (Android version 12 -14), excluding Wear OS. Currently, the inventory app operates at level 34. Users will be prompted to grant permission for any newly integrated components within the app as well with every update.

**Permissions:**

* **Internet Usage:** The app requires access to the internet to enable various features such as syncing data with a cloud server, retrieving updates or information, or facilitating communication with external services. This permission allows the app to establish connections over the internet to fulfill its functionalities.
* **SMS Notifications:** By requesting permission to send SMS notifications, the app intends to utilize this feature to alert users about specific events or updates related to their inventory management. For instance, users will receive notifications every time a new item is added via text messages, should they choose to.

These permissions serve to enhance the app's functionality and user experience by enabling seamless communication and timely updates related to inventory management. However, it's essential for users to review and grant these permissions based on their comfort level with the app's intended usage of these features.

**Monetization:**

Monetizing the inventory mobile app can be achieved through various strategies. For example, implementing a freemium model, offering a basic version for free and a premium version with advanced features via subscriptions, would be effective for attracting users while encouraging upgrades for enhanced functionalities. Furthermore, in-app purchases for additional tools or features, along with non-intrusive ads or sponsorships within the app interface, would provide more revenue opportunities. Also, tailoring enterprise solutions for businesses with specialized needs and offering affiliate marketing opportunities with inventory-related suppliers would further diversify income streams. Lastly, data monetization through aggregated insights and offering white-labeled versions for licensing would present additional avenues for revenue generation. Nonetheless, I would choose a blend of these strategies tailored to user preferences and market demands that optimize the app's revenue potential.