Tanner Gaudes

CS-360

11/28/2023

**Project Three: Launch Plan**

**What will be included in your app's description and what kind of icon will best represent your app once it is made available in the app store?**

The app's description will include a high-level description of the app as well as descriptions of features and instructions regarding how to use them. It is important for users to understand what the app does and how to use it via the description, at least until a dedicated in-app help screen is created. Descriptions of each of the four dedicated screens are listed below this paragraph. Additionally, the app's icon should include an object that represents physical warehouse items, like a cardboard box for example. The user should be able to discern the functionality of the app based on just the icon, and a cardboard box portrays the theme of 'inventory' quite effectively.

**Login Screen**

Upon opening the app, users will be presented with the login screen. The login screen includes fields for username and password, and two buttons to either create a new user account or login as an existing user. Both buttons are disabled until there is valid input in both input fields. A new user can be created with a unique username; if the username is already in use, a new user will not be created. Logging in can be done using a valid username and password combination.

**Item List (Grid) Screen**

The app's main screen is the Item List screen, which is navigated to automatically upon user login. This screen includes a dynamic grid of all items in the database, as well as a button on the top right with a bell icon and a button on the bottom right with a '+' icon. The bell icon will ask the user for permission to send notifications when it is tapped; this will allow the app to send Android notifications whenever an item's quantity is reduced to or below zero. The plus icon will navigate to the Add Item screen, where users can add a new item with a name, price, and initial quantity. Each individual grid item can be long pressed to allow deletion of that specific item via a confirmation dialog. Finally, each individual grid item can be pressed to navigate to the Item Details screen corresponding to this specific item.

**Add Item Screen**

The Add Item screen is relatively simple, allowing users to add a new item with a name, price, and initial quantity. Once all three of the fields for these values have valid input, a button on the bottom of the screen will be enabled that will add the new item to the database and then navigate back to the now-updated Item List screen. The Add Item screen can be backed out of to cancel adding a new item.

**Item Details Screen**

The Item Details screen is like the Add Item screen in that it includes text fields (non-editable) for the item's name, price, and quantity. However, it also includes a button next to the quantity that allows editing the specific item's quantity. Tapping this button will display a dialog allowing the current quantity to be changed using the 'confirm' button or kept the same using the 'cancel' button. Changes to the item's quantity are immediately updated in the database.

**Which version(s) of Android will your app successfully run on? Have you included the most current version? Note that with each version of Android, new components are introduced that add considerations and challenges to development.**

The application requires a minimum API version of 28 and a maximum version of 33; it does not yet support API version 34. With version 28, the application should be able to run on roughly 88.5% of devices. Future updates will support newer API versions like the most current version 34.

**What permissions will your app ask for? Be sure these are only permissions that are necessary for your app to run. For example, does your manifest ask for permission to record phone audio when your app does not use it?**

There is only one permission that the application should request from Android and the user, and that is permission to send notifications. This notification permission will be required for the app to send low-quantity item notifications to the user's device whenever an item's quantity drops to or below zero. Since users and inventory items are stored on internal storage on the device, the application does not require explicit storage permissions like it would if the databases were stored externally. The only required permission (for notifications) is the minimum required permission for the app to run completely.

**What is your plan for monetization of the app. Consider whether your app will include ads, require a one-time payment, both, or neither.**

Monetization of the app could come in one of two viable forms: A one-time purchase or a subscription service. Advertisements may be an annoying distraction that makes the app less appealing to users and thus should be avoided. Other inventory tracker applications like those developed by Square do not include in-app advertisements, so potential users might opt to use those applications instead because there are no annoying advertisements. Additionally, advertising content may be unpredictable and result in occasional inappropriate ads that may be offensive to users; this is something that happens on various websites including popular ones like YouTube.

A single upfront cost is simple and easy and has the added benefit of avoiding annoying or even inappropriate advertisements. However, a one-time purchase means that there may not be a steady source of income for the developer, which may be inviable for an application that must be continuously maintained. Thus, modern software frequently chooses to employ a subscription-based model for monetization (for example, Microsoft Office and Dashlane). This allows for consistent income for app developers, supporting continuous app maintenance in the form of bug fixes, new features, customer support, and so on.