Module 3: Project 1

Shannon Musgrave

Computer Science Dept. of Southern New Hampshire University

CS-360 Mobile Architecture & Programming

Professor Krauss

01/26/2025

The inventory application will serve the purpose of allowing users to manage an inventory using their Android phone. To do this, the program will require a UI to display the necessary information while allowing users to manipulate and adjust the data. The login screen will need some fields and a button that will allow the user to log in or create a new account. Next, the inventory page will display an infinite scroll design so that the list is not limited to a few items. The items will need to be held in a collection, such as a grid or table, that will display them line by line. In addition to these screens, the page will also need to allow the user to add new items and adjust quantities.

Although this application will be for items in a “warehouse,” residential users will still make up a user type since there are many home-based businesses and hobbyists who could make use of the inventory app. The remaining users can be divided into those with industrial or commercial needs. Examples of these would be a person selling crafts online who would like a simple application to keep track of supplies, a small coffee shop owner managing stock levels of coffee beans and pastries, or a farmer using the application to maintain their equipment's parts.

Each user segment will have different expectations, and the application must strive to meet as many of these as possible without experiencing “scope creep.” For instance, a farmer might wish to have a part number along with the item name, while the coffee shop owner may wish to have a name and description. To accommodate both, a nondescriptive “note” field will be available that can be used for either. In each case, the application can be used to help an individual be successful in their endeavor and aid them in their goal of increasing productivity through better management of material. To use this app, all users are assumed to have an average skill level with mobile devices, and in addition to this, the inventory can be kept on local device storage, meaning offline use will be supported, which is particularly beneficial for users in areas with unreliable internet access.

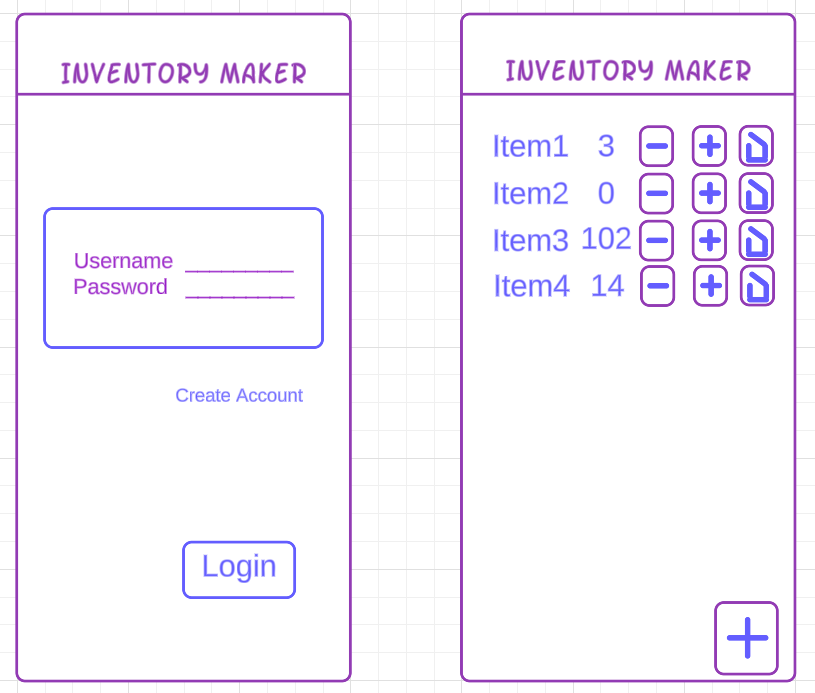
The app will feature two screens: a login screen used to log in or create a new user account and an inventory screen where the user will spend most of their time. Because of the simple two-screen layout, a tab-bar, or some other advanced navigating method will not be necessary. The login screen will follow the familiar pattern of text views and edit views which will be necessary to enter the username and password. Additionally, a button will be used to submit the information and progress to the inventory screen. If the user has never created a password, the user will be notified, and a third edit view will be visible for password confirmation.

The inventory screen will feature a constraint layout that will hold a main linear vertical layout. Within this linear layout, child views will be dynamically added for each row. This child view will hold inventory items, comprising the item name, note, quantity, and three buttons to increment/decrement the quantity, and to delete it. A floating action button will be anchored to the bottom right corner of the screen using the main constraint layout; this button will allow a new inventory item to be inserted into the list. Finally, if an item is reduced to zero, a toast will be generated to let the user know. The item quantity will be an unsigned integer since it does not need to hold negative values, and once the value is at 0 or Integer.MAX\_VALUE, the buttons will be disabled. In addition, colors can be utilized to indicate this and to emphasize a quantity of zero.

Navigation between screens will utilize the login button which will take the user to the inventory screen if successful. If the user suspends the application, they will continue to be logged in and will reopen the application to the inventory screen; however, if they stop the application, it will default to the login screen where they must log in again. To handle this, the app uses Android's onSaveInstanceState method to temporarily store the user's login status during suspension. Upon app termination or forced closure, all login information is cleared from memory to ensure security, prompting a new login when the app is restarted.

The app will utilize an analogous color scheme, which uses colors that are next to each other on the color wheel which “naturally look pleasant together” and create "no jarring clashes for the eye" (Leo, 2023). These colors are indigo and blue, which will have a high contrast against the background. The app will also ensure the font size is larger than 12sp to align with best practices. To design for vision accessibility, developer.android.com states that one should “ensure your app's content is as legible as possible by checking color contrast and text sizing, and that components are visually comprehensible” (Accessibility, 2023).

Rough draft of Inventory Maker.



The code will reside in two activities, which will power the login and inventory screens. In addition to this, each activity will also have a compositional, or "has a," relationship with a model class that will represent the business logic. For instance, the inventory screen will utilize an InventoryActivity that will hold, as a property, an InventoryModel. This model will be responsible for any actions performed on the inventory. The activity code must override several methods to allow the proper behavior to be programmed for the app's lifecycle, such as onCreate and onDestroy. The flow of data between the UI and the model will reside in the activity, which will use methods like findViewById to get a reference to the specified component. The data that will make up an inventory item will consist of the name, note, and amount, composed of strings and an unsigned integer and conversely, the login page will use strings for both the username and password hash.

**References**

Leo, C. W. (2023, September 21). What is an analogous or adjacent color scheme? Color with Leo. <https://www.colorwithleo.com/what-is-an-analogous-or-adjacent-color-scheme/>

Accessibility. (2023). Android Developers. <https://developer.android.com/design/ui/mobile/guides/foundations/accessibility#design-vision>

‌