Stephen Cardone

Prof. Jeff Phillips

CS-360

March 21, 2021

Mobile2Apps : Inventory Application Development Proposal

**Purpose and Goals:**

For this project, I have chosen the Inventory Application. The primary purpose of this application is to “track items in a warehouse.” Below is a list of goals for this project that will be met to fulfill this application:

* Inventory app will be available on Android
* The app will require a login to be able to restrict actions by user
* Inventory app will allow users to create new products to track
* Users can increment/decrement inventory as items arrive or leave
* All inventory data will persist on device when the app is closed or the phone is power cycled.
* The application will notify the user whenever an item’s stock goes to 0

**Users and User Roles:**

Now that the goals of the project are laid out, we can explore the users and user types that we can foresee using the application. An inventory application can be utilized by a variety of people. By keeping a simple design with simple, intuitive features, we will be able to supply a useful product to all of them. One example user would be a store with a warehouse. The warehouse manager could keep track of the inventory for items that are sold in the storefront. This way customers could be updated on inventory without someone having to go check the warehouse. Another example user would be of a home business. A simple UI such as the one I will design will allow for a small business to track their products effectively without having additional unnecessary features or cumbersome advertisements. For users like these, we could become an essential part of their daily productivity. By maintaining the simple and intuitive design style, we can perform this important role for the users while allowing them to quickly return to their other duties since the UI is clear and easy to use. Now that we have described some example users, here are the prospective user roles that the application will require and a I have split the example users into each category:

* Level 0 (read only): Can only see stock values and cannot change values. ( Front end sales clerk for checking current inventory)
* Level 1 (read/write): Can see stock values, and increase/decrease stock values based on shipment or sold items. (Warehouse workers)
* Level 2 (admin): All of the above, plus additional features such as reports and audit information. (Store Manager, Small business owner)

**Screens and features:**

With our goal of a user centered UI, we need to take the user goals listed above and create a system of views that will be able to accomplish those goals. First we will need a **LoginView:** This will be the first view the user sees. It will allow them to login to the application and send them to the next view

**InventoryView (This is the default view after login):** This view will display the list of all inventory. Ideally it will have a search bar that allows users to search for Item number, or key words to quickly find a specific product. At the bottom of this view will be the Buttons that navigate to the views to Add/Remove items that are tracked in inventory, as well as increase/decrese the stock of a given item.

**Add/Remove Item view (Accessible from a button on the InventoryView):** A view that allows the user to add a new item to inventory. It will display editText views that the user will fill out and a button to add the items. The view will also have a Remove button that will allow the user to remove items from the inventory based on the part number.

**Increase/Decrease Inventory View (Accessible from a button on the InventoryView):** This view will bring up an interface that allows the user to select a part number, as well as a number (positive ore negative) and allows the user to edit the current inventory count for a given item.

**Code Design:**

The login view will have a 2 editText fields that accept username and password. The password field will use the password type to abstract the real text. There will be a submit button that triggers a callback in the java code that accepts the username and password as parameters and verifies them agains the known accounts. If the account is real, the user will go to the next page, otherwise they will be notified to re-enter credentials. The loginView will also have a button to Add a new account.

The inventoryView will be a grid view that contains the list of all the items in inventory as with stock value as well as any other metadata we want to include (Description, price etc.) The user will be able to scroll down the list. At the bottom will be buttons that are always on top. Hitting one of these buttons will trigger a callback that prompts the activity to change intent to the corresponding view (Either add/remove items or edit stock views).

The Add/Remove stock view will have editText fields and when a user taps the “Add” button, there will be a callback in the app that accepts the new data info as parameters. This function will add the item into the inventory and update the database. The remove will behave the same way but will remove items from the inventory database.

The Edit Stock view will be very similar to the add/remove stock view. It will be laid out with edit text fields that the user can fill in and a button to update the inventory. Upon button tap, a callback function will receive parameters for the Item, and the requested adjustment. This function will update the inventory data and update the database.

We also have an additional requirement for a feature to notify whenever an inventory item goes to 0. To accomplish this, there will be an inventory class responsible for interacting with the database. Whenever a change is made to the database, we will check for a 0 value on the items that have been edited. If the new updated balance is 0, we will know that the item is out of stock and notify the user via a toast message, or other visual indicator in the inventory view.