**Mobile Computing - AndroidOS Final Project Proposal**

**Team Members: Nick Pierce, Matthew Berry, Jared Corder**

**App Name: Smart Shopper**

**Overview:**

One of the most arduous, and dragging, aspects about shopping is locating where all the items are. New customers are always staring at undescriptive signs and asking themselves, “why is this item not here?”. Our team’s app, **Smart Shopper**, is designed to facilitate that predicament, and allow store managers to have the flexibility of moving item locations without all the hassle. This app is comprised of 6 tabs.

Tab Zero (Welcome Screen tab): Initial view displayed to both consumers and admin. Users will select which role and corresponding store and be “intent” over to the appropriate subsequent view. A spinner exist to facilitate the selection.

Tab One (Store Screen tab): A virtual, noninteractive, display of the stores schematic. Inside the display, aisles will be individually labeled with their corresponding department name. Non-aisle departments- such as produce, meat, dairy, etc.- will simply retain a label of its name under its virtual location. Below will be a fixed, navigation bar to permit ease in traversing to other tabs in the app; the home tab on the navigation bar is this tab.

Tab Two (Search tab): When opened, only a title “Search item” and a search bar will initially appear. Intuitively, users will subsequently enter the item’s name in the search bar (e.g.: “Cheez-its”). The search bar retains an auto-complete feature to predict customer queries. Partial queries (“Cheez” instead of “Cheez-it” are accepted); best matches will be determined and a subset will display on tab one- showing all locations of matches and displaying the aisle number and/or department section- varying on if the department retains aisles or not. A fixed navigation bar and a search again button exist on the bottom.

Tab Three (Admin Login): When opened, the app’s logo, username, and password fields are displayed. The admin is prompted to enter a username and password to login into the admin features tab.

Tab Five (Admin Hub): When transferred from Admin Login tab, users are given there employee information. Two buttons will be displayed, each linking to their functionality. One button will go to the create an admin (this is only visible to store admins). The other button will go to product modifier

Tab 6 (Product modifier): Once delivered to this tab from the Admin Login tab, the admin can alter three things: the location of an item, the appending of a new item, and the removal of an existing item. These features are encapsulated in the first three buttons: add, remove, modify. The user will put the barcode in and click one of three buttons: Add, Update or Remove. Clicking all three of these buttons will display the following fields below: Store Name, Vendor, Department, Isle, Search tags. If it is an update or delete action there will be a check to see if that barcode exists, and alert the user if it does not. For update, the fields will be editable and the user can edit the fields accordingly (this may change once the database is created and we establish foriegn keys). If it is a delete the fields are not editable. The user is just seeing them to confirm this is what they want to delete.

If it is an add there will be a check to make sure that the barcode is not in the database. If the check is valid, it will allow the fields to be edited.

Once the user is done, they click the submit button. A status will appear if the submit works successfully. Users can return to hub with the back to hub button

Tab seven(Admin modify): This field allows store admins to create admins.the admin can alter three things: the update an admin, creating a new admin, and the removal of an existing admin. These features are encapsulated in the first three buttons: add, remove, modify. The user will put the admin id in and click one of three buttons: Add, Update or Remove. Clicking all three of these buttons will display the following fields below: Name, Employee id (not editable), Password, and a rank checkbox that indicates if that admin has rank perms,an update or delete action there will be a check to see if that admin id exists, and alert the user if it does not. For update, the fields will be editable and the user can edit the fields accordingly (this may change once the database is created and we establish foriegn keys). If it is a delete the fields are not editable. The user is just seeing them to confirm this is what they want to delete.

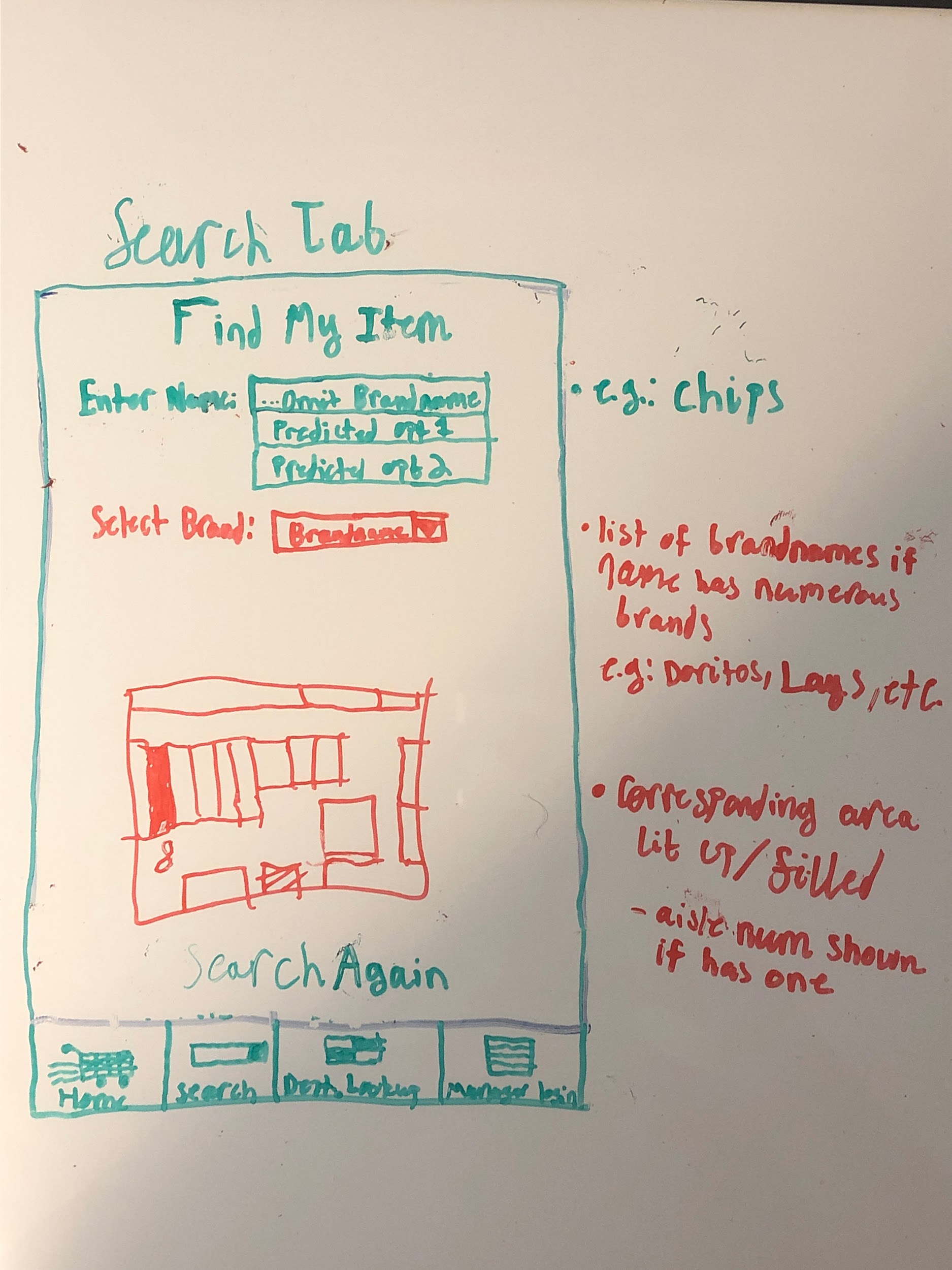
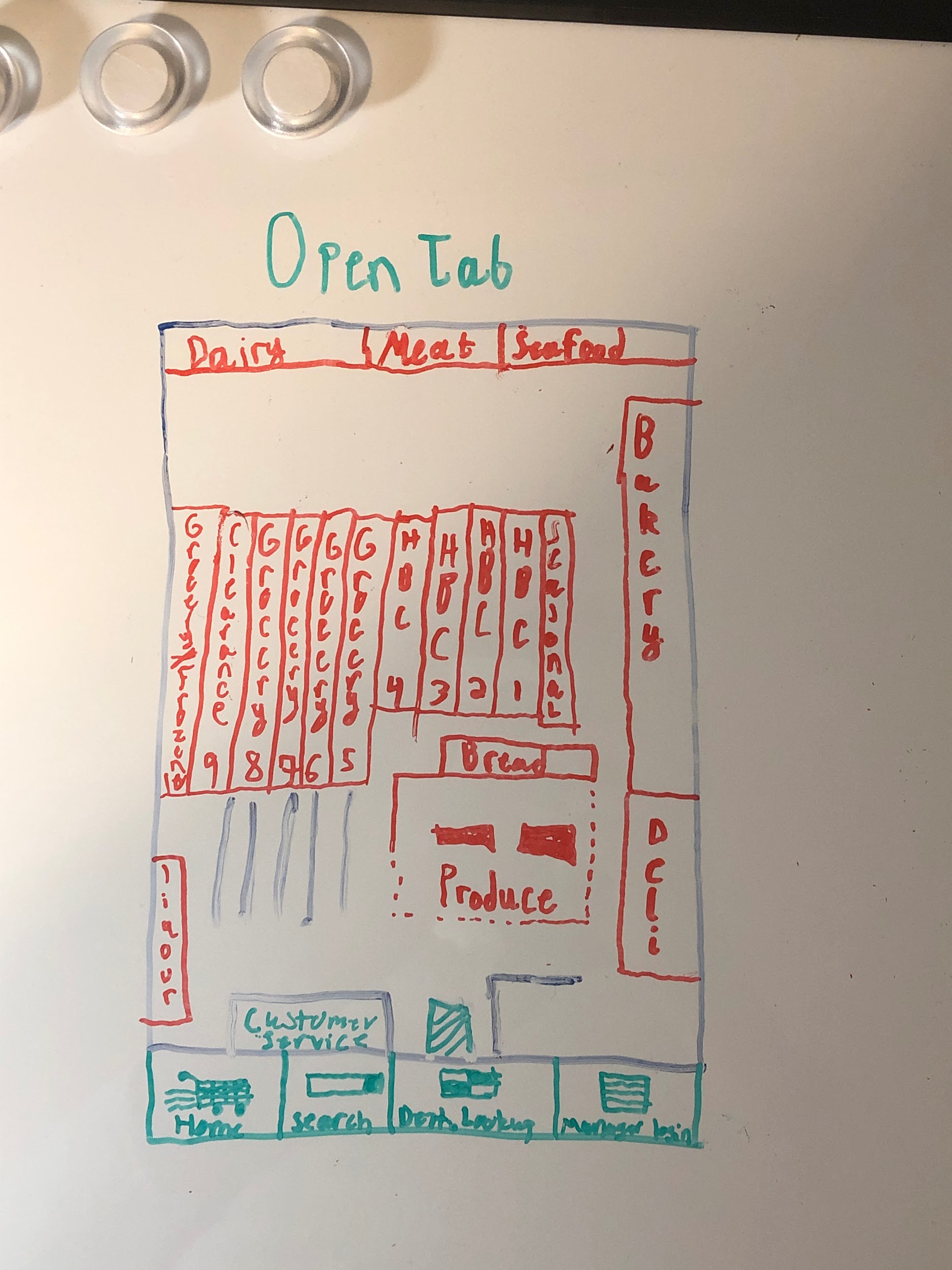
If it is an add there will be a check to make sure that the admin id is not in the database. If the check is valid, it will allow the fields to be edited.

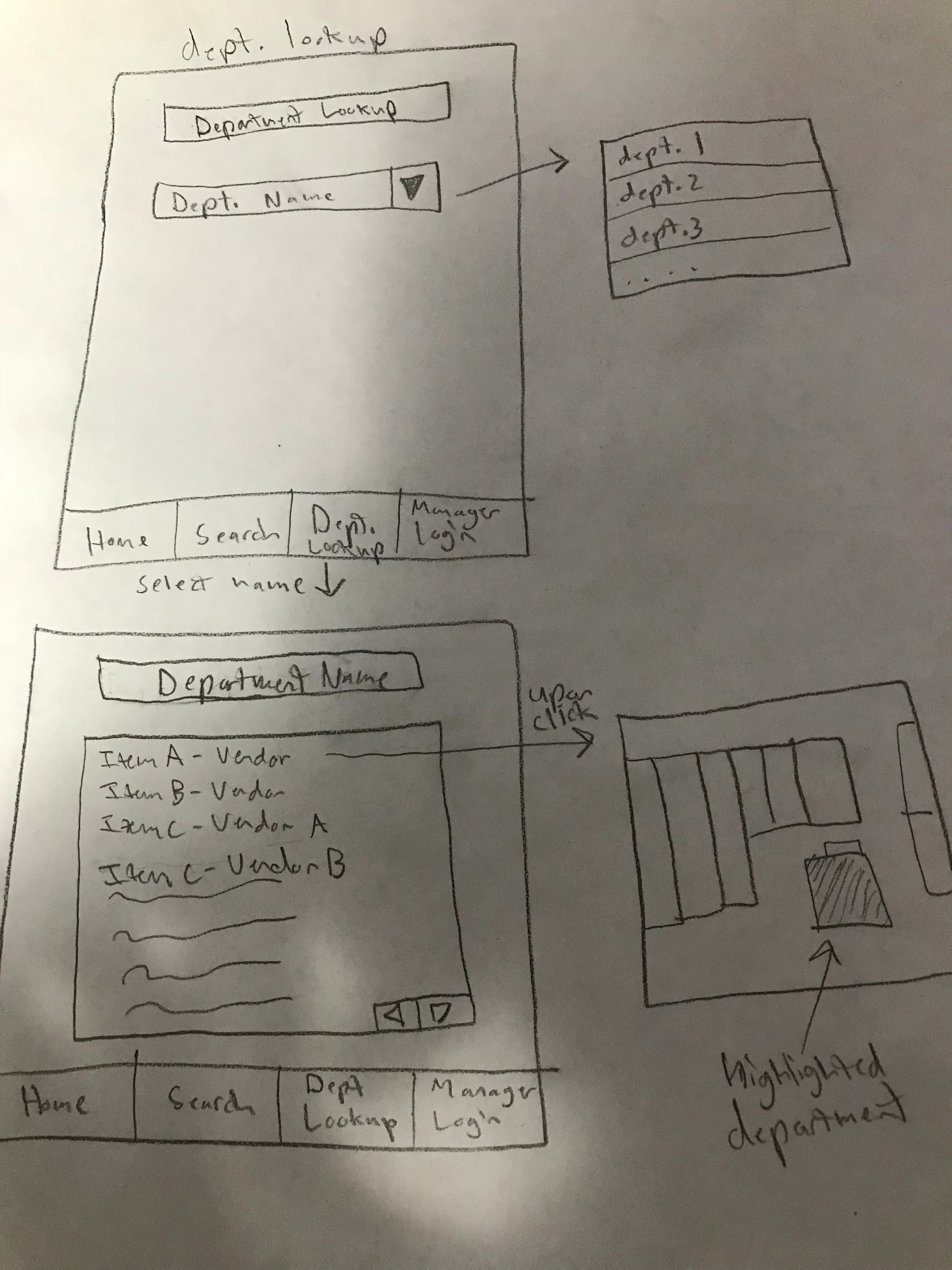
Once the user is done, they click the submit button. A status will appear if the submit works successfully. Users can return to hub with the back to hub button

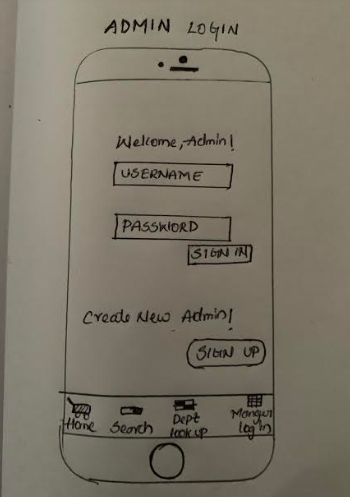
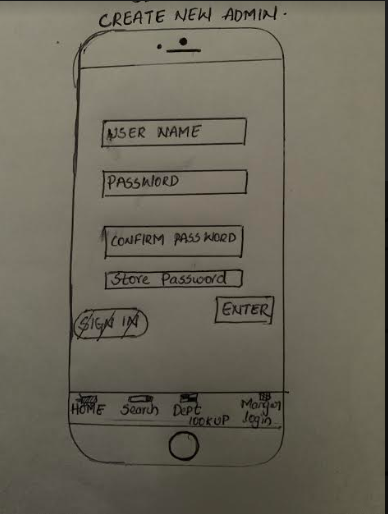
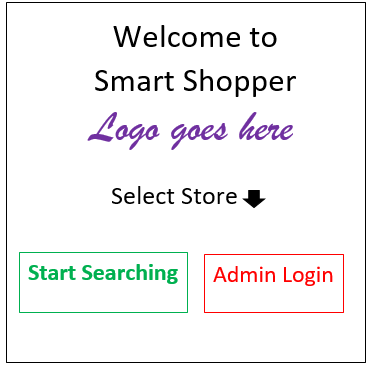
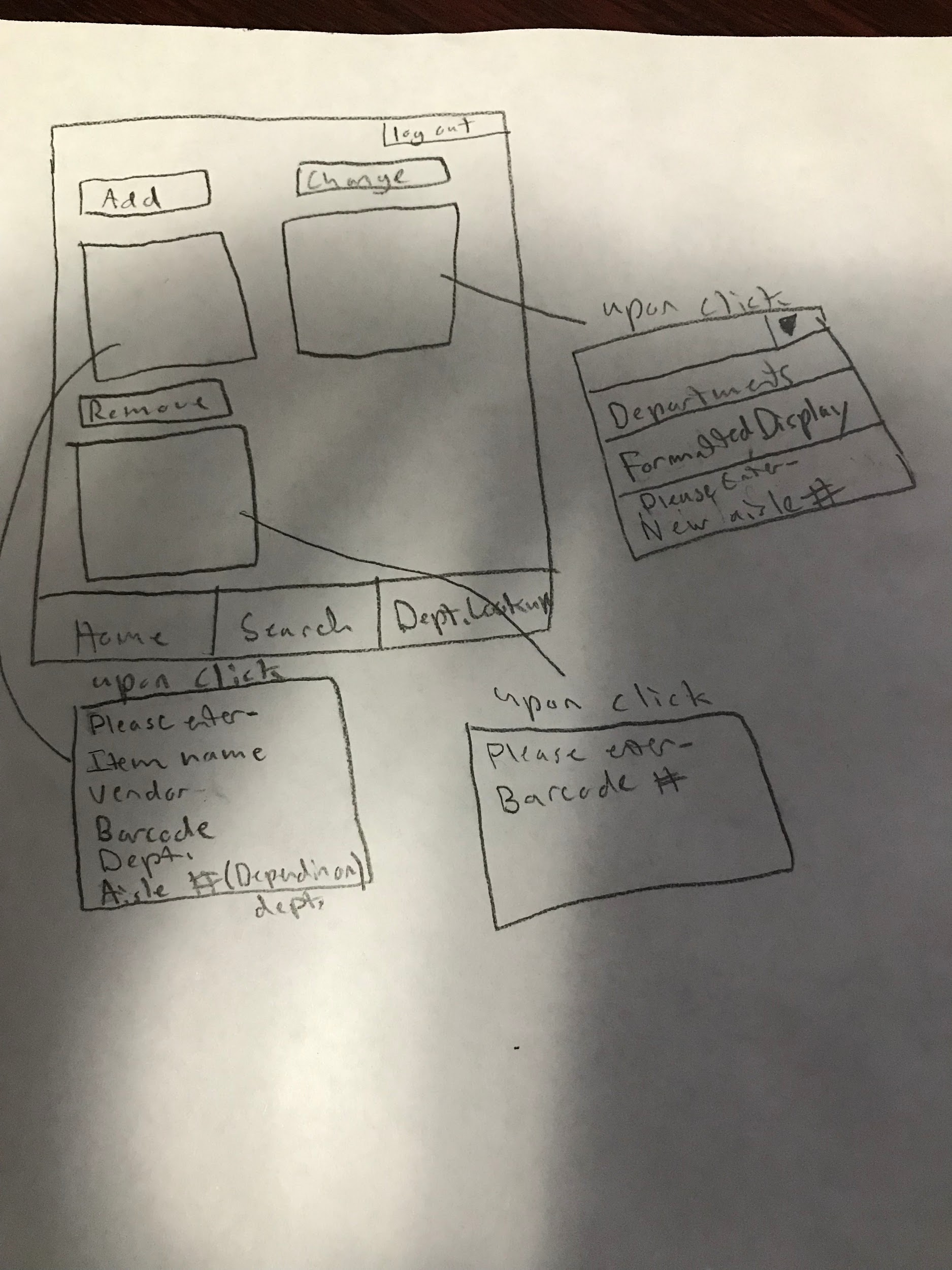
**Risky/External Components:**

1. Utilizes adapter/DLL from MBAAS (remote data persistence)
2. Invokes architectural design and patterns
3. May utilize responsive design to be optimal across all sizes

**Sketches:**

****



****

**Flow:**

Initial activity: welcome screen (store selection).

* Can use nav bar on bottom to navigate to any layout aforementioned above on user side only after initial intent from starting activity. If traversed to login for admin, can enter valid login to proceed to admin activity layouts w/o nav bar
  + If opted into to render new admin, transferred to new admin + store authentication and item layouts
  + If the admin clicks on the logout button at any time then they will be transferred back to the home screen
* If search action is invoked on user side then they are conditionally taken to the the store map screen where they can start search
  + When search is created a text view appears to enter the desired product.
  + That item is then searched and result is determined. **The location of said product will light up, with other locations becoming transparent, to indicate visually where the item is at along with text view w/ auxiliary information.**