4-4 Assignment

UI Design

Alex Kastigar

7/27/25

CS-360-16014-M01 Mobile Architect & Programming

4-4 Assignment: UI Design

I reworked the Inventory App's main grid screen since that's where people spend most of their time and where speed and readability are particularly vital. My goal was to allow warehouse managers and floor clerks to locate, filter, and update items with speed, and to bring low-stock situations to their attention without extra taps.

I put an AppBar at the top with the app logo, Search icon, and overflow menu for application-wide settings. Below that is a row of filter chips labeled "All," "Low Stock," and any user-defined categories (e.g. "Beverages," "Electronics"). These enable users to filter the grid immediately by tapping a chip, crucial for clerks who need to prioritize high-priority items.

The foundation of the screen is a RecyclerView in a two-column grid of CardView items. Every card displays:

1. An ImageView for an optional product icon
2. A TextView for the item name, in Material's headline style for hierarchy
3. A Quantity badge. A Chip with color-coded backgrounds (green for ample, orange for low, red for none)
4. Two small ImageButtons ("–" and "+") at the bottom for direct count changes

Swiping a card to the left reveals a Delete action, following standard list‑swipe conventions. A FloatingActionButton (+) floats over the bottom‑right corner to create new items. Finally, a BottomNavigationView anchors the screen with three tabs: Inventory, Alerts (new), and Settings, which keeps key destinations a single tap away.

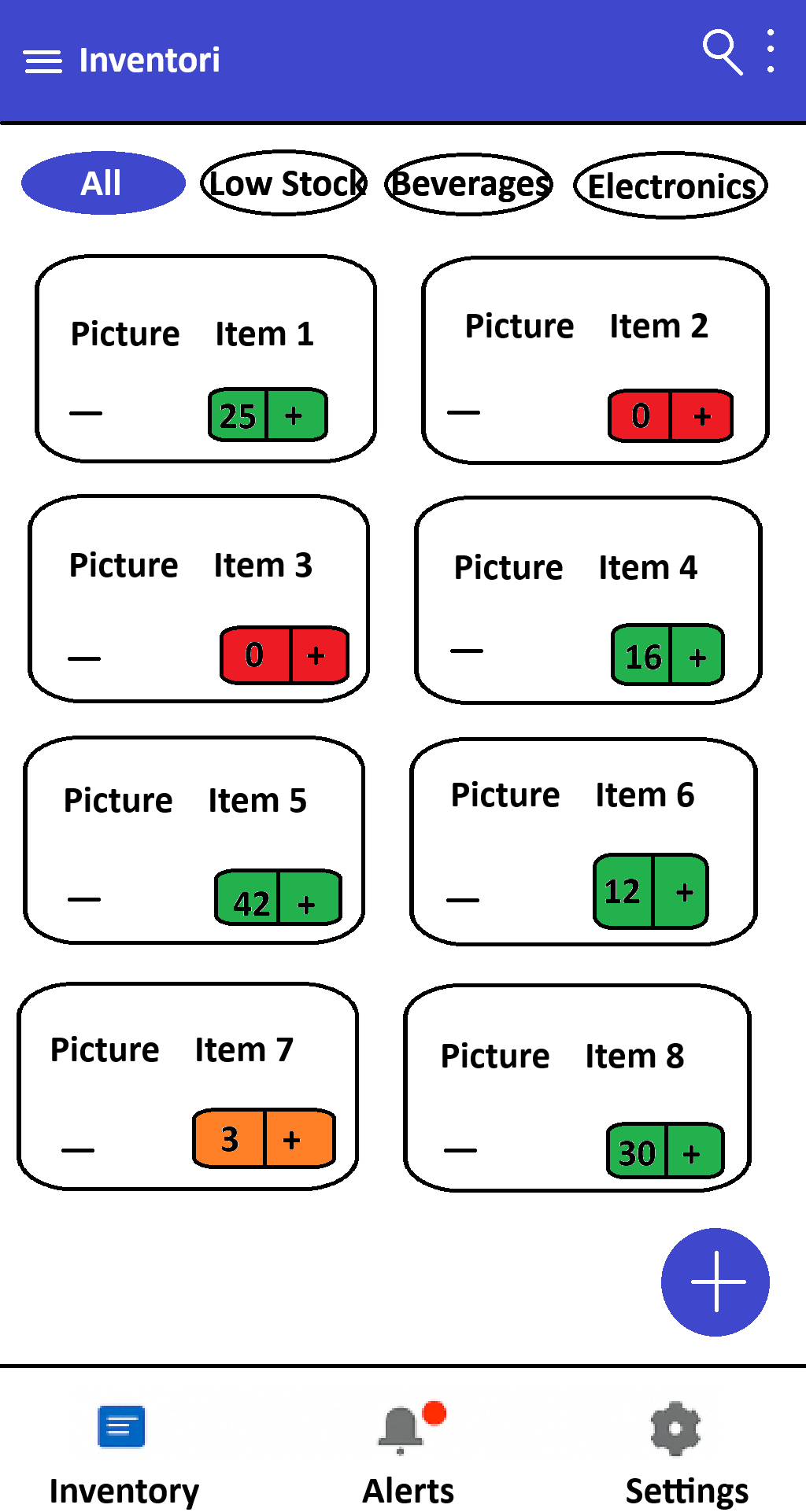
I used Material's type scale (headline for item names, body text for badges) to direct the eye naturally from name to quantity. The stock quantity badge chip utilizes Material color semantics to convey stock levels at a glance, adhering to Android's convention on color -> status relationships (Material Design Layout). All actionable elements (chips, cards, buttons) meet the 48 dp minimum tap target size. Grouping filter chips and tabs into distinct bars reduces cognitive load. Consistency & Affordance: Swipe‑to‑delete and FAB interactions follow platform conventions, so users immediately understand actions.

A filter chips and search icon satisfy the user need of instant discovery, enabling managers and clerks to locate items without scrolling through hundreds of cards. Quantity badges with color coding provide proactive notification by making low‑stock items visually stand out, so that no one misses a count of zero. Placing plus/minus buttons directly on each card satisfies frictionless updates, the second key user need. Putting navigation in a BottomNavigationView ensures that adding items, checking alerts, or toggling settings is never a back‑and‑forth menu hunt.

I added an Alerts tab to show a list of zero‑stock items in one place. I wasn’t sure if how central I needed this but it felt right to have it ready in one tap instead of an obtuse filter system.

To avoid accidental quantity adjustments, tapping the card (outside of the plus/minus buttons) expands it to a detail view rather than adjusting quantity. This novel use of card expansion separates navigation from instant actions.

By reorganizing screens with obvious grouping, sensible navigation, and color‑based status cues (while following Android's Material guidelines for layout, components, and typography) this redesign makes inventory management easier for both experienced power users and intermittent operators.



References:  
Google LLC. (n.d.). Design & Plan. Android Developers. Retrieved July 20, 2025, from <https://developer.android.com/design>