GitHub Username: TheBaileyBrew

Inventory Manager

Description

Inventory Manager is a simple Android app, intended for managing all your stock & inventory needs. This application allows users to create all the details of their inventory item, and manage levels with clean visual guides. The application stores data in a cloud server and can be accessed from multiple devices via unique user logins. Each user utilizing the app can be assigned a role and have access to different data or views.

Intended User

The intended user for this app is anyone who has a collection of items to manage in an inventory and wants to keep track of usage and be notified when it's time to reorder supplies again.

Features

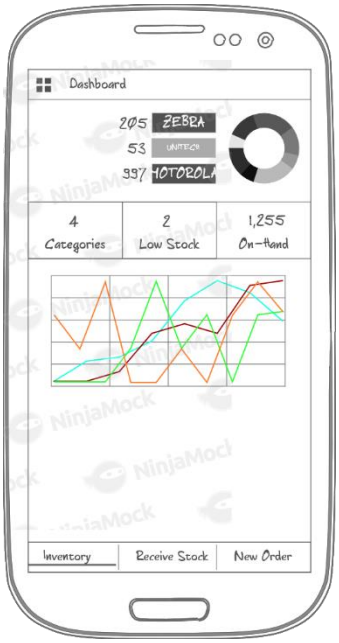
- Saves inventory levels

- If items have barcode, barcodes can be scanned and attached t individual items in inventory
- Visual representation of stock levels

User Interface Mocks

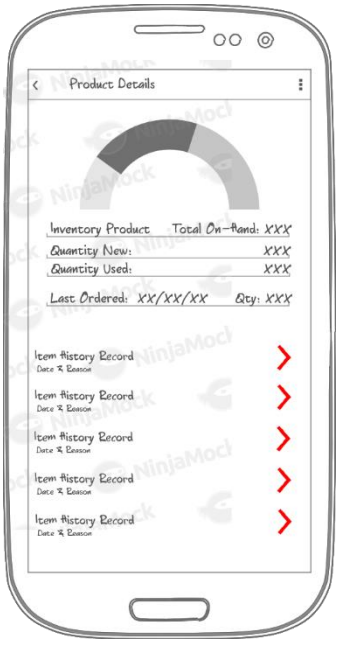
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1



The Dashboard is the primary screen with basic details about the inventory itself. This view shows current quantities of either individual items or categories (default selected as user setting)

Screen 2



The Product Details Screen shows overall details about each individual item category. Bar at top will show percent of total available and numbers below will represent the totals. Recycler/List at bottom will show history of the category (Orders received, returned, checked-in)

Add as many screens as you need to portray your app's UI flow.

Key Considerations

How will your app handle data persistence?

Data for user details will be stored locally via Room. Data for inventory will be serviced by Firebase (with local Room replica for offline use).

Describe any edge or corner cases in the UX.

For example, how does the user return to a Now Playing screen in a media player if they hit the back button?

Describe any libraries you'll be using and share your reasoning for including them.

Picasso, Expandable Menu, DrawerFrame, Xzing.

Picasso will be used for image displaying. Expandable Menu will be used for pop-up display on item clicks. DrawerFrame will be used to show the hidden left nav menu. Xzing will be used for camera intent and barcode scanning.

Describe how you will implement Google Play Services or other external services.

Firebase Storage, Authentication + Cloud Messaging

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

First I need to prepare the framework designs. Then create the repo, migrate to AndroidX and add the library implementations

Task 2: Implement UI for Each Activity and Fragment

- Create the Login Activity + Logic
- Create the Main Dashboard
- Develop the UI for breakaway screens

Task 3: Work out backend logic

- Incorporate libraries and external resources
- Link Firebase
- Set up Room Instance + ViewModels

Task 4: Debug & test

- Test on multiple devices, validate onSave/onRestore