**How to Use this Template**

1. Create a new document, and copy and paste the text from this template into your new document [ Select All → Copy → Paste into new document ]
2. Name your document file: "**Capstone_Stage1**"
3. Replace the text in green

---

**GitHub Username**: ShowYoungg

# Inventory Management

## Description

Inventory Management is an app that manages inventory, helps keep track of the flow of inventory i.e how inventories of different sort go in and out, the unit cost price and sales price; and generates accounting records such as Purchase and Sales Journals, Account Receivables and Account Payables, Price List, Profit and Loss Account.

This app will ensure:

- Accounting record of stocks are kept
- Price list that will feature cost price and selling price will be featured
- Stock inflow and outflow management

- Cash Book and Bank Statement are prepared
- Account Receivables and Payables are prepared
- Trading, Profit and Loss account is generated.
- Accounting records of all sales and purchases are kept
- Accounting records for debtors and creditors are kept
- Product pictures are also featured
- All records can be exported to csv file which will be saved on device memory

## Intended User

This app is intended for use by, but not limited to the following;
- Stock Keepers
- Warehouse Managers
- Supermarkets
- Grocery Stores and Pharmacies.
- Book Keepers

## Features

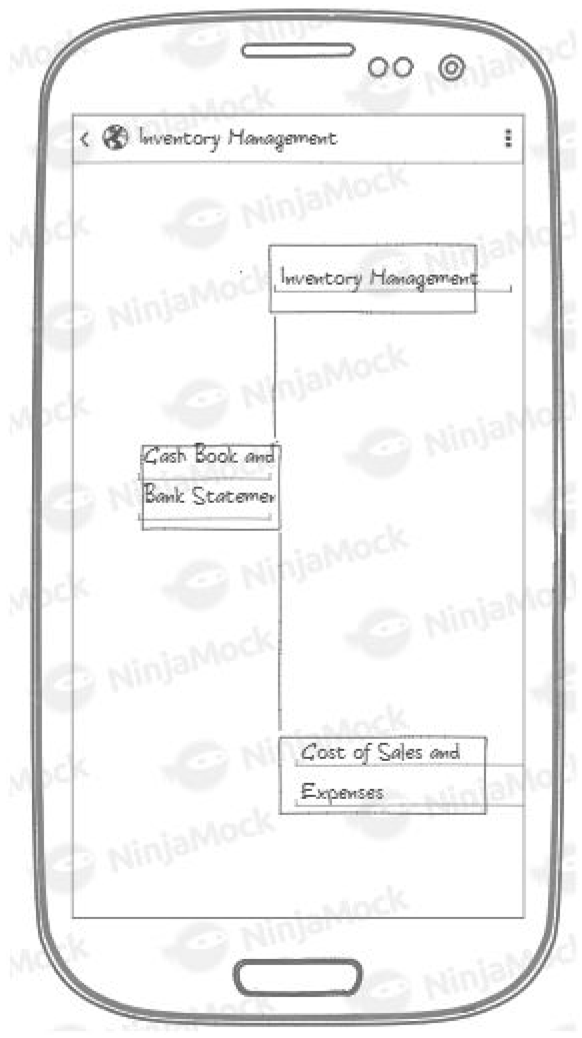
The following are some of the app features;

- Saves information about stocks i.e cost price and selling price, quantity, name and numbers sold (outgone stock) and purchased (stock in) in a database.
- Takes and saves product pictures.
- User can export data into csv file and save such file on the device memory.
- Data is backed up offline on the device.
- The app is capable of generating profit and loss account, cash book and bank statement, account receivables and payables, price list and sales and purchases journals.

## ● User Interface Mocks

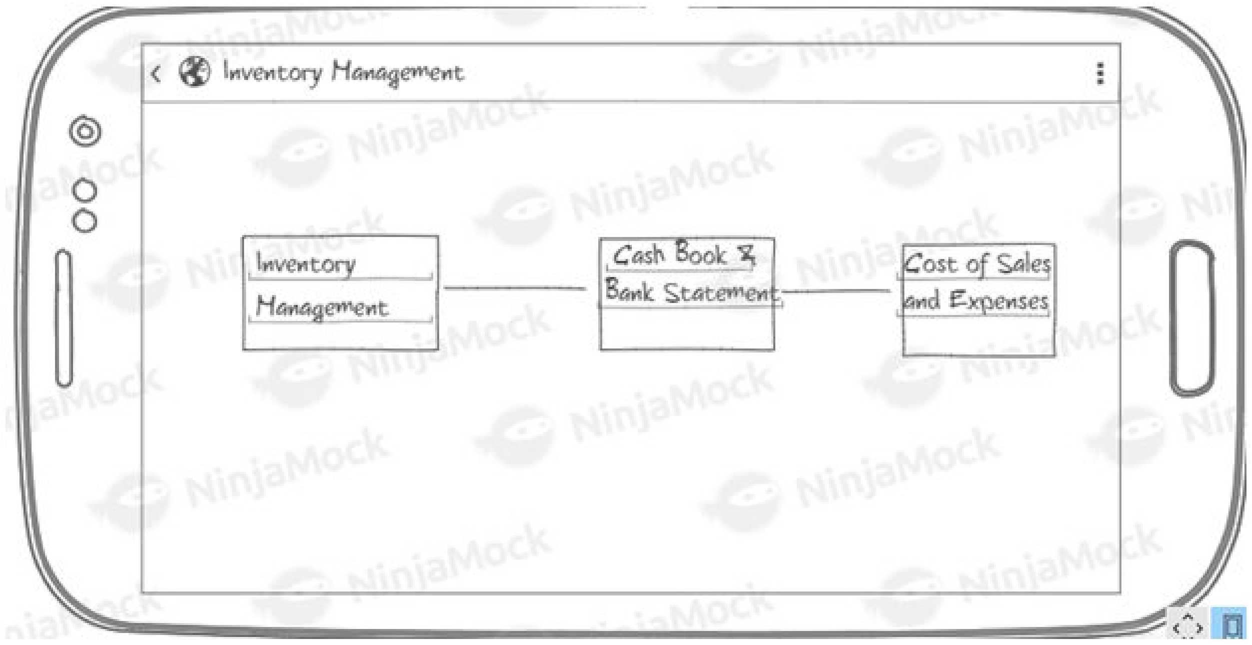
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

## Screen 1 (Home Page -- Portait)



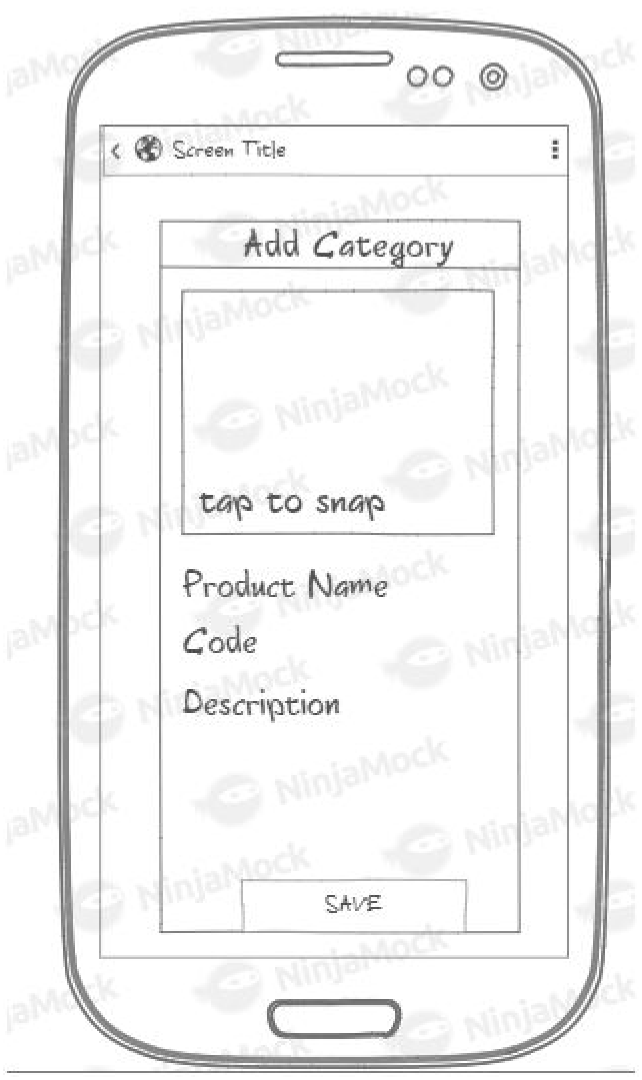
This is the home screen in portrait mode, it contains three ImagesButtons (Inventory Management, Cash Book and Bank Statement and Cost of Sales and Expenses).
These are three ways users can input data; depending on the user's objective, either or all of the functions which the images represent can be used in a transaction.

## Screen 1 (Home Page -- Landscape)



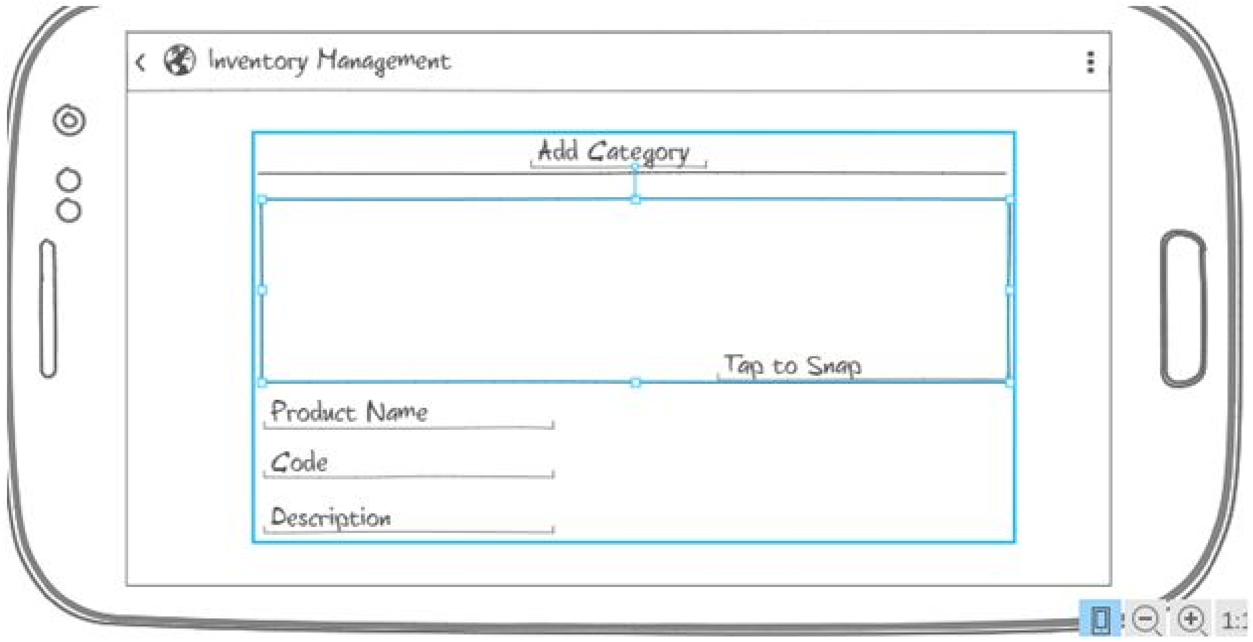
This is the home screen in landscape mode, it contains three ImagesButtons (Inventory Management, Cash Book and Bank Statement and Cost of Sales and Expenses).
These are three ways users can input data; depending on the user’s objective, either or all of the functions which the images represent can be used in a transaction.

## Screen 2A (InventoryManagementActivity)



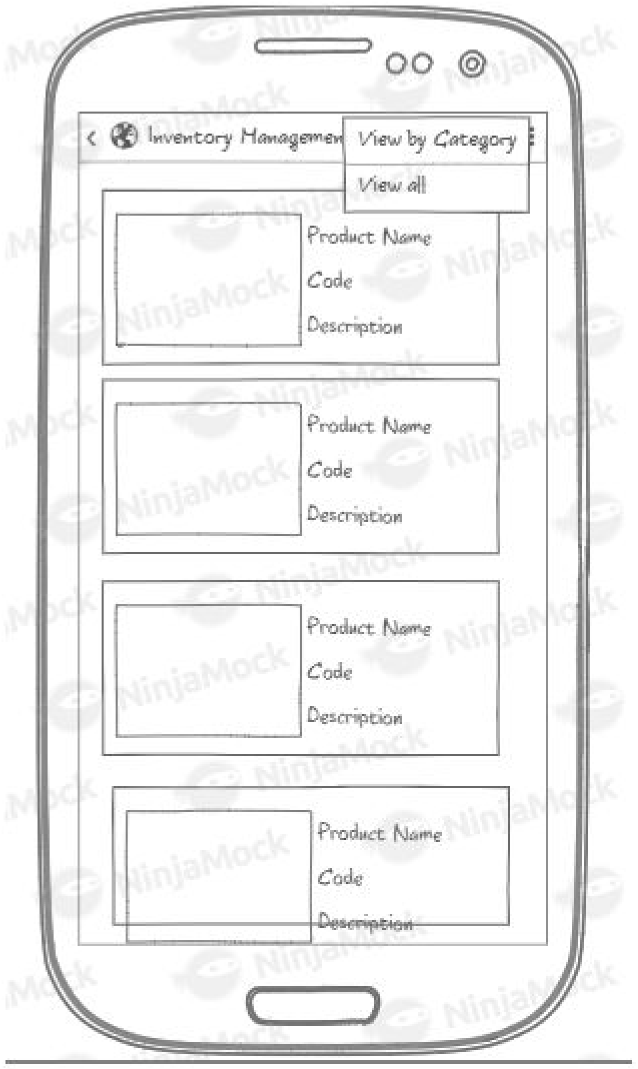
From the home screen, if the first image button (Inventory Management) is clicked, this screen (InventoryManagementActivity) is opened, the activity displays a dialog bar which requests for product information with a button to save the information.

**Screen 2A (InventoryManagementActivity – Landscape)**



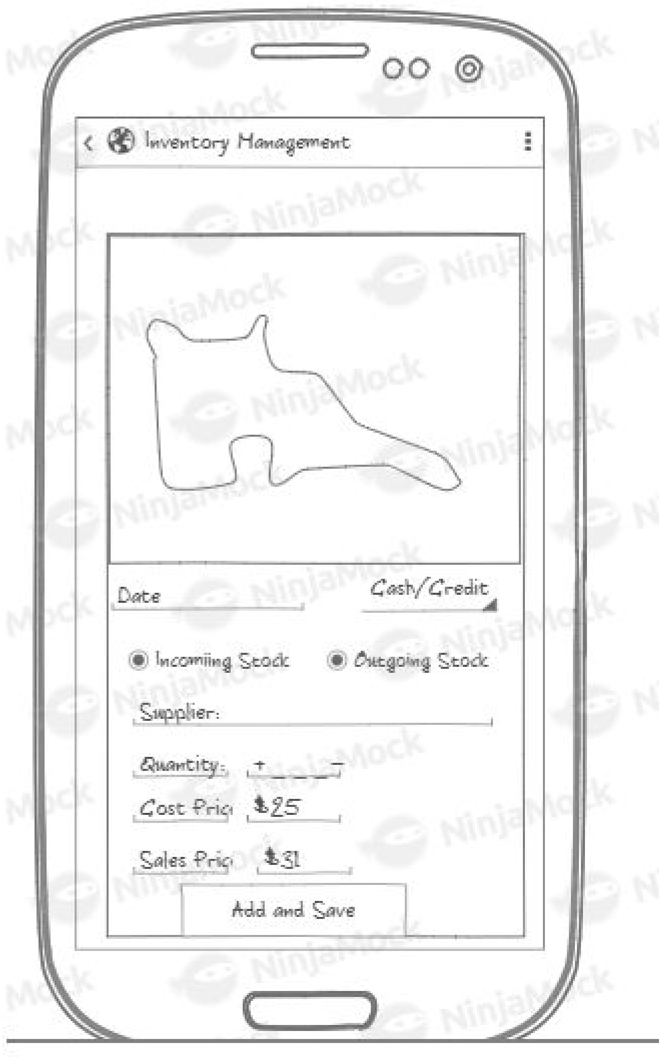
From the home screen, if the first image button (Inventory Management) is clicked, this screen (InventoryManagementActivity) is opened, the activity displays a dialog bar which requests for product information with a button to save the information.

## Screen 2B (InventoryManagementActivity)



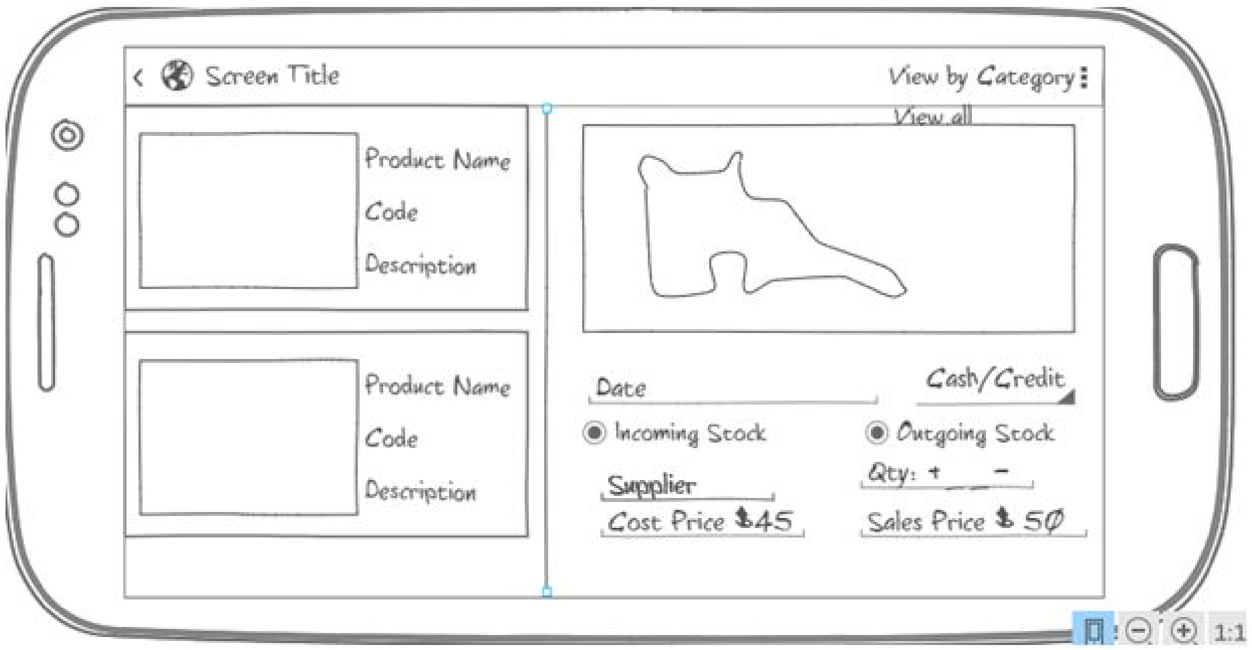
When users save information gathered by the dialog box in screen 2a above, the information will be shown in this activity, this is the InventoryManagementActivity covered by the dialog box in screen 2a above.
Users can view the information saved, either by product category or as entered, as a list.

## Screen 3A (ProductTransactionFragment)



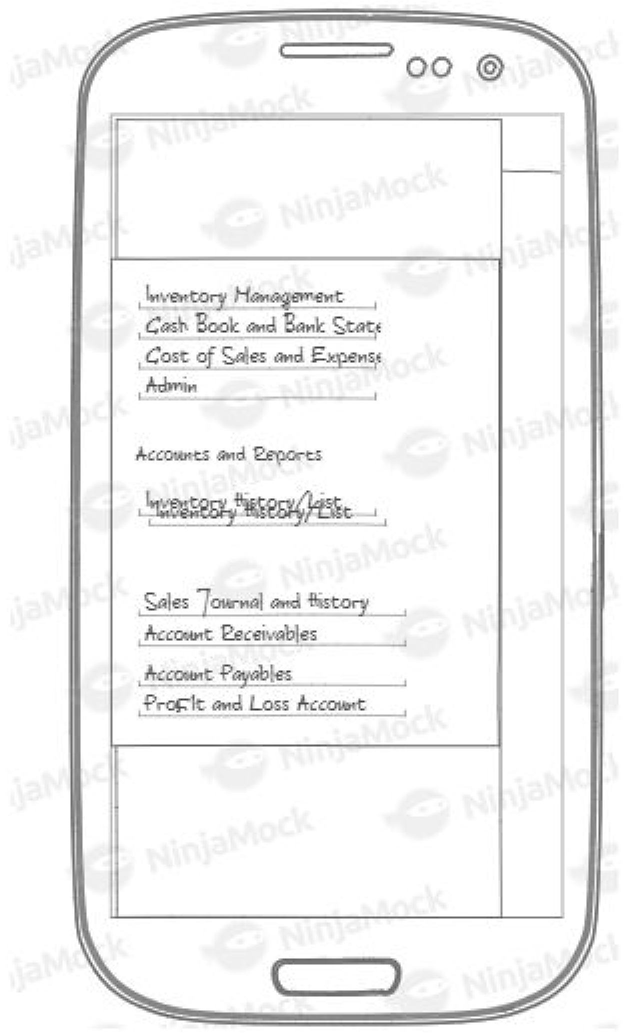
When users click on an item in the list on screen 2a (InventoryManagementActivity) above, ProductTransactionFragment is launched, this fragment will display all the information relating to the item selected in the list for users to perform a sales (outgoing stock) or purchase (incoming stock) transaction.

## Screen 3B (InventoryManagementActivity with ProductTransactionFragement on landscape)



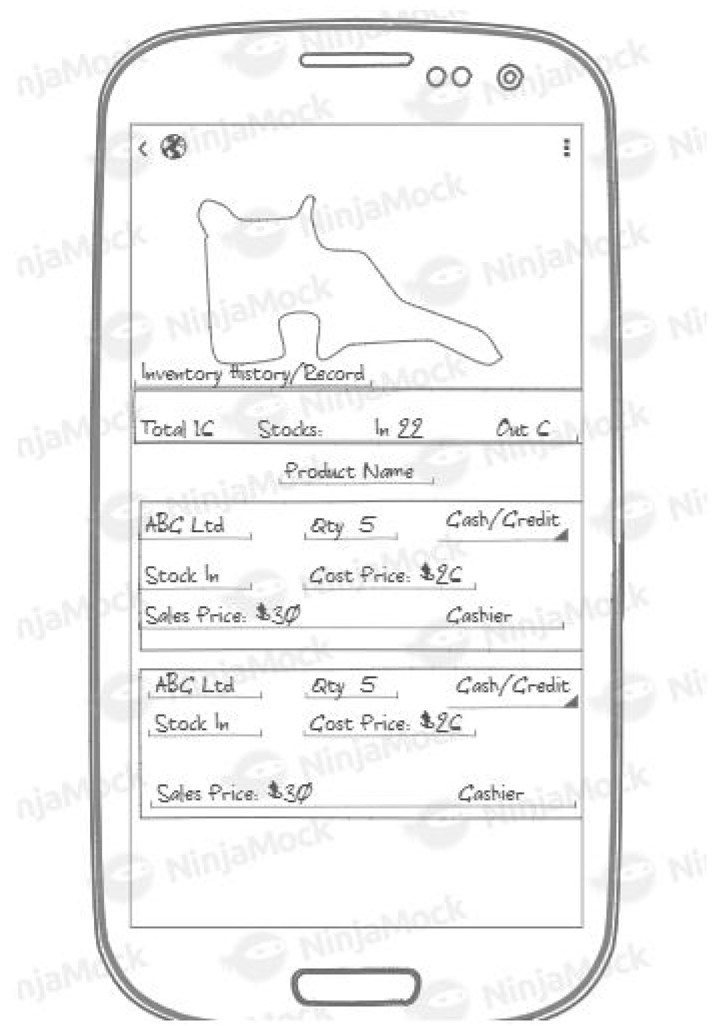
This is the combination of Screen 2B and Screen 3A on a landscape mode.
When users click on an item in the list on the left, its details shows up in the fragment on the right.

## Screen 3C (NavigationDrawer)



Users can navigate to other activities or fragments using navigation drawer; this is accessible for Screen 2B (InventoryManagementActivity) and Screen 3B (InventoryManagementActivity with ProductTransactionFragement on landscape) above.

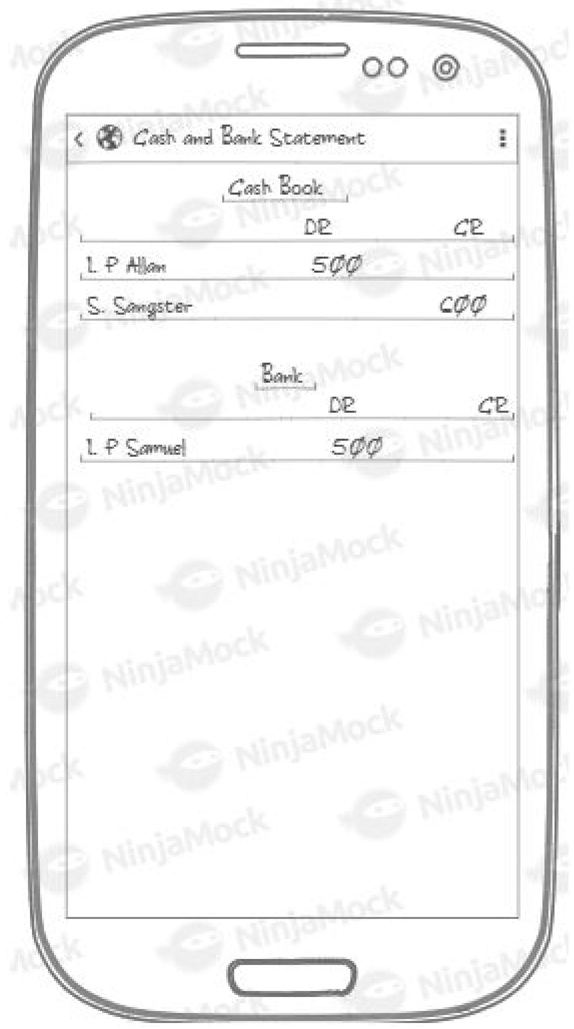
## Screen 4 (InventoryHistoryActivity)



From the navigation drawer, if users click on “Inventory History/Record”, InventoryHistoryActivity will be launched.
This activity shows;

1. the summary of inventory; i.e Total stock remaining, total incoming stock, total outgoing stock.
2. the history and record of inventory ---the outgoing and the incoming.

## Screen 5 (CashAndBankActivity)



Cash and Bank Statement

Cash Book

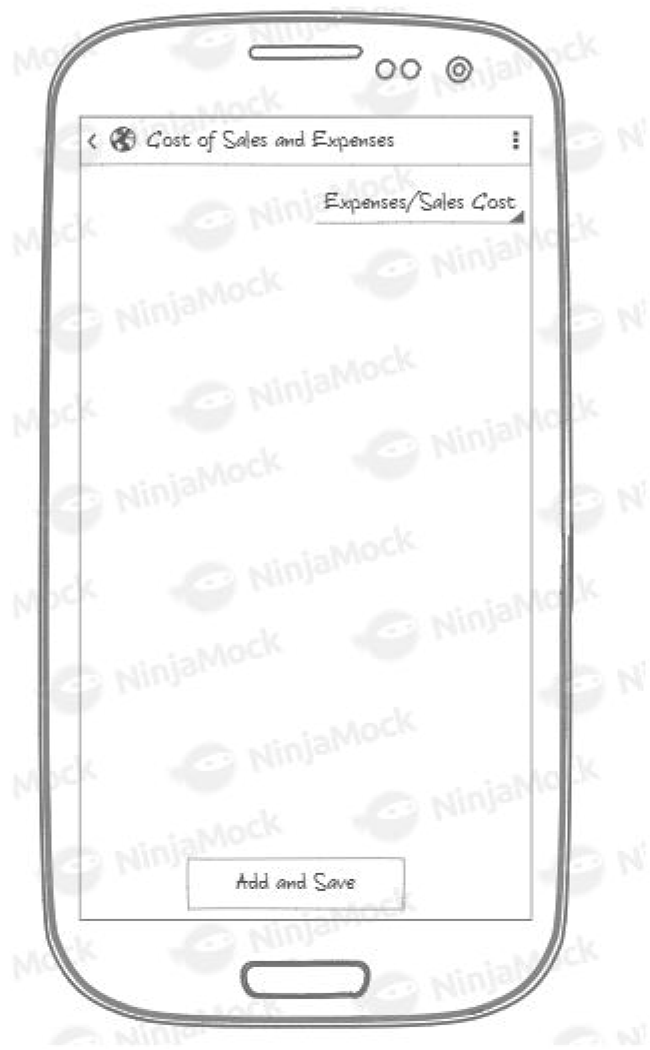
| | DR | CR |
|---|---|---|
| L P Allan | 500 | |
| S. Sangster | | 600 |

Bank

| | DR | CR |
|---|---|---|
| L P Samuel | 500 | |

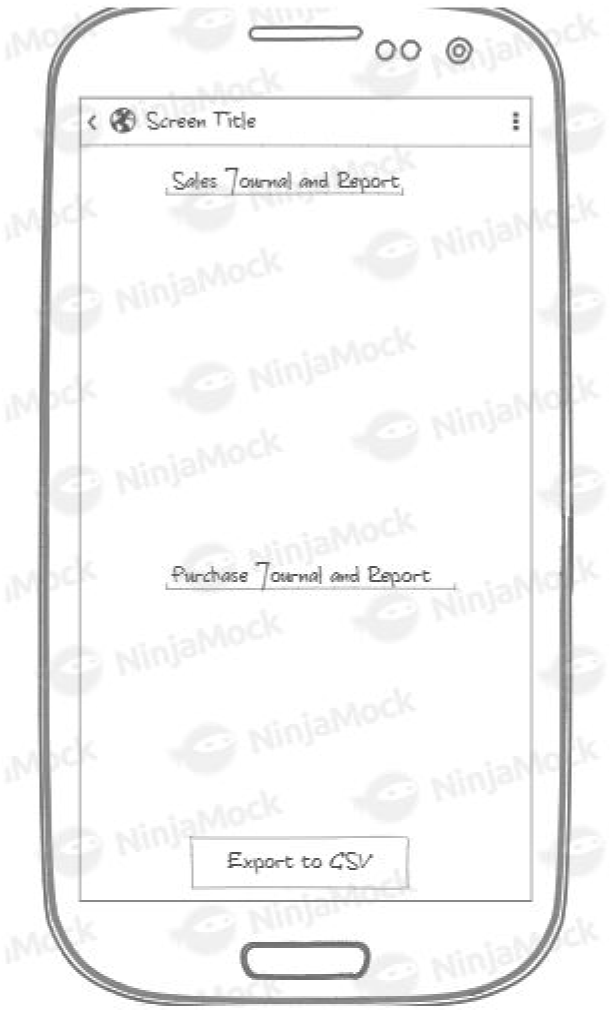
From the navigation drawer, if users click on “Cash Book and Bank Statement”, CashAndBankActivity  will be launched.
This activity shows the statement of bank account and cash book.

## Screen 6 (SalesCostAndExpensesActivity)



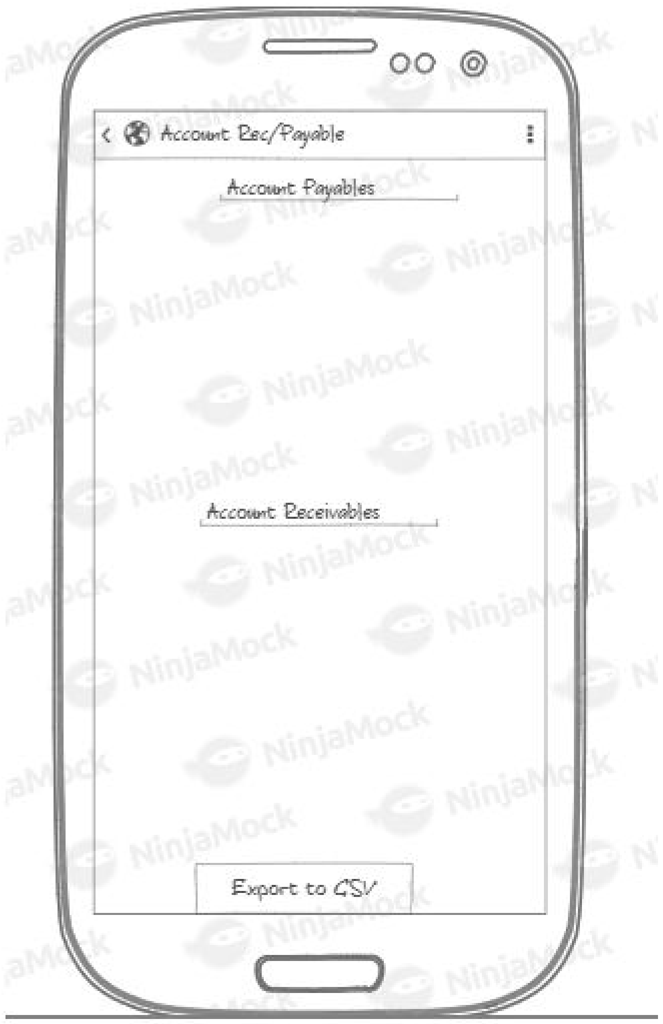
From the navigation drawer, if users click on “Cost of sales and Expenses”, SalesCostAndExpensesActivity will be launched.
Users can add expense, cost of sales and other costs, this will aid the generation of profit and loss account.

## Screen 7 (SalesAndPurchaseHistoryActivity)



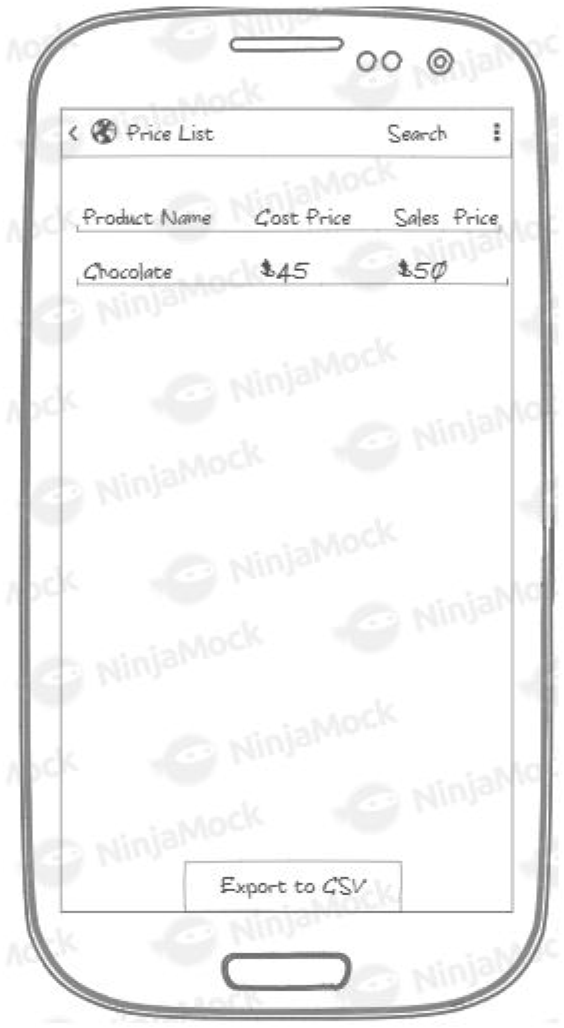
From the navigation drawer, if users click on “Sales and Purchase Journal”, SalesAndPurchaseHistoryActivity will be launched.
Users can view all purchases and sales made.

## Screen 8 (PayablesAndReceivablesActivity)



From the navigation drawer, when users clicks on either “Account Payables” or “Account Receivables”, PayablesAndReceivablesActivity will be launched, users can view all the account receivables and payables.
Users can export this data into a CSV file which will be saved on the users device storage.

## Screen 9 (PriceListActivity)

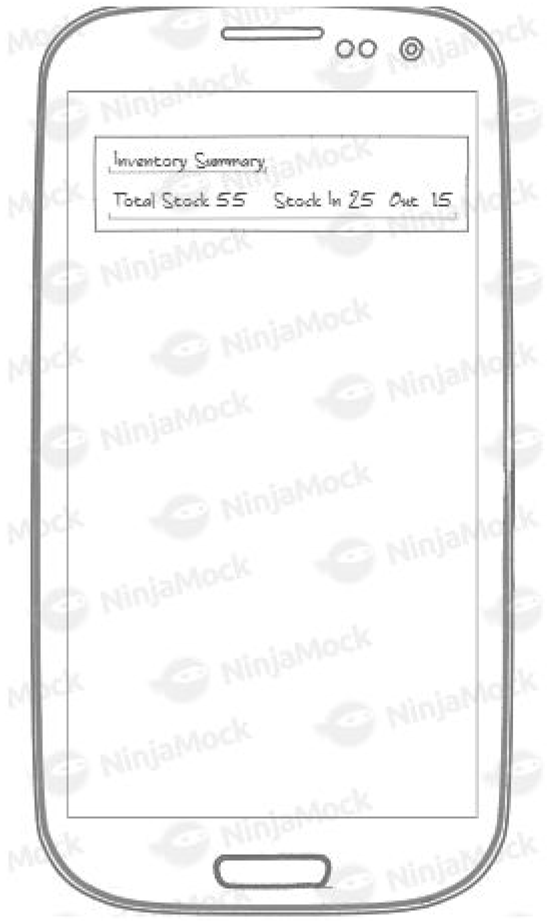


This activity shows products' price lists. Users can also search a particular product or transaction by product name or code.

## Screen 10 (Firebase Auth UI)

Firebase Auth UI will be used and generated by lines of code. Users will be able to access the app by signing in with either a Gmail account or email and password.

## Screen 11 (App Widget)



The app widget shows the summary of inventory: Total stock remaining, total stock purchased or brought in and total stock taken out or sold.
Users can tap on the widget to launch InventoryManagementActivity.
IntentService will be used to update the widget.

## Key Considerations

### How will your app handle data persistence?

With regard to data persistence, the app will use
1. Room Database to store stock information offline
2. SharedPreferences and SavedInstanceState will be used to persist data where the use of database may not be necessary or cumbersome; for example, to persist the state of an app which will not exceed the time span of an activity lifecycle like configuration change.

### Describe any edge or corner cases in the UX.

Sign-In and Sign-Out activities are not built but generated by the lines of code from Firebase Authentication UI; hence, the screen is not included above.

Top navigation arrow in the toolbar will be implemented in all activities and fragments; this will allow users to navigate back to previous activities or fragments.

NavigationDrawer is accessible from InventoryManagementActivity by swiping on the screen from left to right.

The widget shows the summary of stock: total stock remaining, stock brought in and taken out. Users can click on the widget to launch InventoryManagementActivity. IntentService will be used to handle the communication between the widget and the app.

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

The app will incorporate the following libraries, and not limited to;
- OpenCSV (version: 4.3.2) for exporting stock data into csv file.
- Picasso (version:2.5.2)for loading and caching images.
- Room Database for storing and backing up data offline

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

- Firebase Authentication as a Google Play API will be used to control users access to the app. Users will be able to sign in either with Google mail or email and password and sign out.
- Google AdMob as a Google Play API to incorporate interstitial and/or banner ads in the app

## 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

This project will be set up using Android Studio 3.0 with Gradle Build 3.0.0 and the app will be named InventoryManagement.

- This app will be written solely in Java Programming Language.
- Android Studio 3.0.0 will be used to build the app.
- Gradle version 3.0.0 will be used to build the app.
- While setting up the project, the app will target the minimum api level 15, the compile and target SDK version will be api level 26.
- After the project set-up, the app module build.gradle file will be altered so as to accommodate all the necessary dependencies and library dependencies in order to develop the app; such dependencies, apart from android supports and widgets libraries, are Picasso (version: 2.5.2), google play service, google admob, Firebase among many others.
- The app will use Android Support Library: appcompat-v7:26.1.0.
- Google Play services version:11.0.4 will be employed for the app to feature google play services like Google Admob e.t.c.
- Google AdMob version:11.0.4 will be used in the app.
- Firebase Authentication version:11.0.4 will be used in the app
- OpenCSV  version:4.3.2 as a library to generate csv files will be put in the gradle dependency block for use
- The gradle file will be synced.
- The app will feature stable version of gradle, gradle dependencies and Android Studio.

● Later versions of libraries may be used if one or more is found to favour resource management and the app effectiveness.

## Task 2: Implement UI for Each Activity and Fragment

● Build UI for MainActivity as shown in Screen 1 (Home Page) above: the home page will have three image buttons (Inventory Management, Cash Book and Bank Statement and Cost of Sales and Expenses). These three image buttons are the entry point into the app.

● Build UI for InventoryManagementActivity as shown in Screen 2a and 2b above. From the home screen, if the first image button (Inventory Management) is clicked, this screen (InventoryManagementActivity) is opened, the activity displays a dialog bar which requests for product information with a button to save the information.

● Build UI for ProductTransactionFragment . When users click on an item in the list on screen 2a (InventoryManagementActivity) above, ProductTransactionFragment is launched, this fragment will display all the information relating to the item selected in the list for users to perform a sales (outgoing stock) or purchase (incoming stock) transaction.

● Build a UI on landscape mode to accommodate InventoryManagementActivity on the left half of the screen and ProductTransactionFragment on the right half of the screen as shown in Screen 3b above.

● Make a NavigationDrawer as shown in Screen 3c above; users can navigate to other activities or fragments using navigation drawer; this is accessible for Screen 2B (InventoryManagementActivity) and Screen 3B (InventoryManagementActivity with ProductTransactionFragement on landscape) above.

● Build UI for InventoryHistoryActivity; From the navigation drawer, if users click on "Inventory History/Record", InventoryHistoryActivity will be launched.
This activity shows the summary of inventory; i.e. Total stock remaining, total incoming stock and total outgoing stock, the history and record of inventory --- the outgoing and incoming.

● Build UI for CashAndBankActivity as shown in Screen 5 above. From the navigation drawer, if users click on "Cash Book and Bank Statement", CashAndBankActivity will be launched.
This activity shows the statement of bank account and cash book.

- Build UI for SalesCostAndExpensesActivity as shown in Screen 6 above.  From the navigation drawer, if users click on “Cost of sales and Expenses”, SalesCostAndExpensesActivity will be launched.
Users can add expense, cost of sales and other costs, this will aid the generation of profit and loss account.

- Build UI for SalesAndPurchaseHistoryActivity as shown in Screen 7 above. From the navigation drawer, if users click on “Sales and Purchase Journal”, SalesAndPurchaseHistoryActivity will be launched.
Users can view all purchases and sales made.

- Build UI for PayablesAndReceivablesActivity as shown in Screen 8 above. From the navigation drawer, when users clicks on either “Account Payables” or “Account Receivables”, PayablesAndReceivablesActivity will be launched, users can view all the account receivables and payables.
Users can export this data into a CSV file which will be saved on the users device storage.

- Build UI for PriceListActivity as shown in Screen 9 above. This activity shows products’ price lists. Users can also search a particular product or transaction by product name or code.
- Firebase Authentication UI is generated with lines of code and not need be designed and built since it is generated by code.
- Implement interstitial adds on some activities transitions i.e MainActivity to InventoryManagementActivity etc. Banner ads may also be incorporated.
- Build the widget for the app.

## Task 3: Create Layout Files for All Activities and Fragments

- Create layout for MainActivity
- Create layout for InventoryManagementActivity
- Create layout for ProductTransactionFragment
- Create layout for landscape that will house both InventoryManagementActivity and ProductTransactionFragment.
- Create layout for NavigationDrawer.
- Create layout for InventoryHistoryActivity
- Create layout for CashAndBankActivity.
- Create layout for SalesCostAndExpensesActivity
- Create layout for SalesAndPurchaseHistoryActivity
- Create layout for PayablesAndReceivablesActivity
- Create layout for PriceListActivity.
- Create layout for the app widget.

## Task 4: Define App Theme and Resources(colours, strings, dimensions, and style)

- Extract all colours into colour resource file in the value folder.
- Extract all hard-coded strings into strings resource file in the value folder
- Extract all hard-coded dimemsions into dimens resource file in the value folder
- Define the app style in the resource file for styles including the app theme
- The app theme will use App Compat theme

## Task 5: Apply Material Design

- On the homepage (MainActivity) implement fade effect such that the UI will fade upon image button click before launching the InventoryManagementActivity.
- Surfaces effect and elevations will be applied to the list items in the InventoryManagementActivity such that when a list item is clicked, a surface effect will be shown.
- Transition from InventoryManagementActivity to ProductTransactionFragment will incorporate parallax effect on the image views.
- Elements in ProductTransactionFragment will have animation (slide from right to left) upon launch
- All other activities will have slide effect.

## Task 6: App Testing

- The app will be tested on at least a phone, a tablet and emulators of different configurations in order to ensure all functionalities work as expected.

## Task 7: How to Handle Backend Communication

- The app will use AsyncTask and/or IntentService to handle backend communications; i.e communication between the app and its widget will be handled using IntentServices.

## Task 8: App Signing and Key Generation

- The app will be signed and release key generated

Add as many tasks as you need to complete your app.

---

**Submission Instructions**

● After you’ve completed all the sections, download this document as a PDF [ File → Download as PDF ]

○ Make sure the PDF is named “**Capstone_Stage1.pdf**”

● Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

● Create a new GitHub repo for the capstone. Name it “**Capstone Project**”

● Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”