mInventory

GitHub Username: rashidaattar

mInventory (Tentative name)

Description

For those who do not want to invest in an ERP system, mInventory is just what they need!
Always wanted to get rid of all that pile of sheets having all information of your inventory?
Or ever missed the delivery date of your orders ? Or simply just wanted an app to manage your inventory? mInventory is just what is needed. It's free and is completely offline.

Intended User
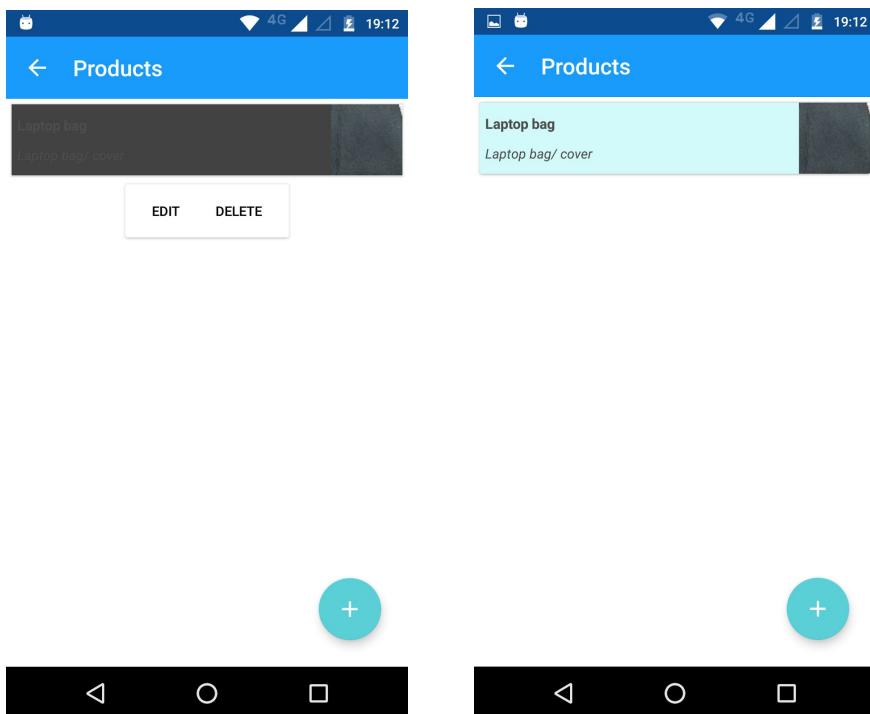
Business owners, traders, shopekeepers.

Features

- Stores Product information with images
- Stores customer and vendor information
- Keeps track of all the orders and notifies when the orders are due.
- Notifies when the stock of a particular product is low.
- Free of cost
- Saves data offline.
- Contains a Widget with a list of pending orders.

User Interface Mocks
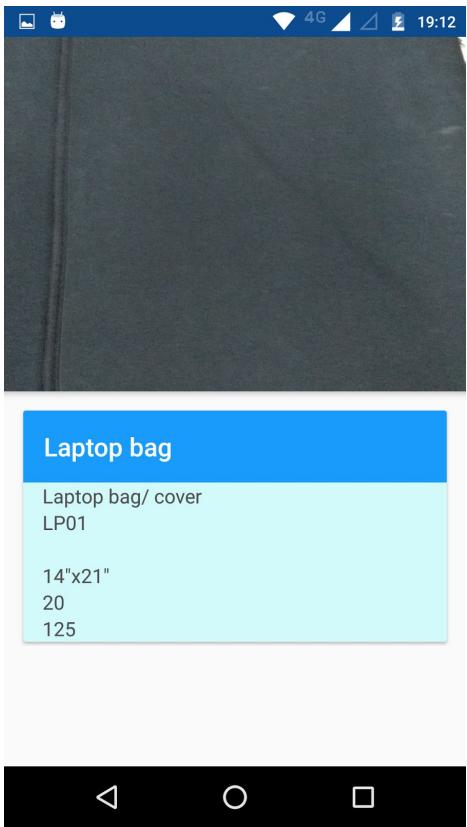
Screen 1

Products list view : enables Editing/Deleting from list view.
Similar views for Customer, vendor and order list.
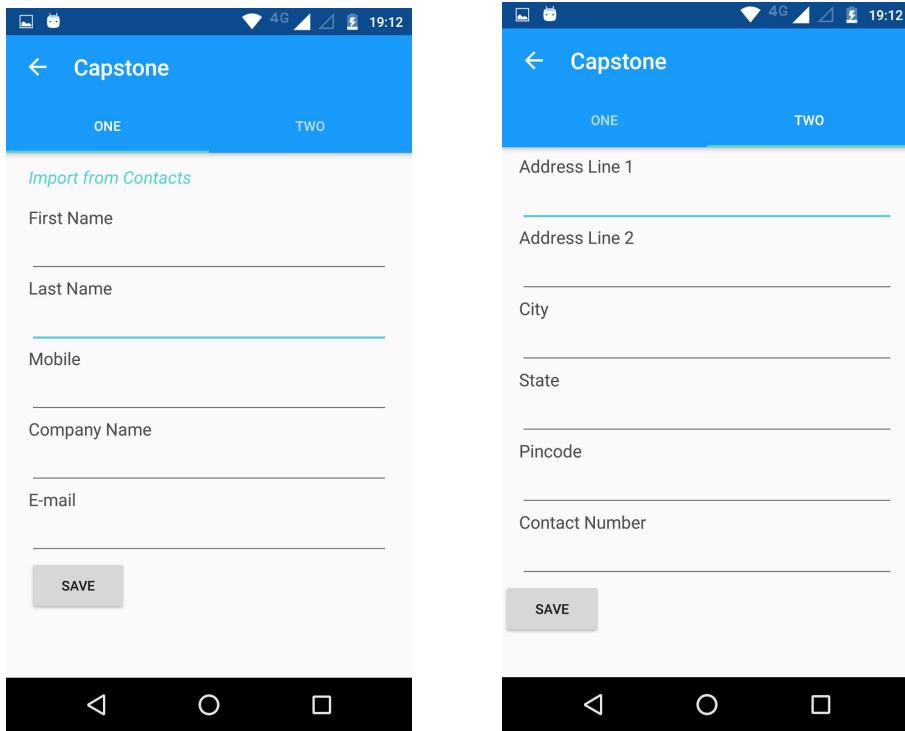
Screen 2
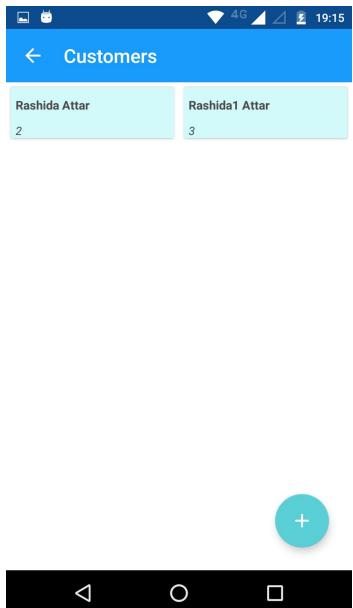
Product detail view : Uses Appbar layout to display the image saved for that product. Displays default dummy image if no image is saved for that product.

Screen 3
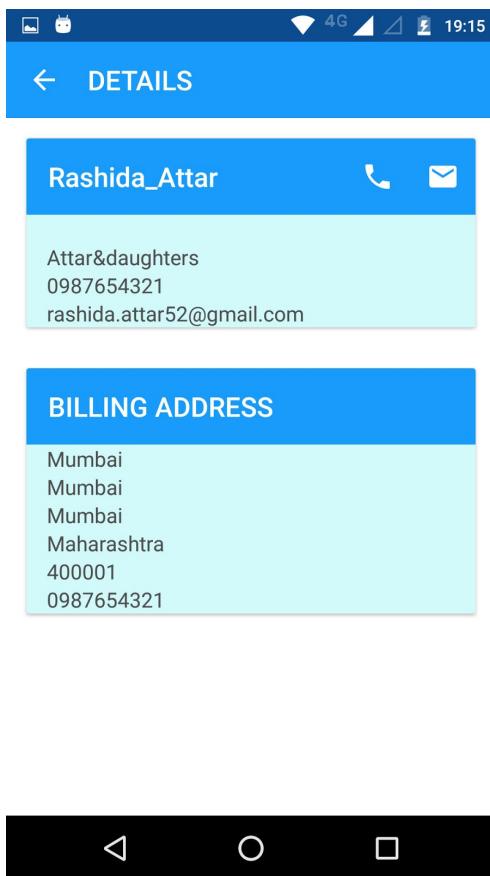
Form to add Customer and Vendor details: Includes contact information and address details.

Screen 4
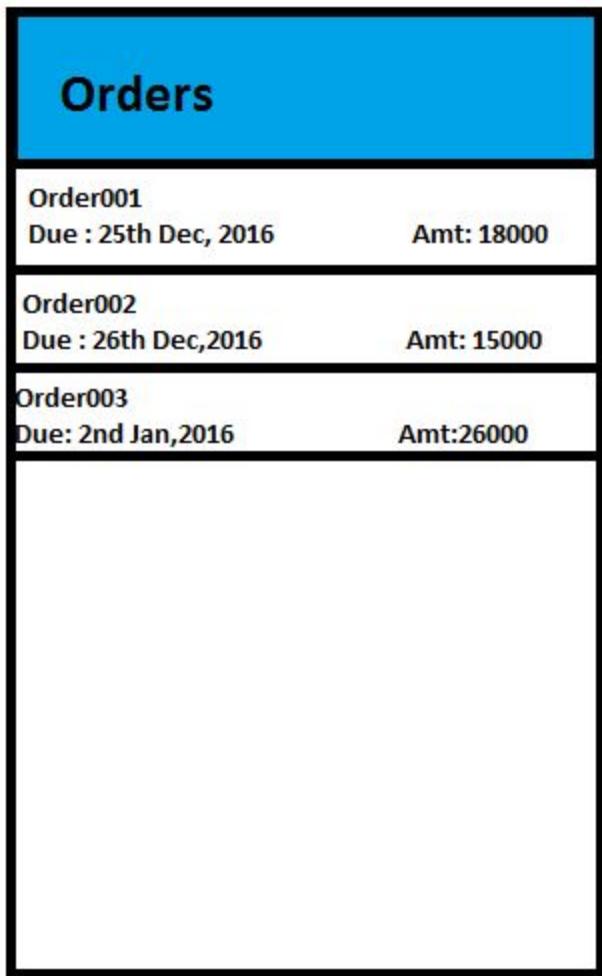
Customer and vendor list views.Functionality similar to product list view.

Screen 5

Customer and vendor detail view. Enable calling and email feature. Also, to integrate google maps to get directions to the address.

Screen 6

Orders

Order001 Due : 25th Dec, 2016	Amt: 18000
Order002 Due : 26th Dec,2016	Amt: 15000
Order003 Due: 2nd Jan,2016	Amt:26000

Probable UI for Widget. The widget displays pending orders in ascending order of the due date.

Key Considerations

How will your app handle data persistence?

App makes use of content providers, has an inventory database created to store all the inventory related data.
App also makes use of Shared Preferences to store persistent data.

Describe any corner cases in the UX.

App implements separate UI for tablets.

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

Butterknife : used to bind views.
Schematic : to implement content providers.
Material Design libraries : card view, tab layout, appbar layout.

Describe how you will implement Google Play Services.

1. Ad-Mob : to display interstitial and banner ads
2. Google Maps : to navigate to the required address.

Next Steps: Required Tasks

Task 1: Project Setup

Write out the steps you will take to setup and/or configure this project. See previous implementation guides for an example.

You may want to list the subtasks. For example:
- Configure libraries
- Something else

If it helps, imagine you are describing these tasks to a friend who wants to follow along and build this app with you.
1. Finalize the functionalities of the app.
   a. Save products
   b. Save customer details
   c. Save vendor details
   d. Save order details
2. Design the database schema.
3. Create an Android Project.
4. Develop the landing page.

Task 2: Implement UI for Each Activity and Fragment

1. Design UI for the forms :
   a. save/edit products
   b. save/edit customer and vendor information

    c. save/edit order details
  2. Design UI for the list view :
    a. UI for product details.
    b. UI for vendor details
    c. UI for customer details
    d. UI for order details
  3. Implement content providers using Schematic

Task 3: Integrate Google play services.
  1. Integrate Google AdMob.
  2. Integrate Google maps.

Task 4: Implement AsyncTask to search for products while adding an order.
Simultaneously check the stock level of that product.

Task 5: Develop UI.
Develop the UI for all the pages.

Task 6: Design Tab UI.

Task 7: Integrate Widget provider.