



[< Back to Android Basics Nanodegree by Google](#)

Inventory App, Stage 2

REVIEW

CODE REVIEW 5

HISTORY

Meets Specifications

Project tested on

Phone: Nexus 5x api 26

Tablet: Nexus 10 api 26

Nice work on the project! 😊

Layout

The app contains activities and/or fragments for the user to:

- Add Inventory
- See Product Details
- Edit Product Details
- See a list of all inventory from a Main Activity

Multiple actions listed above can be combined in a single activity.

The user navigates between the activities and/or fragments using one or more of the following navigation patterns - Navigation Drawer, View Pager, Up/Back Navigation, or Intents.

All of the correct functions are implemented.

In the Main Activity/Fragment, each list item displays the Product Name, Price, and Quantity.

Each list item also contains a Sale that reduces the total quantity of that particular product by one (include logic so that no negative quantities are displayed).

The Product Detail Layout displays the Product Name, Price, Quantity, Supplier Name, and Supplier Phone Number that's stored in the database.

The Product Detail Layout also contains buttons that increase and decrease the available quantity displayed.

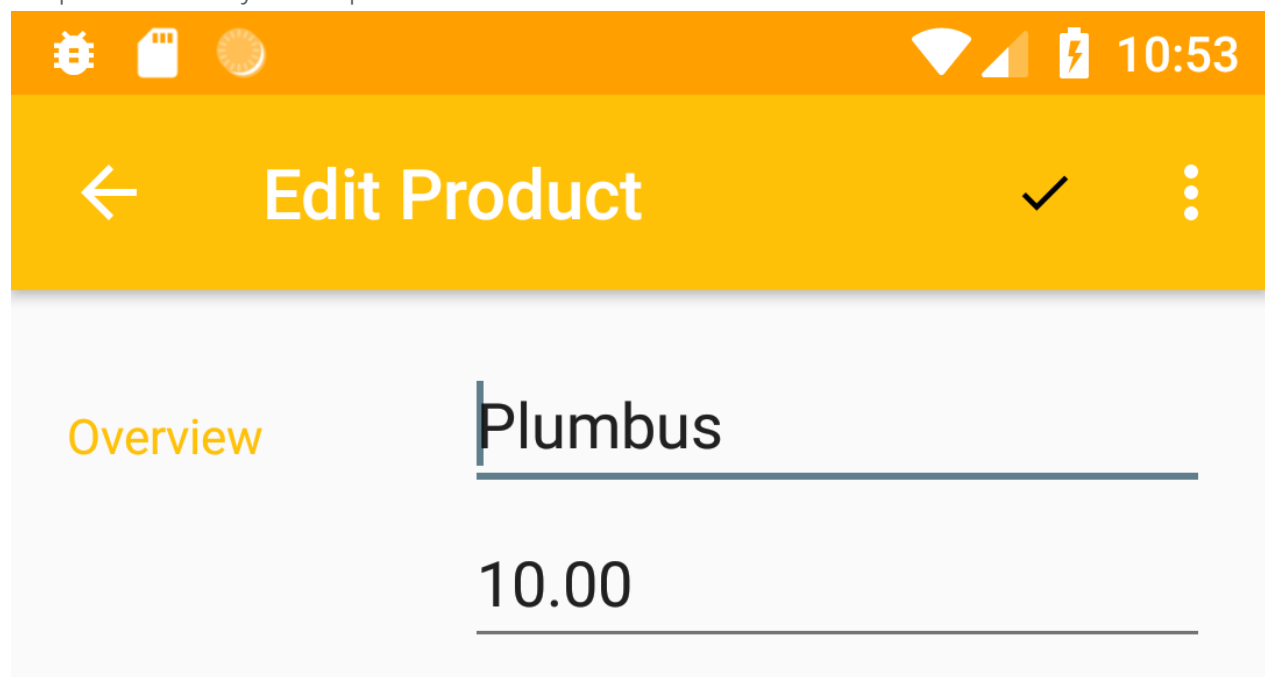
Add a check in the code to ensure that no negative quantities display (zero is the lowest amount).

The Product Detail Layout contains a button to delete the product record entirely.

The Product Detail Layout contains a button to order from the supplier. In other words, there exists a button to contains a button for the user to contact the supplier via an intent to a phone app using the Supplier Phone Number stored in the database.

Note: Including the Product Image is an optional feature. The functionality to pick a picture from the gallery is beyond the scope of this program, but students who are interested in learning this functionality may implement it.

The product detail layout is implemented.



Quantity

-

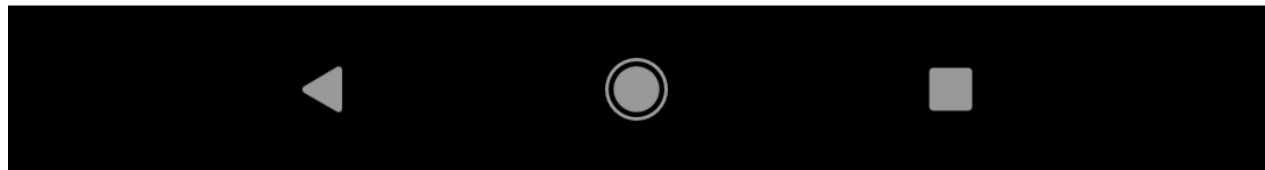
5

+

Supplier

Plumbus Inc.

1-800-888-5555



When there is no information to display in the database, the layout displays a TextView with instructions on how to populate the database (e.g. what should be entered in the field, which fields are required).

The code adheres to all of the following best practices:

- Text sizes are defined in sp
- Lengths are defined in dp
- Padding and margin is used appropriately, such that the views are not crammed up against each other.

Correct units are used. Padding/margin is used appropriately.

Functionality

The code runs without errors. For example, when user inputs product information (quantity, price, name), instead of erroring out, the app includes logic to validate that no null values are accepted. If a null value is inputted, add a Toast that prompts the user to input the correct information before they can continue.

I was not able to spot any errors.

The Main Activity displaying the list of current inventory contains a ListView that populates with the current products stored in the table.

The Main Activity contains an Add Product Button prompts the user for product information and supplier information which are then properly stored in the table.

Before the information is added to the table, it must be validated -

In particular, empty product information is not accepted. If user inputs invalid product information (name, price, quantity, supplier name, supplier phone number), instead of erroring out, the app includes logic to validate that no null values are accepted. If a null value is inputted, add a Toast that prompts the user to input the correct information before they can continue.

In the Edit Product Activity, user input is validated. In particular, empty product information is not accepted. If user inputs invalid product information (name, price, quantity, supplier name, supplier phone number), instead of erroring out, the app includes logic to validate that no null values are accepted. If a null value is inputted, add a Toast that prompts the user to input the correct information before they can continue.



Plumbus

\$10.00 Quantity: 0

SALE

Plumbus

\$10.00 Quantity: 0

SALE

Plumbus

\$10.00 Quantity: 0

SALE



The `EditActivity` is implemented and includes all of the correct information.

In the Main Activity that displays a list of all available inventory, each List Item contains a `Sale` Button which reduces the available quantity for that particular product by one (include logic so that no negative quantities are displayed).

When a user clicks on a List Item from the Main Activity, it opens up the detail screen for the correct product.

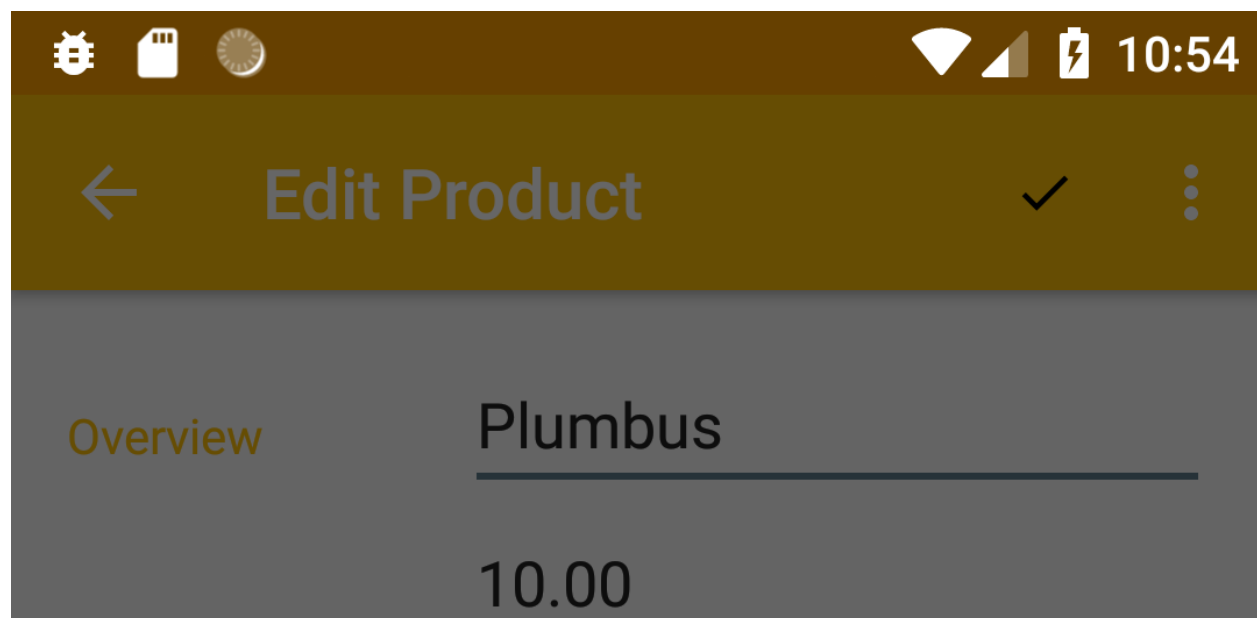
In the Detail View for each item, there are Buttons that correctly increase or decrease the quantity for the correct product.

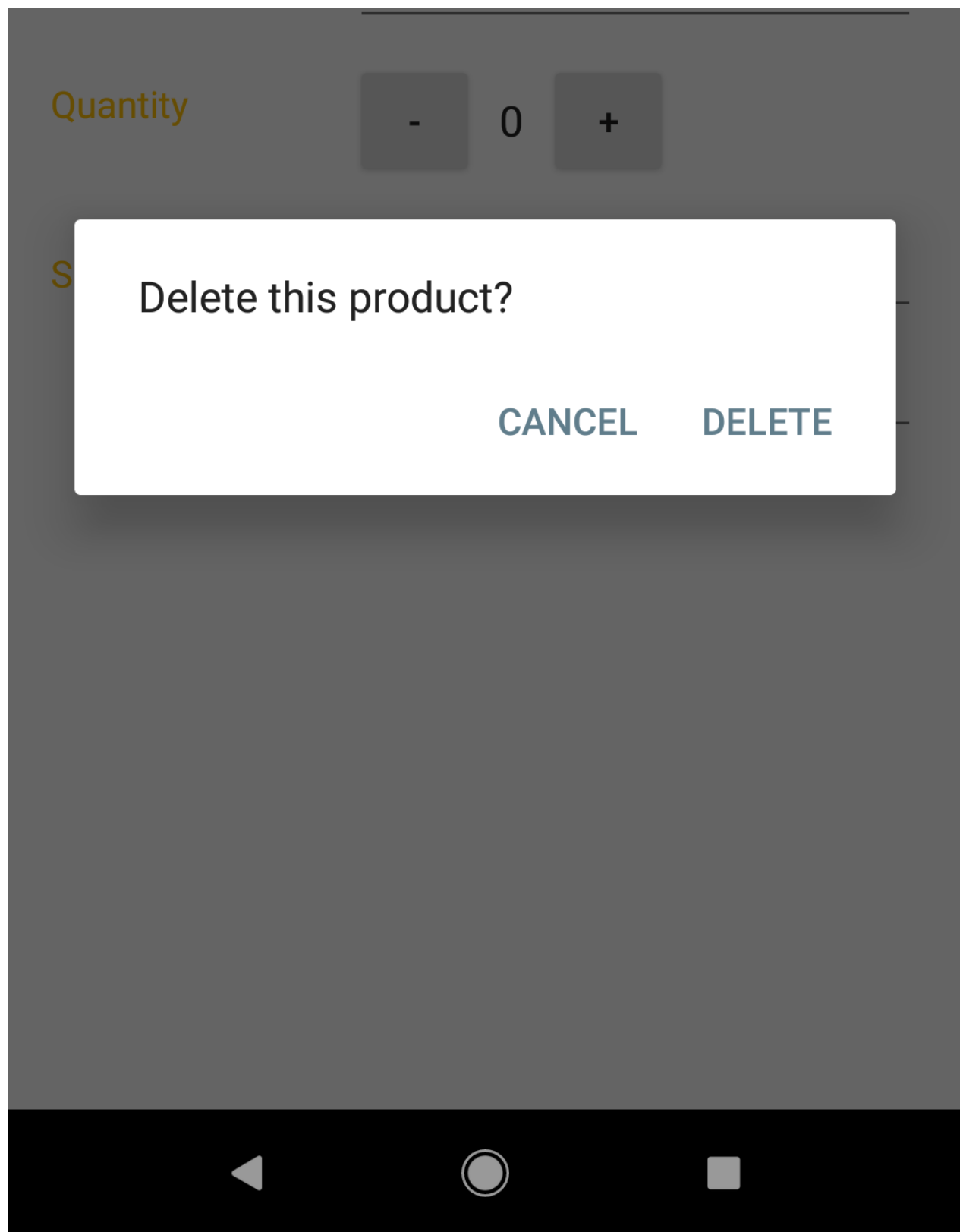
Add a check in the code to ensure that no negative quantities display (zero is the lowest amount).

The student may also add input for how much to increase or decrease the quantity by if not using the default of 1.

The Detail Layout contains a button for the user to contact the supplier via an intent to a phone app using the Supplier Phone Number stored in the database.

In the Detail Layout, there is a Delete Button that prompts the user for confirmation and, if confirmed, deletes the product record entirely and sends the user back to the main activity.





Confirmation is prompted for.

The intent of this project is to give you practice writing raw Java code using the necessary classes provided by the Android framework; therefore, the use of external libraries for core functionality will not be permitted to complete this project.

(i.e. Database and content provider libraries are not allowed for this project. Butterknife and similar libraries for ease of coding are allowed)

Code Readability

All variables, methods, and resource IDs are descriptively named such that another developer reading the code can easily understand their function.

All identifiers are descriptive.

The code is properly formatted i.e. there are no unnecessary blank lines; there are no unused variables or methods; there is no commented out code.

The code also has proper indentation when defining variables and methods.

Proper formatting is in place.

 [DOWNLOAD PROJECT](#)

5

[CODE REVIEW COMMENTS](#)



[RETURN TO PATH](#)