Assignment #2

Cash Register Application

Objectives:

- Use Android Studio to create Android app for cash register;
- Use Text View, Buttons and other UI components;
- Use constraint layout, relative layout and table layout;
- Create click listener and manage events;
- Create models to manage application data;
- Handel incorrect user's input.
- Multiple page navigation and passing data between activities.

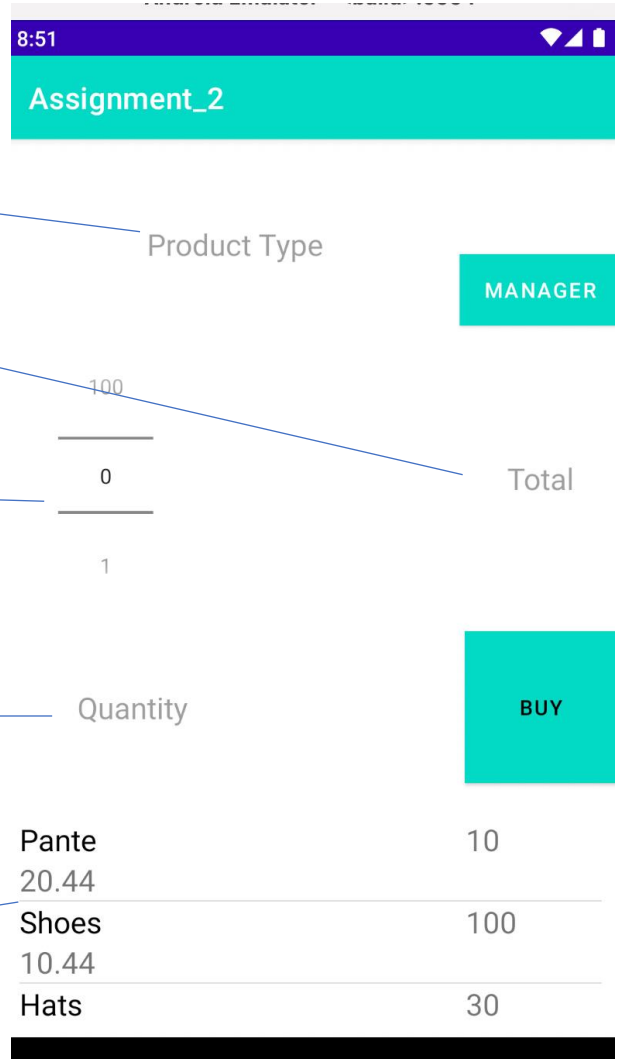
Cash Register Application (Part 1)

Part 1:
UI

- Constraint Layout
- Number Picker
- 1 List View
- 3 Text Views
- 1 Buy Button

Buy the amount that the user entered

Each Product has name, quantity and price attributes.



Selected product TextView

Total amount TextView

Quantity NumberPicker

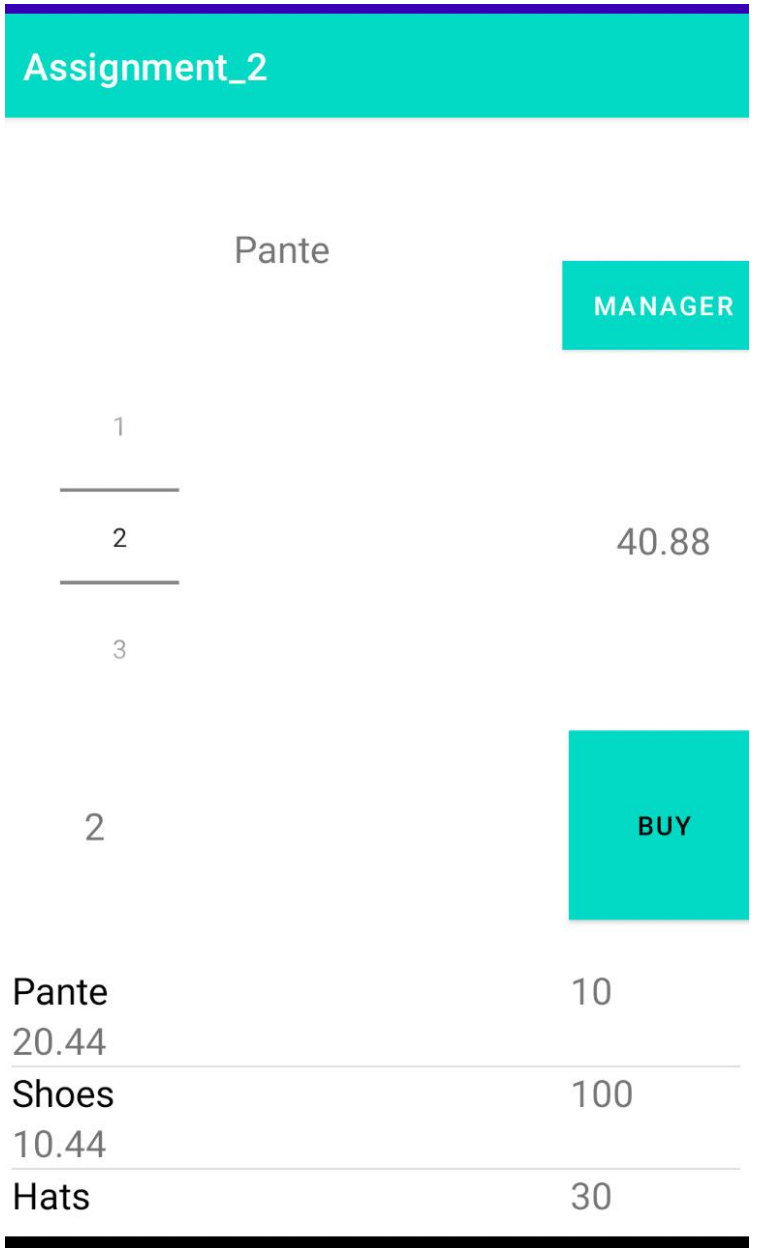
Selected Quantity TextView

Products Names, Quantity and Price List View

Cash Register Application (Part 1)

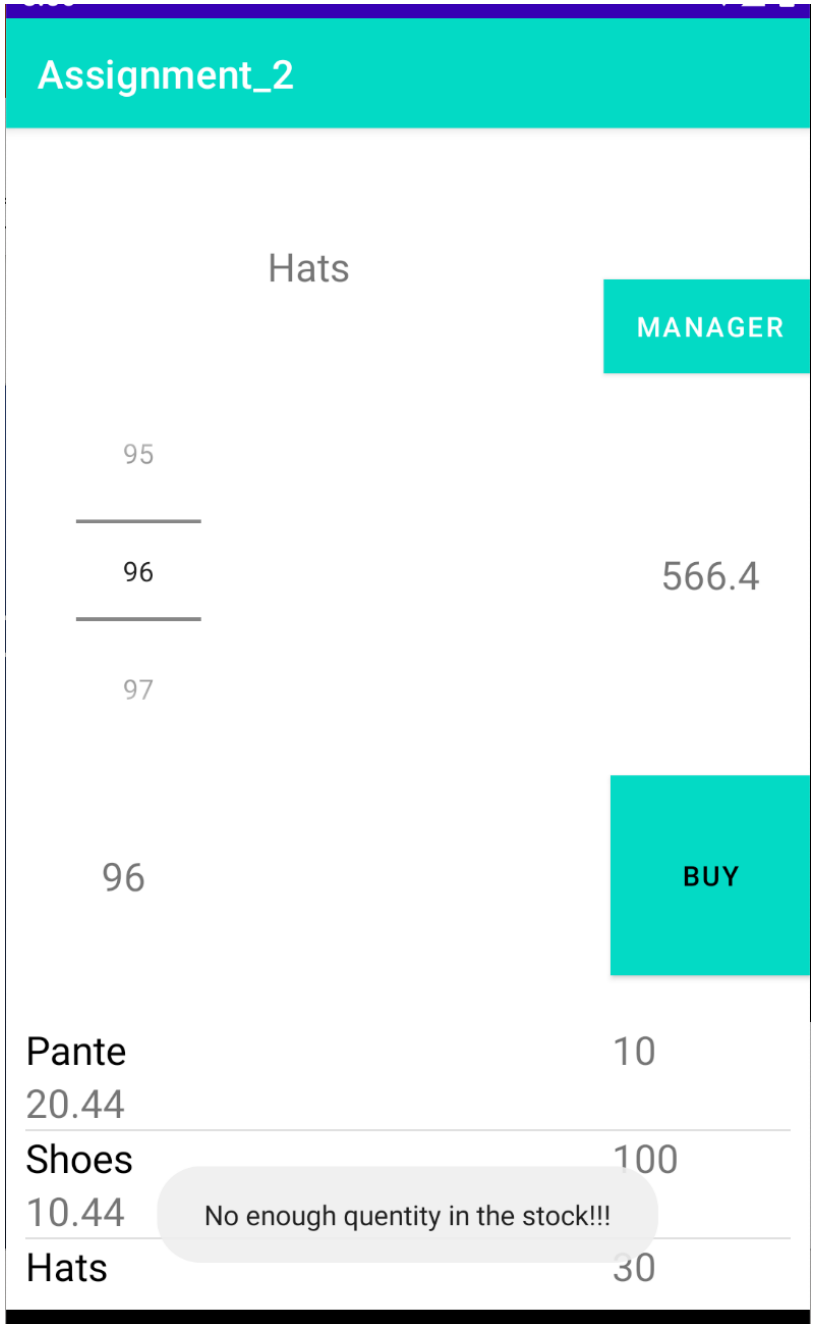
Interactive with UI:

- When the user selects one product from the list, the product textview must updated to have the product name;
- When the user selects the quantity from the number picker, the quantity textview must be updated to have the selected quantity;
- Total label should show the total price of that purchase
- Total = amount * item price



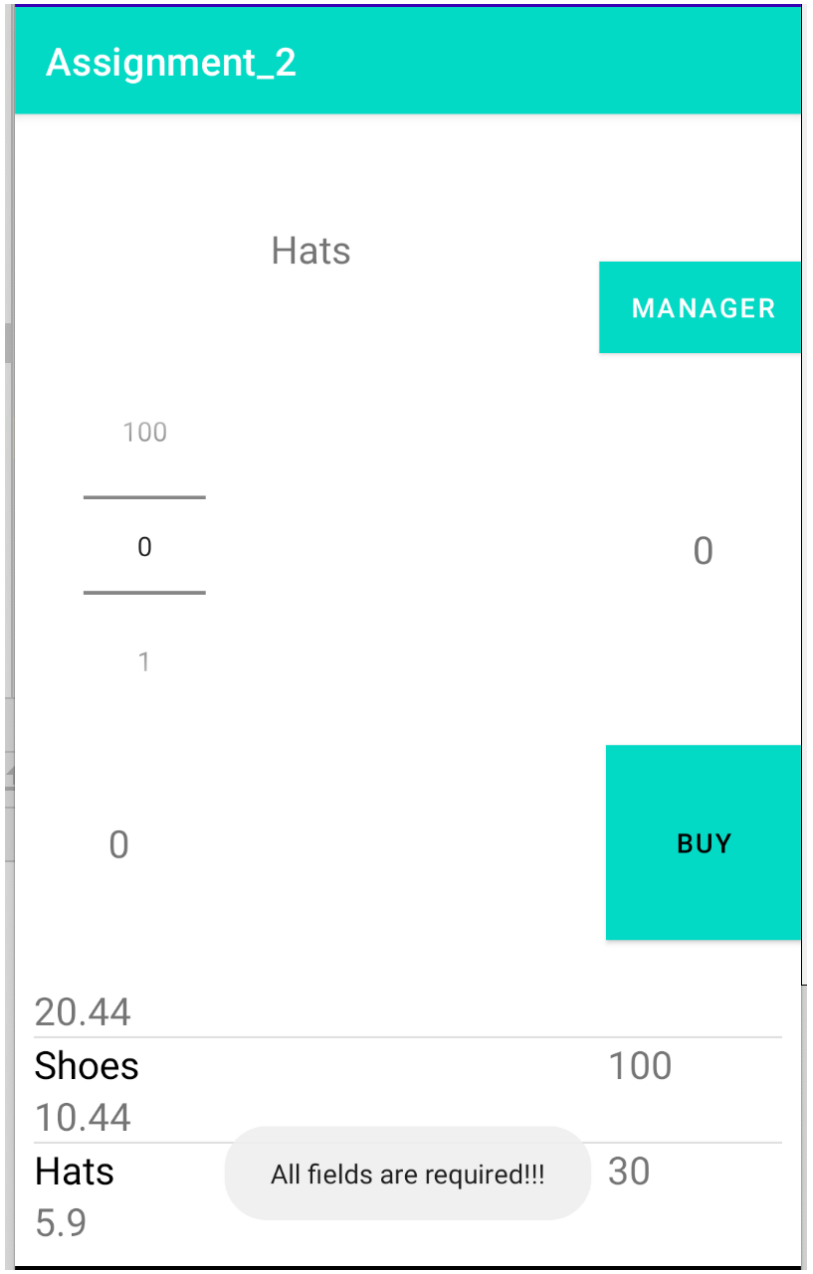
Cash Register Application (Part 1)

If the user selects a quantity which is more that the available quantity in the stock, a toast popped up with an error message.



Cash Register Application (Part 1)

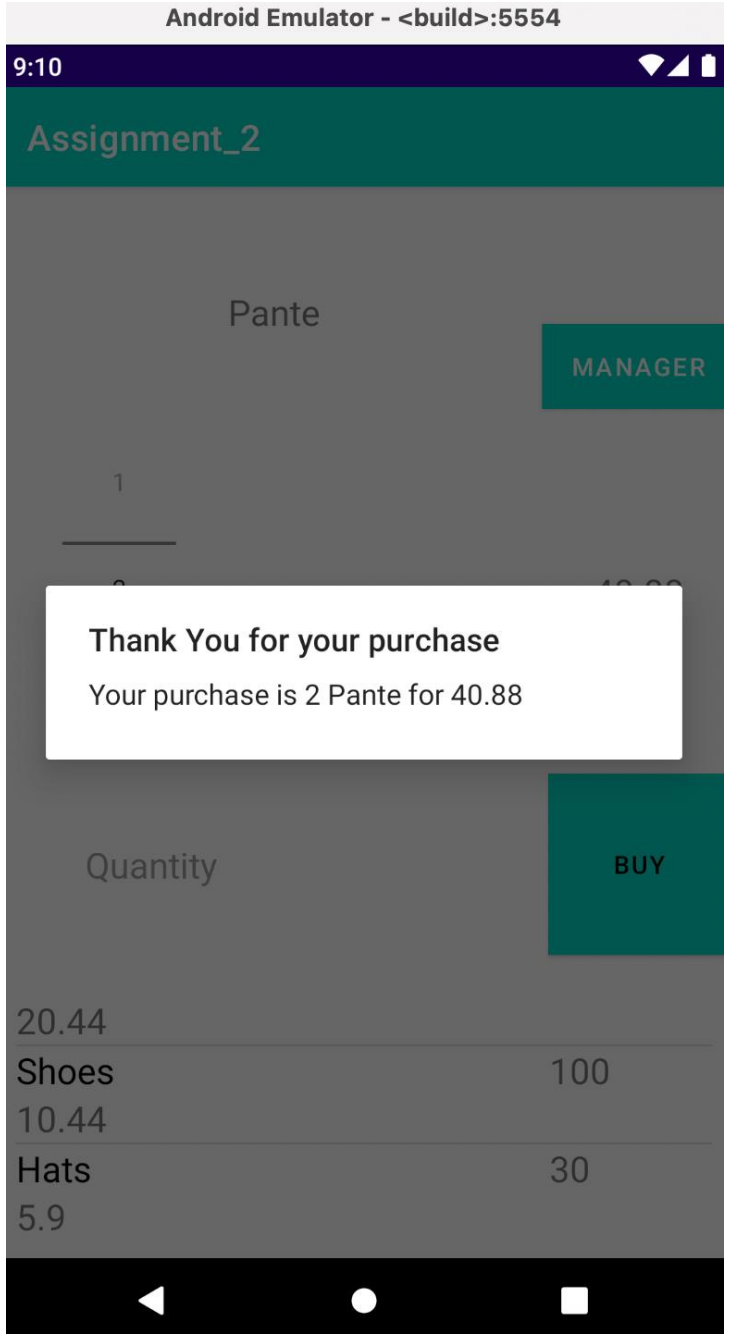
If the user clicks on Buy button without selecting a product or a quantity, an error message must appear.



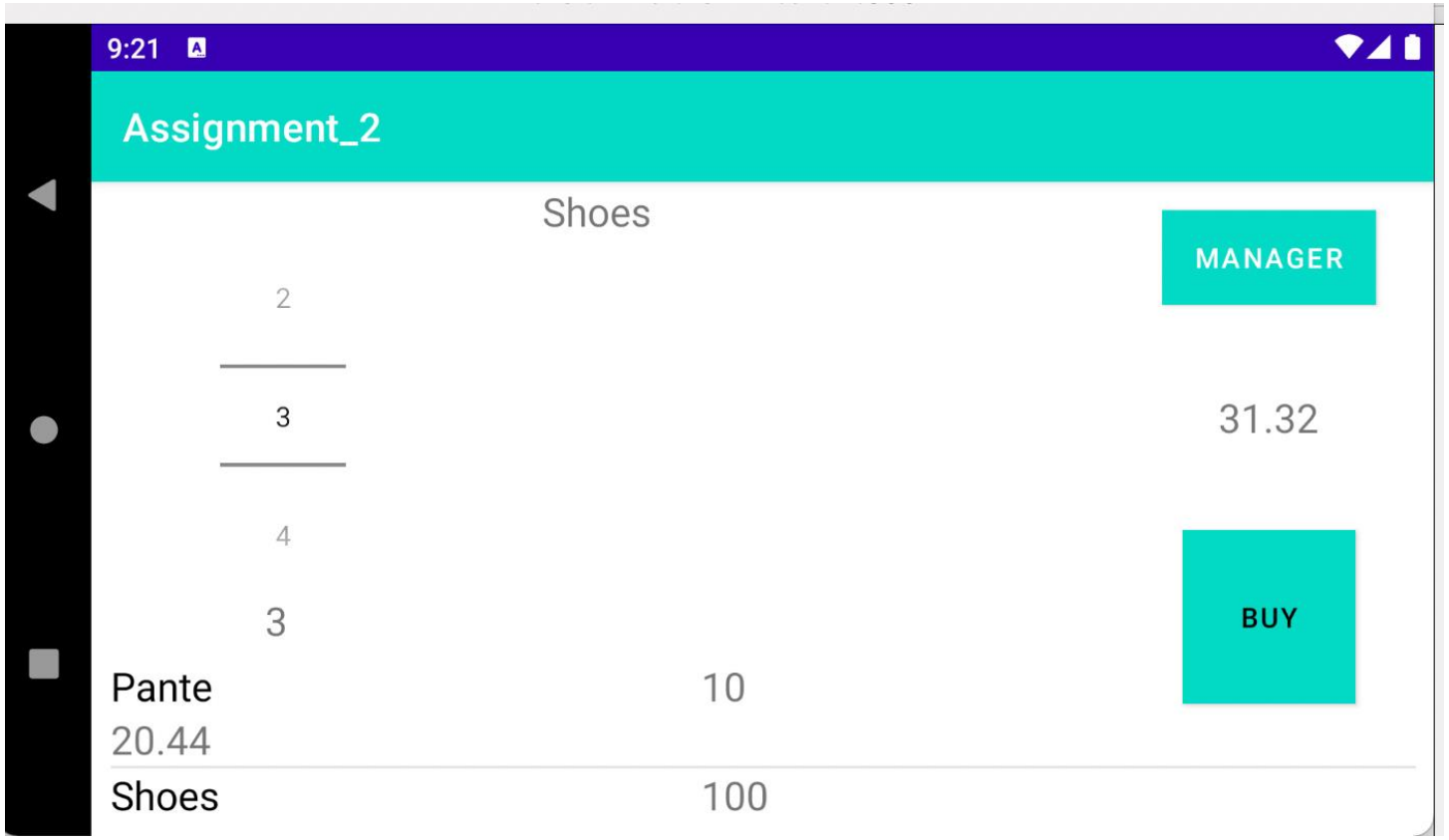
Cash Register Application (Part 1)

- When the user clicks on Buy Button the quantity of the purchased item should be updated and the user interface should be reset to accept a new purchase.

- New quantity = old quantity – purchased amount



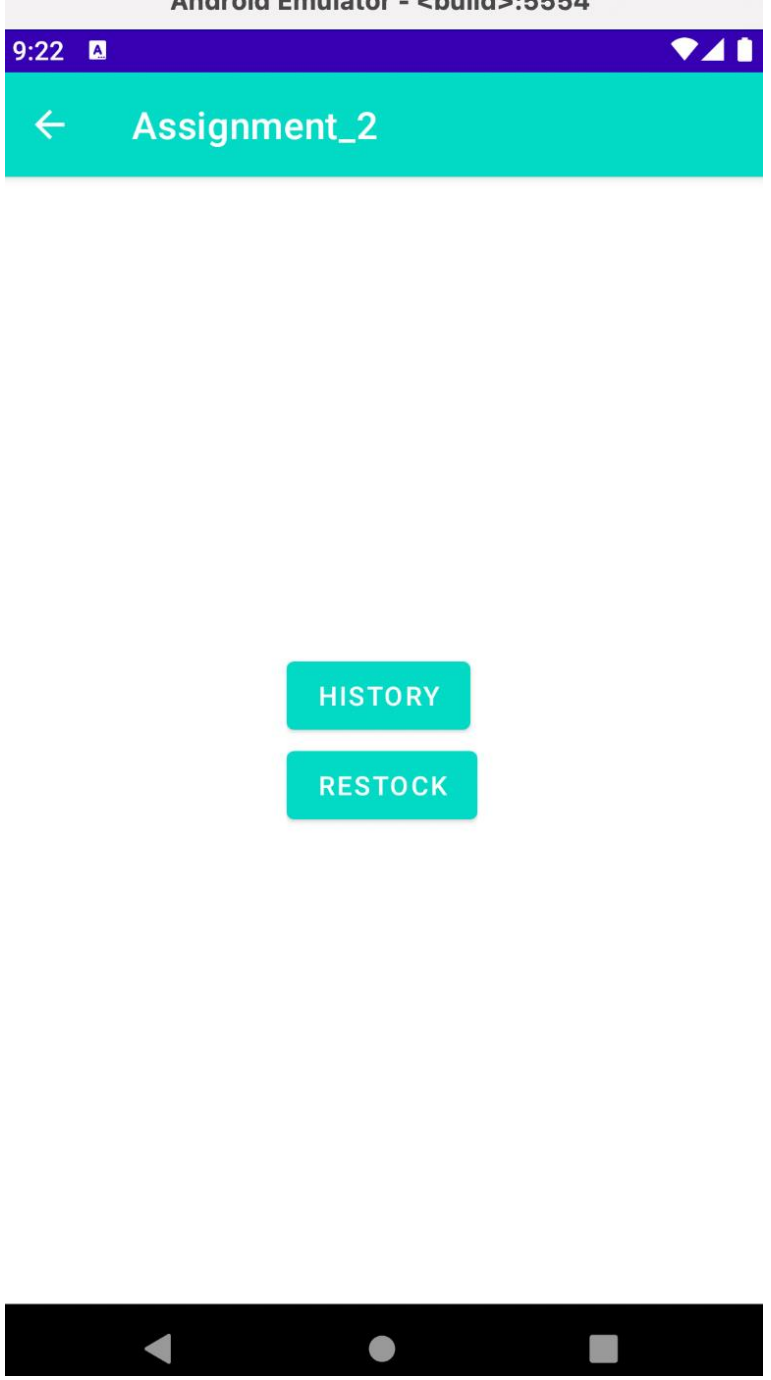
Cash Register Application (Part 1)



Support Landscape and Portrait orientation in your first activity, using Constraint Layout.

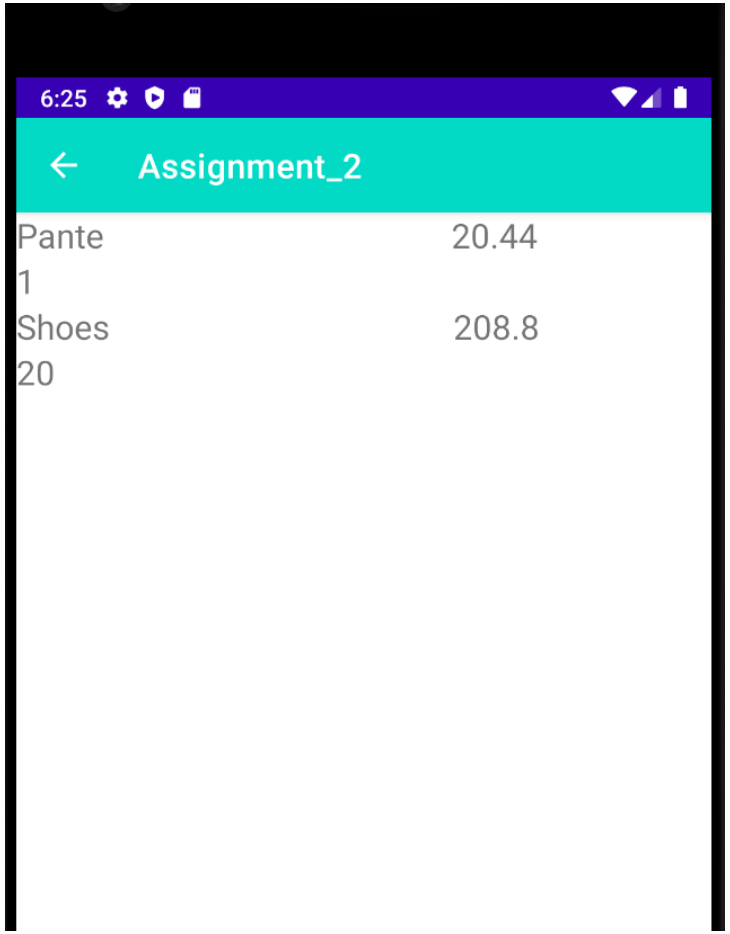
Cash Register Application (Part 2)

- Take Cash Register From Part 1
  Rearrange as you wish so all these Components would fit
  3. Add a manager Button
- When manager button is clicked The next page (Manager Panel) will appear.
- The Manager page has one History Button.

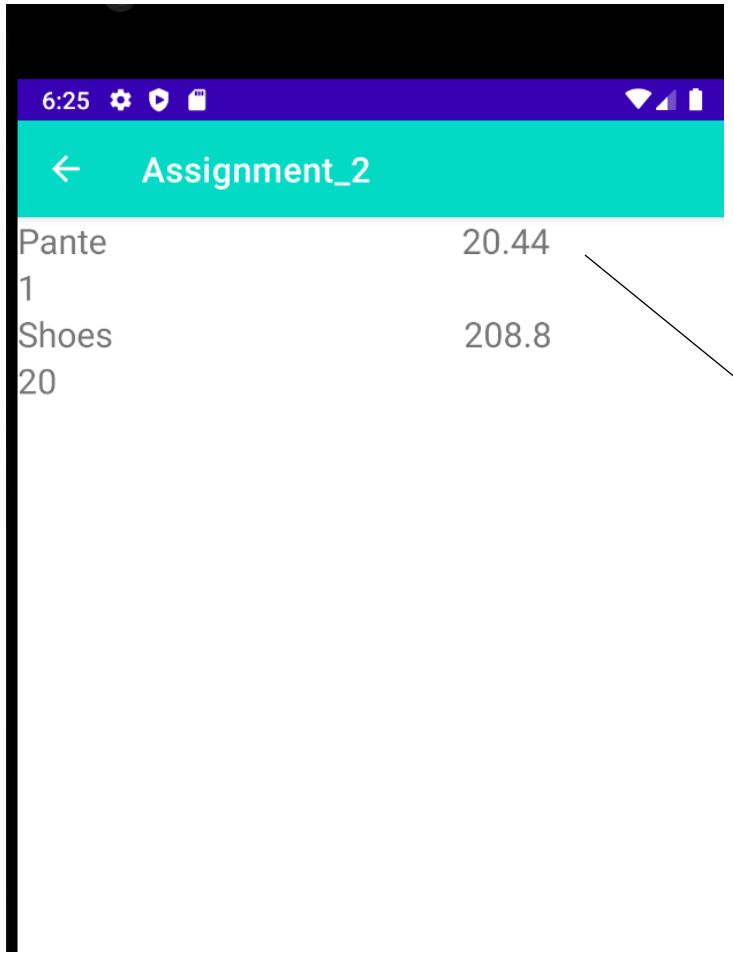


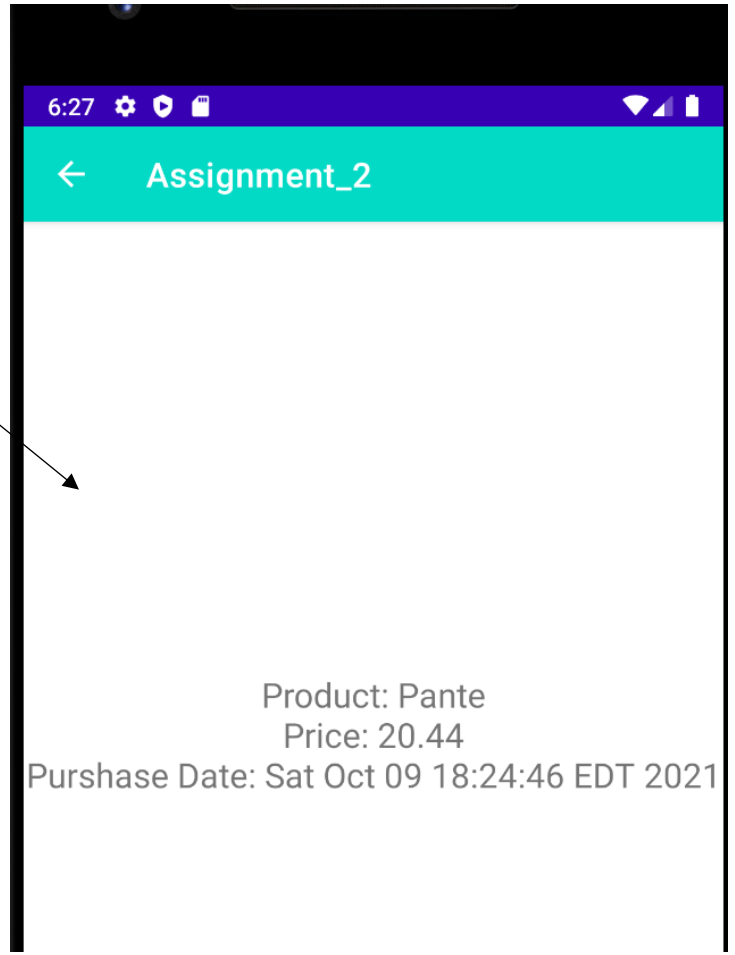
History List

- History Page list all purchased products, quantities and total price.
- When the user selects one item the app navigates to another page with more details about that purchase.
- Note: You will need to add the model class To be able to obtain some sort of history Object and show it here.
- Your history class (object) should have these properties(Name, Quantity, Total price, purchase date).
- History Table must use recycler view and adapter.



History Details





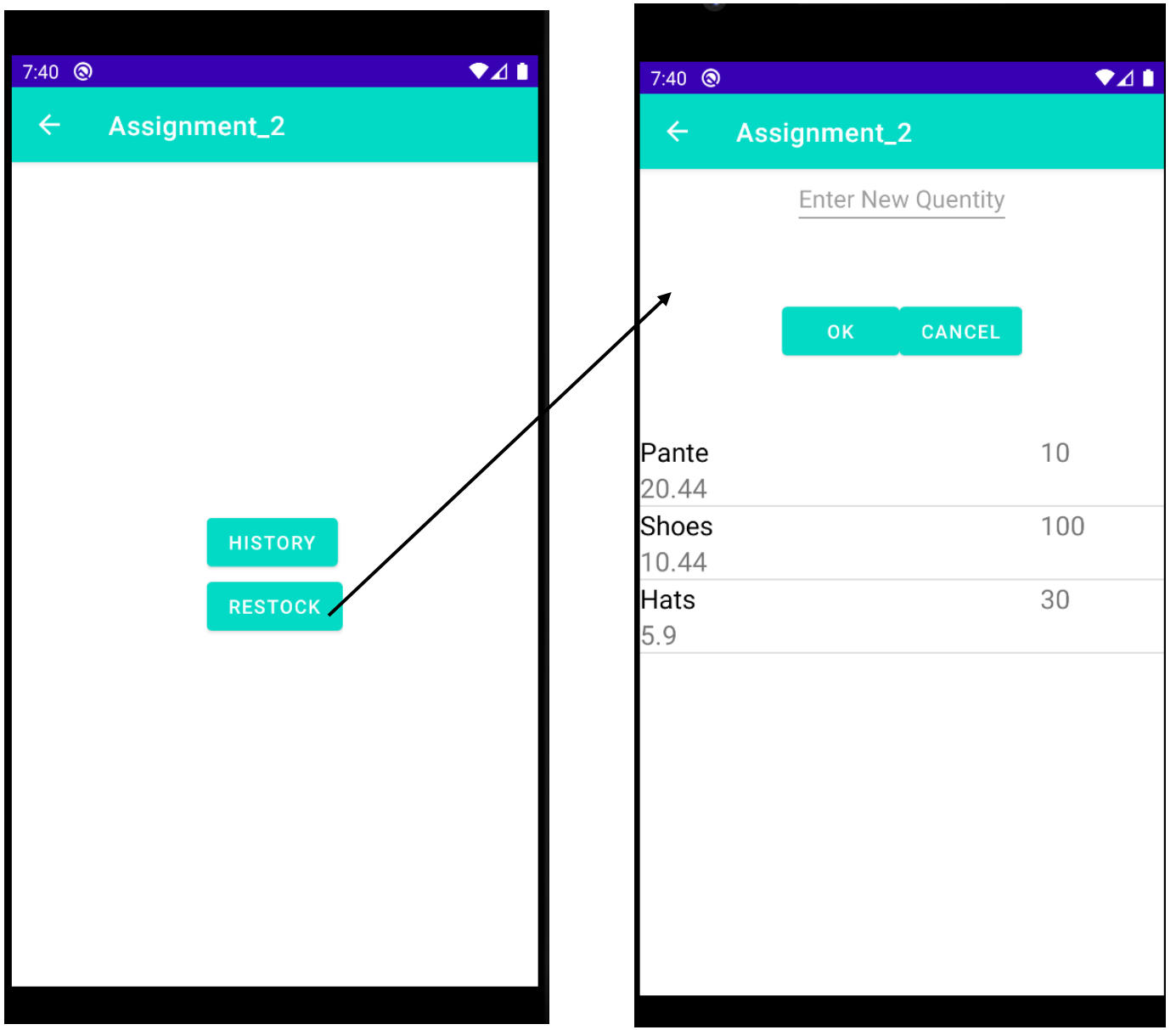
- When the user clicks on one purchases, the app will navigate to detail page, where the user could check the purchase date.
- The use could click on Back Button to come back to previous page.

Restock Activity

Restock Activity Layout:

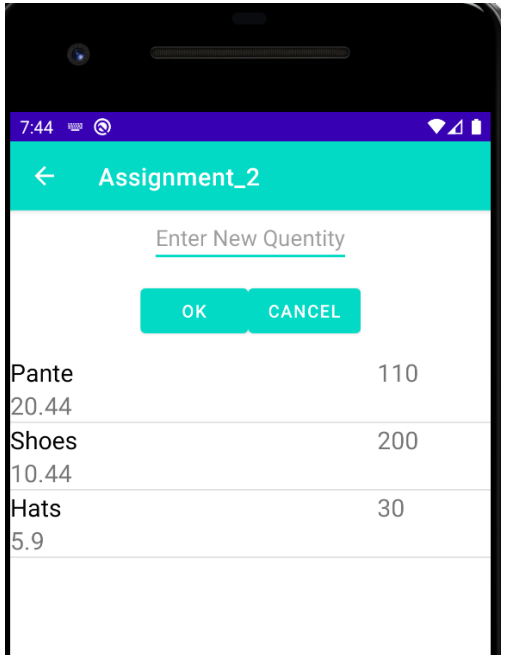
- Edit text, to enter the new quantity.
- Two buttons, ok and cancel.
- List View shows the stock list of product

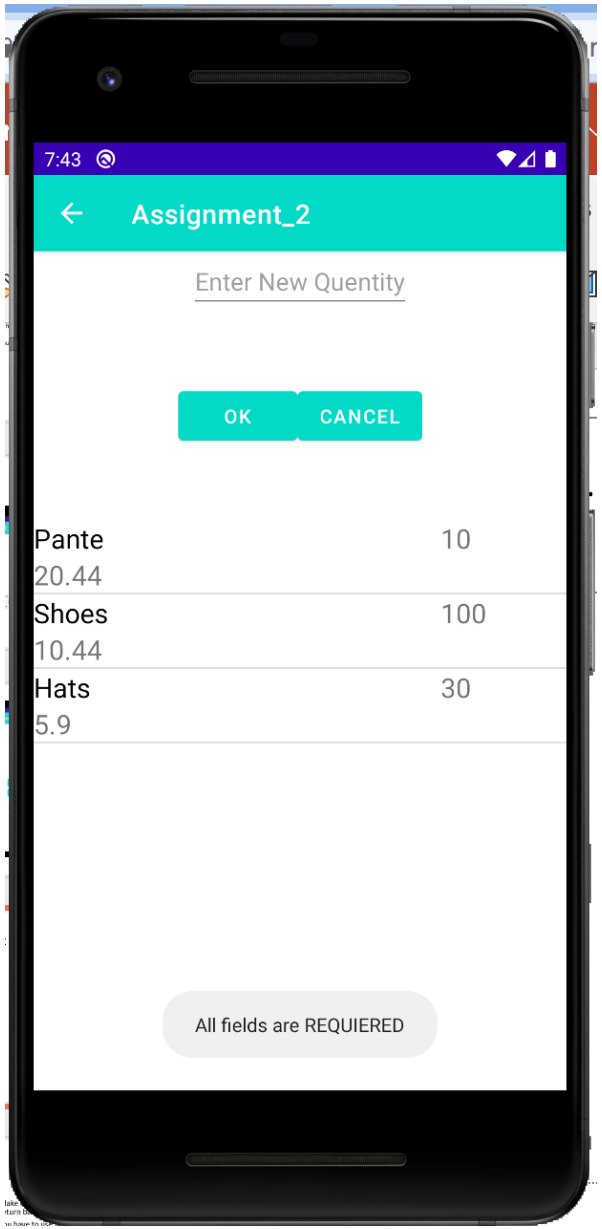
When the user select a product and enter a new quantity, it will be added to the existing quantity and update the stock.



Validating User's input

1. If the user clicks on OK button without selecting an item or entering a quantity, a Toast with an error message must appear.
2. The user should be able to update the quantity of an item by adding the inserted value to the existing quantity.
3. Clicking on Cancel must return the user to the Manger page.





Important Notes to consider

- Make sure to see the updated list when you return to Main Activity after updating quantity in restock activity.
- You have to use Recycler View and Adapter in History List.
- Make sure to design all layout using constraint layouts.
- You have to rotate your device during the video demonstration.