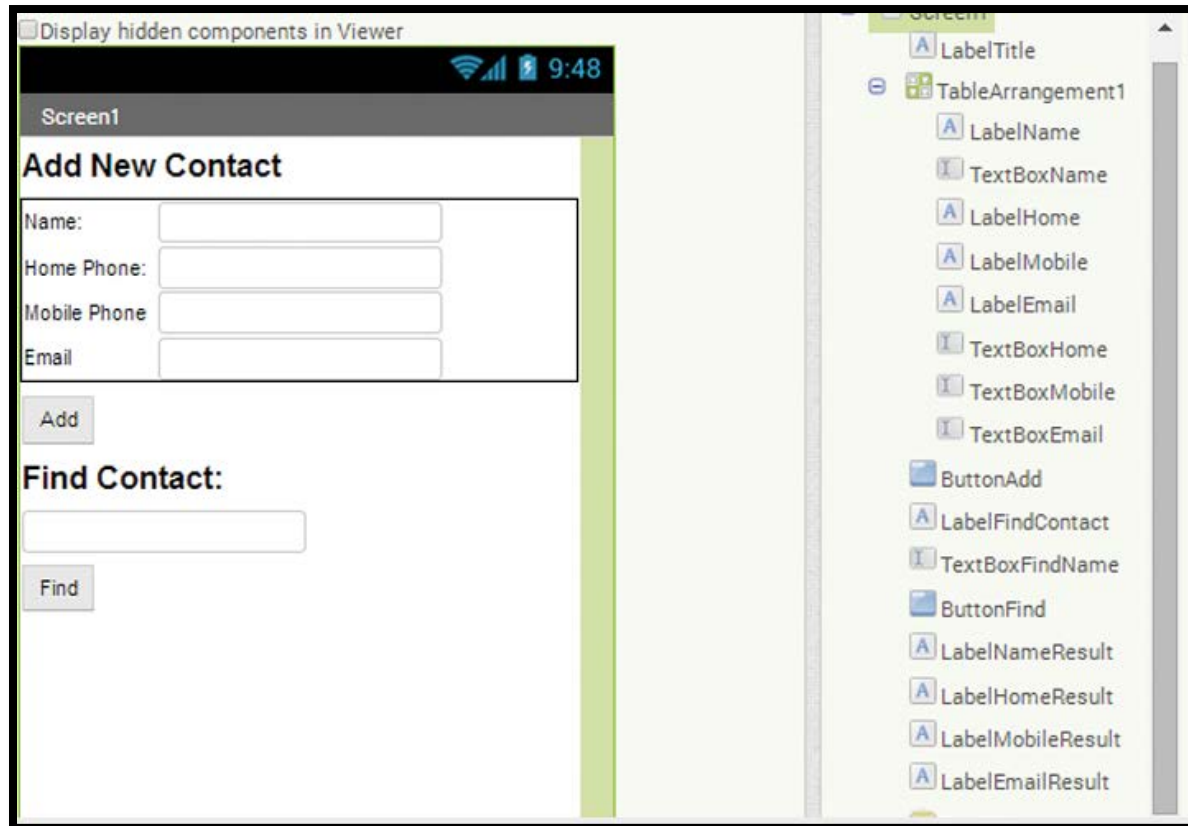


Storing a List as a Value in a Tag-Value Pair. Part 2

Now let's modify the interface so that a user can type in a name, press the Find Button, and then have that contact's information pulled back out of the TinyDB and presented back onto the device.



Step 1: Add a Label, rename it LabelFindContact, and change its Text property to Find Contact. Set the font size to 20 and mark it as bold.

Step 2: Add a TextBox, rename it TextBoxFindName, and set the Hint property to Enter Name.

Step 3: Add a Button, rename it ButtonFind, and set the Text property to Find.

Step 4: Add four Labels and rename them LabelName, LabelHomePhone, LabelMobilePhone, and LabelEmail. Clear the Text properties of each.

Step 5: Now you will program the when ButtonFind.Click do event handler.

- When the user clicks the Find Button, we will first retrieve the value from the TinyDB using the name that was entered as the tag.
- Because the value was stored as a list, we will have to iterate through the list to extract the values.
- As we extract each value, we will populate the Labels on the interface.
- Go to ButtonFind and select the when ButtonAdd.Click do block.

Step 6: Now let's set up an empty list variable to hold the results (name, home number, mobile number, and email) after we search and find a value in the TinyDB.

- Go to the Variables drawer, select an initialize global name to block, click on name and rename the variable.
- Results, and plug a create empty list block from the Lists drawer.

Step 7: Retrieve the value from the TinyDB, as follows:

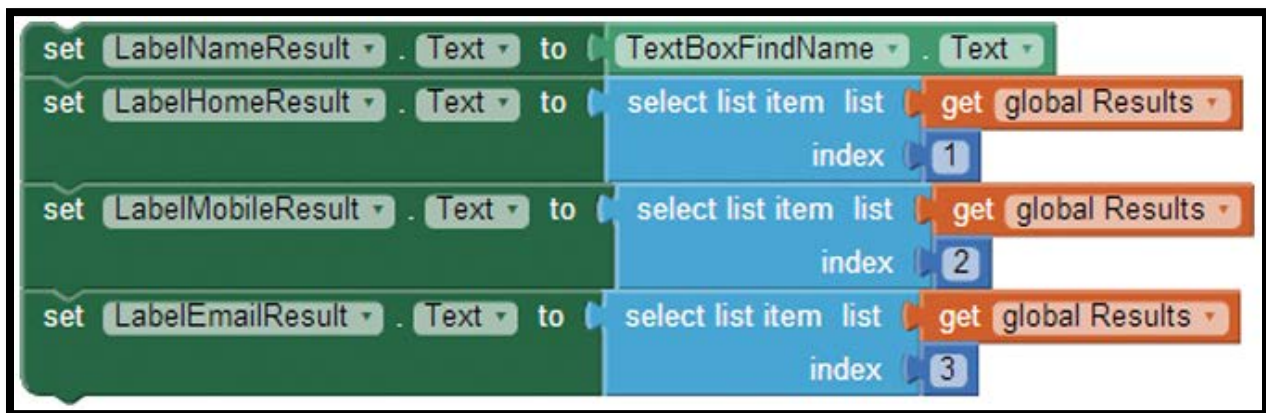
- Hover over the variable name ContactInfo in the initialize global ContactInfo block and select a set global ContactInfo to block.
- Go to TinyDBContacts and select the TinyDBContacts.GetValue block. Plug in the TextBoxFindName.Text block as the tag, and then plug the get value block into the ContactInfo block.
- Create an empty text block to use for the valueIfTagNotThere slot.
- Plug the entire block into the when ButtonFind.Click do block.



Step 8: Now we want to pull out the information in the list that we stored as our value for the tag. We can use the select list item in list block from the Lists drawer to retrieve the items and populate the appropriate Labels.

- Find the set LabelHomeResult.Text block from its drawer and place it in the Blocks Editor.
- In the Lists drawer, select the select list item in list block and plug it into the set LabelHomeResult.Text block.
- Hover over the variable name Results in the initialize global Results block and select a get global Results block. Plug that into the select list item in list block in the first slot.
- Because the home number is the first thing stored in the results list, use a number 1 block for the index argument.

Step 9: Repeat Step 8 for set LabelMobileResult.Text and LabelEmailResult.Text, but be sure to change the index arguments to 2 and 3 respectively.



Step 10: Place the Label set blocks into the ButtonFind.Click event handler.