App Inventor Workshop 2

Objectives

- Create a registration app
- Use lists to store data
- Use a web db to save your users
- Use maps to show where your users are

Types of Databases

Types of databases:

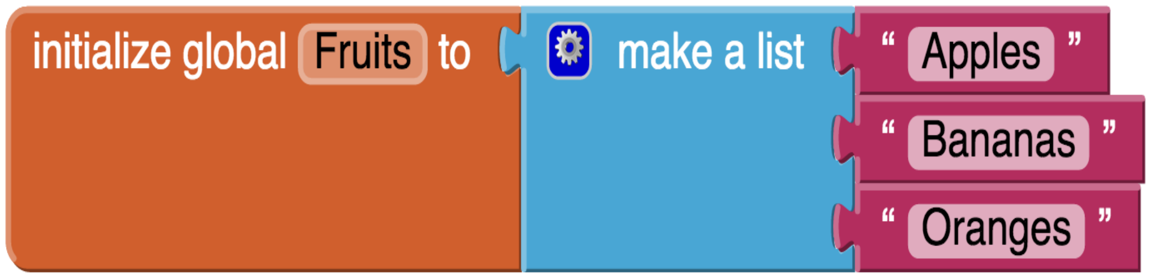
- TinyDB stores data directly on the phone and will only be used by that phone
    - Uses tag, value pairs to store the data
    - Tag to get the data back

- TinyWebDB stores data on a web database that can be shared among phones
    - Uses tag, value pairs to store the data
    - Tag to get the data back

- FusionDB stores data on a web database that can be retrieved
    - Uses columns and rows to store the data, like Excel
    - Flexible retrieval

Organizing data

- Lists can hold multiple pieces of data and they're easy to get data from.
- You may have made a to-do list or a grocery list before, and lists in programming are very similar.

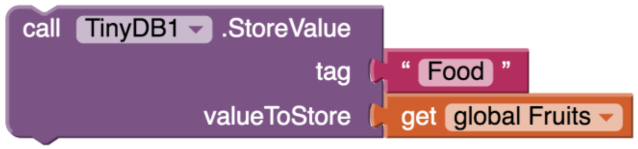


```
initialize global Fruits to  make a list  " Apples "
                                          " Bananas "
                                          " Oranges "
```

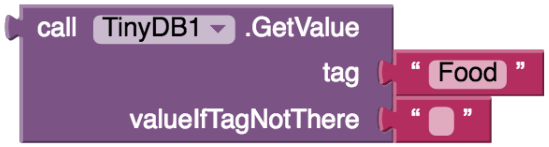
List Name: Fruits

- Apples (Index = 1)
- Bananas (Index = 2)
- Oranges (Index = 3)

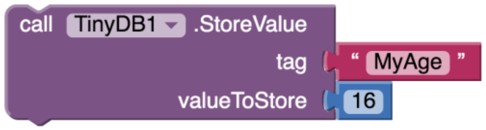
Using Lists in a Database



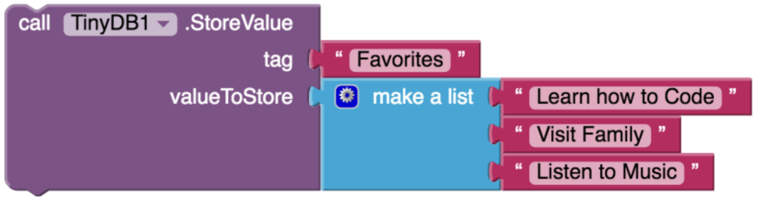
```
call TinyDB1 .StoreValue
    tag "Food"
    valueToStore get global Fruits
```



```
call TinyDB1 .GetValue
    tag "Food"
    valueIfTagNotThere ""
```



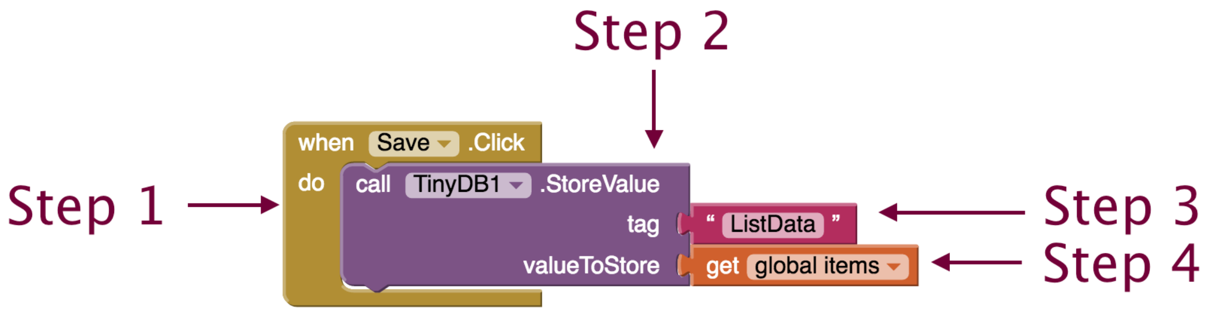
```
call TinyDB1 .StoreValue
    tag "MyAge"
    valueToStore 16
```



```
call TinyDB1 .StoreValue
    tag "Favorites"
    valueToStore make a list "Learn how to Code"
                             "Visit Family"
                             "Listen to Music"
```

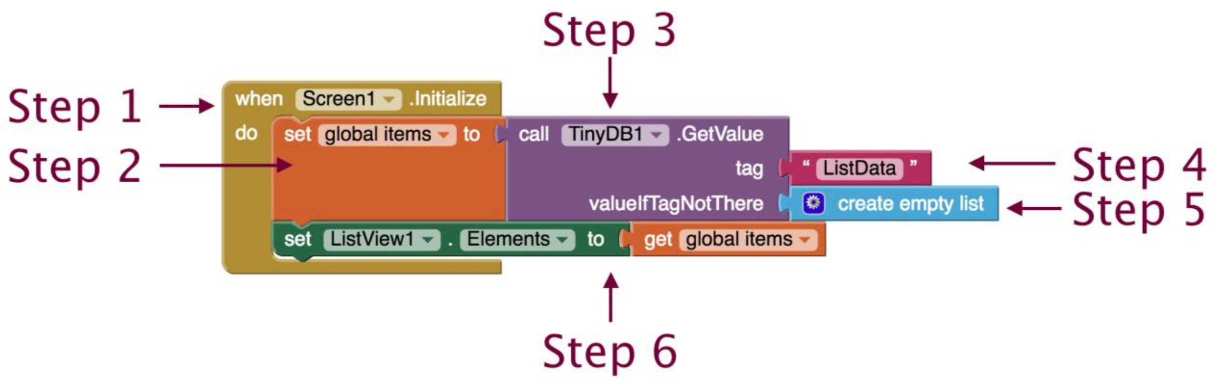
Saving the list for next time

1. Grab the when "when button.Click" block for your save button
2. Add the "call TinyDB1.StoreValue" block
3. Add a "tag" name
4. Add your *items* variable to "valueToStore"



Using a saved list

1. Grab the "when Screen1.Initialize" block
2. Grab a "set 'variable name' to" block and set the variable to be your *items* list
3. Call the database using "call TinyDB1.GetValue" block
4. Enter the tag name you used to save the list for the "tag"
5. Put the "create empty list" block for "valueIfTagNotThere"
6. To view your list "set ListView1.Elements to" block and attach it to your *items* var


Step 3
Step 1
Step 2
when Screen1 .Initialize
do set global items to call TinyDB1 .GetValue
tag " ListData "
valueIfTagNotThere create empty list
set ListView1 . Elements to get global items
Step 4
Step 5
Step 6

TinyWebDB

- TinyWebDB is like TinyDB but in the cloud
- Default TinyWebDB is shared by everyone
- https://appinvtinywebdb.appspot.com/
- Good for developing and testing your app

- If you want your own TinyWebDB
  - http://appinventor.mit.edu/explore/content/custom-tinywebdb-service.html

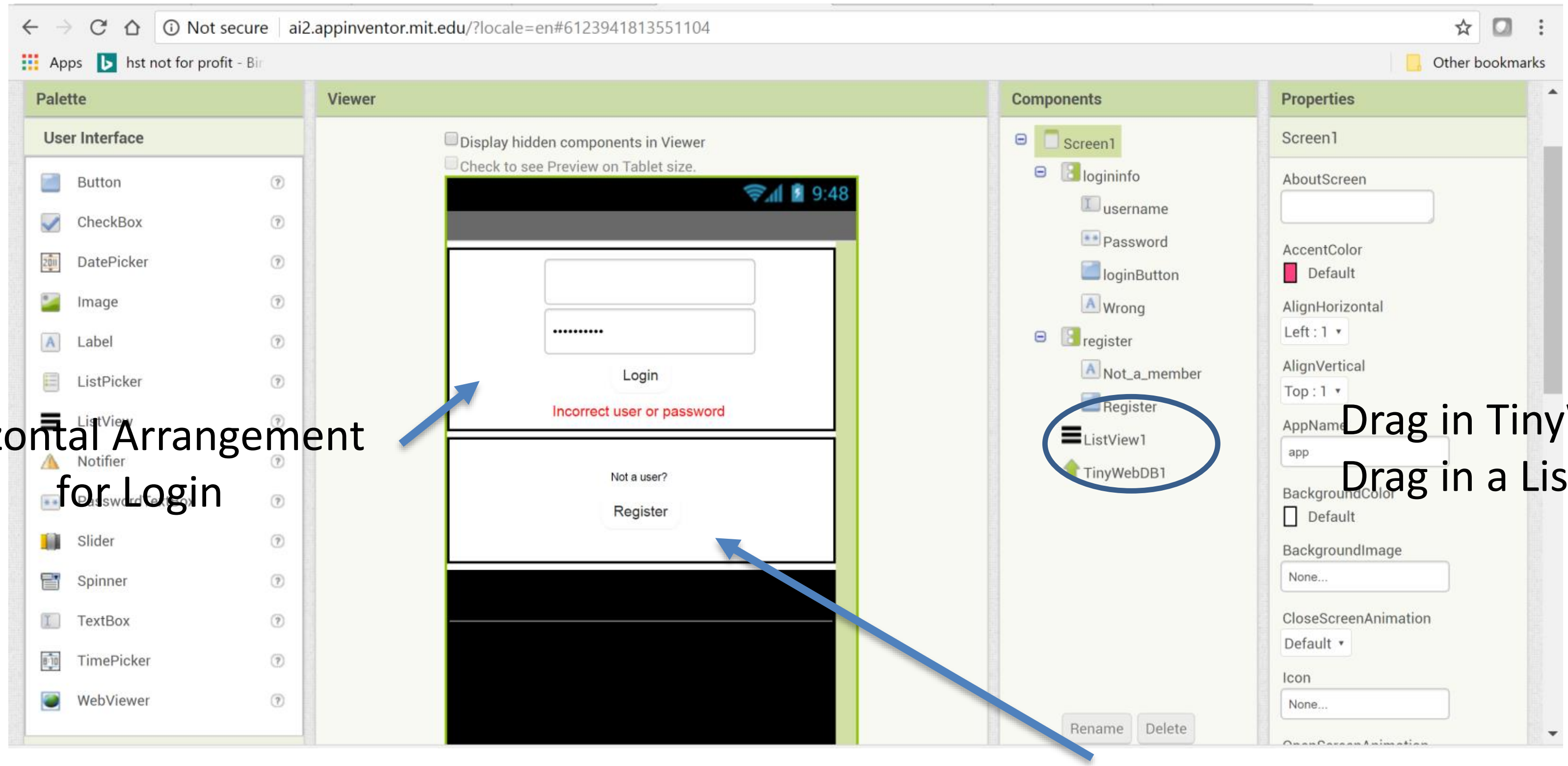
Building our app

- Register new users
- Login for existing users
- Map where the user is
  - Show directions from current location

Components we will use

- TinyWebDB to store our registered users
- List to store the information about our users
- Maps to show where our users are
  - Activity starter to call the google maps app
- Notifier to show messages

Creating a Login Screen



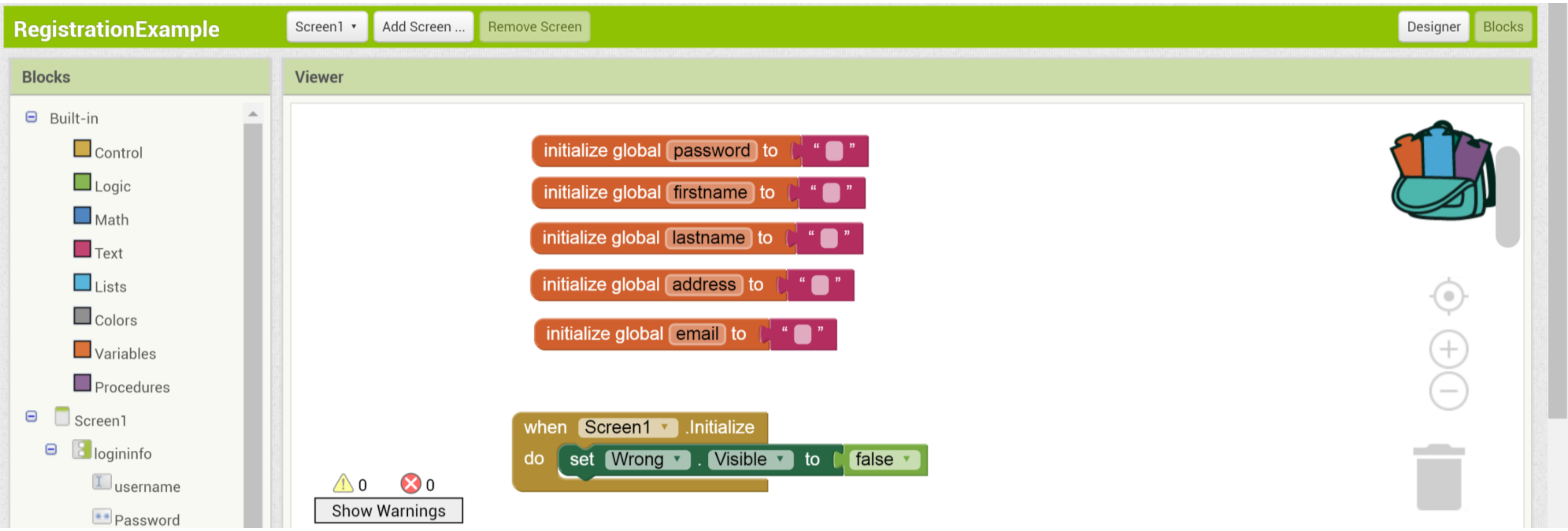
Horizontal Arrangement for Login

Drag in TinyWebDB

Drag in a List Viewer

Horizontal Arrangement for Register

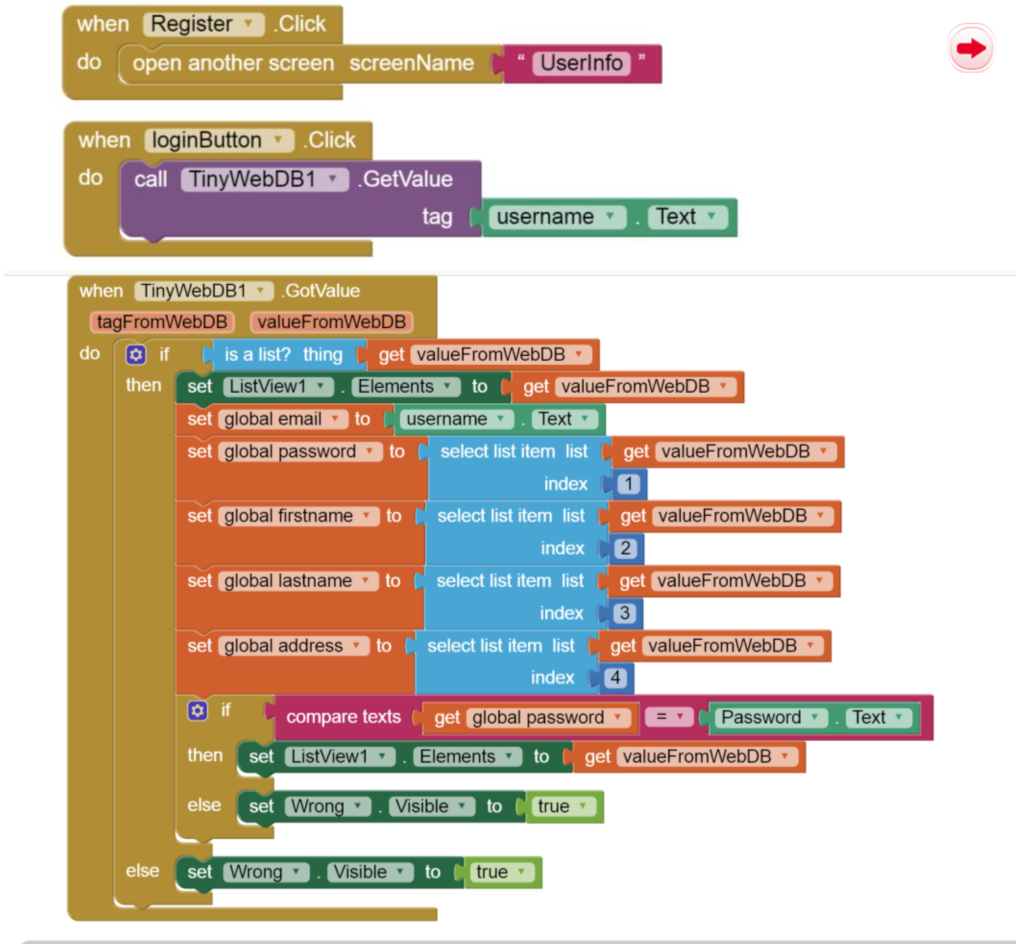
Login blocks



```
initialize global password to " "
initialize global firstname to " "
initialize global lastname to " "
initialize global address to " "
initialize global email to " "

when Screen1 .Initialize
do  set Wrong . Visible to false
```

Initialize our variables and Start with the error message turned off

Logging the user in



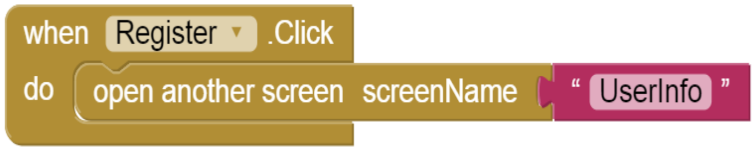
```
when Register .Click
do  open another screen  screenName  " UserInfo "

when loginButton .Click
do  call TinyWebDB1 .GetValue
                          tag  username . Text

when TinyWebDB1 .GotValue
 tagFromWebDB  valueFromWebDB
do  if  is a list?  thing  get valueFromWebDB
    then  set ListView1 . Elements to  get valueFromWebDB
          set global email to  username . Text
          set global password to  select list item  list  get valueFromWebDB
                                                    index  1
          set global firstname to  select list item  list  get valueFromWebDB
                                                     index  2
          set global lastname to  select list item  list  get valueFromWebDB
                                                    index  3
          set global address to  select list item  list  get valueFromWebDB
                                                   index  4
          if  compare texts  get global password  =  Password . Text
          then  set ListView1 . Elements to  get valueFromWebDB
          else  set Wrong . Visible to  true
    else  set Wrong . Visible to  true
```

- If they are logging in, get the value from tinywebdb

Check that the value is a list, set our variables, and make sure they used the right password

Show the list

Registering Users

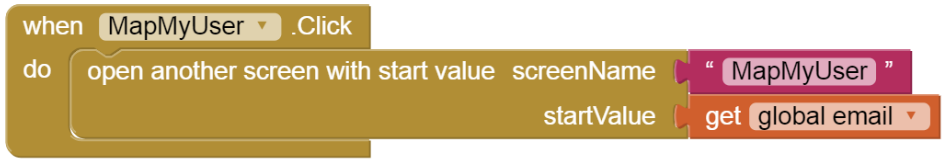


```
when Register .Click
do  open another screen screenName "UserInfo"
```

- Call the UserInfo screen to register
- Decide what information you want from the user
- We will use a list to keep the information – the list will be the value we store
- We will use their email as the tag

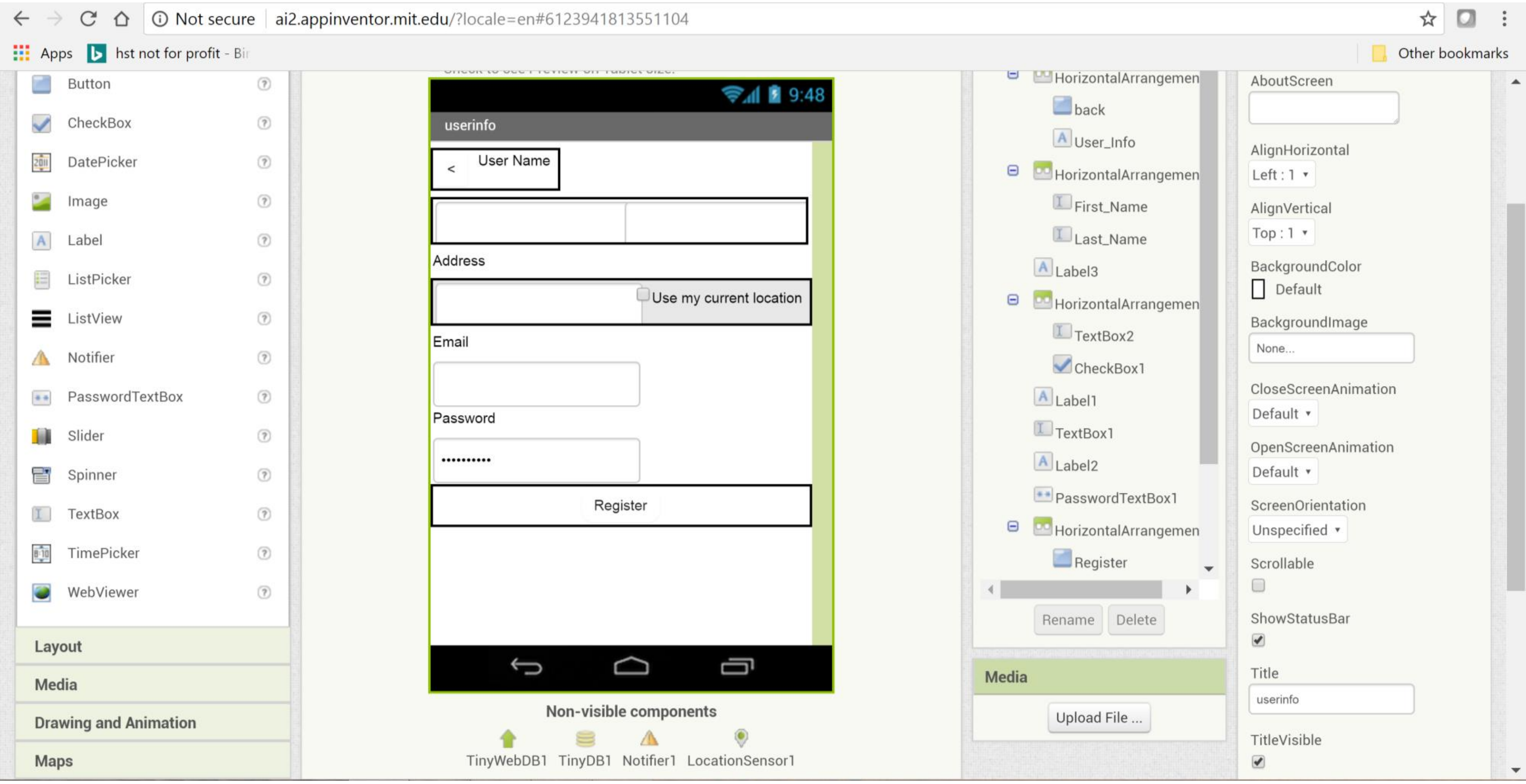
MapMyUser

- Call the mapping screen passing in the user's email



```
when MapMyUser .Click
do  open another screen with start value  screenName  " MapMyUser "
                                           startValue  get global email
```

Creating a Registration Screen



AboutScreen

AlignHorizontal: Left : 1

AlignVertical: Top : 1

BackgroundColor: Default

BackgroundImage: None...

CloseScreenAnimation: Default

OpenScreenAnimation: Default

ScreenOrientation: Unspecified

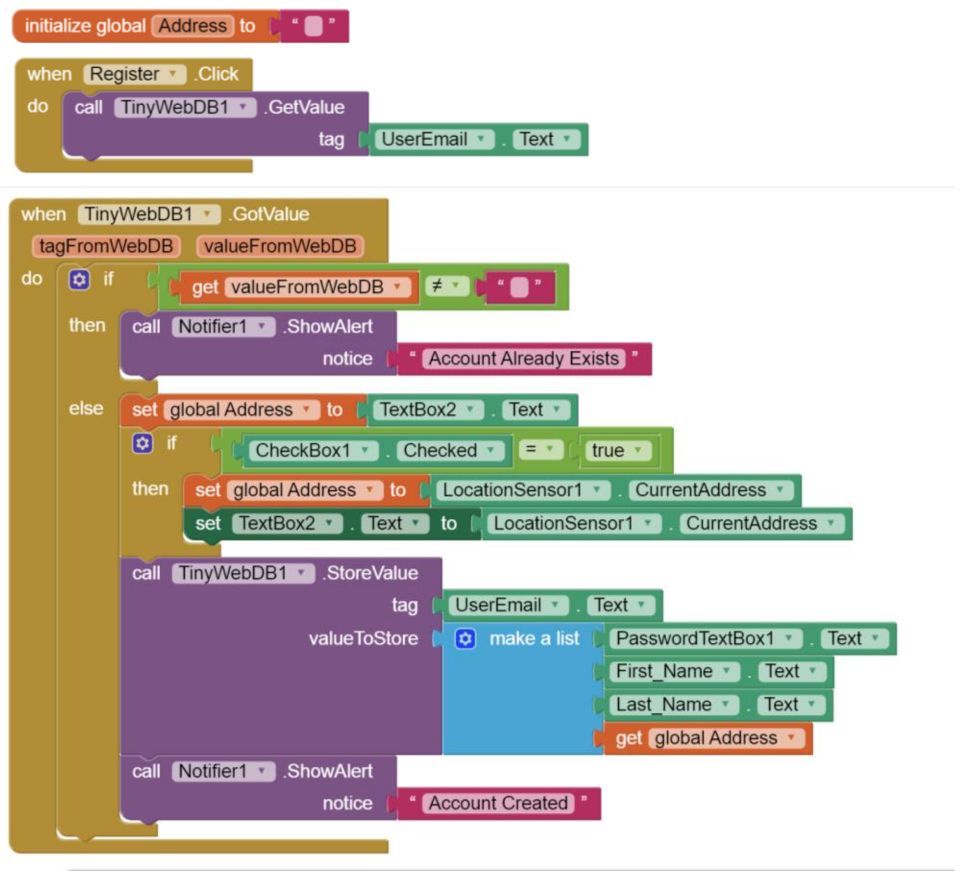
Scrollable: ☐

ShowStatusBar: ✓

Title: userinfo

TitleVisible: ✓

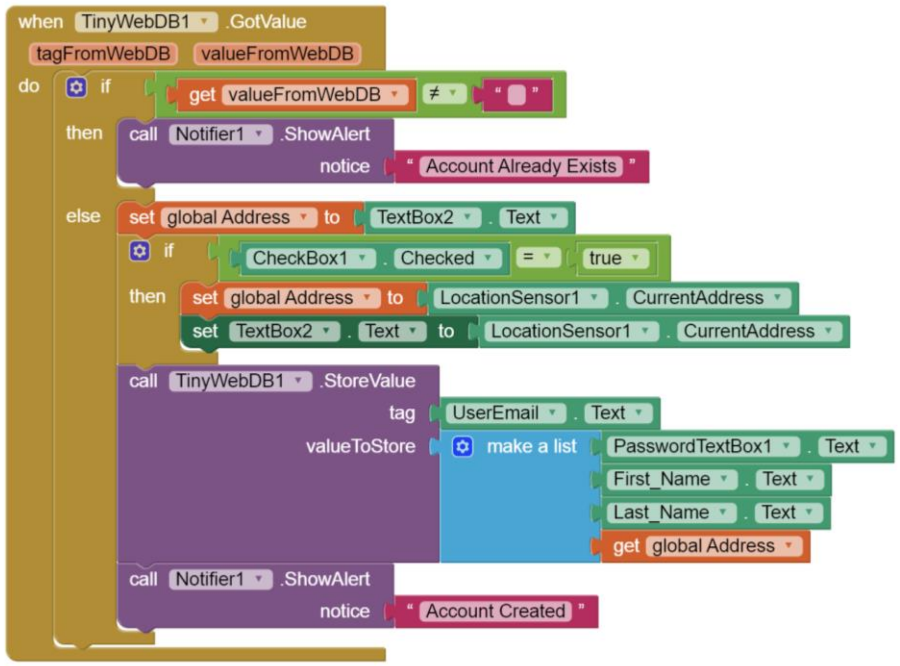
Registration blocks



```
initialize global Address to " "

when Register.Click
do  call TinyWebDB1.GetValue
        tag  UserEmail.Text

when TinyWebDB1.GotValue
  tagFromWebDB  valueFromWebDB
do  if  get valueFromWebDB ≠ " "
    then  call Notifier1.ShowAlert
              notice  " Account Already Exists "
    else  set global Address to TextBox2.Text
          if  CheckBox1.Checked = true
          then  set global Address to LocationSensor1.CurrentAddress
                set TextBox2.Text to LocationSensor1.CurrentAddress
          call TinyWebDB1.StoreValue
              tag  UserEmail.Text
              valueToStore  make a list  PasswordTextBox1.Text
                                         First_Name.Text
                                         Last_Name.Text
                                         get global Address
          call Notifier1.ShowAlert
              notice  " Account Created "
```

- Check that it is a new user
- Set their address if they picked current location
- Store the user information in TyinWebDB

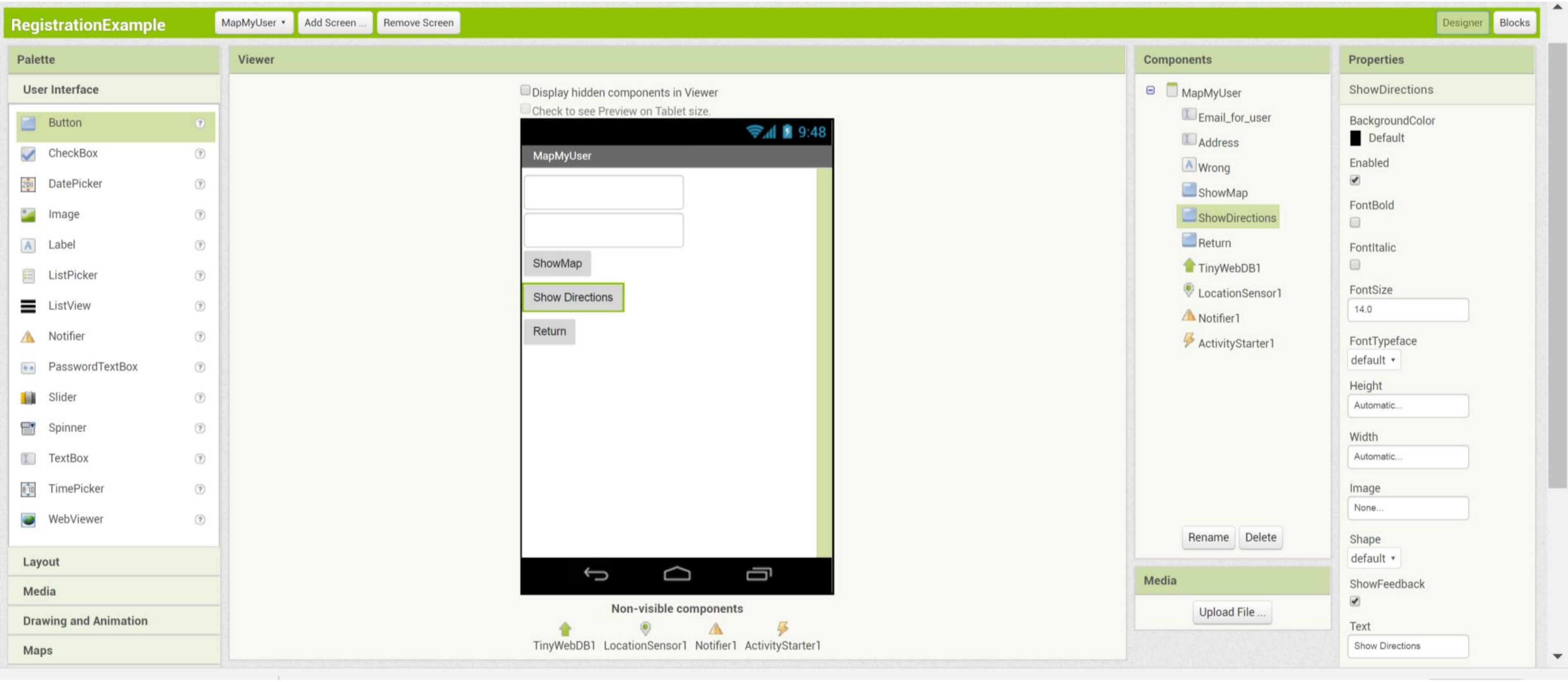
Storing in TinyWebDB



```
when TinyWebDB1 .GotValue
  tagFromWebDB  valueFromWebDB
do  if  get valueFromWebDB ≠ " "
    then  call Notifier1 .ShowAlert
                notice  " Account Already Exists "
    else  set global Address to TextBox2 . Text
          if  CheckBox1 . Checked = true
          then  set global Address to LocationSensor1 . CurrentAddress
                set TextBox2 . Text to LocationSensor1 . CurrentAddress
          call TinyWebDB1 .StoreValue
                tag  UserEmail . Text
                valueToStore  make a list  PasswordTextBox1 . Text
                                           First_Name . Text
                                           Last_Name . Text
                                           get global Address
          call Notifier1 .ShowAlert
                notice  " Account Created "
```

- Store the user info in a list
- The UserEmail is the tag
- Password is list item 1
- First name is list item 2
- Last name is list item 3
- Address is list item 3
- On Screen1 and MapMyUser use those list items to get the values back

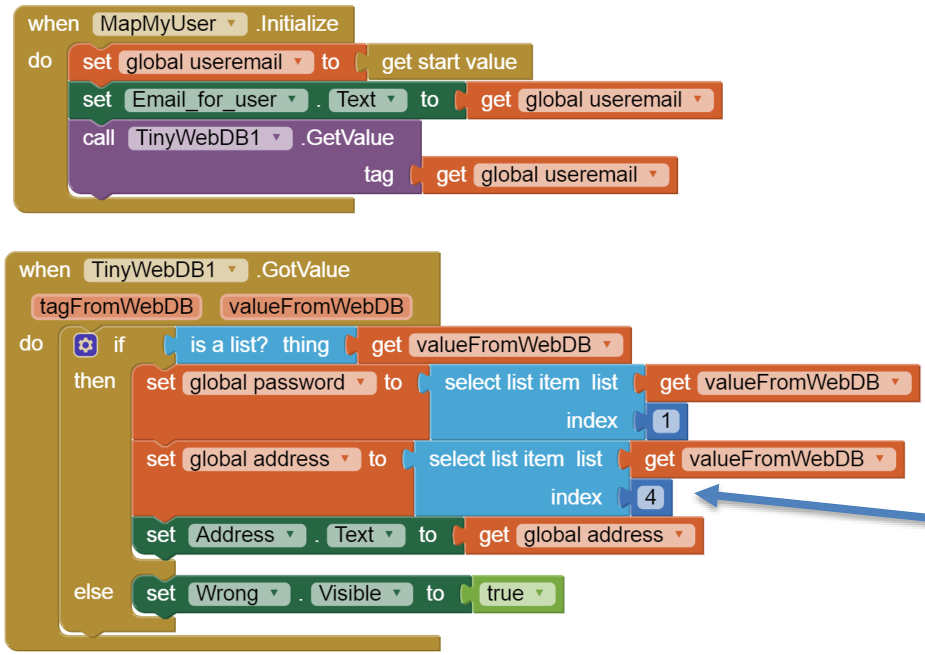
Creating a Map Screen



BackgroundColor: Default
Enabled: ✓
FontBold: ☐
FontItalic: ☐
FontSize: 14.0
FontTypeface: default
Height: Automatic...
Width: Automatic...
Image: None...
Shape: default
ShowFeedback: ✓
Text: Show Directions

Mapping blocks

- Getting the user info – the starting value for the email was passed in from the first screen



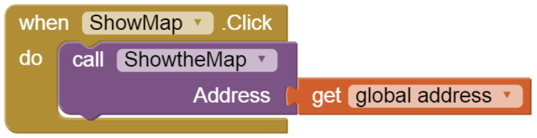
```
when MapMyUser .Initialize
do  set global useremail to  get start value
    set Email_for_user . Text to  get global useremail
    call TinyWebDB1 .GetValue
                         tag  get global useremail

when TinyWebDB1 .GotValue
 tagFromWebDB  valueFromWebDB
do  if  is a list? thing  get valueFromWebDB
    then  set global password to  select list item  list  get valueFromWebDB
                                                   index  1
          set global address to  select list item  list  get valueFromWebDB
                                                  index  4
          set Address . Text to  get global address
    else  set Wrong . Visible to  true
```

Get the User's info from TinyWebDB

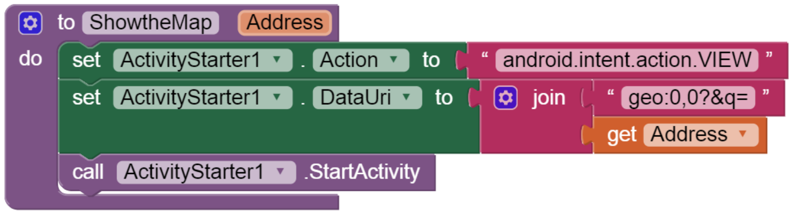
Remember the address was the Fourth item in the list

Map the users location

- When the user clicks to show the map call the activity starter to start the map.
- We need to create a procedure for the map
- Pass in the users address and start the map with that address



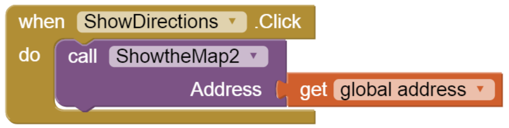
```
when ShowMap .Click
do  call ShowtheMap
         Address  get global address
```



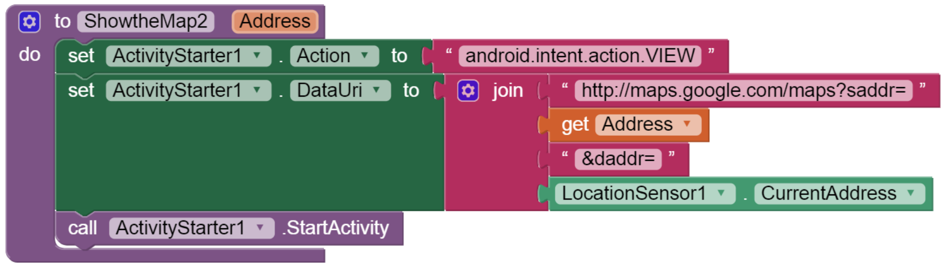
```
to ShowtheMap Address
do  set ActivityStarter1 . Action to  " android.intent.action.VIEW "
    set ActivityStarter1 . DataUri to  join  " geo:0,0?&q= "
                                             get Address
    call ActivityStarter1 .StartActivity
```

Show Directions

- When the user clicks directions you need to pass the start and destination addresses
- saddr is the start, daddr is the destination
- Call the map to go from their address to the current location

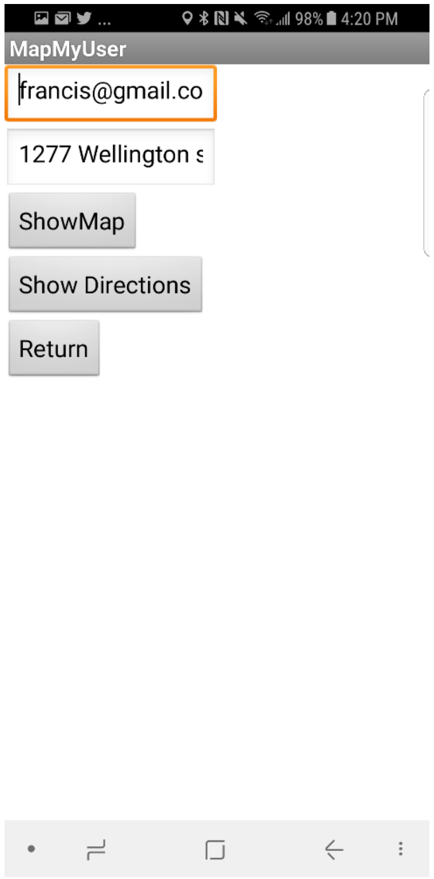


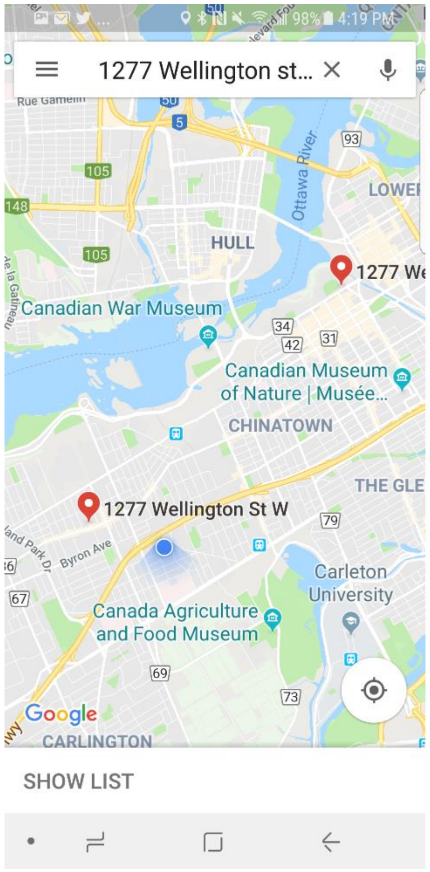
```
when ShowDirections .Click
do  call ShowtheMap2
         Address  get global address
```

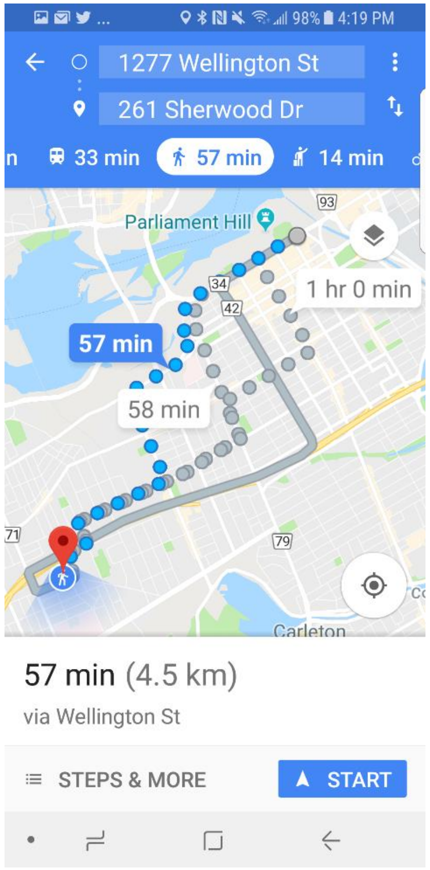


```
to ShowtheMap2 Address
do  set ActivityStarter1 . Action to  " android.intent.action.VIEW "
    set ActivityStarter1 . DataUri to  join  " http://maps.google.com/maps?saddr= "
                                             get Address
                                             " &daddr= "
                                             LocationSensor1 . CurrentAddress
    call ActivityStarter1 .StartActivity
```

Your completed app



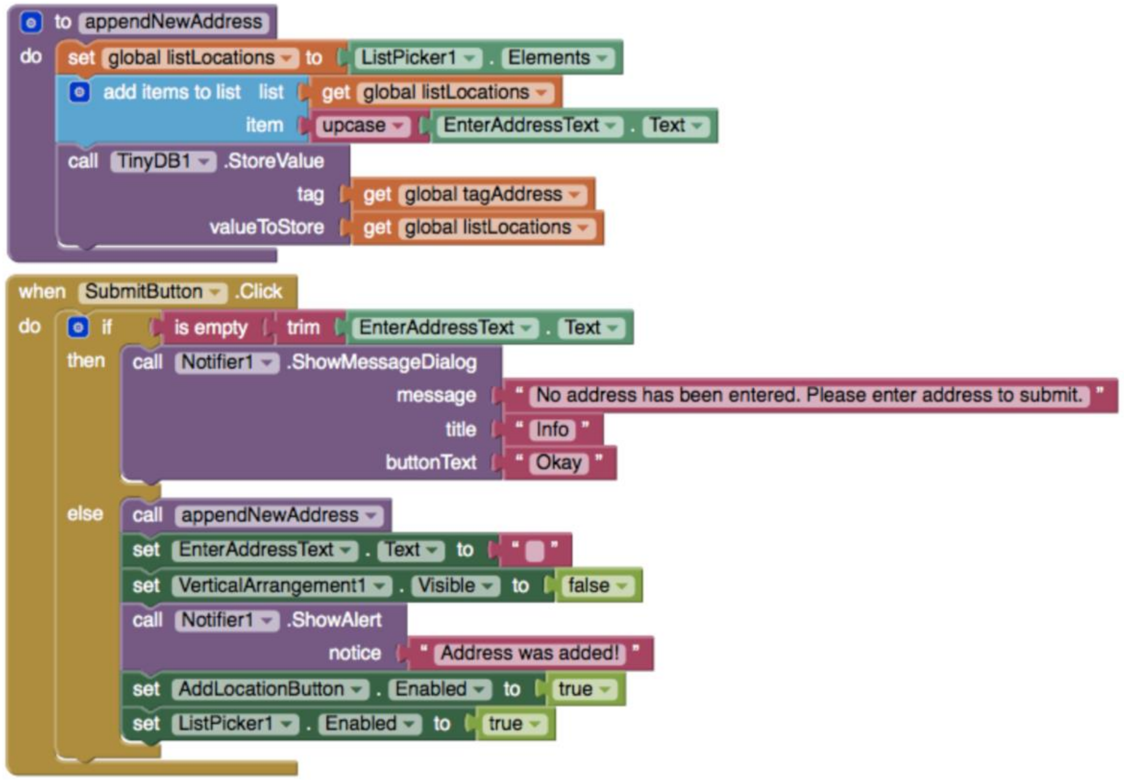




More with maps

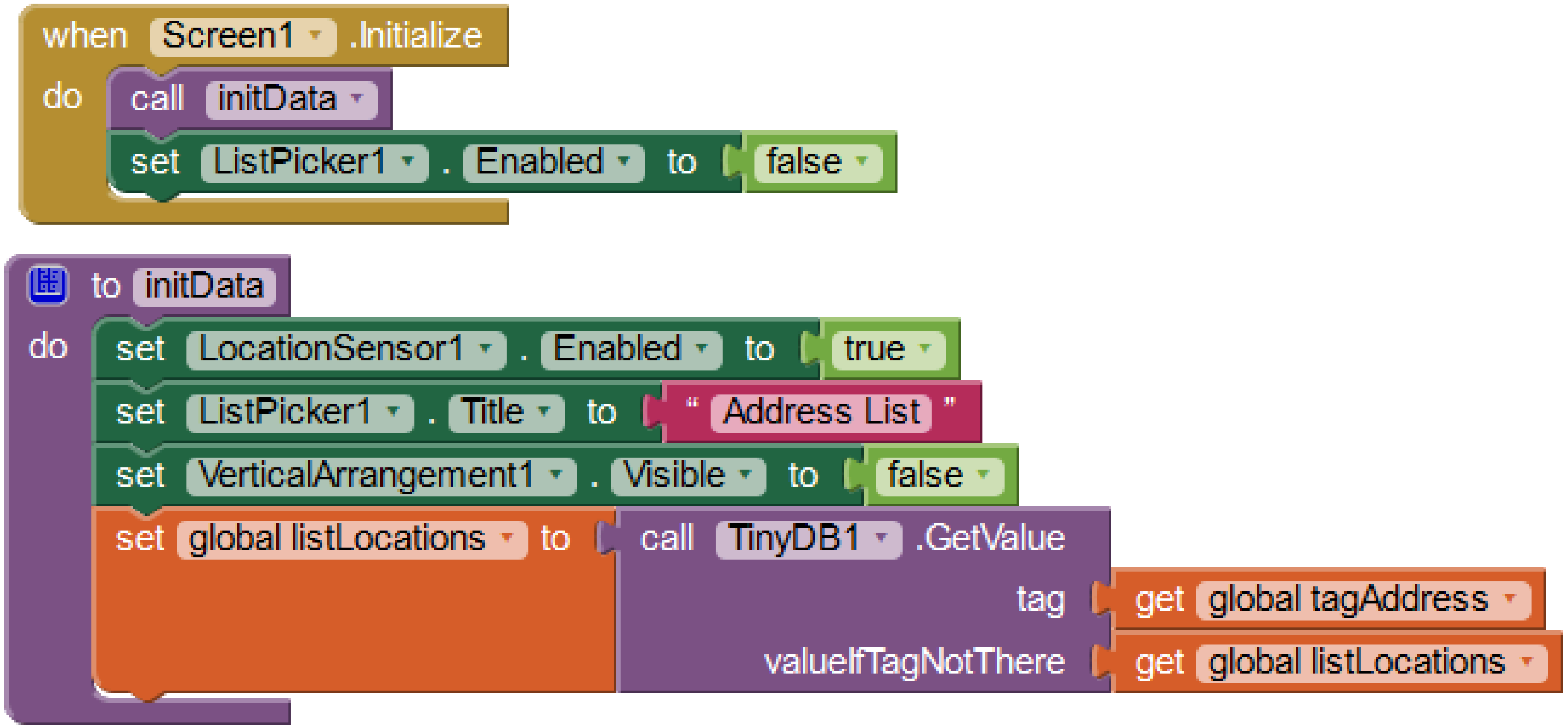
- Letting the user choose which address to map
  - Want to show where a job opportunity is
  - Want to show where a sports event is
  - Want to show where to drop off charity items …
- Create a list with the addresses
- Use the list picker to choose which one to map

Creating an address list



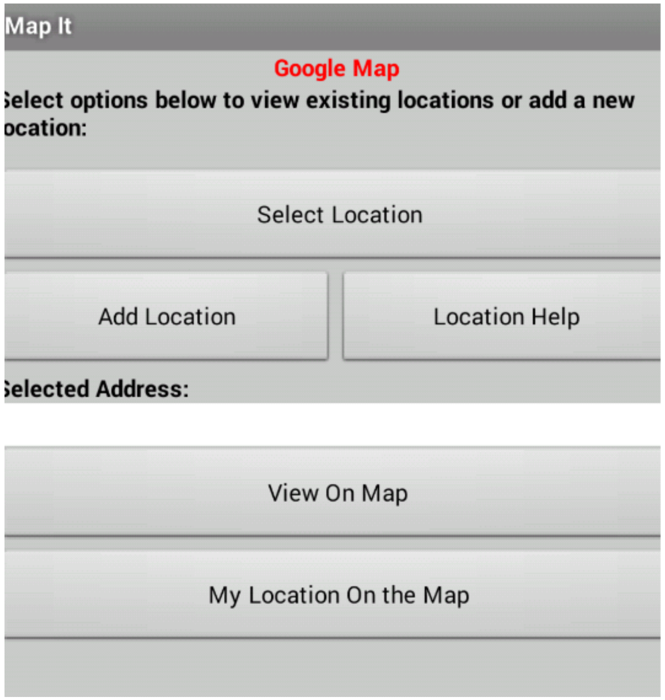
```
to appendNewAddress
do  set global listLocations to ListPicker1 . Elements
    add items to list  list  get global listLocations
                       item  upcase  EnterAddressText . Text
    call TinyDB1 .StoreValue
                 tag         get global tagAddress
                 valueToStore get global listLocations

when SubmitButton .Click
do  if  is empty  trim  EnterAddressText . Text
    then  call Notifier1 .ShowMessageDialog
                  message    " No address has been entered. Please enter address to submit. "
                  title      " Info "
                  buttonText " Okay "
    else  call appendNewAddress
          set EnterAddressText . Text to " "
          set VerticalArrangement1 . Visible to false
          call Notifier1 .ShowAlert
                  notice " Address was added! "
          set AddLocationButton . Enabled to true
          set ListPicker1 . Enabled to true
```

Retrieving an address list



```
when Screen1 .Initialize
do  call initData
    set ListPicker1 . Enabled to false

to initData
do  set LocationSensor1 . Enabled to true
    set ListPicker1 . Title to " Address List "
    set VerticalArrangement1 . Visible to false
    set global listLocations to call TinyDB1 .GetValue
                                             tag  get global tagAddress
                                 valueIfTagNotThere  get global listLocations
```

Selecting an address


Map It

Google Map

elect options below to view existing locations or add a new ocation:

Select Location

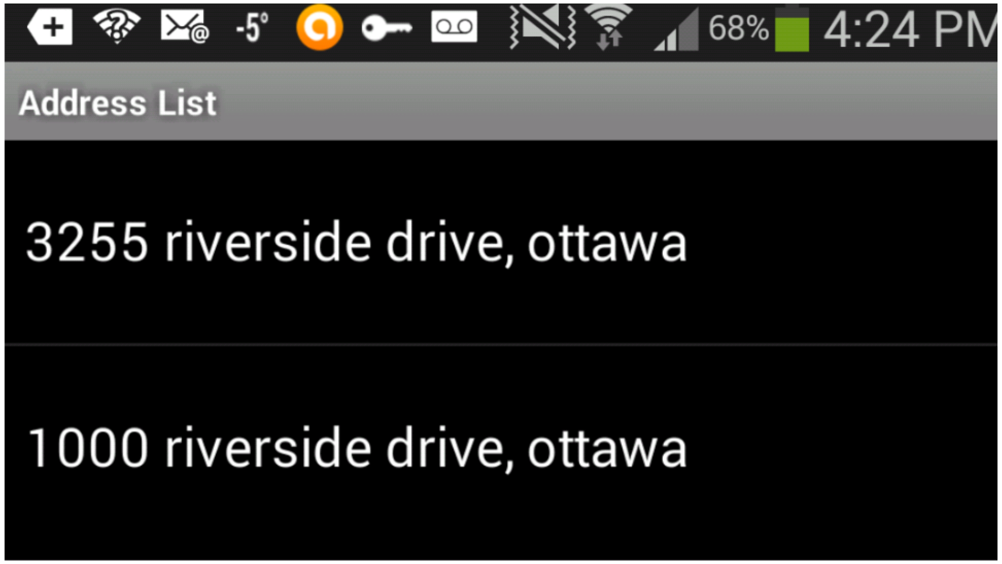
Add Location

Location Help

elected Address:

View On Map

My Location On the Map

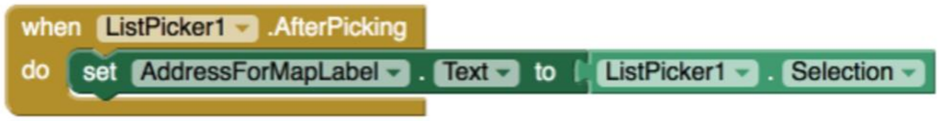

-5° 68% 4:24 PM

Address List

3255 riverside drive, ottawa

1000 riverside drive, ottawa

When the user clicks on ListPicker1 'Select Location' and selects an address, this action calls the blocks below:



```
when ListPicker1 .AfterPicking
do  set AddressForMapLabel . Text to ListPicker1 . Selection
```

Exploring maps

- http://appinventor.mit.edu/explore/ai2/android-wheres-my-car.html

- http://appinventor.mit.edu/explore/displaying-maps.html