Homework #2:  
Contact App



By Sean W. Caldwell

CSUID: 2595953

Date: 09/16/2019

Table of Contents

[Idea: 3](#_Toc19539421)

[Layout: 3](#_Toc19539422)

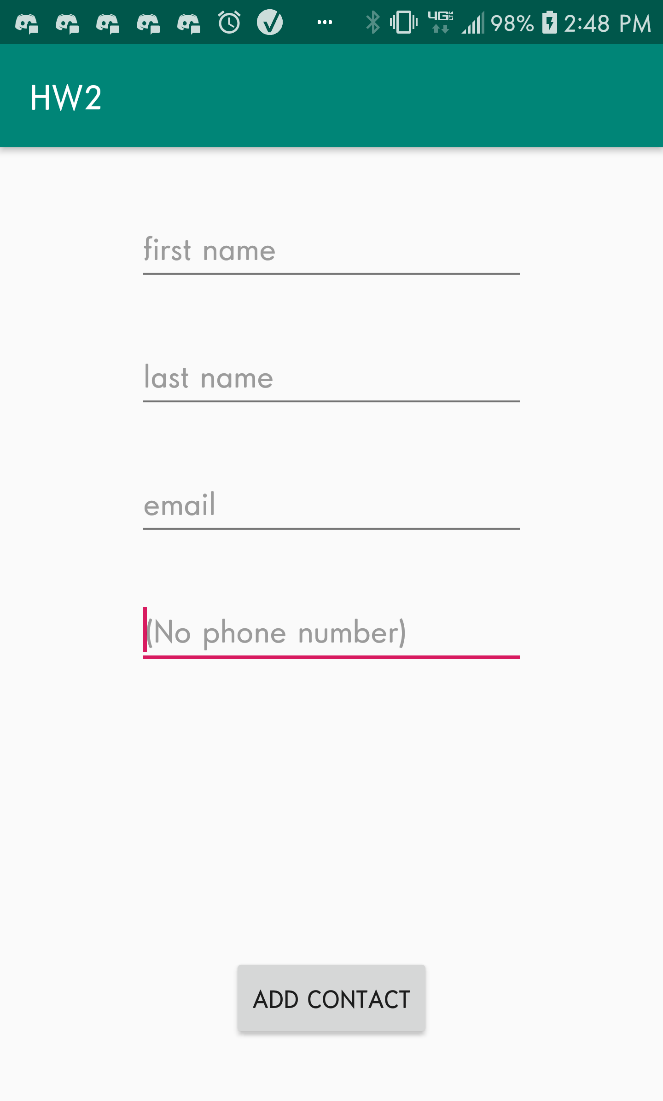
[Data Structure: 4](#_Toc19539423)

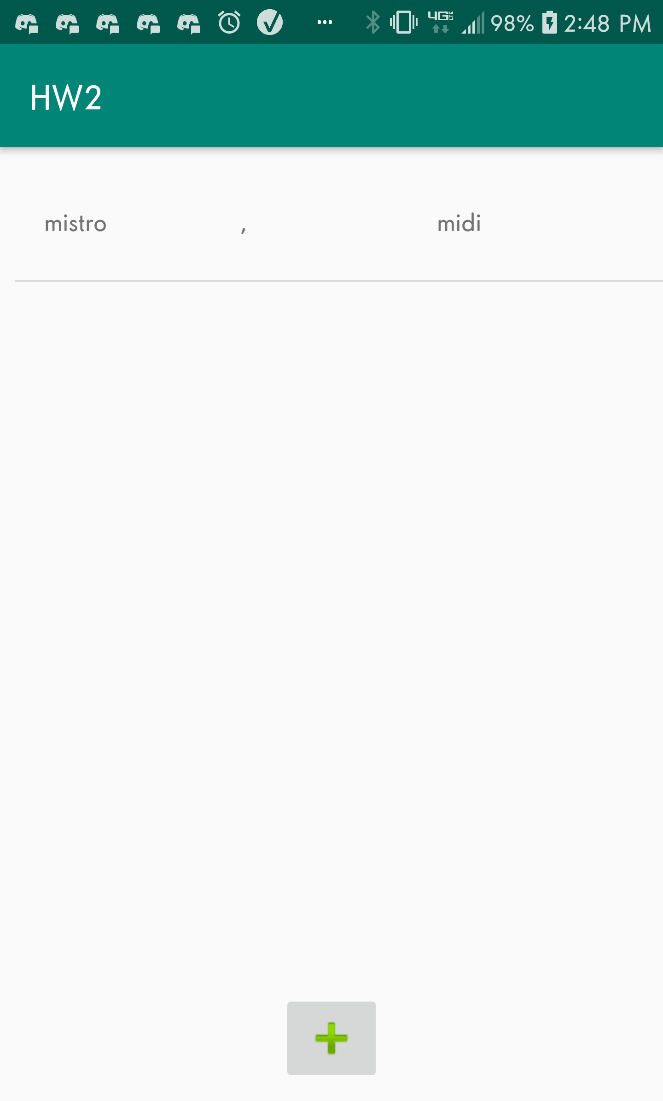
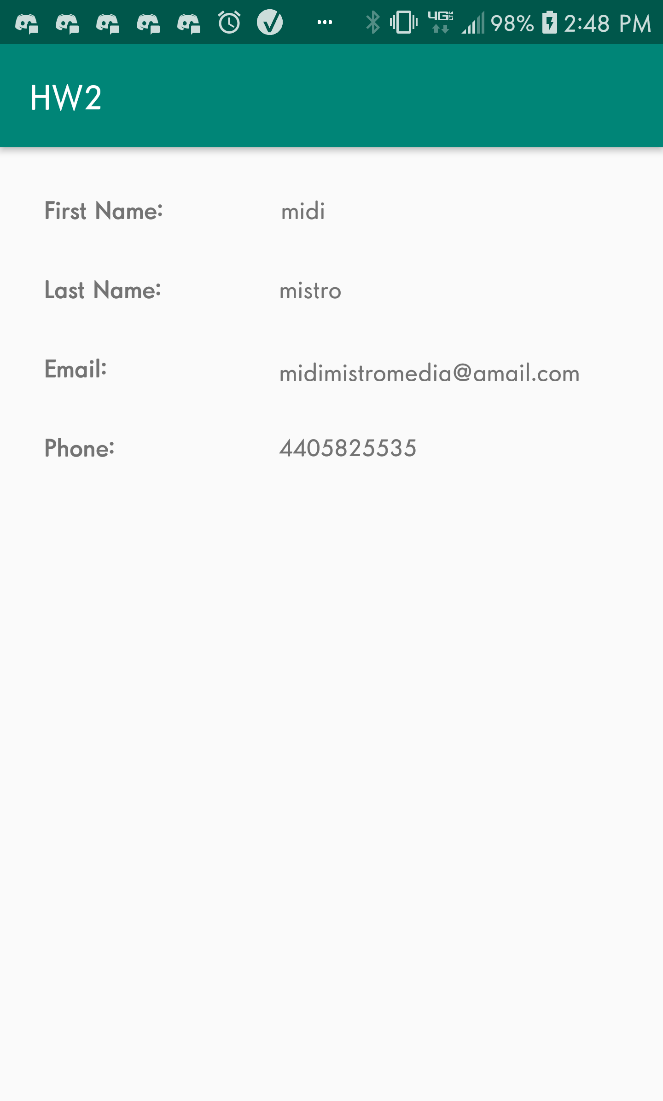
[Implementation Logic: 5](#_Toc19539424)

# Idea:

The idea behind this app was to create a non-persistent simplistic contacts app. For this app, I decided to make it a 90% functional simplistic contacts app. I did not include contact removal or contact persistence. Unlike what was provided in class or what was provided in Android, I decided to create my own custom “Contact” class, which means I could define a single contact as individual variables with their own variables, rather than a string that may not stay static for future expansion.

# Layout:

Like any contact app, my contact app displays a list of contacts. It does this by using a ListView and custom ListView Adapter. The user starts off with a blank screen that says there are no contacts. To add a contact, the user should press the green “+” button to add a new contact. Completion of contact is not necessary for my app. The green “+” takes the user from the main activity to the “AddContact”, where a user can add the new contact. Although not necessary for this homework, by clicking the contact name, the user can then go into the “ViewContact” activity where they can view the whole contact info. To exit out of this 3rd activity, all the user need do is press the back button.



# Data Structure:

Contact: (used for storing a contact; made possible by creating a custom class)

String FirstName

String LastName

String email

String phoneNumber

Boolean No-ContactsFlag (used for checking whether there were any contacts or not and providing the appropriate display).

# Implementation Logic:

To implement the back and forth between activities, I used a JSON string to share the contact, and used intent extras to share/extract the data, depending on the direction of the data. “AddContact” returns a json string of the new contact to the main activity, which updates the listivew when “AddContact” triggers the OnActivityResult. The ViewContact does not return anything as it is only meant to display the contact. I used my own custom Contact class to manage the contacts, making it easier to implement and manage a contact across activities.