Contacts Project Specs

Your objective is to create a Contacts App similar to the one on your phone. Your project will have multiple files and should be well organized. Mostly likely you will not be able to finish this project today. The most important part is to try to get all the main components in place so you can see what it’s like to build a more complex project. Add this project to your Github.

You’ll be building a program that models the Contacts app on your phone. Take a look at your Contacts app for a point of reference. You’ll create three files (contact.py, contact\_manager.py, and contact\_ui.py)

Contact Class

* Update your contact \_\_init\_\_ function to take all the potential optional parameters it might need. Ie, mobile\_phone, work\_phone, email, twitter\_handle, etc.
* Your contact class should have the ability to send a text! *(For now, it will just print out a message if a mobile phone number is available.)*

Contact Manager

You will create a new file called contact\_manager.py. It will make use the Contact class as a module. The purpose of contact\_manager is to manage the contacts list. Create a list to hold the contacts.

* It can:
  + show current contacts
  + add a new one
  + edit an existing one
  + delete an existing one
* Special cases:
  + Remember to handle special cases like upper/lower case names.
  + For this exercise, assume that you cannot have a duplicate contact with the same first and last name. Make sure to print out an error message when necessary.
* Make sure to test your functions in a main() function.

Contact UI

This module’s purpose is to provide a textual UI to interact with the Contact Manager. You will import contact\_manager.py. Take a look at the last lecture’s exercise Part 2 to help you through this.

* The UI(user-interface) menu will have the following options for a user to choose from running with a while loop:
  + **"0 - Main Menu"**
  + **"1 - Show all contacts"**
  + **"2 - Add a new contact"**
  + **"3 - Edit a contact"**
  + **"4 - Delete a contact"**
  + **"5 - Exit the program."**

Bonus Challenges

* Write contacts to a text file after adding, editing, or deleting.
* Read contacts from a text file when your program starts up.
* Look up how to send a text, using the Twilio API.

Project planning and pseudocode

1. Spend 10-15 mins alone by outlining your project. Use pseudocode to break down each file and then the functions in each file. Discuss your plan with your partner. Try to create the files and skeleton code that you think you’ll need.