

THOUGHT PROCESS & ALGORITHM

INTRODUCTION

This document serves as a documentation of my rationale in building the algorithm and the shell script using the Bourne again shell to simulate an address book. This address book implementation has the following features:

1. Create New Contact

Allows users to create a new contact to be added to the address book.

2. View Contacts

This feature allows a user to view all contacts that have been saved in the address book.

3. Edit Contacts

This allows users to make edits/ changes to existing contact records.

4. Search Contact

This allows users to search for a specific contact from the existing contacts in the address book's record.

5. Delete Contact

This feature allows users to delete contacts from the existing contacts in the address book's record.

Rationale

I first broke down the problem into small batches of deliverables.

Question

Create an Address Book program using the bourne or bourne-again shell. It should use functions to perform the required tasks. It should be menu-based, allowing you the options of:

Search address book

Add entries

Remove / edit entries

You will also need a "display" function to display a record or records when selected.

Requirements:

Search: When the user searches for "Harry", the script should identify and display all "Harry" records. It is up to you whether this search is in Surname only, or in the entire record.

Add: Input the data (Name, Surname, Email, Phone, etc). If it appears to be a duplicate, edit the existing record. Save the record into the data file when the user confirms.

Remove: Enter search criteria, narrow it down to one, confirm, then remove that record.

Edit: As remove, but use the existing record for default entries. For example, if you change the phone number, the session may look like this, if you only want to change Harry Porter's Phone Number: Name [Harry Porter]

Phone [+234] 123456781

Email [hp@hpseries.org]

Remove the old record, and add the new one. Alternatively, edit the existing record.

Identified Deliverables.

Below are some of the deliverables I identified:

- Create a Menu showing the user available options
- Create a function to Add Contacts to the Address Book file.
- Create a function to View/Display all the contacts saved in the Address Book file.
- Create a function to Remove/Delete Contacts from the Address Book file. ●
- Create a function to search for a contact from the existing contacts in the Address Book file.
- Create a function to Edit contacts that are already saved in the address book.
- Create a way to exit the address book

For each of the deliverables, I have included the tasks that should be done.

● *Create a Menu showing the user available options*

This Menu should allow users to select options to either add a new contact, view available contacts, search for a contact, edit an existing contact, deleting a contact & even exiting the address book Menu.

● *Create a function to Add Contacts to the Address Book file.*

Add_Contacts() function aims to allow users to add their contact information using the contacts name, phone number & email address. This shall then add the user's information to the address book

● *Create a function to View/Display all the contacts saved in the Address Book file.*

View_Contacts() function is supposed to display a list of all the contacts that have been saved into the address book for the user to view

● *Create a function to Remove/Delete Contacts from the Address Book file.*

Delete_Contact() function is meant to allow a user to search for and delete contact information that they wish to remove from the address book.

● *Create a function to search for a contact from the existing contacts in the Address Book file.*

Search_Contact() function is supposed to allow users to search for contact information they wish to find from the address book file.

- **Create a function to Edit contacts that are already saved in the address book.**

Edit_Contact() function is meant to allow users to make edits/changes to existing contacts in the address book file.

- **Create a way to exit the address book**

Exit functionality allows the user to close & exit from the address book Main menu

ALGORITHM

1. Display the menu of options for the user to select.
 1. Add New Contact
 2. View Contacts
 3. Edit Contacts
 4. Search Contact
 5. Delete Contact
 6. Exit Address Book
2. **If** User_Option == 1.
then
 - a. Request user for Contact Information i.e Contact Name, Phone Number & Email address
 - b. Add Contact to the Address book file
3. **If** User_Option == 2.
then
 - a. Lists all Contacts Saved in the Address Book file.
4. **If** User_Option == 3.
then
 - a. Prompts the user to search for the contact to edit.
 - b. If the contact exists in the address book go to **c**, **else** return error message.
 - c. Prompt user to select the line number of the contact information to be edited.
 - d. Loop through the file line by line to find the user's target line to be edited.
 - e. When that line is found, then allow the user to input new edits to the desired contact information line.

f. Then finally save the edited information to the address book updating it.

5. **If** User_Option == 4.

then

- a. First allows the user to search for their desired contact information they wish to delete.
- b. If the desired contact information is found, then it is removed from the address book file.

6. **If** User_Option == 5.

then

- a. Prompts the user for the contact information to be deleted. To find it, the user must enter the contact to be deleted in the same letter case as saved in the address book.
- b. Should the desired contact be found, it will then be deleted from the address book.

7. **If** User_Option == 6

then

- a. Break from the while loop displaying the AddressBook Menu & Options, effectively closing & exiting the address book