Requirements and User Interface for a Simple Address Book

Requirements Statement

The software to be designed is a program that can be used to maintain an address book. An address book holds a collection of entries, each recording a person's first and last names, address, city, state, zip, and phone number.

It must be possible to add a new person to an address book, to edit existing information about a person (except the person's name), and to delete a person. It must be possible to sort the entries in the address book alphabetically by name or by ZIP code. It must be possible to search for an entry containing a given text in the person's name or other information about the person. It must be possible to print out all the entries in the address book in "mailing label" format.

It must be possible to create a new address book, to open a disk file containing an existing address book, to save an address book to a disk file, or to print an address book using standard New, Open, Save, Save As ... and Print File menu options. Of course, the program's File menu will also have a Quit option to allow closing all open address books and terminating the program.

It must be possible to sort the contents of the address book in alphabetical order by the person's last name, or by zip code order. It must be possible to search the address book for the occurrence of a given string in a person's name or any of the information fields about the person.

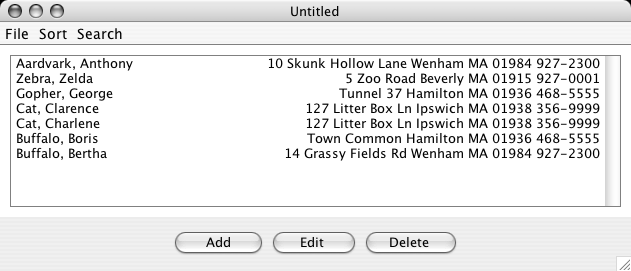
The initial requirements call for the program to only be able to work with a single address book at a time; therefore, if the user chooses the New or Open menu option, any current address book will be closed before creating/opening a new one. A later extension might allow for multiple address books to be open, each with its own window which can be closed separately, with closing the last open window resulting in terminating the program. In this case, New and Open will result in creating a new window, without affecting the current window.

The program will keep track of whether any changes have been made to an address book since it was created or read from disk, or was last saved (as the case may be). The 'Save' option in the 'File' menu is enabled just when the address book has been changed since it was created or read from disk, or was last saved. It is disabled following the creation of a new address book, the reading of an existing address book from disk, or a save/save as operation. (If any of these operations is cancelled by the user or fails due to an input-output error, the "changed" status of the address book is not altered.) The program will offer the user the opportunity to save changes when a changed address book is closed either explicitly or as a result of choosing to create/open another or to quit the program.

The program will keep track of the file that the current address book was read from or most recently saved to, will display the file's name as the title of the main window, and will use that file when executing the Save option. When a New address book is initially created, its window will be titled "Untitled", and a Save operation will be converted to Save As ... - i.e. the user will be required to specify a file.

# User Interface

Because this is to be a "standard GUI" style application, some attention needs to be given to the user interface at this point. A user interface like the following might be adopted. Not shown in the screen shot are the options in the "File" menu (New, Open, Save, Save As ..., Print, and Quit), the "Sort" menu (Sort by Name and Sort by ZIP) and the "Search" menu (Find and Find Again). For the "Edit" and "Delete" buttons, the user must first select a person in the scrolling list of names, and then can click the appropriate button to edit/delete that person.



[Reference to case study](http://www.cs.gordon.edu/courses/cs112/Projects/Project3/index.html)