Design and implementation of an “Address Book” management system.

Note : This is an assignment you are encouraged to implement in your free time. It surely will give true clarity of OOP if done following the principle of the Object Orientation Programming. It is expected that once you have implemented this assignment, you need to revisit your implementation again and again. Use C++ as preferred language for this to develop. Attempt this problem in phase by phase manner.

Requirement :

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, zip, phone number and email id.

Phase 1 : 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 last name (with ties broken by first name if necessary), or by ZIP code (with ties broken by name if necessary). It must be possible to print out all the entries in the address book .

Phase 2 : It must be possible to create a new address book, to open a disk file containing an existing address book to close an address book, and to save an address book to a disk file, using standard New, Open, Close, Save and Save As ... File menu options. The program's File menu will also have a Quit option to allow closing all open address books and terminating the program.

Phase 3 : 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.

Phase 4 : The program will keep track of whether any changes have been made to an address book since it was last saved, and will offer the user the opportunity to save changes when an address book is closed either explicitly or as a result of choosing to create/open another or to quit the program.

Phase 5 : 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.