SIMPLE ADDRESS BOOK

http://www.cs.gordon.edu/courses/cs211/AddressBookExample/index.html

Requirement (1/3)

- A program to maintain an address book
- Each entry records a person's first and last names, address, city, state, zip, and phone number
- Functions:
  - add a new person to an address book
  - edit existing information about a person (except the person's name)
  - delete a person
  - sort entries in address book alphabetically by last name/zip code
  - print entries
  - create a new address book
  - open a disk file containing an existing address book
  - save (as) an address book to a disk file

Requirement (2/3)

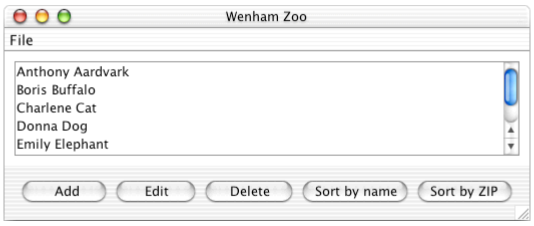
- The initial requirements call for the program to only be able to work with a single address book at a time; therefore, if the user chooses the New or Open menu option, any current address book will be closed before creating/opening a new on
- The program will keep track of whether any changes have 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

Requirement (3/3)

- 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

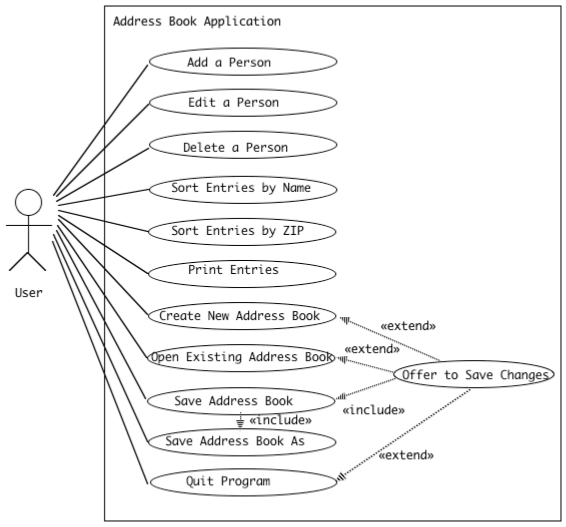
- Not shown in the screen shot is a File menu with New, Open, Close, Save, Save As ..., Print, and Quit options. For the "Edit" and "Delete" buttons, the user must first select a person in the scrolling list of names, and then can click the appropriate button to edit/delete that person



Anthony Aardvark
Boris Buffalo
Charlene Cat
Donna Dog
Emily Elephant

5

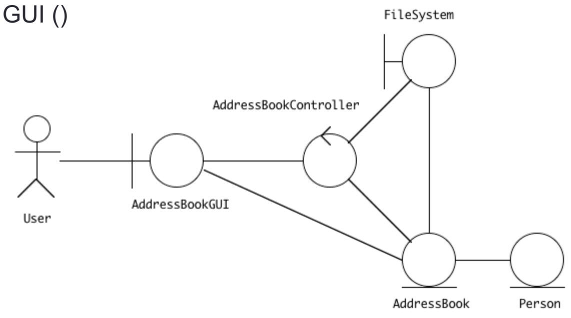
Use cases



6

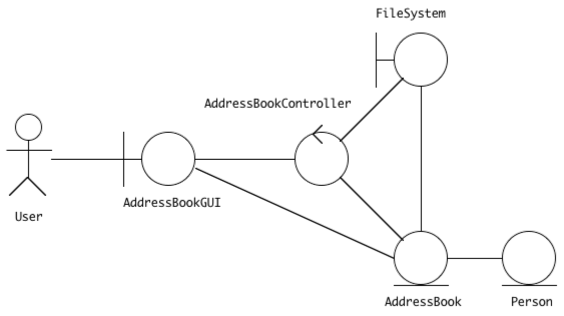
Analysis

- AddressBook: current address book that the program is working with
- Person: one of the people in the current address book.
- AddressBookGUI: the interface between the address book system and the human user.
- FileSystem: the interface between the address book system and the file system on disk.
- AddressBookController: carries out the use cases in response to user gestures on the GUI ()



7

Analysis



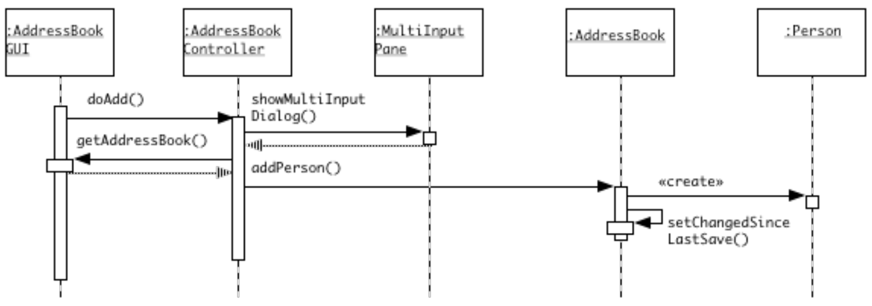
8

CRC Cards for class AddressBookController

Responsibilities	Collaborators
Allow the user to perform the Add a Person Use Case	AddressBook
Allow the user to perform the Edit a Person Use Case	AddressBook
Allow the user to perform the Delete a Person Use Case	AddressBook
Allow the user to perform the Sort Entries by Name Use Case	AddressBook
Allow the user to perform the Sort Entries by ZIP Use Case	AddressBook
Allow the user to perform the Create New Address Book Use Case	AddressBook
Allow the user to perform the Open Existing Address Book Use Case	FileSystem
Allow the user to perform the Save Address Book Use Case	AddressBook FileSystem
Allow the user to perform the Save Address Book As ... Use Case	FileSystem
Allow the user to perform the Print Entries Use Case	AddressBook
Perform the Offer to Save Changes Extension when needed by another Use Case	AddressBook

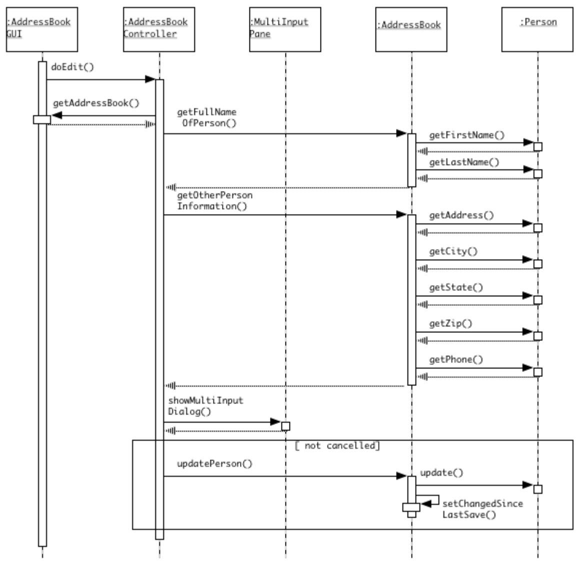
9

Add a Person Use Case Sequence Diagram



10

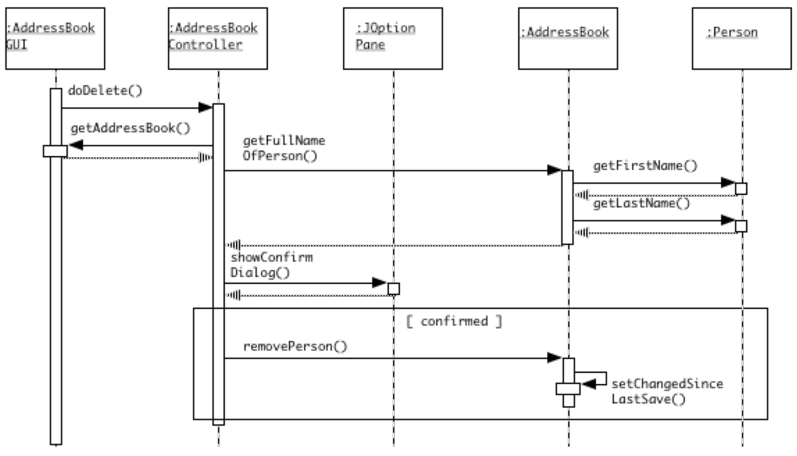
Edit a Person Use Case Sequence Diagram



If there is no selected name, none of the above is done; instead, an error is reported

11

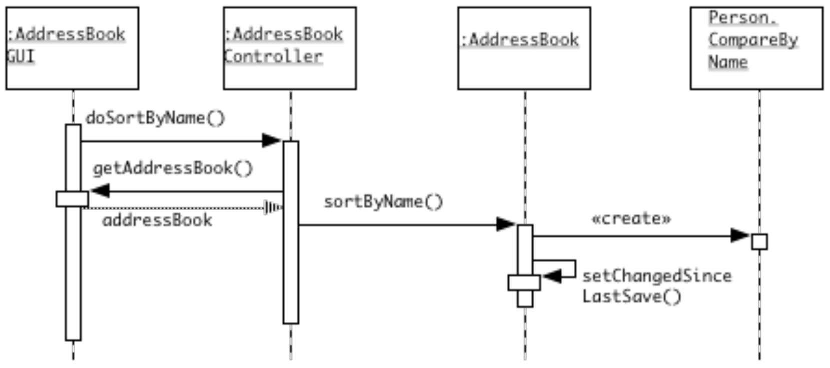
Delete a Person Use Case Sequence Diagram



If there is no selected name, none of the above is done; instead, an error is reported

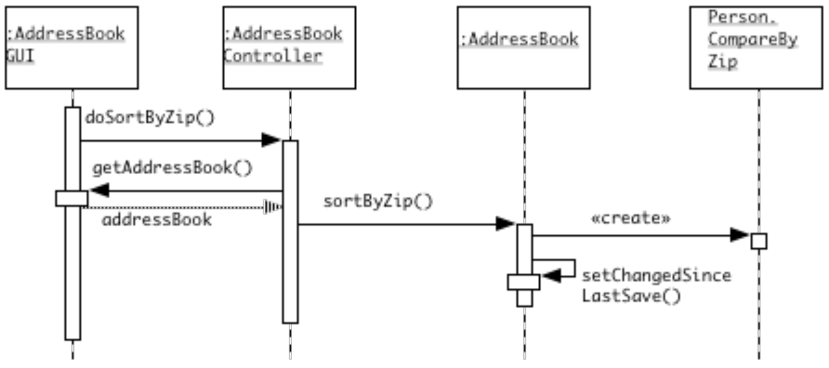
12

Sort Entries By Name Use Case Sequence Diagram



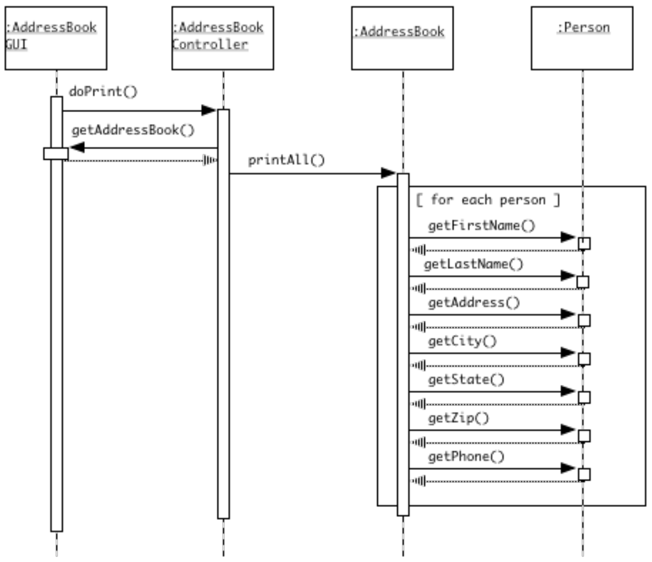
13

Sort Entries By Zip Use Case Sequence Diagram



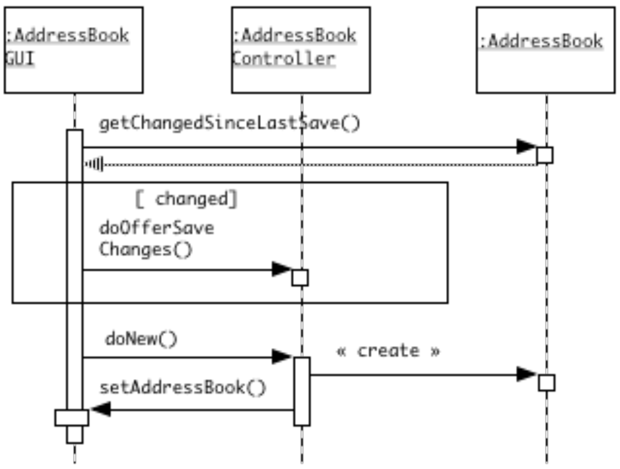
14

Print Entries Use Case Sequence Diagram



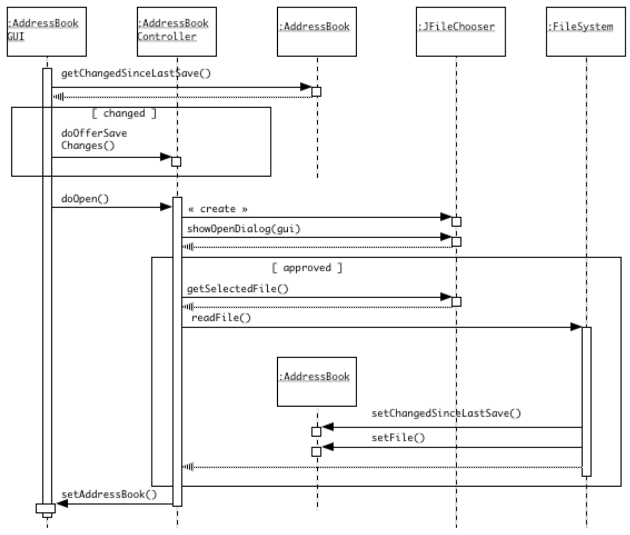
15

Create New Address Book Use Case Sequence Diagram



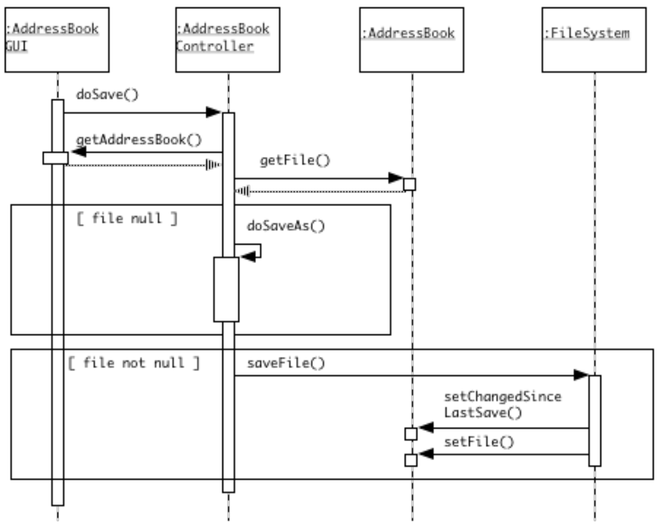
16

Open Existing Address Book Use Case Sequence Diagram



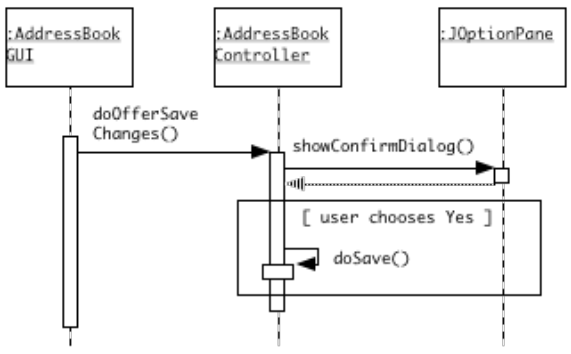
17

Save Address Book Use Case Sequence Diagram



18

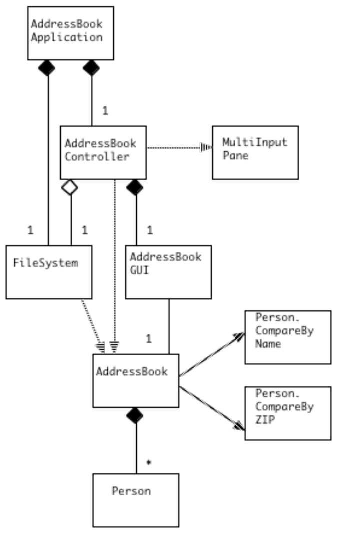
Offer To Save Changes Extension Use Case

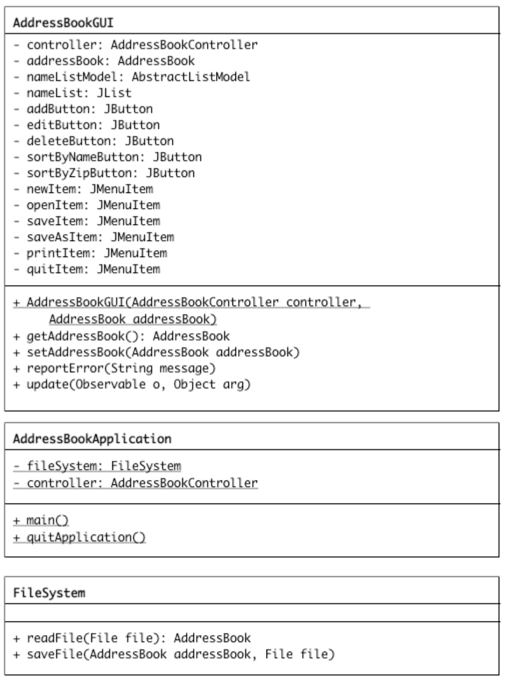


If the user chooses "Cancel" in the confirm dialog, an InterruptedException is thrown

19

Class Diagram

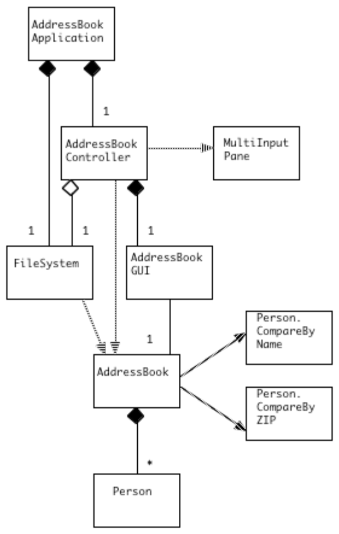




```
AddressBookGUI
- controller: AddressBookController
- addressBook: AddressBook
- nameListModel: AbstractListModel
- nameList: JList
- addButton: JButton
- editButton: JButton
- deleteButton: JButton
- sortByNameButton: JButton
- sortByZipButton: JButton
- newItem: JMenuItem
- openItem: JMenuItem
- saveItem: JMenuItem
- saveAsItem: JMenuItem
- printItem: JMenuItem
- quitItem: JMenuItem

+ AddressBookGUI(AddressBookController controller,
     AddressBook addressBook)
+ getAddressBook(): AddressBook
+ setAddressBook(AddressBook addressBook)
+ reportError(String message)
+ update(Observable o, Object arg)

AddressBookApplication
- fileSystem: FileSystem
- controller: AddressBookController

+ main()
+ quitApplication()

FileSystem

+ readFile(File file): AddressBook
+ saveFile(AddressBook addressBook, File file)
```

20

Class Diagram



```
AddressBook
- collection: Person [] or Vector
- count: int (only if an array is used for collection)
- file: File
- changedSinceLastSave: boolean

+ AddressBook()
+ getNumberOfPersons(): int
+ addPerson(String firstName, String lastName, String address,
            String city, String state, String zip, String phone)
+ getFullNameOfPerson(int index): String
+ getOtherPersonInformation(int index): String[]
+ updatePerson(int index, String address, String city,
               String state, String zip, String phone)
+ removePerson(int index)
+ sortByName()
+ sortByZip()
+ printAll()
+ getFile(): File
+ getTitle(): String
+ setFile(File file)
+ getChangedSinceLastSave(): boolean
+ setChangedSinceLastSave(boolean changedSinceLastSave)
```

```
Person
- firstName: String
- lastName: String
- address: String
- city: String
- state: String
- zip: String
- phone: String

+ Person(String firstName, StringlastName, String address,
         String city, String state, String zip, String phone)
+ getFirstName(): String
+ getLastName(): String
+ getAddress(): String
+ getCity(): String
+ getState(): String
+ getZip(): String
+ getPhone(): String
```

21