BACHELOR PROJECT

PROJECT TITLE

Address Book System(ABS)

NAME: Farzana Akter		ID: 151-35-1075	
NAME:		ID:	
Supervisor Name	:	Md. Alamgir Kabir	
Department:	:	Software Engineering	
Faculty of Science and Information Technology			

December 2017

Table of Contents

Chapter 1 ---------- *5*

1. Introduction ---------- *6*

1.1 About the System ---------- **6**

1.2 Purpose ---------- **6**

1.3 Scope ---------- **5**

1.4 Vision ---------- **7**

1.5 Why this system is necessary? ---------- **7**

1.6 Proposed Solution ---------- **7**

Chapter 2 ---------- *8*

2. System Analysis ---------- *9*

2.1 Use Case Model ---------- **9**

2.2 Actor Goal List ---------- **9**

2.3 Use Case Description (Brief) ---------- **10**

2.3.1 Manage Person ---------- 10

2.3.2 sort entries ---------- 11

2.3.3 Manage Address Book ---------- 12

2.4 System Sequence Diagrams ---------- Error! Bookmark not defined.

2.4.1 Manage Person ---------- **Error! Bookmark not defined.**

2.4.2 Sort Entries ---------- **Error! Bookmark not defined.**

2.4.3 Manage Address Book ---------- **Error! Bookmark not defined.**

2.5 Domain/Conceptual Model ---------- Error! Bookmark not defined.

2.6 Activity diagram ---------- Error! Bookmark not defined.

Chapter 3 ---------- *Error! Bookmark not defined.*

3. System Design ---------- *Error! Bookmark not defined.*

3.1 Sequence Diagrams ---------- **20**

3.1.1 Manage the System ---------- **Error! Bookmark not defined.**

3.2 Class Diagram ---------- **20**

Chapter 4 ---------- ***21***

4. Implementation ---------- *Error! Bookmark not defined.*

4.1 Tools &Technologies ---------- Error! Bookmark not defined.

4.2 Project Link ---------- **22**

Table of Figure

Figure 1: Use case diagram of Address Book..... 9
Figure 2: System Sequence Diagram(Manage Address Book)**Error! Bookmark not defined.**
Figure 3: System Sequence Diagram(Manage Person) .....**Error! Bookmark not defined.**
Figure 4: System Sequence Diagram(Sort Entries) .....**Error! Bookmark not defined.**
Figure 5: Domain or Conceptual Diagram of Address Book ..**Error! Bookmark not defined.**
Figure 6: Activity Diagram.....**Error! Bookmark not defined.**
Figure 7: Sequence Diagram of Address Book..... 19
Figure 8: Class diagram 20

Chapter 1

Introduction

1.Introduction:

The name of my system is “Address Book”. The system 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.

1.1. About The System:

1.Add New Person. (First and last name, address, city, state, zip, and phone number).

2.Delete Person.

3. Edit existing Person. (Information except the person's name).

4.Sort the Entries (by first and last name or ZIP code).

1.2. Purpose:

Now the age of modern Science and people wants to lead an easy life. They use all the modern and scientific things in their daily life and don’t want to even use pen and paper without basic needs because all the information’s are now software basis and it is easy to keep, remember and supply in various needed place like job interview sector, passport office or something else which can be very needed and important. But it is mandatory for all people to keep personal information safe and secure. An address book is a **book** like a **database** used for keep people’s personal information. Each entry usually consists of a few standard **fields** (for example: first name, last name, **address**, city, state, zip, **mobile phone** number). Most such systems store the details in alphabetical order of people's names or zip code. If any person doesn’t want keep his/her information he/she can delete the system easily. It is easy to use and secure also.

1.3. Scope:

It's an Address Book system. Here the system will help user to manage the person's information and can add them in the digital book application for addressing. Here will be no admin and people will create account for themselves as a new profile and also can edit, update, delete a person.

1.4. Vision

When a user wants to use this system he or she will surely have benefited. The vision of this system is to add persons as catalog entry with every possible information so that user can enlist all the contacts perfectly to find any person that user added his or her system to reducing the workload and less time consuming.

1.5. Why this system is necessary?

ABS is the system designed so that users can add a huge number of person's details so that he or she can contact them easily. This system is needed to find a new person whose are connected with the user. Another positive side is user can find anytime any person that are listed in the system by the user.

1.6. Proposed Solution

The ABS system there will be a process where user can edit and update a profile suggestion by the peoples. ABS make the top reviewed persons list in ascending order. The user just review and the system will automatically enroll them by following the reviewed order.

Chapter 2

System Analysis

1. System Analysis

1.1. Use Case Model

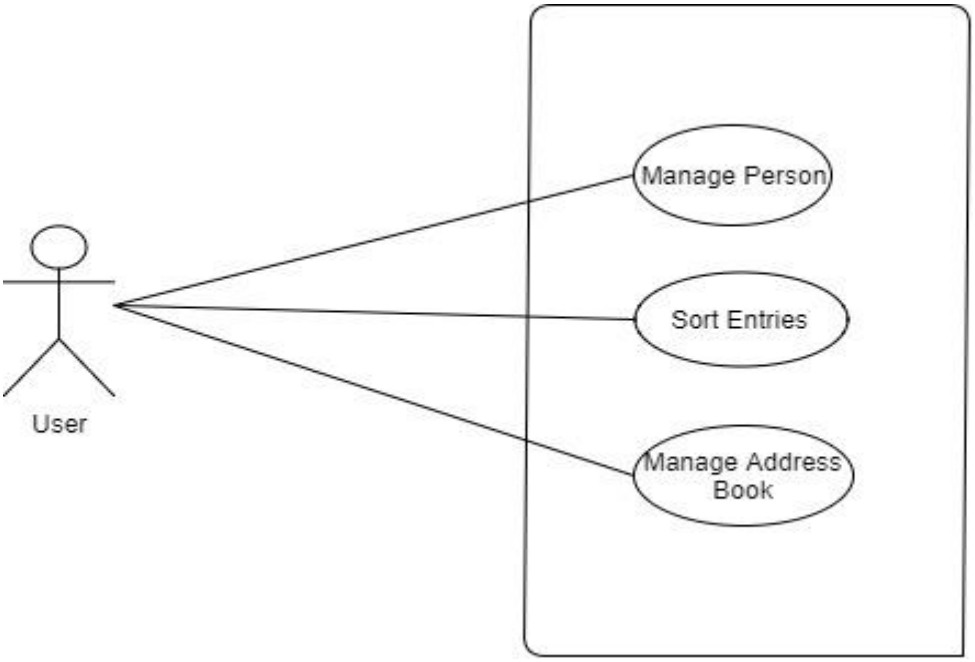

Figure 1: Use Case diagram of address book system

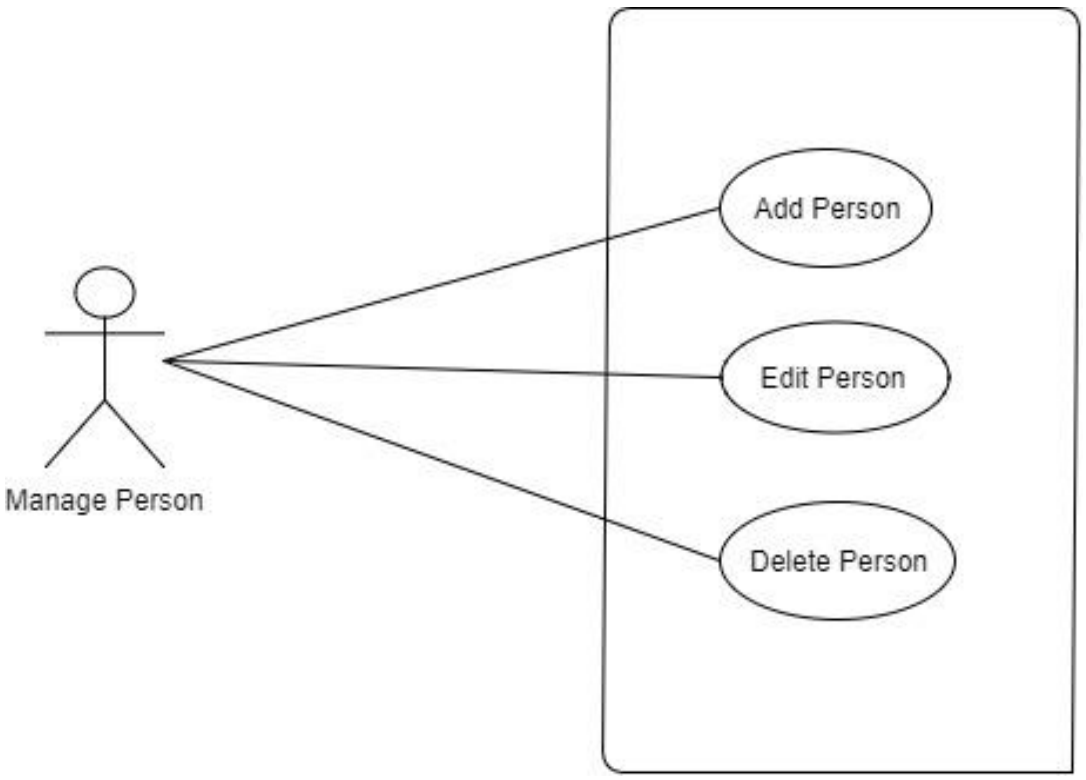

Figure 1.1.Use Case(Mange Person)

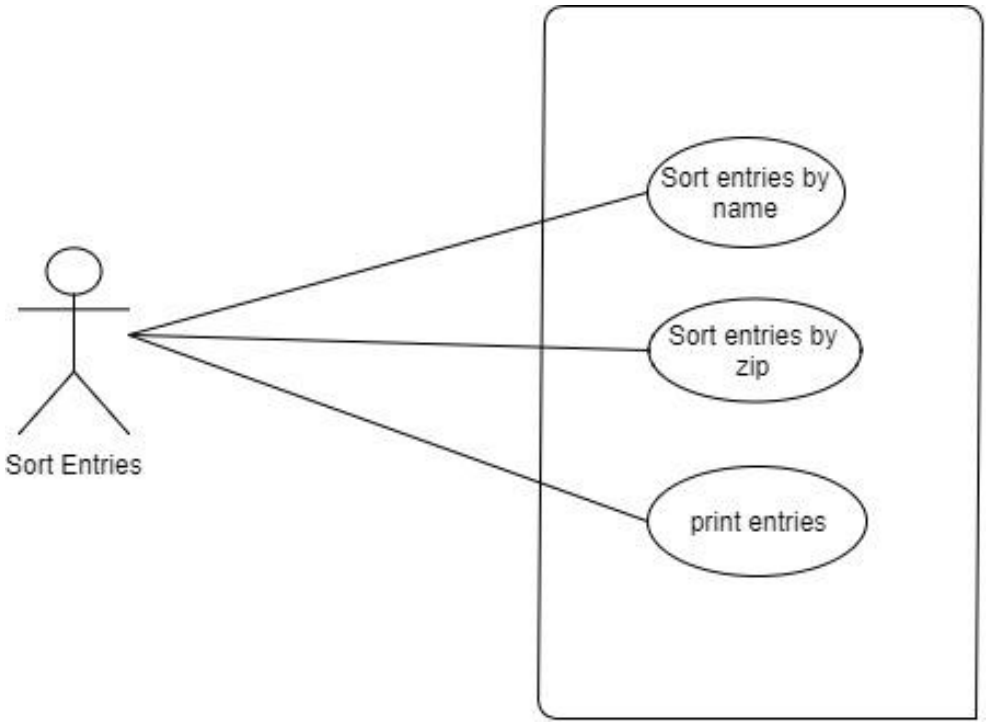

Figure1.2.Use case(Sort Entries)

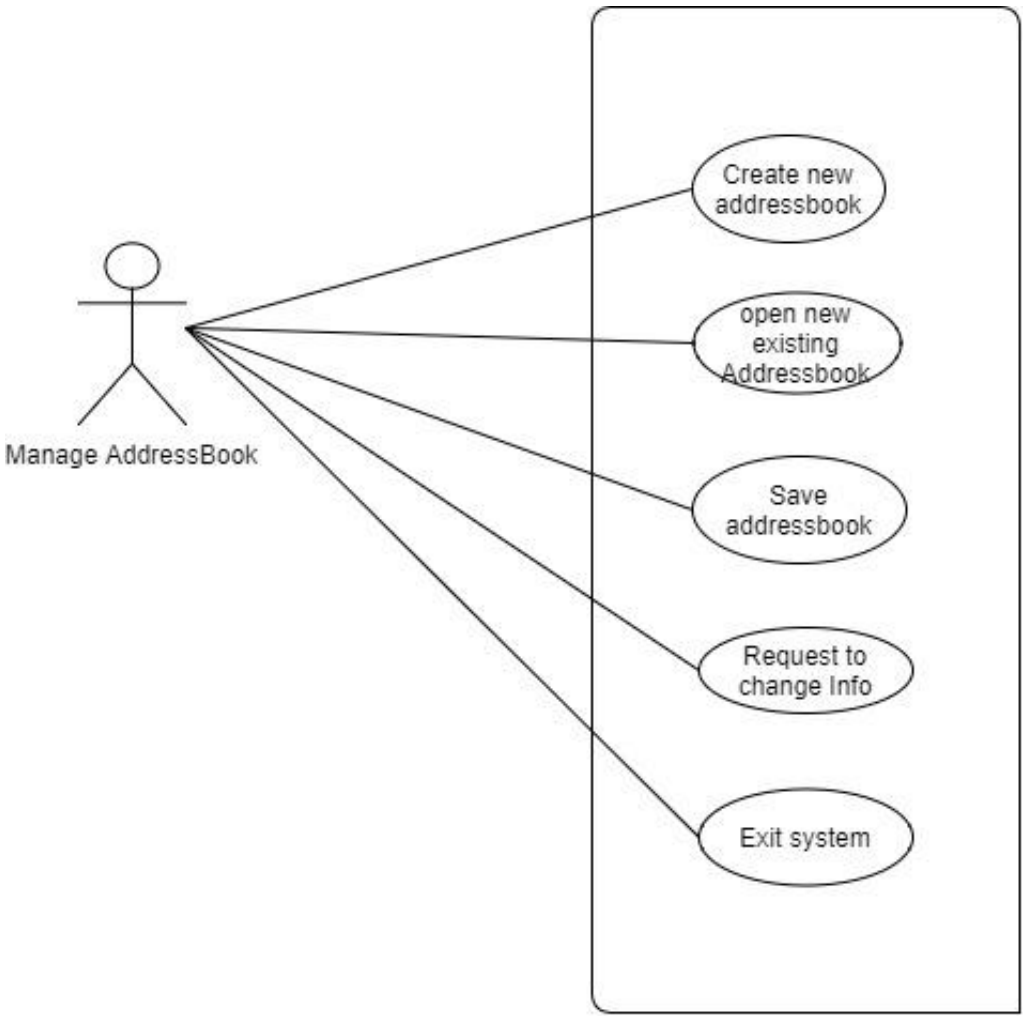

Figure1.3.Use case(Manage Address Book)

1.1. Actor Goal List

Actor	Goal
1. User or Admin	1. Add Person 2. Edit Person 3. Delete Person 4. Sort Entries by last name 5. Sort Entries by zip 6. Print Entries 7. Request to Change Info 8. Create New Address Book 9. Open New Existing Address Book 10. Save Address Book 11. Exit System

1.2. Use Case Description

1.2.1. Manage Persons

Use Case Name:	Manage Person	
Scenario :	Add person, Edit person, Delete Person in an address book	
Brief Description:	When a person wants to add in address book by clicking add button he/she can easily add that system. Then he/she can edit his/her info by edit button. If anyone don't want to stay that system then he /she can remove her address book profile by using delete button .	
Actor:	User	
Precondition:	User must exist and each recording about user must hold a person's first and last name. Anyone edit his/her info but his/her name.	
Post condition:	Address book will not save as long as the user click the save or save as button.	
Flow of events:	Actor	System
	1. Request to add profile in address book. 2. If all the information place is fill up then user can be added. 3. if the user wants to edit name then edit information properly. 4. Request to delete profile from address book.	1.1. Add person as a user properly. 2.1. Save all the information correctly. 3.1. Edit information without name. 4.1. Delete the profile if any user doesn't want to keep his/her profile in address book.
Exception Condition:	2.1. If all information places are not filled up, then system will not work.	

1.3.2.Sort Entries

<table>
<tr><td>Use Case Name:</td><td colspan="2">Sort Entries</td></tr>
<tr><td>Scenario :</td><td colspan="2">Sort entries</td></tr>
<tr><td>Brief Description:</td><td colspan="2">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).</td></tr>
<tr><td>Actor:</td><td colspan="2">User</td></tr>
<tr><td>Precondition:</td><td colspan="2">Person must have use his/her first and last name. If there is needed zip code then it is also needed to fill up that place.</td></tr>
<tr><td>Post condition:</td><td colspan="2">Address book will not save as long as the user click the save or save as button.</td></tr>
<tr><td rowspan="2">Flow of events:</td><td>Actor</td><td>System</td></tr>
<tr><td>1. Person have to request for adding his/her name according to first and last name.
2. request to add zip code
3. request to make entries sort by name.
4. request to make entries sort by zip code.</td><td>1.2. Add person as his /her name.

2.1. add zip code properly.
3.1. make entries sort by name.

4.1. sort entries by zip code.</td></tr>
<tr><td>Exception Condition:</td><td colspan="2">2.1. If system can’t add name and zip code properly or can’t make entries sort that would be problem.</td></tr>
</table>

1.3.3. Manage Address Book

Use Case Name:	Manage address book	
Scenario :	Create or open new address book and then save or quit.	
Brief Description:	It must be possible to create a new address book, to open a file containing an existing address book to close an address book, and to save an address book. The program's File menu will also have a Quit option to allow closing all open address books and terminating the program.	
Actor:	User	
Precondition:	the current address book contents must have been changed since the last successful New, Open, Save, or Save As operation was done, the Offer to Save Changes extension is executed. And also user can't cancel the operation.	
Post condition:	User have to click save button otherwise it'll not be saved.	
Flow of events:	Actor	System
	1. Person have to request to add his/her name newly. 2. Request to add new information in a new profile or exist the previous one.	1.1 Add person as new user 2.1. add all new information in a new profile or kept the previous one.
Exception Condition:	2.1. If system can't add new information, then it would be problematic.	

1.3. System Sequence Diagram

1.3.1. Manage Persons

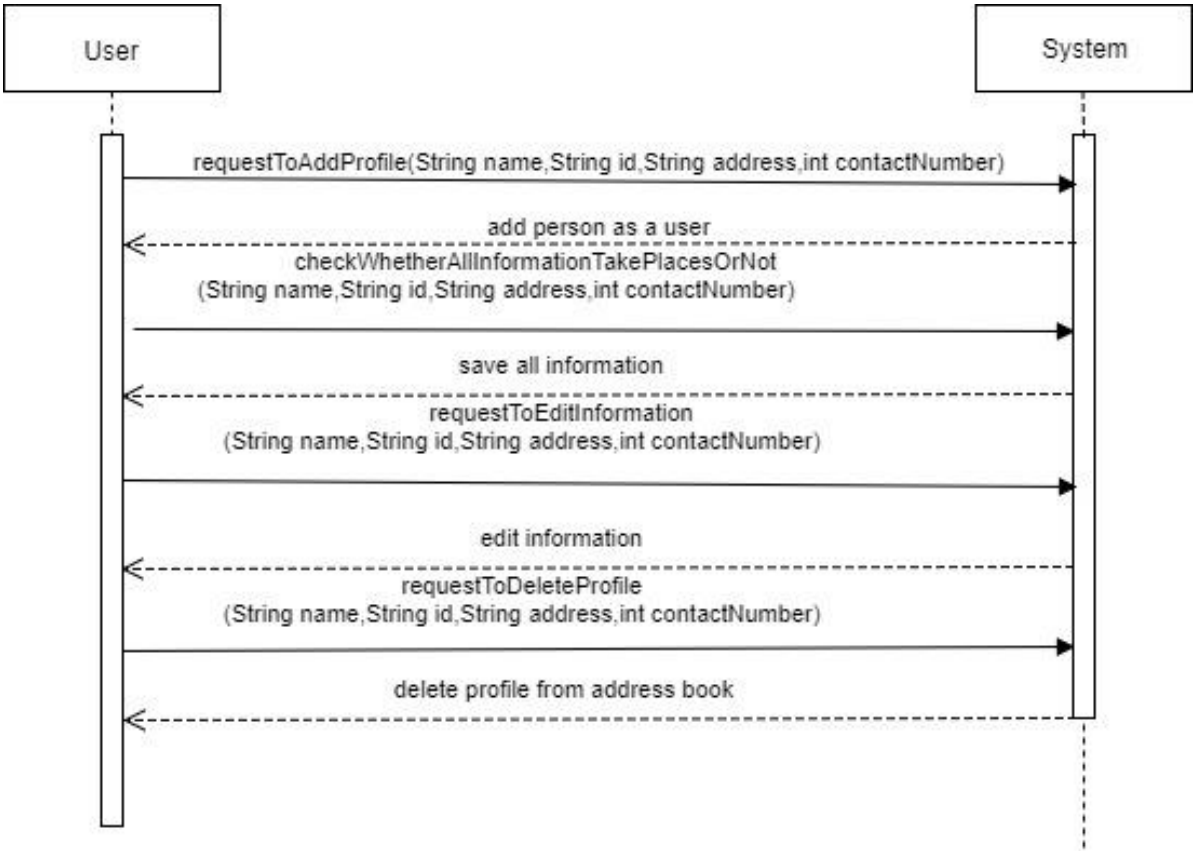

Figure 2: System Sequence Diagram (Manage Person)

1.3.2. Sort Entries

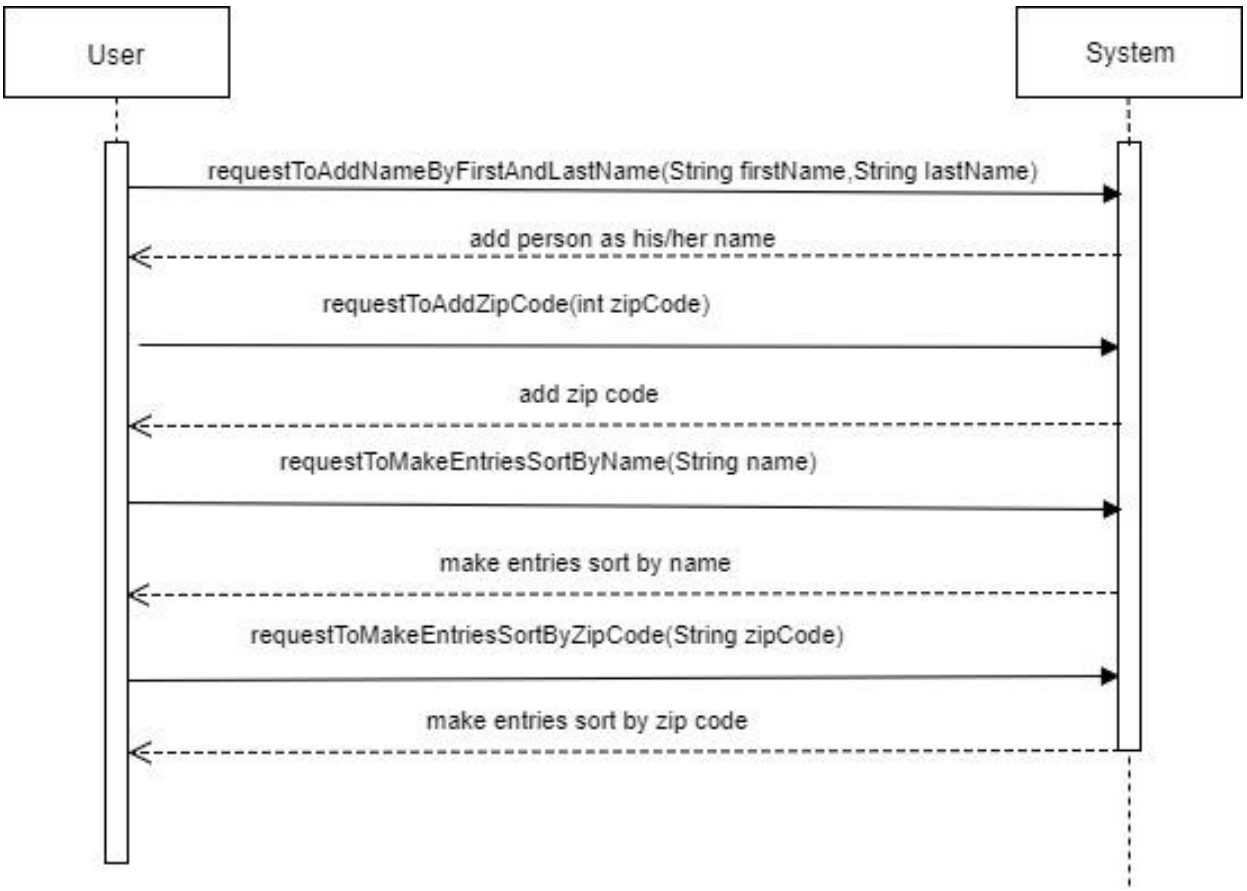

Figure 3: System Sequence Diagram (Sort Entries)

1.3.3. manage Address Book

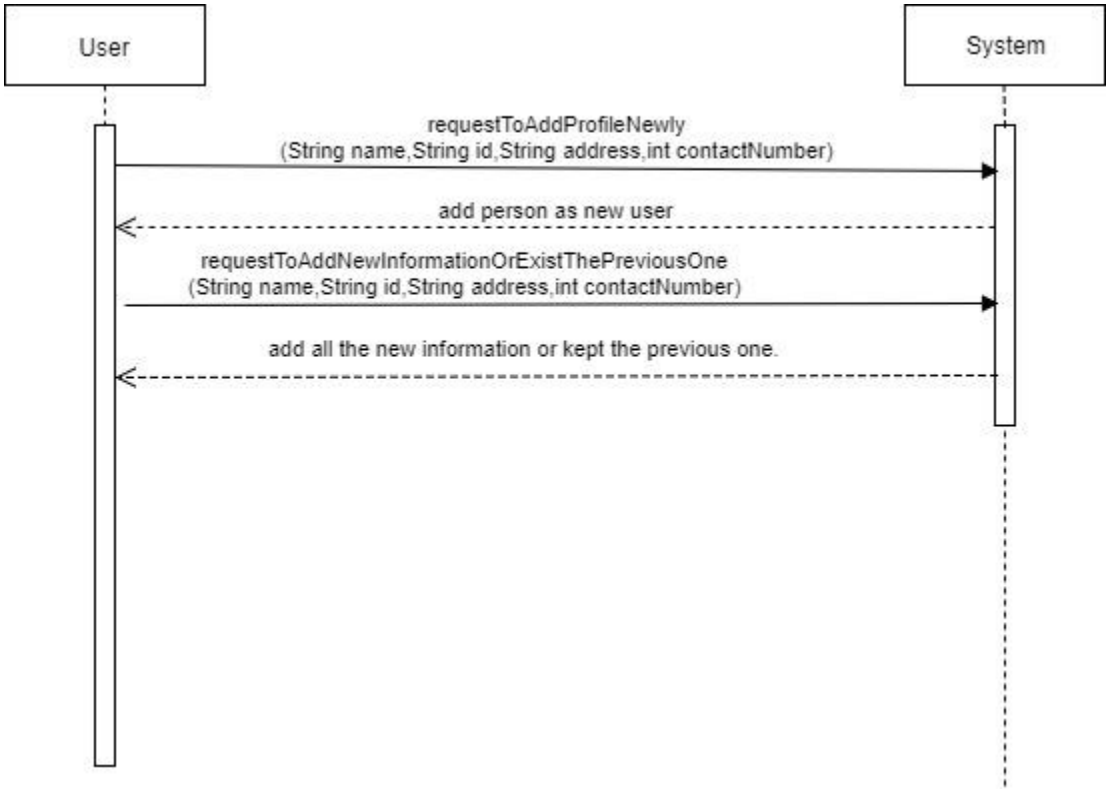

Figure 4: System Sequence Diagram(Manage Address Book)

1.4. Domain or Conceptual model diagram

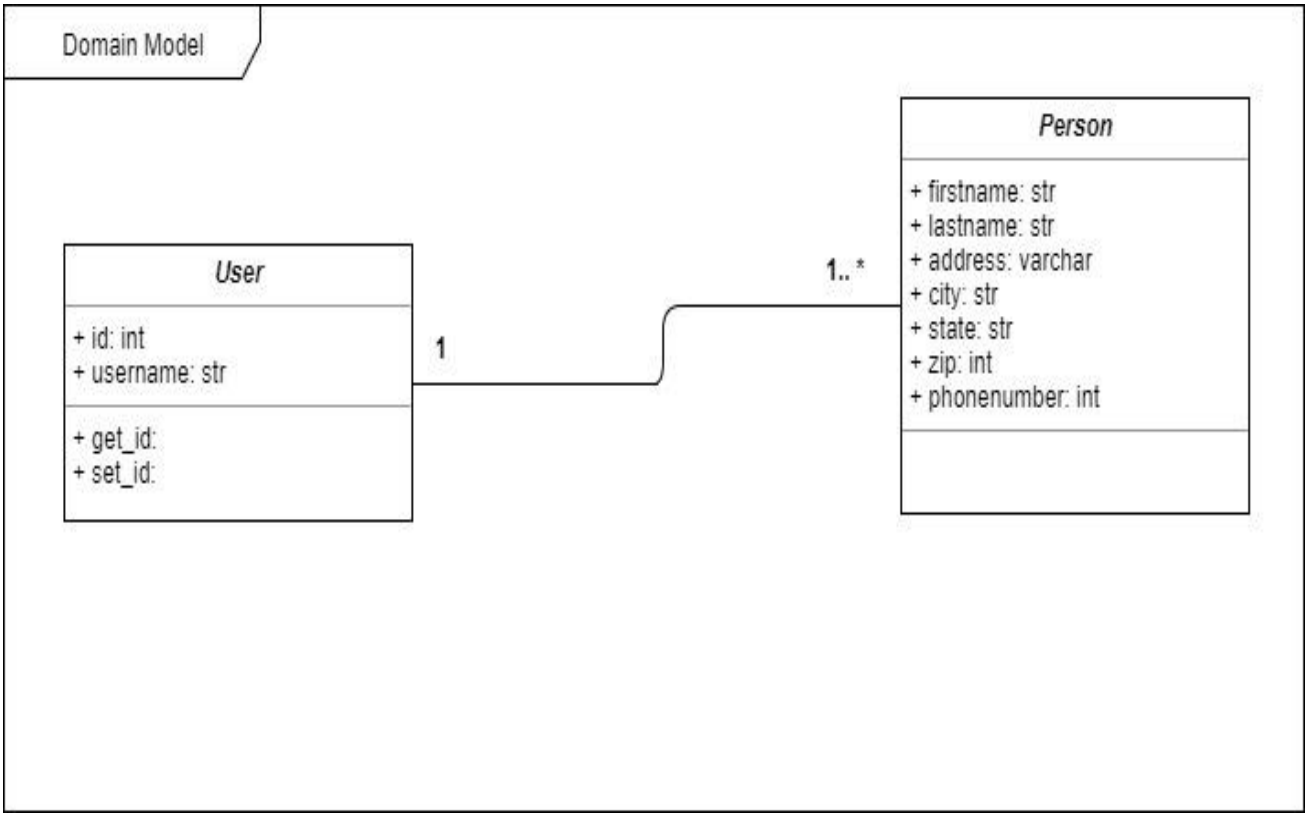

Figure 5: Domain or Conceptual diagram of address book system.

1.5. Activity Diagram

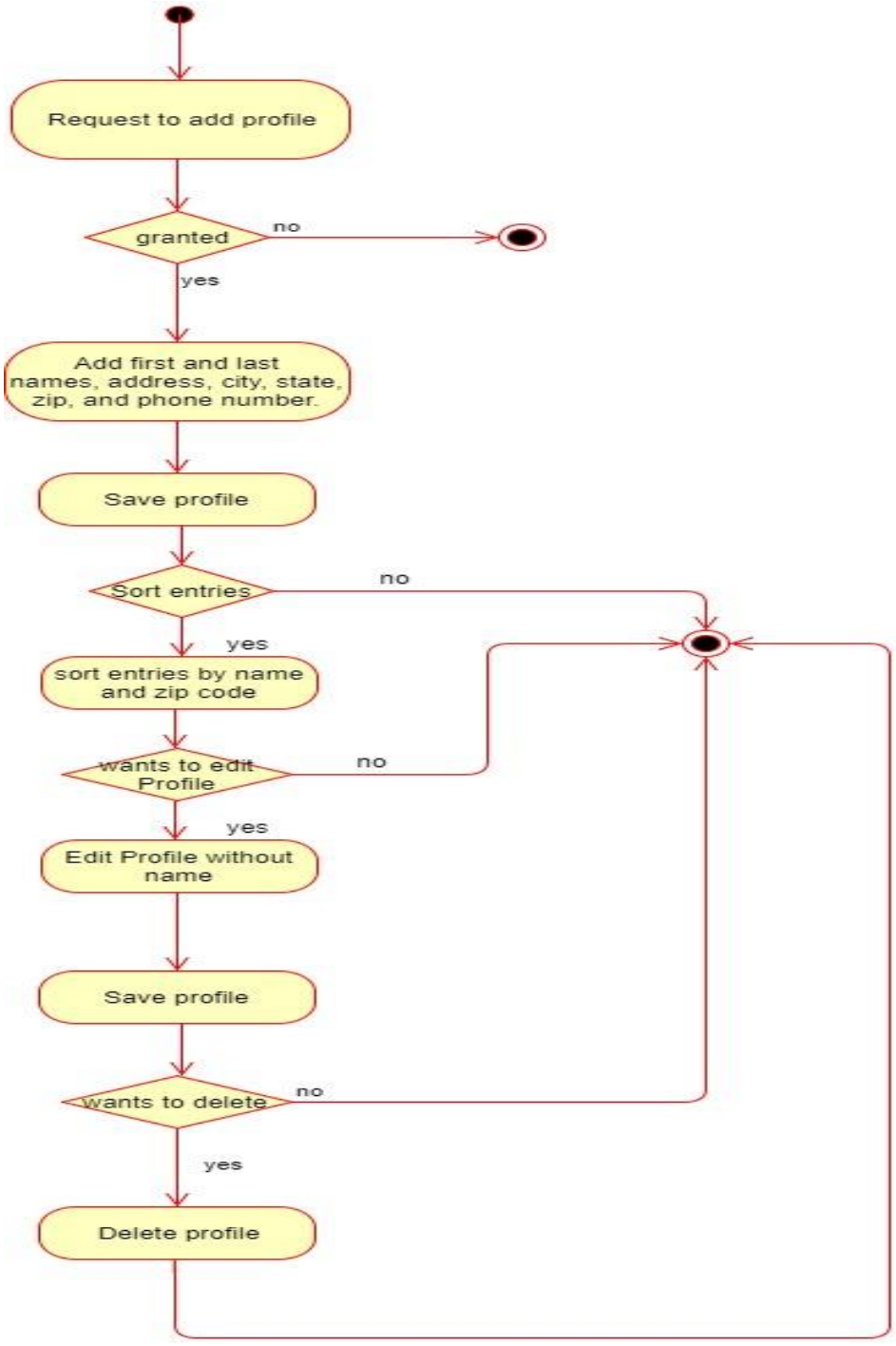

Figure 6: Activity diagram of address book system.

Chapter 3

System Design

2. System Design

Design is a process that uses the product of analysis to produce a specification for implementing a system. Design is the logical description of how a system will work.

Design emphasizes a conceptual solution that fulfills the requirements, rather than its implementation. For example, a description of a database schema and software objects. Design ideas often exclude low-level or "obvious" details obvious to the intended consumers. Ultimately, designs can be implemented, and the implementation (such as code) expresses the true and complete realized design. The term is best qualified, as in object-oriented design or database design.

2.1. Sequence Diagram

The UML includes interaction diagrams to illustrate how objects interact via messages. They are used for dynamic object modeling.

2.1.1. Manage the System

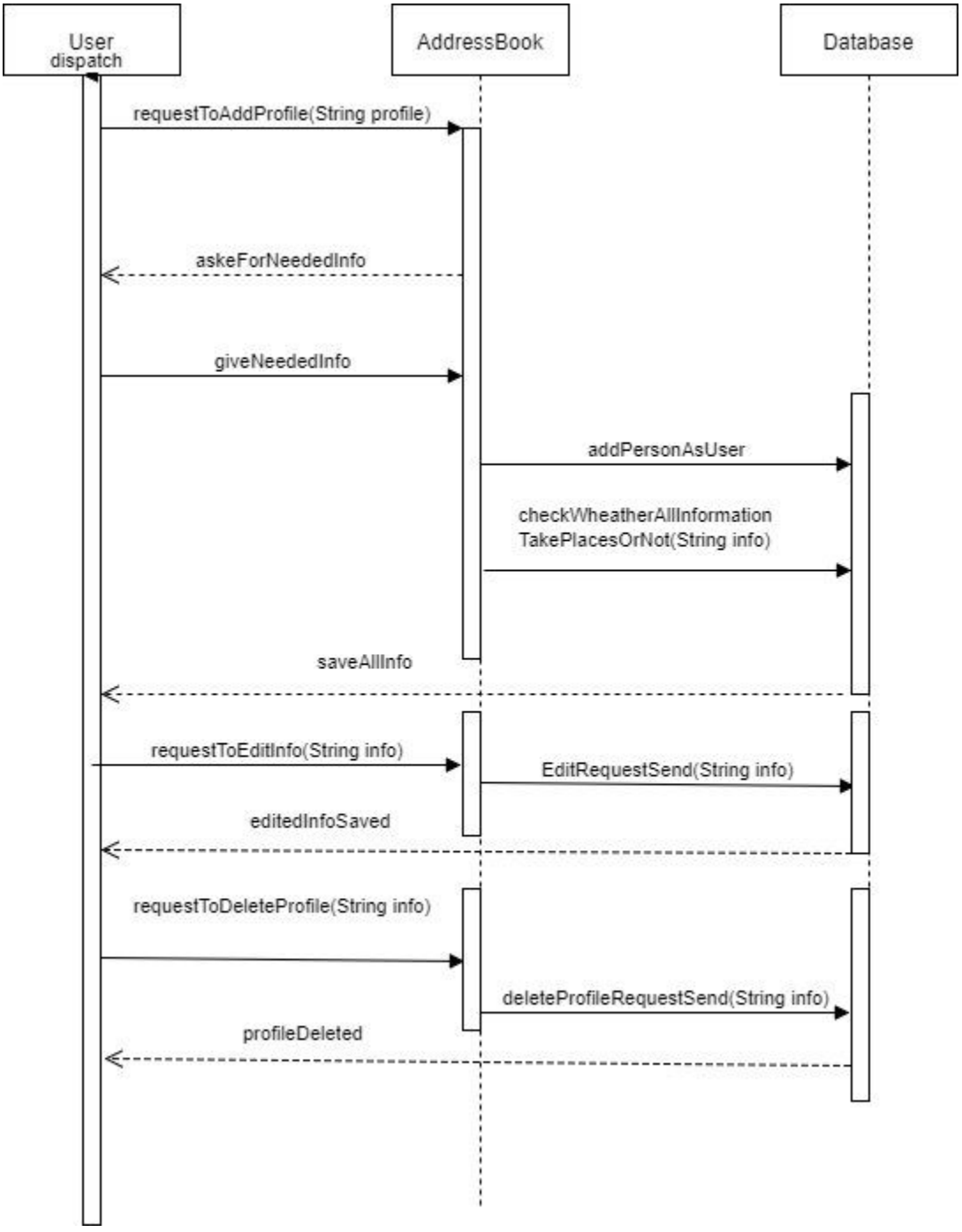

Figure 7: Sequence Diagram(Manage person).

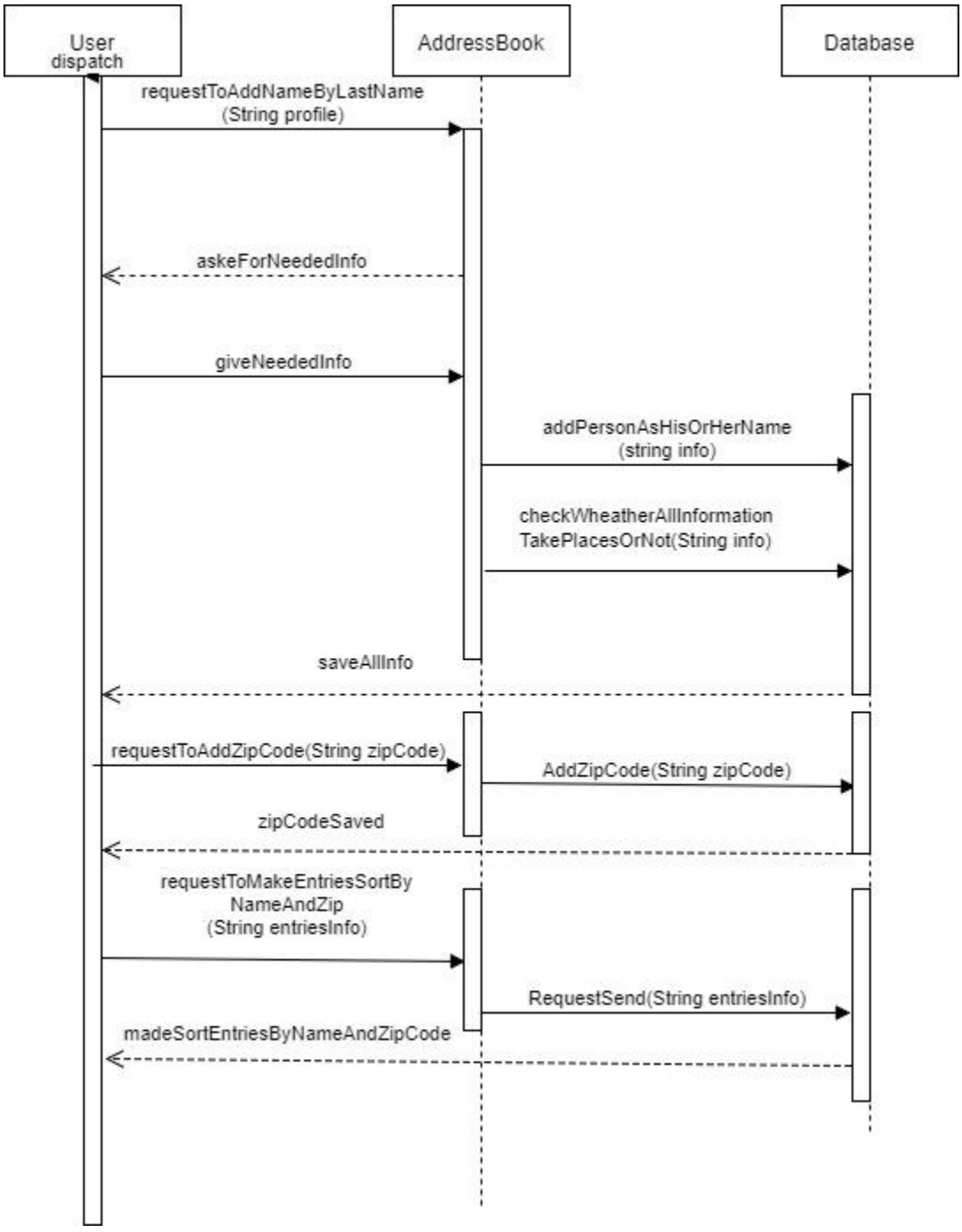

Figure 8: Sequence Diagram (Manage Entries)

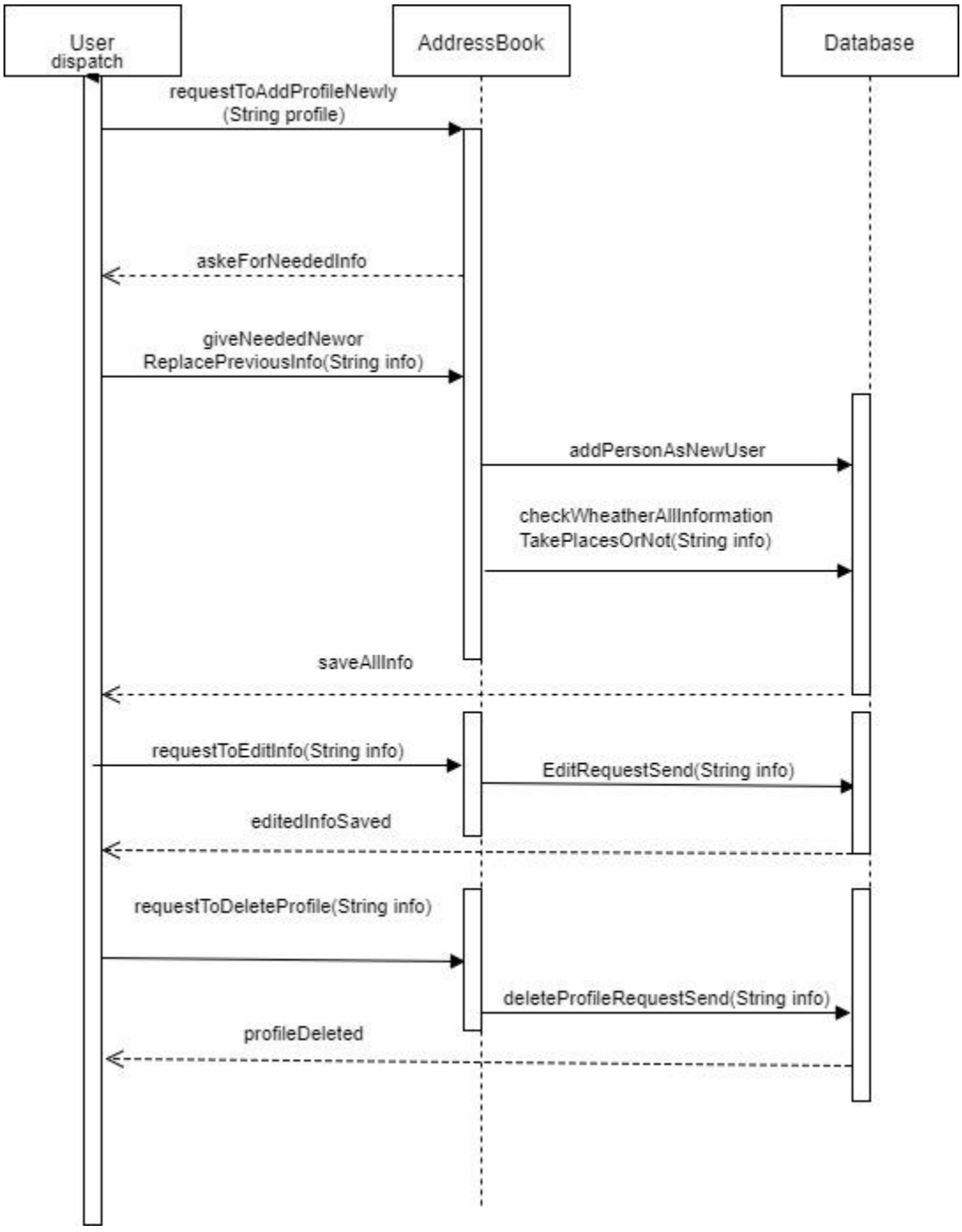

Figure 9: Sequence Diagram: Manage Address Book)

2.2. Class Diagram

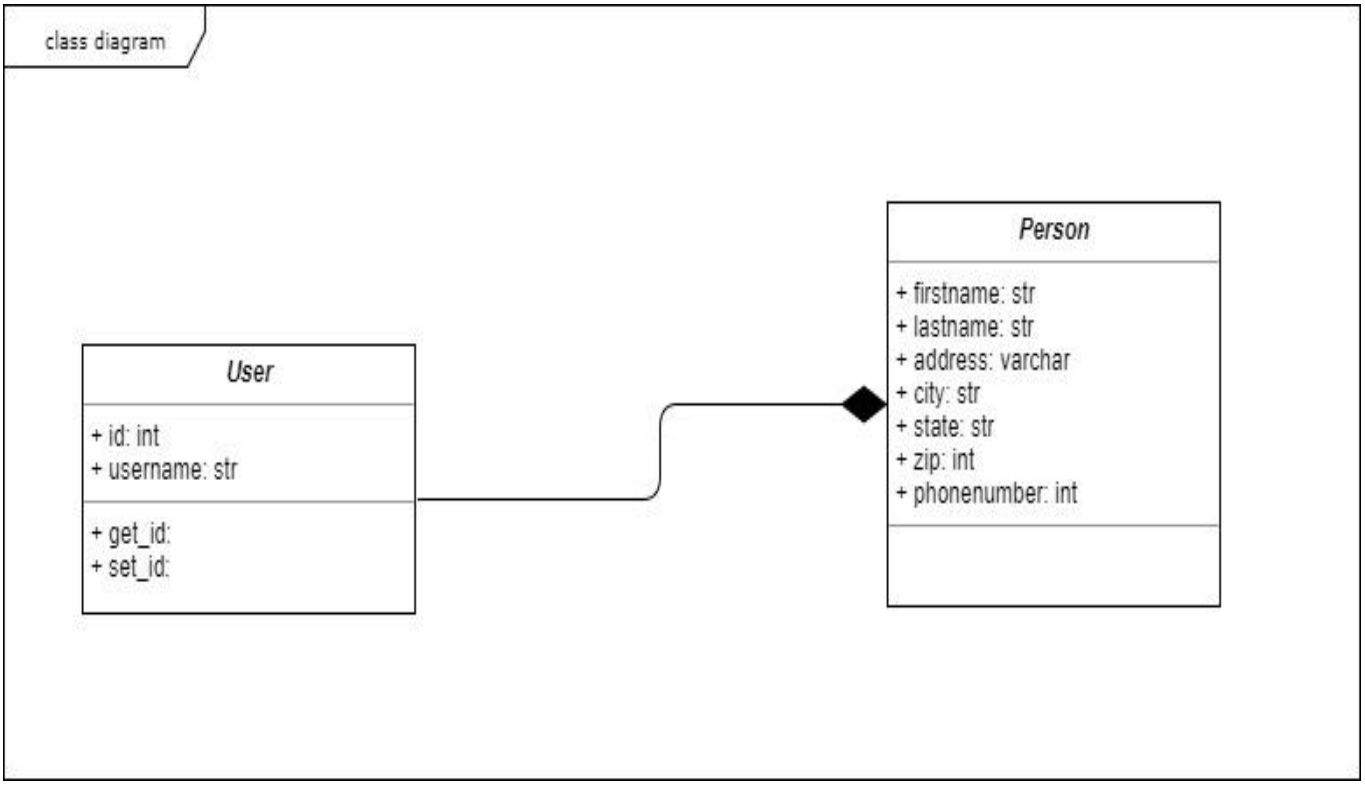

Figure 8: Class diagram of address book system.

Chapter 4

Implementation

3. Implementation

Implementation perspective describes software implementations in a particular technology (such as php). Implementation means programming and building the system, not deploying it.

In the implementation phase, the developer builds the components either from scratch or by composition given the architecture document from the design phase and the requirement document from the analysis phase. The architecture document should give guidance.

3.1. Tools and Technologies

- Following are the tools and technologies used in development of this project:
- PHP
- My SQL
- Atom
- XAMPP
- HTML5, CSS, JavaScript, J-query, Twitter bootstrap

3.2. Project Link

https://github.com/FarzanaAkhi/Address-Book-System