Daffodil International University

Documentation on

Object oriented Software Development(Project 5)

SuperVisor

Md. Alamgir Kabir

Lecturer, Daffodil International University

Submitted By

Md. Ashik Iqbal

ID – 142-35-713

Section- B

Submission date- 09/12/2017

Department of software Engineering,

Daffodil International University

Table of Contents

Chapter-01

Chapter-02

Chapter-03

1. Introduction

Address Book is a contact management software, wherewith we can note the addresses, birthdays, phone numbers, mobile numbers, company name, country city, website address and email addresses of our contacts.
We can multiple users data like the company name, register number, bank account number, tax number etc.

1.1 Purpose

Over fifty percent of newly created small business have gone belly up within a few years, and most of those that survive are just barely doing so. They limp along, year after year, making just enough of a margin to stay in business but never managing to generate enough capital for the improvements, upgrades, or expansions that would allow them to move to the next level. But there are, of course, also new small business that seem to be super changed, they may begin with one or two people working out of a home office, garage, kitchen, or even a local coffee shop, but the next thing we know they're employing dozens or even hundreds of people and their offices or products are everywhere.

There are something that makes difference. This is obviously a huge question and there are plenty of small business gurus who make a great deal of money selling all kinds of answers. But there is one part of the answers that is simple and clear. In order to succeed, a small business must have an effective system of Address Book.

1.2 Scope

Address book has become a foundation stone in any organization, be it in the form of a knowledge management framework, database management, employee file and folder management, digital asset management, record management or nay other form. Future of the content management lies in being handled with the cloud approach, be it private or public, for any infrastructure. We will elaborate on essential features of an industry competent address book. Cloud servers can be used for better content delivery and workflow. We propose a new model through which the usage of an address book can be enhanced to knowledge management through a private/public cloud. A study of 13 different address book are covered in the project. These features will be used to conclude the best address book system from a user and developers.

1.3 Vision

Our vision is to deliver best user experience while using the address book software, to build a place where people can come to save their identity and feel safe. In addition, to provide experiences for our users and partners, across all of their interactions with Microsoft, that they value and recognize, and enable them to realize their full potential

1.4 Why Address book system is needed?

Without any form of contact address book, important contact data can get lost in many areas of anyone's business. Either the data is lost deep down in an employee's email inbox which you can not access, or the data is on business card that can easily get lost in the office. No address book means spending valuable time searching for addresses, telephone numbers, and business cards in order to react to customer needs. In the worst case you may customers due to inefficient customer care. With a professional address book software you no longer have to rely on e-mails and business cards, but can collect and maintain all necessary data in one glass.

1.5 Propose solution

When using address book software, our business's some problems might be solved , such as below,

- collecting, storing, editing/updating, and easily finding centrally stored customer data(name, address, telephone number, e-mail address, etc).
- keeping track of all appointments and documents and documents. As all data is saved centrally, editing data can be done by anyone in our business without creating duplicate files. This makes sharing and synchronising information very easy,

all vital data is always up to date.

-email integration. This allows you to easily transfer data from emails to your software.

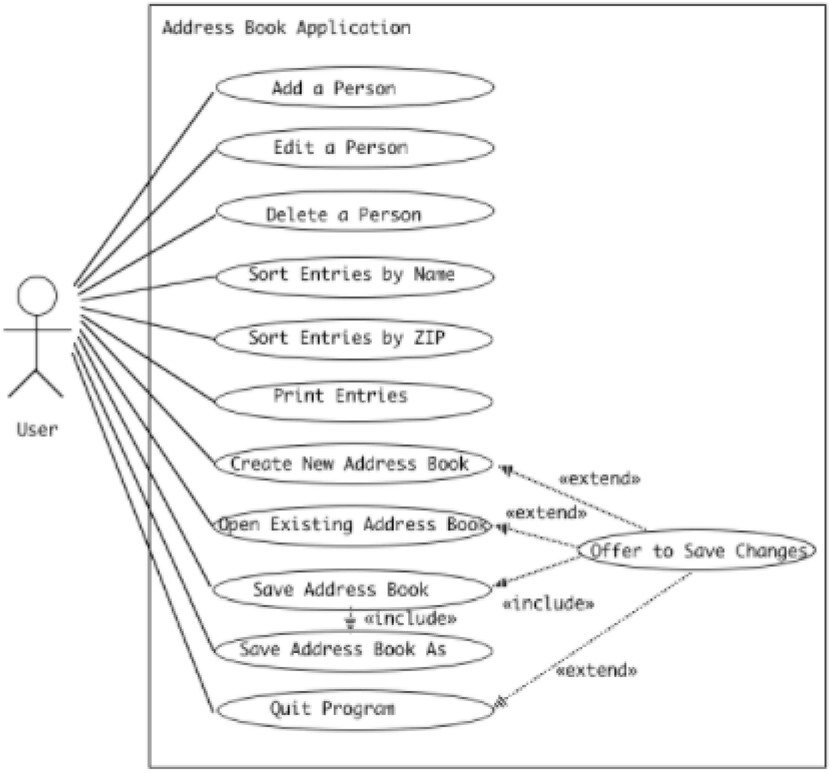
2. System Analysis

Every system has some goals. In order to complete the goals, we need to design the goals. There are some ways to design the goals. Use case modeling is one of them. A use case is a methodology used in system analysis to identity, clarify, and organize system requirements. The case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. It consists of a group of elements(for example, classes and interfaces) that can be used together in a way that will have an effect larger than the sum of separate elements.

2.1 Actor goal list

Actor	Goal
User	- Add a person -Edit a person -Delete a person -Sort Entries by name -Print entries -create new address book -save address book- - quit program

2.2 Use case model

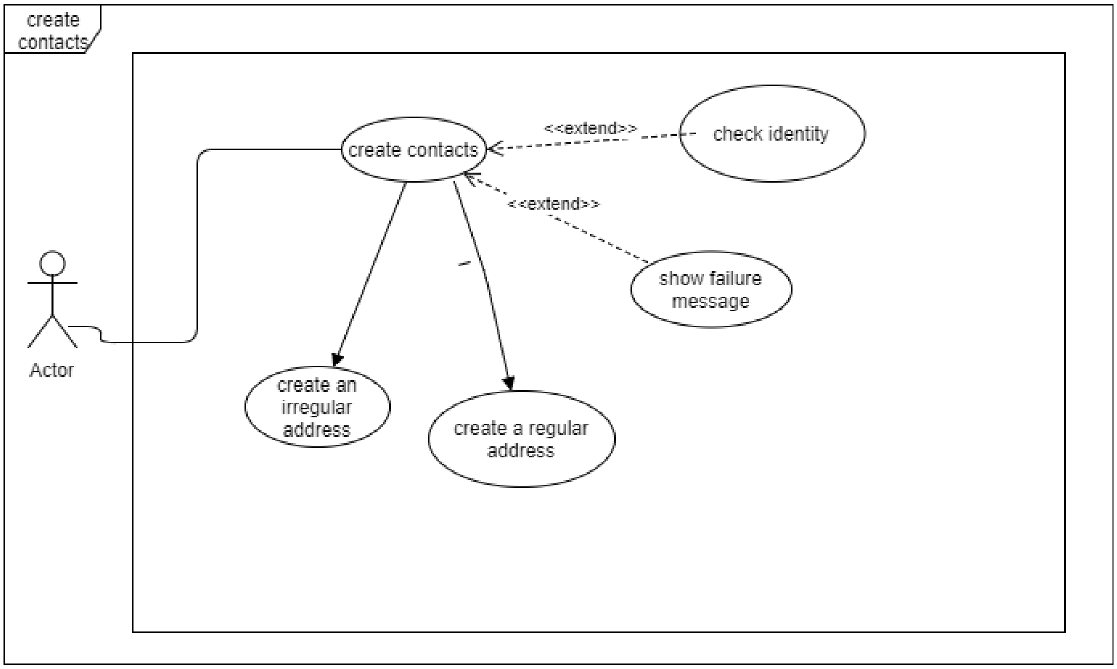


2.3 Use case description (brief)

Brief description of every use cases will be given here

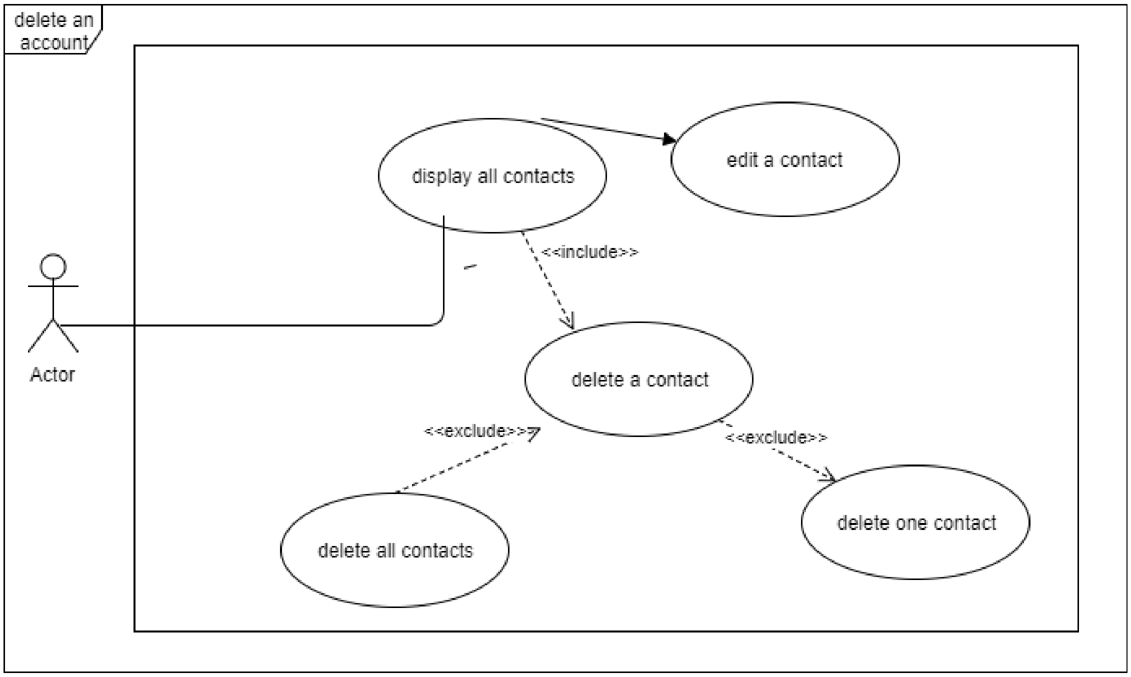
2.3.1 Create a new address

When a new request arrives, then user add the request on the system.



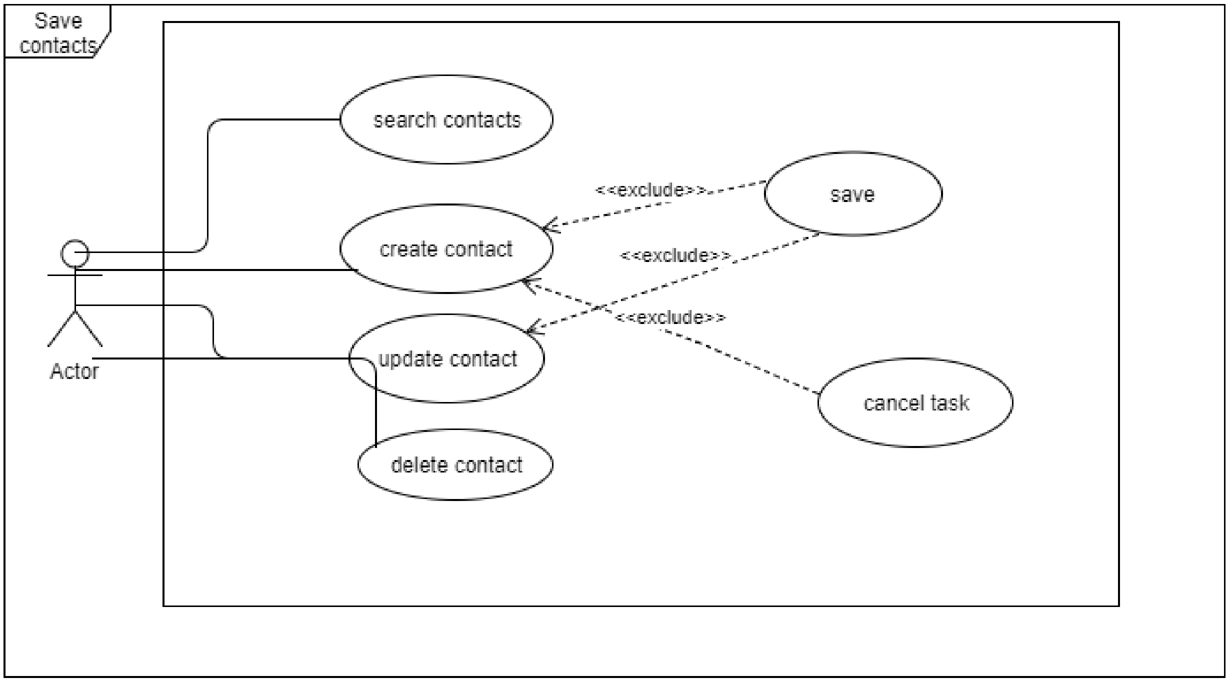
2.3.2 Delete an account

User can delete an account permanently by this feature.



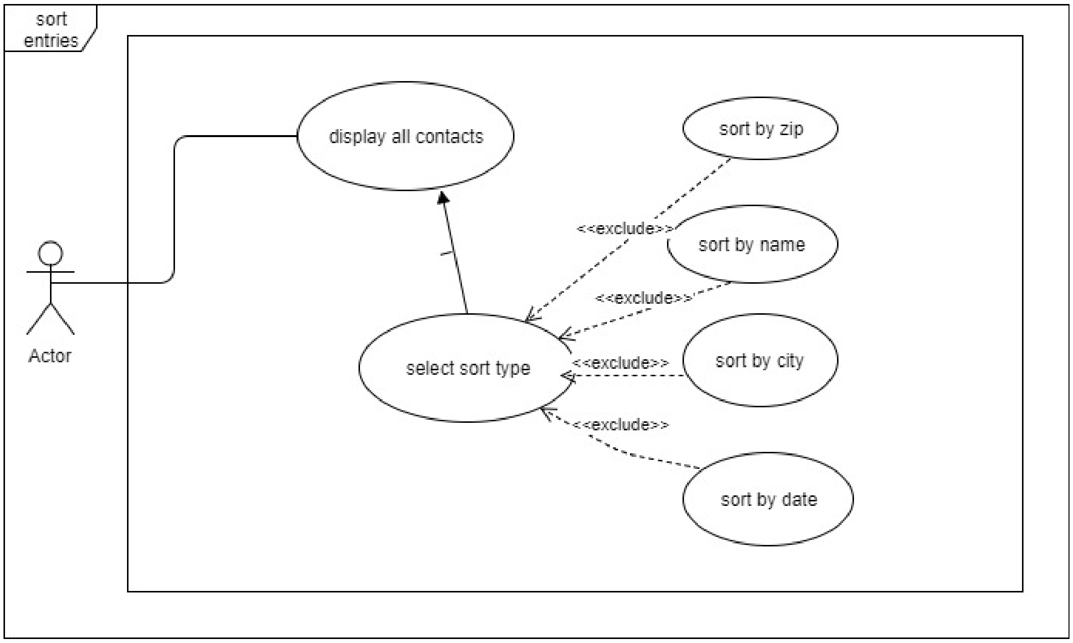
2.3.3 Save contacts

Every account needs to save on the system in order to use software usability



2.3.4 Sort entries

Whether an user selects to sort, this case will enable the feature.



2.4 Use case description detailed

In order to get the total recognition of our project, we need a clear cut description of the use cases. The below tables are going to show the actor goals in a detailed way.

2.4.1. Manage Address

<table>
<tr><td>Name</td><td>Manage Address</td></tr>
<tr><td>ID</td><td>001</td></tr>
<tr><td>Description</td><td>The user can do three types of work here. Firstly, he can add new address. Secondly he can edit/delete address. Finally, he can sort all the contacts.
At the time of sorting, he can do it by zip address.</td></tr>
<tr><td>Actors</td><td>User</td></tr>
<tr><td>Organizational benefits</td><td>Assist to do operation on contacts. Otherwise modification can not happen.</td></tr>
<tr><td>Preconditions</td><td>Users have to login in the system first.</td></tr>
<tr><td>Post conditions</td><td>Users selects an option to do operations. Otherwise operations will be invalid</td></tr>
<tr><td>Main flow</td><td><table><tr><th>Actor</th><th>System</th></tr><tr><td>1.users first login into the system
2.Then selects an option
3. After saving all operations, users will exit</td><td>1. provide user interface with login form
2. System will save all changes</td></tr></table></td></tr>
<tr><td>Alternate flow</td><td>1. If customer does not login, he can not see the options.
2. If he does not press option, hen contacts will like the first
3. If users fail to do sort, the quick searching will be difficult</td></tr>
</table>

2.4.2 Print Address

<table>
<tr><td><b>Name</b></td><td>Print Address</td></tr>
<tr><td><b>ID</b></td><td>002</td></tr>
<tr><td><b>Description</b></td><td>Sometimes, user might have to print the address like a document. The print option will print the specific request contact in order to meet customer needs. The option will set a link to printer with software.</td></tr>
<tr><td><b>Actors</b></td><td>User</td></tr>
<tr><td><b>Organizational benefits</b></td><td>At any time printing might need. This requirement will do this work easier.</td></tr>
<tr><td><b>Frequency of usage</b></td><td>Infrequent</td></tr>
<tr><td><b>Preconditions</b></td><td>Users selects an option to do operations.</td></tr>
<tr><td><b>Post conditions</b></td><td>User needs to exit after doing an operation. As, the program might print more than one address.</td></tr>
<tr><td><b>Main flow</b></td><td><table>
<tr><td><b>Actor</b>
1.users first login into the system
2.Then selects an option
3. After printing , users will exit</td><td><b>System</b>
1. provide user interface with login form
2. System will print the user request</td></tr>
</table></td></tr>
<tr><td><b>Alternate flow</b></td><td>1. If customer does not login, he can not see the options.
2. If he does not press stop option after requested printing, then contacts will be printed one by one.
3. If users fail to do sort, the quick searching will be difficult</td></tr>
</table>

2.4.3 Open/Close Address Book

<table>
<tr><td><b>Name</b></td><td>Open/ close address book</td></tr>
<tr><td><b>ID</b></td><td>003</td></tr>
<tr><td><b>Description</b></td><td>User login to open the software</td></tr>
<tr><td><b>Actors</b></td><td>User</td></tr>
<tr><td><b>Organizational benefits</b></td><td>With a view to using or exit, the requirement will be used</td></tr>
<tr><td><b>Frequency of usage</b></td><td>Very frequent.</td></tr>
<tr><td><b>Preconditions</b></td><td>Users selects an option to do operations.</td></tr>
<tr><td><b>Post conditions</b></td><td>User have to open the system first.</td></tr>
<tr><td><b>Main flow</b></td><td><table><tr><td><b>Actor</b>
1.users first login into the system
2.Then selects an option
3. After printing , users will exit</td><td><b>System</b>
1. provide user interface with login form
2. System will save the user requests.</td></tr></table></td></tr>
<tr><td><b>Alternate flow</b></td><td>1. If customer does not login, he can not see the options.
2. If he does not press save option, then operations will be lost.</td></tr>
</table>

2.4.4 Save Address

<table>
<tr><td><b>Name</b></td><td>Save address</td></tr>
<tr><td><b>ID</b></td><td>004</td></tr>
<tr><td><b>Description</b></td><td>User save an address after creating.</td></tr>
<tr><td><b>Actors</b></td><td>User</td></tr>
<tr><td><b>Organizational benefits</b></td><td>User requested contacts will be saved here. That's how everything saved on the computer</td></tr>
<tr><td><b>Frequency of usage</b></td><td>Very frequent.</td></tr>
<tr><td><b>Preconditions</b></td><td>Users selects an option to do operations.</td></tr>
<tr><td><b>Post conditions</b></td><td>User have to open the system first.</td></tr>
<tr><td><b>Main flow</b></td><td><table><tr><td><b>Actor</b>
1.users first login into the system
2.Then selects an option
3. After printing, users will exit</td><td><b>System</b>
1. provide user interface with login form
2. System will save the user requests.</td></tr></table></td></tr>
<tr><td><b>Alternate flow</b></td><td>1. If customer does not login, he can not see the options.
2. If he does not press save option, then operations will be lost.</td></tr>
</table>

2.4.5 Print most recently saved.

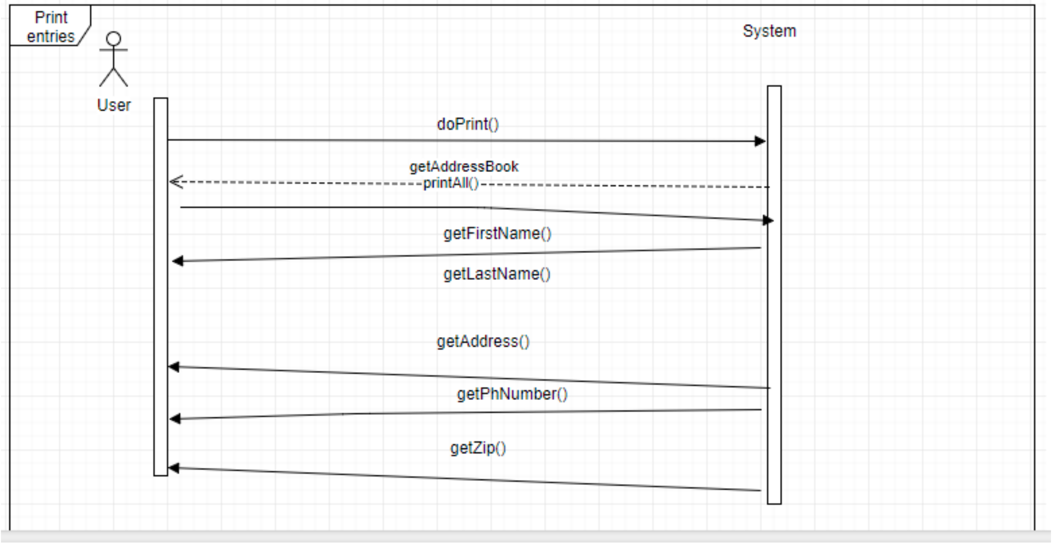
<table>
<tr><td>Name</td><td>Save address</td></tr>
<tr><td>ID</td><td>005</td></tr>
<tr><td>Description</td><td>Users will see which are recently saved.</td></tr>
<tr><td>Actors</td><td>User</td></tr>
<tr><td>Organizational benefits</td><td>User can also do search contacts from here.</td></tr>
<tr><td>Frequency of usage</td><td>Infrequently</td></tr>
<tr><td>Preconditions</td><td>User have to open the system first.</td></tr>
<tr><td>Post conditions</td><td></td></tr>
<tr><td>Main flow</td><td><table><tr><td><b>Actor</b>
1.users first login into the system
2.Then selects a most recently saved option
3. After that, he will see the contacts which he has recently saved.</td><td><b>System</b>
1. provide user interface with relevant options
2. System will show the customer needs.</td></tr></table></td></tr>
<tr><td>Alternate flow</td><td>1. If customer does not login, he can not see the options.</td></tr>
</table>

2.5 System sequence diagram

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various run time scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modeling a new system. In this project we do need some system sequence diagram in order to realize the inside operations in a system.

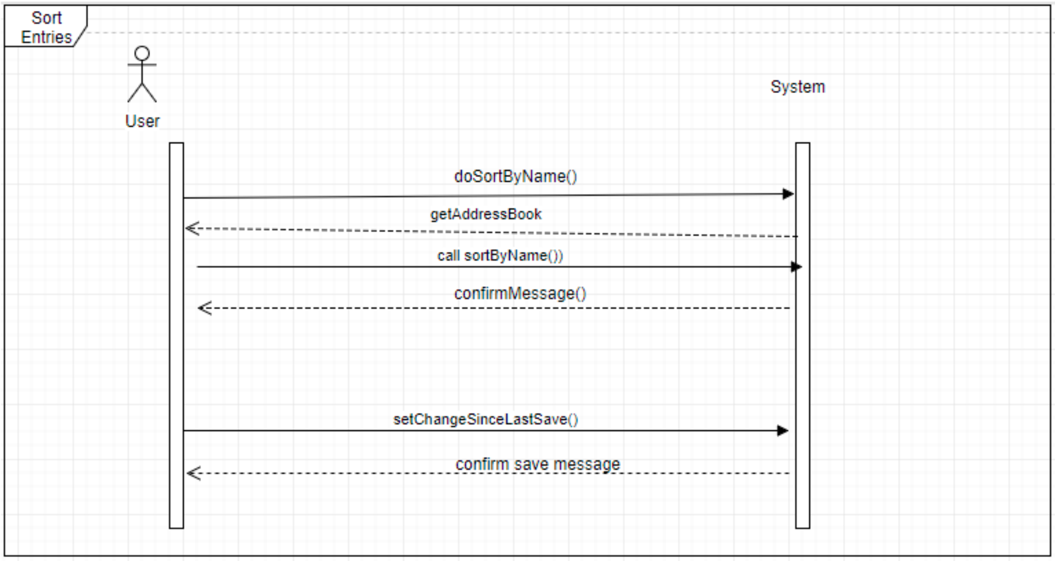
2.5.1 Print entries system sequence diagram

at the last stage of the stage, user will get first name, last name, address, phone number, zip etc.



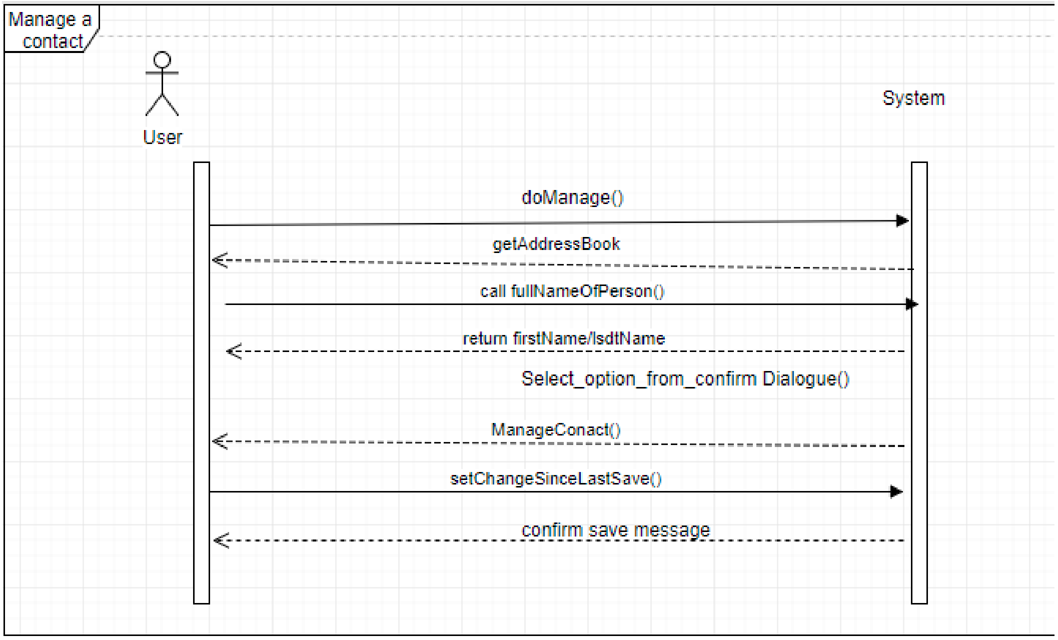
2.5.2. Sort entries

If any user wanna use sorting feature, then this requirement will help him to sort this needs
Below diagram will give us a total view.

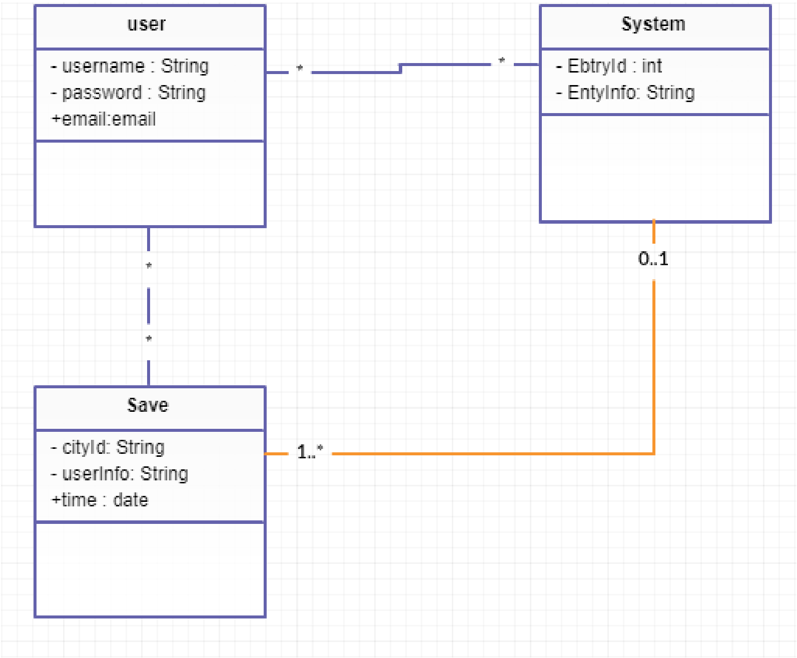


2.5.3 Manage a contact

Various tasks will be done by this requirement like update, delete, edit etc.



2.6 Domain Model



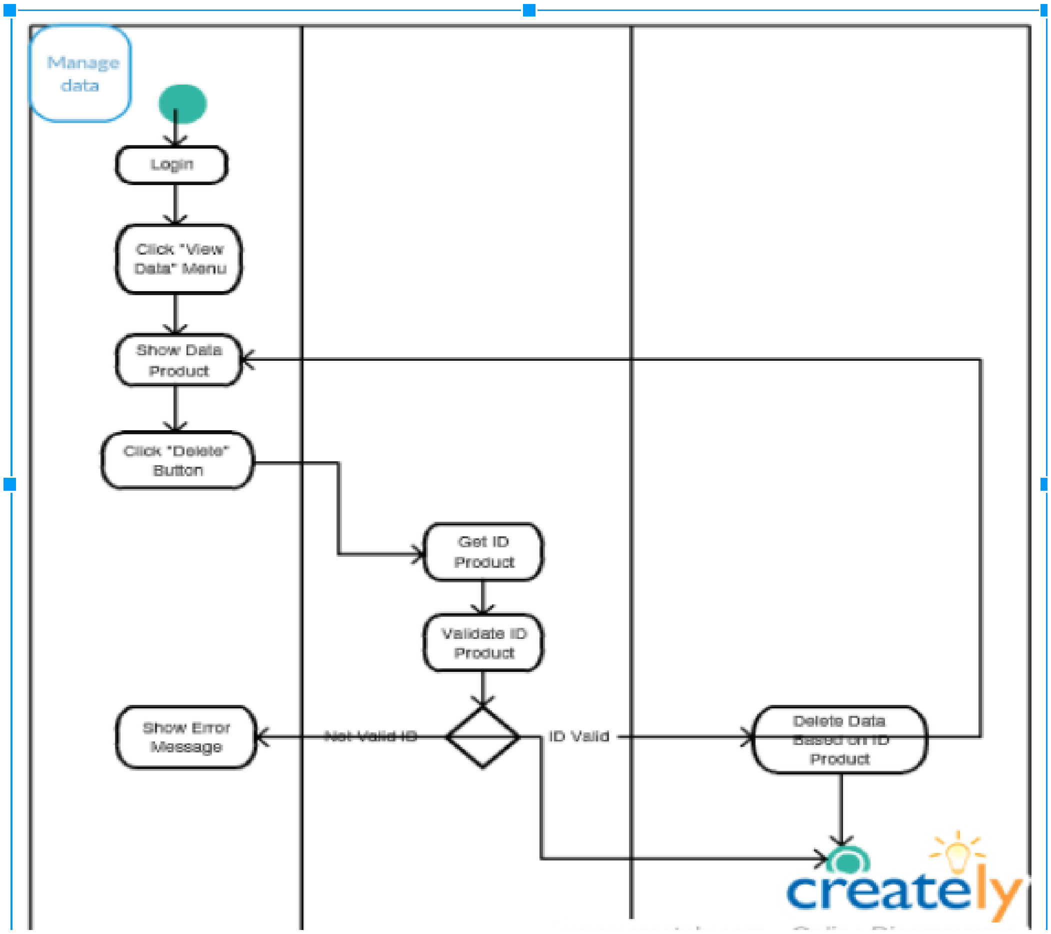
2.7 Activity diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.

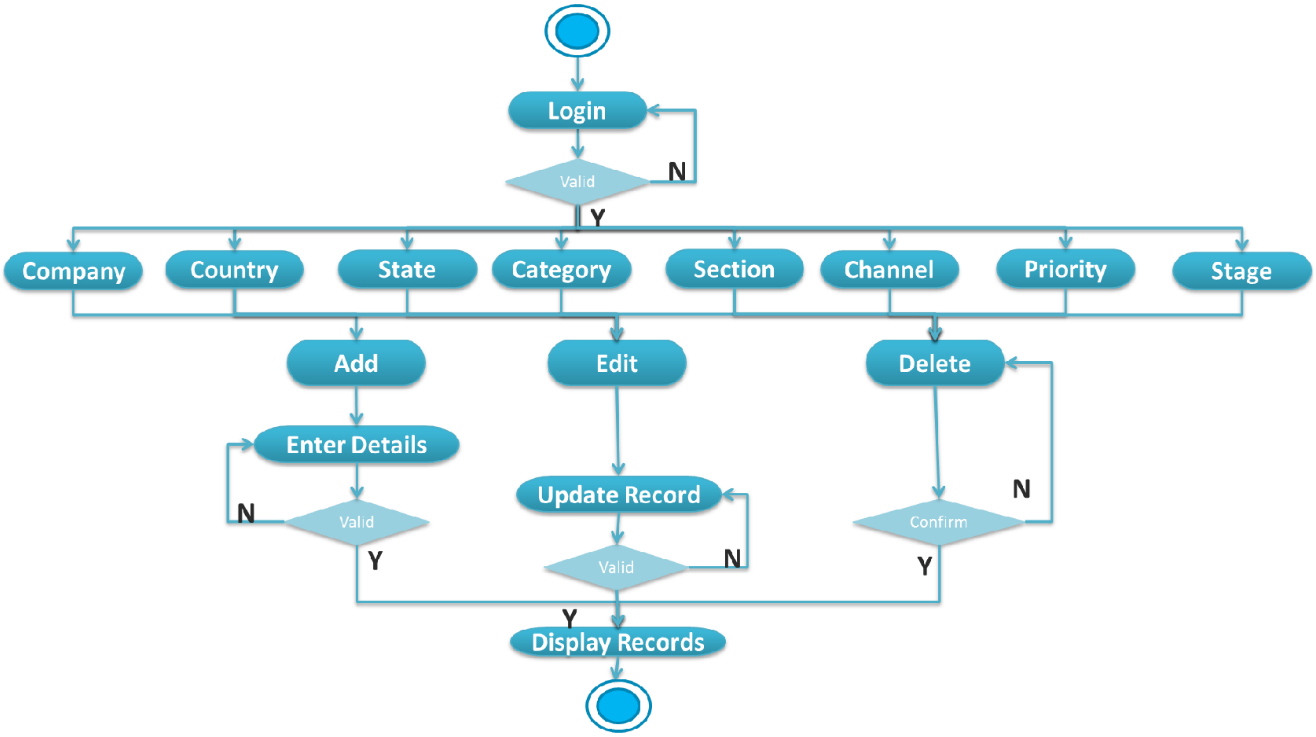
Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. We can describe our system by drqwing activity diagrams. Those will be shown below.

2.7.1 Manage data activity diagram



2.7.2 Display all records



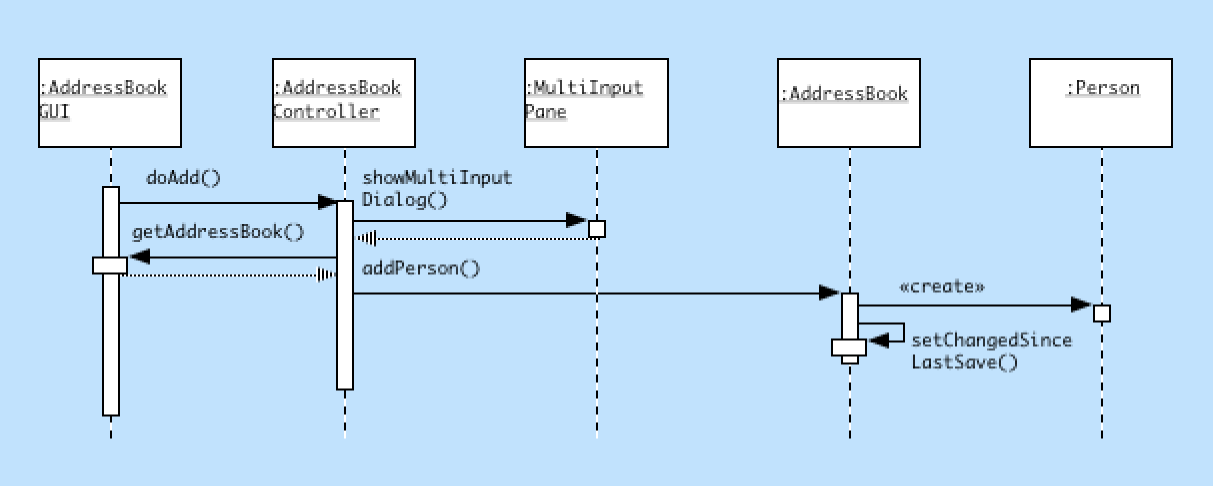
3. System design

Software design is the process by which an agent creates specification of a software artifact, intended to accomplish goals, using a set of primitive components and subject to constraints. Software design may refer to either “all the activity involved in conceptualizing, framing, implementing, commissioning, and ultimately modifying complex systems” or “the activity following requirements specification and before programming, as a stylized software engineering process.

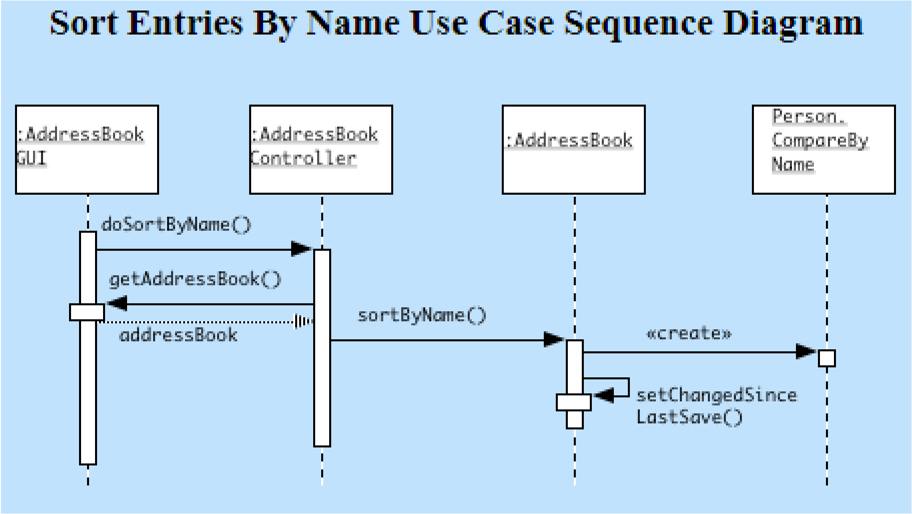
3.1 Sequence diagram

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They’re also called event diagrams. A sequence diagram is a good way to visualize and validate various run time scenarios. This can help to predict how a system will behave and to discover responsibilities as a class may need to have in the process of modeling a new system. Our contact management has some sequential behaviors.Those will be implemented in the next sections one by one

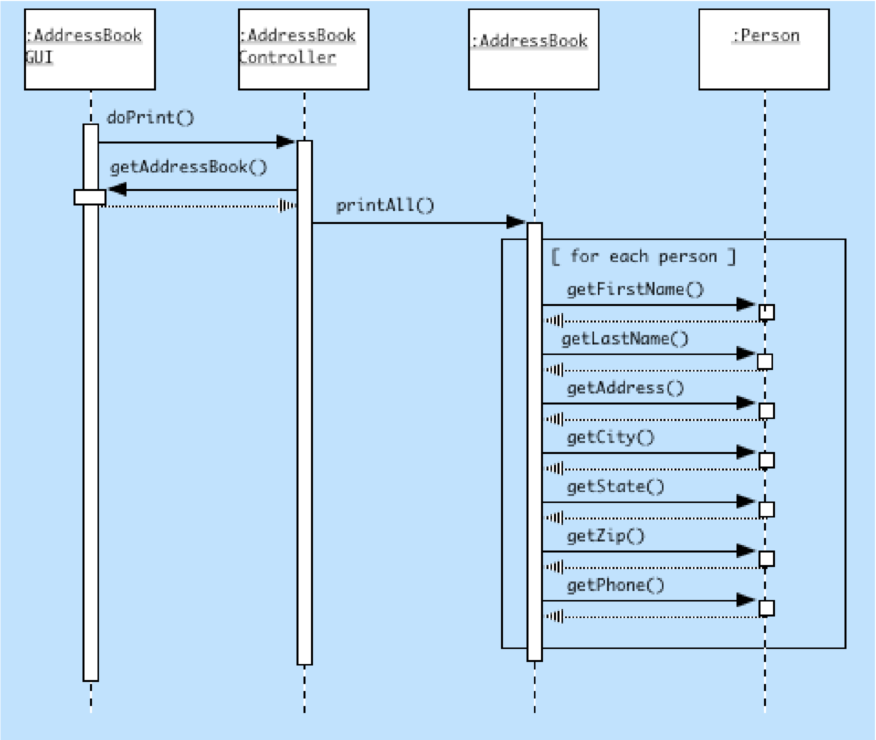
3.1.1 Add a contact



3.1.2 Sort Entries

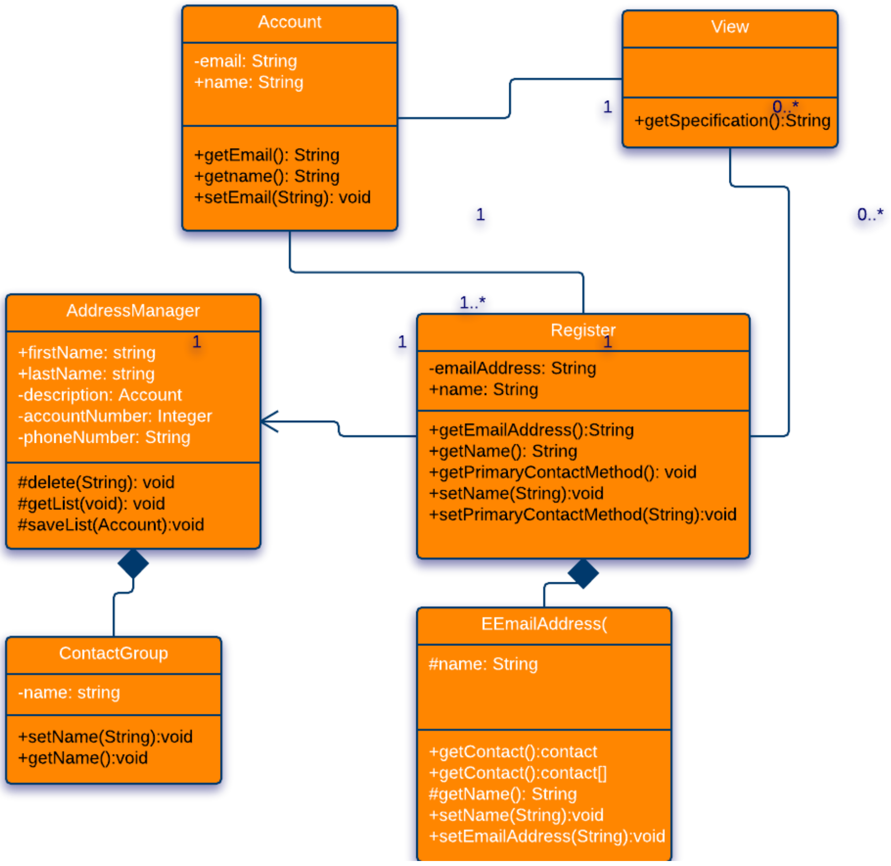


3.1.3 Print Entries



3.2 Class diagram

A class diagram models the static structure of a system. It shows relationship between classes, objects, attributes, and operations. Our project needs a class diagram which is shown below.


Account
-email: String
+name: String
+getEmail(): String
+getname(): String
+setEmail(String): void
View
+getSpecification():String
1
0..*
1
0..*
1..*
1
1
1
AddressManager
+firstName: string
+lastName: string
-description: Account
-accountNumber: Integer
-phoneNumber: String
#delete(String): void
#getList(void): void
#saveList(Account):void
Register
-emailAddress: String
+name: String
+getEmailAddress():String
+getName(): String
+getPrimaryContactMethod(): void
+setName(String):void
+setPrimaryContactMethod(String):void
ContactGroup
-name: string
+setName(String):void
+getName():void
EEmailAddress(
#name: String
+getContact():contact
+getContact():contact[]
#getName(): String
+setName(String):void
+setEmailAddress(String):void