Course Project

Address Book

Course Code: SWE 331

Course Title: Object Oriented Software Development

Name: Md. Robin Khan

ID: 151-35-908

Supervisor name: Alamgir Kabir

Department: Software Engineering

Faculty of science and Information Technology

December 2017

Table of Contents

List of Tables

List of Figures

Chapter 01

1. Introduction

1.1 About the system

This system helps one to manage his address books. There is different types of books including family, office etc. Each address book holds a collection of entries including their name, email, city, contact number and address. To get these facilities user must register with the system first.

1.2 Purpose

To manage the details information of different types of persons easily is the purpose of this system. As it is a web application, so it will be available and reachable from anywhere and anytime. So one can find out the details about the desired person book by searching in this system very easily.

1.3 Scope

- ✓ Anybody can register with the system by providing their basic information like name, email and password.
- ✓ The system will be well secured.
- ✓ Users can manage their books and entries very easily.
- ✓ There is also a search option in books menu and in each book which response immediately.
- ✓ There is less chance of losing the information as it will store them in the database.
- ✓ It gives user very user-friendly environment.

1.4 Vision

To provide users with the facilities like manage address books, search an entry, access to the system from anywhere anytime by logging into their account very easily etc. is the vision of this system.

1.5 Why is the system necessary

To fulfill the following demands the system is necessary:

1. To add unlimited entries in different types of address books.
2. To keep details of the entries safe and private.
3. To edit and update existing information of any entry.

4. To search and find out any book or any entry with details from a huge collection.
5. To delete an entry or a collection of entry at a time.

1.6 Proposed solution

User must log into the system first for privacy purpose after registration. There he/she can add address book with a type from a dropdown list for managing books and their entries easily. The user also can edit and delete any entry or a collection of entry. There is also a quick search option in each entry where user can search any entry by name. And when the user want to log off he/she can do that very easily.

Chapter-02

2. System analysis

2.1 Actor goal list

Table 1: Actor goal list

Actor	Goals
Registered User	Add book Manage book Manage entry Search book Search entry
Non-registered User	Registration

2.2 Use case model

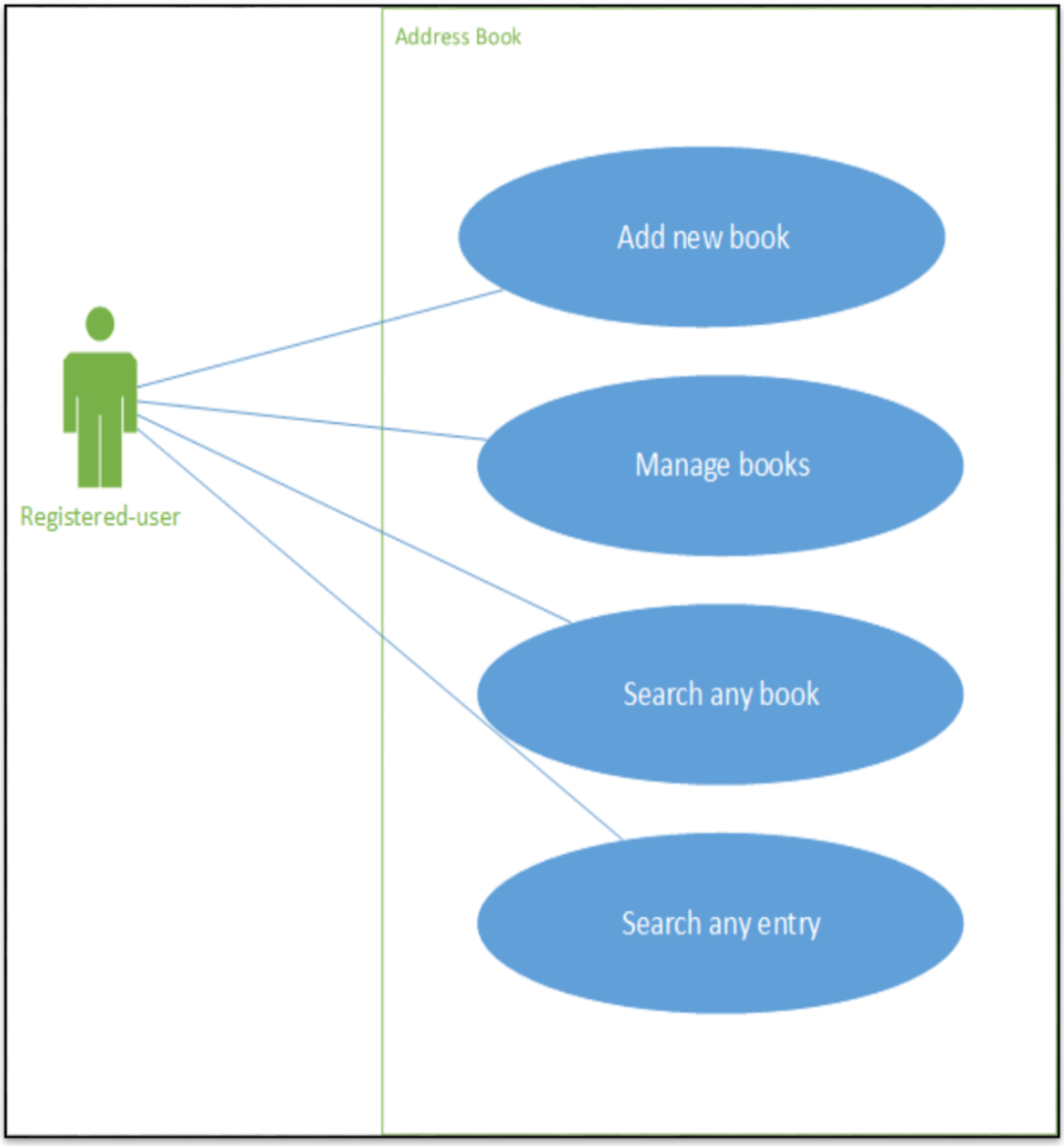

Figure 1: use case model

2.3 Use case description (brief)

2.3.1 Add new book

Registered user will first log into the system by providing email and password. Then he/she can add new book giving a name and choosing a type. In every book there is

add entry menu also. There he/she have to enter name, city, phone number and address.

2.3.2 Manage books

This is also for registered users. After having so many books with huge amount of entry an user will have the privilege to manage all books and entries including edit and delete.

2.3.3 Search any book

In books menu registered users will find a search option for any book by name or by type.

2.3.4 Search any entry

In each book to find any specific entry there is also another search option for registered user and it is possible to search any entry by their name or city or phone number or any other field.

2.4 Use case description

2.4.1 Add new book

Table 2: Add new book use case

Name:	Add new book	
Scope:	Creates a new book	
Level:	User level	
Primary actor:	Registered user	
Precondition:	Registration and login	
Post-condition:	A book with entry and their details should be stored in the database.	
Main flow:	Actor	System
	1. Log into the system. 2. Click add new book. 3. Give name, select type and submit.	1.1 Authenticate and welcome the user. 2.1 Display book type. 3.1 Adds a new book
Alternative Scenario:	1. Wrong input when log in. 2. Don't enter any required field.	1.1 Displays wrong message. 2.1 Shows required message under the fields.

2.4.2 Manage books

Table 3: Manage books

Name:	Manage books

Scope:	Delete, edit and update books and their entries	
Level:	User level	
Primary actor:	Registered user	
Precondition:	Registration and login	
Post-condition:	All changes should be applied in the database if the changes are valid.	
Main flow:	Actor	System
	1. Log into the system. 2. Select delete any book or entry. 3. Select edit a book or an entry.	1.2 Authenticate and welcome the user. 2.1 Show confirmation box. 3.1 Apply changes immediately.
Alternative Scenario:	1. Wrong input when log in. 2. Don't enter any required field.	1.2 Displays wrong message. 2.1 Shows required message under the fields.

2.4.3 Search any book

Table 4: Search any book use case

Name:	Search any book	
Scope:	To find out any specific book by name or by type	
Level:	User level	
Primary actor:	Registered user	
Precondition:	Registration and login	
Post-condition:	All changes should be applied in the database if the changes are valid.	
Main flow:	Actor	System
	1. Log into the system. 2. Search a book by name. 3. Search a book by type.	1.3 Authenticate and welcome the user. 2.1 Show relevant results. 3.1 Show relevant results.
Alternative Scenario:	1. Wrong input when log in. 2. Don't enter any valid search item.	1.1 Displays wrong message. 2.1 No result found.

2.4.4 Search any entry

Table 5: Search any entry

Name:	Search any entry
Scope:	To find out any specific entry of a book by name or any other field.
Level:	User level
Primary actor:	Registered user
Precondition:	Registration and login
Post-condition:	All changes should be applied in the database if the changes are valid.

Main flow:	Actor	System
	1. Log into the system. 2. Search an entry by name or any other field.	1.4 Authenticate and welcome the user. 2.1 Show relevant results.
Alternative Scenario:	1. Wrong input when log in. 2. Don't enter any required field.	1.3 Displays wrong message. 2.1 Shows required message under the fields.

2.5 System Sequence Diagram

2.5.1 Add new book (Main flow)

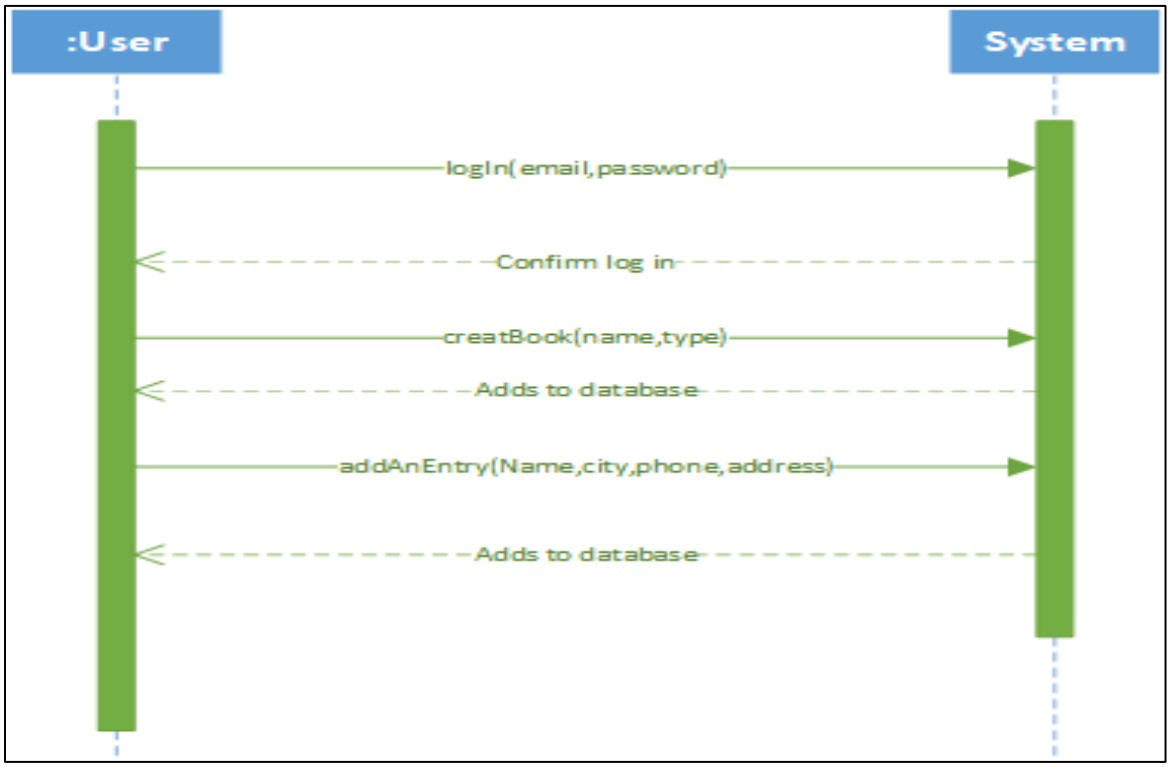

Figure 2: Add new book (success scenario)

2.5.2 Add a new book (Alternative scenario)

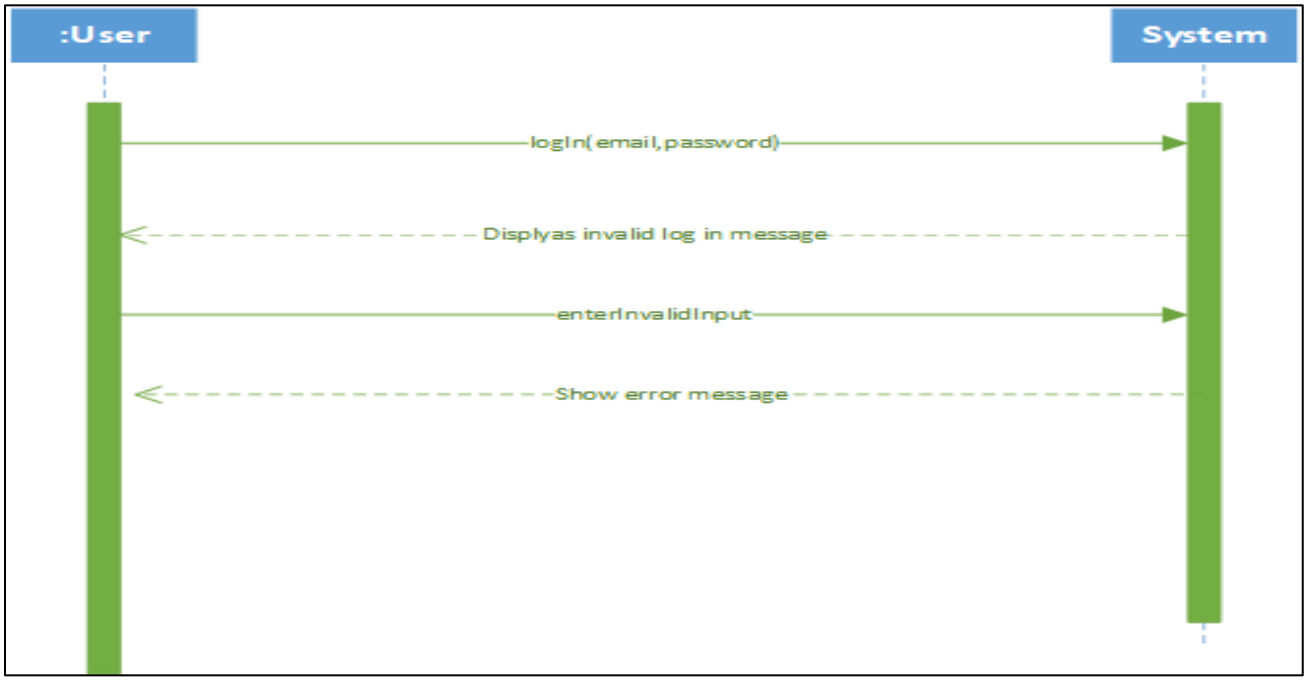

Figure 3: Add new book (Alternative scenario)

2.5.3 Manage book (Main flow)

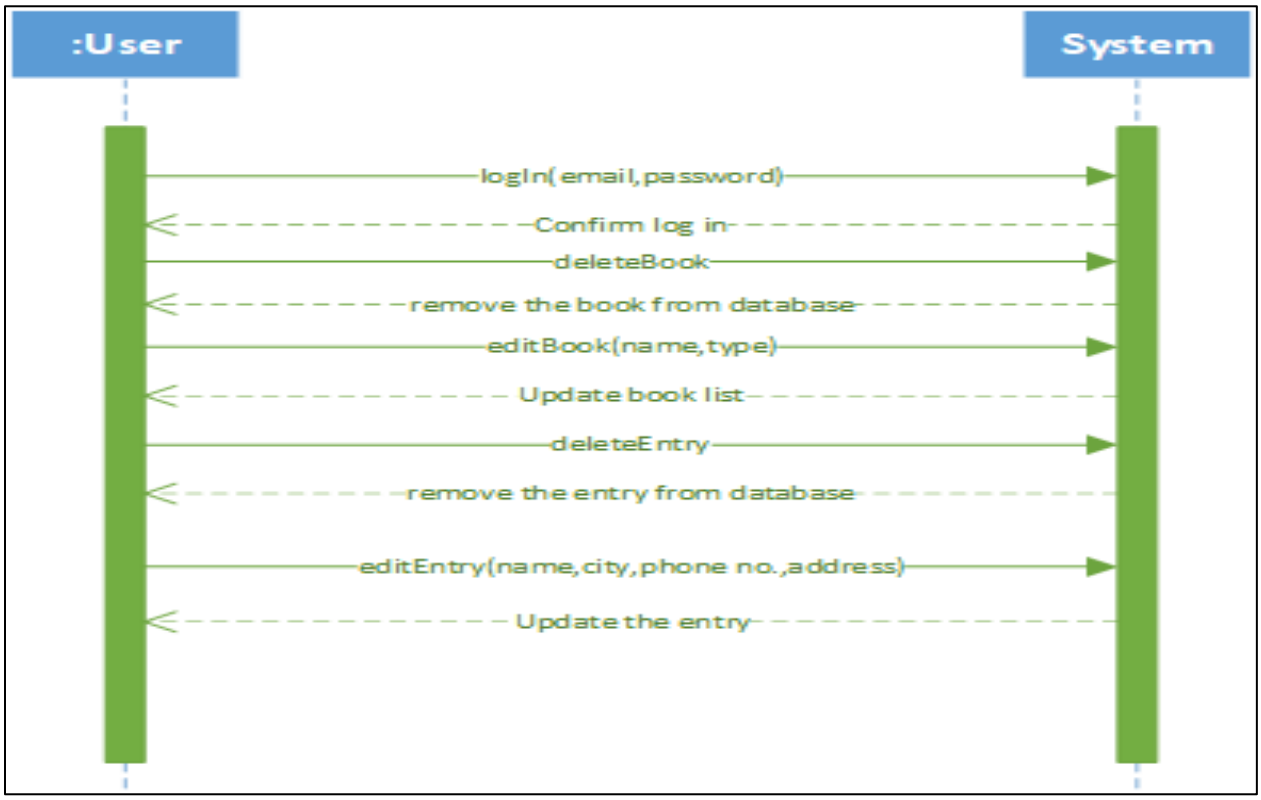

Figure 4: Manage books (success scenario)

2.5.4 Manage books (Alternative scenario)

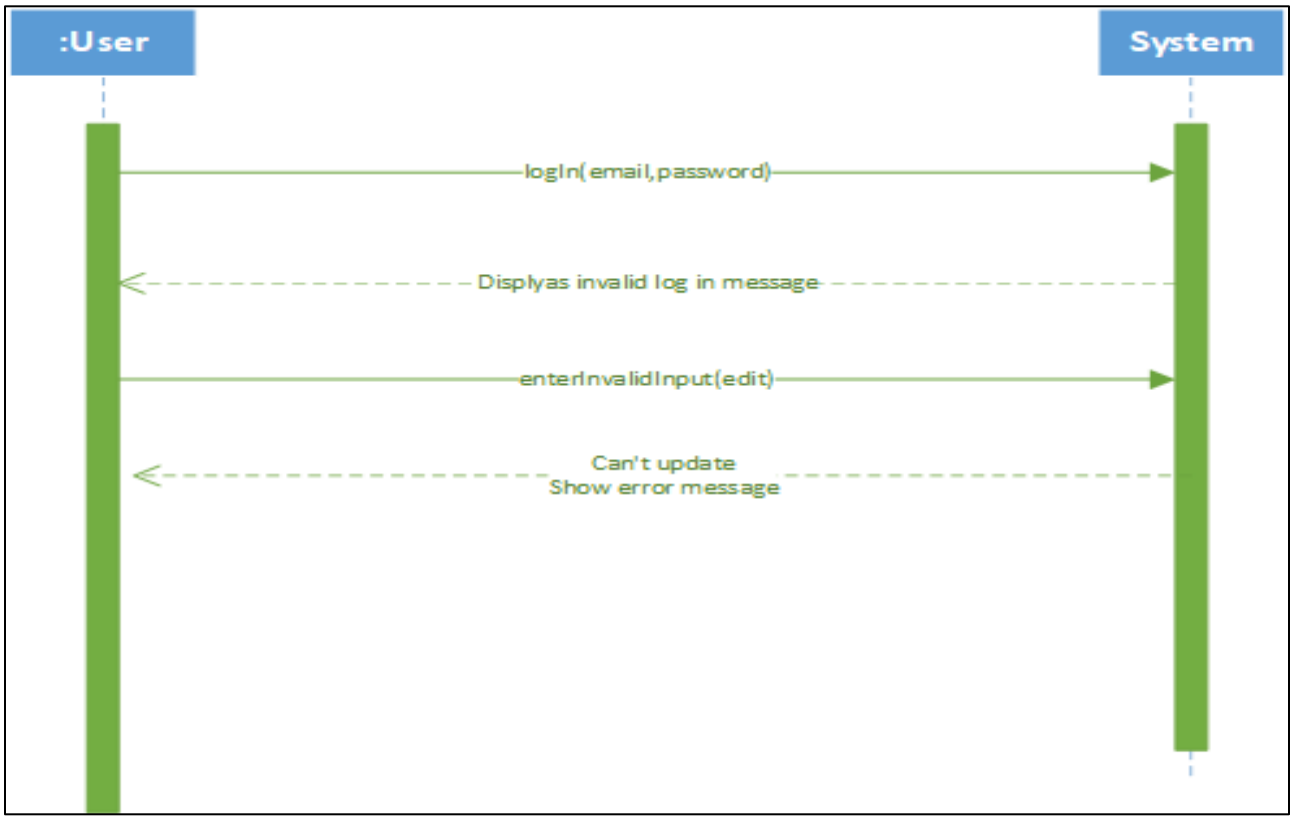

Figure 5: Manage books (Alternative scenario)

2.5.5 Search any book (Main flow)

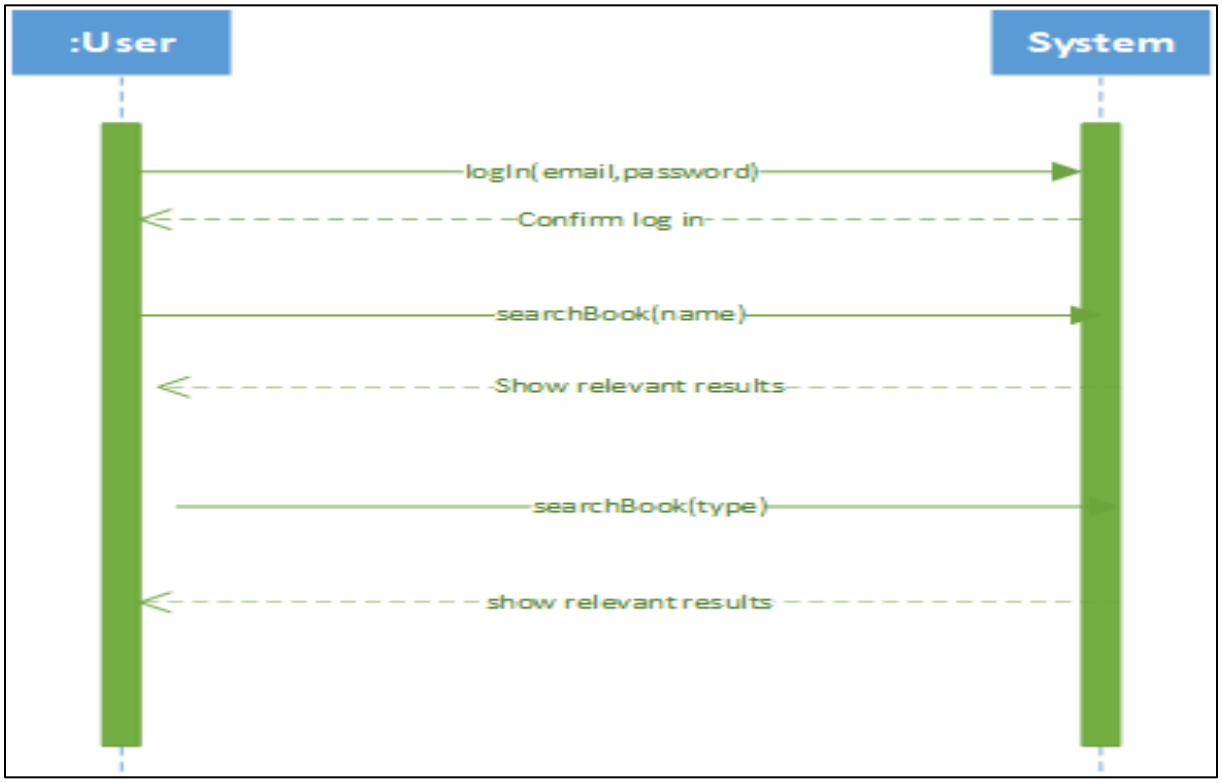

Figure 6: Search any book (success scenario)

2.5.6 Search any book (Alternative scenario)

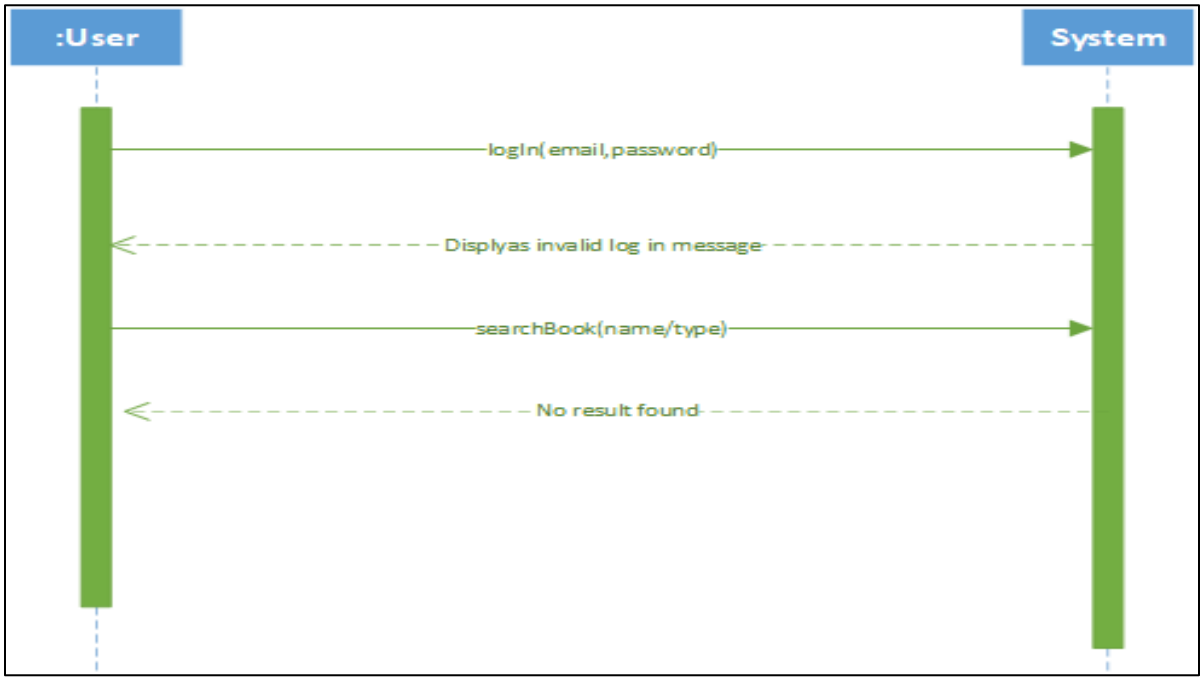

Figure 7: Search any book (Alternative scenario)

2.5.7 Search any entry (Main flow)

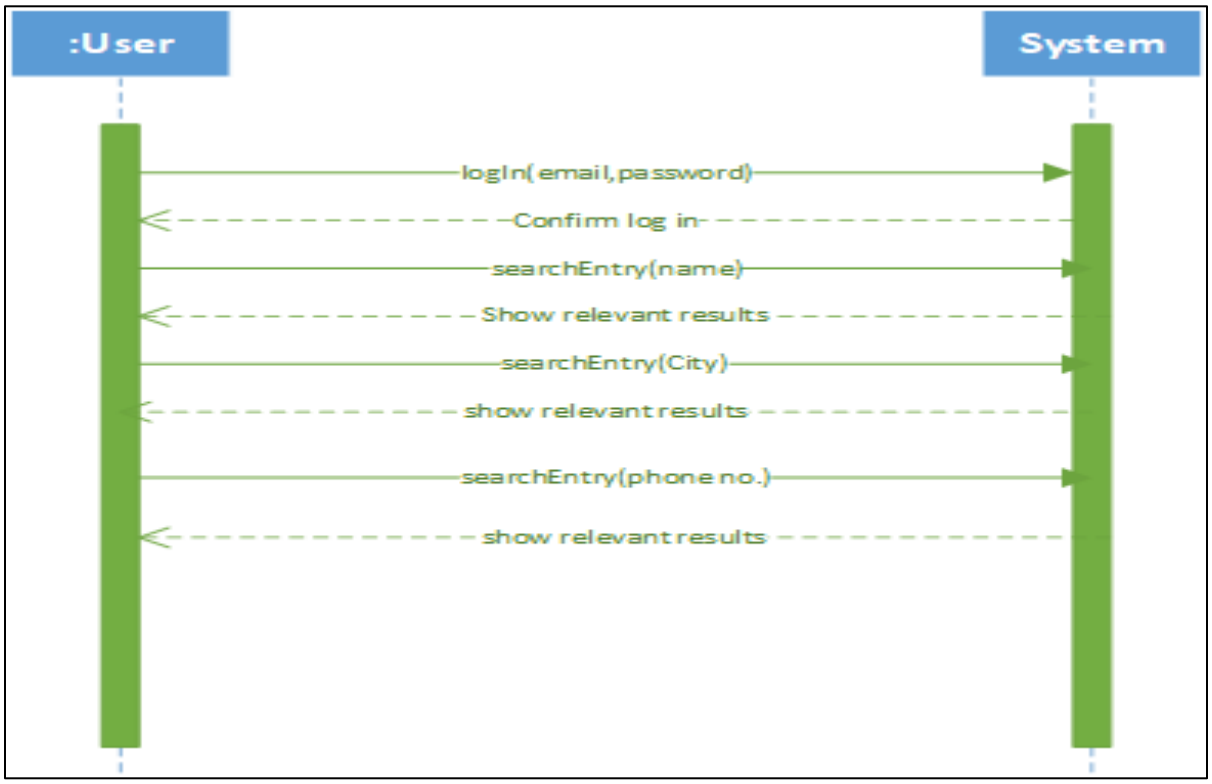

Figure 8: Search any entry (success scenario)

2.5.8 Search any entry (Alternative scenario)

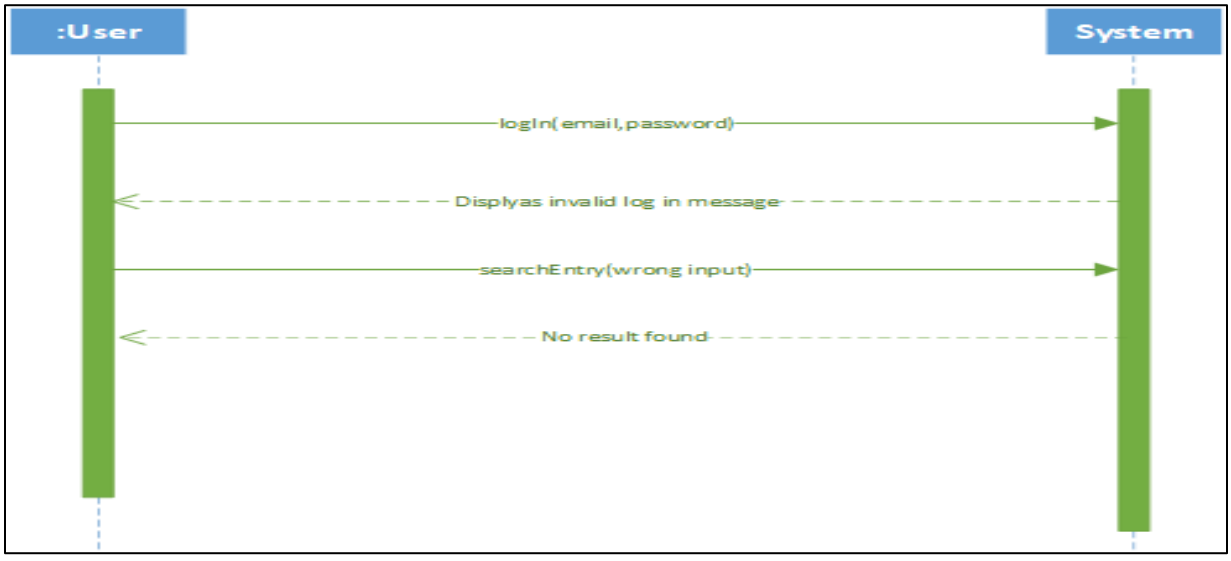

Figure 9: Search any entry (Alternative scenario)

2.6 Domain model

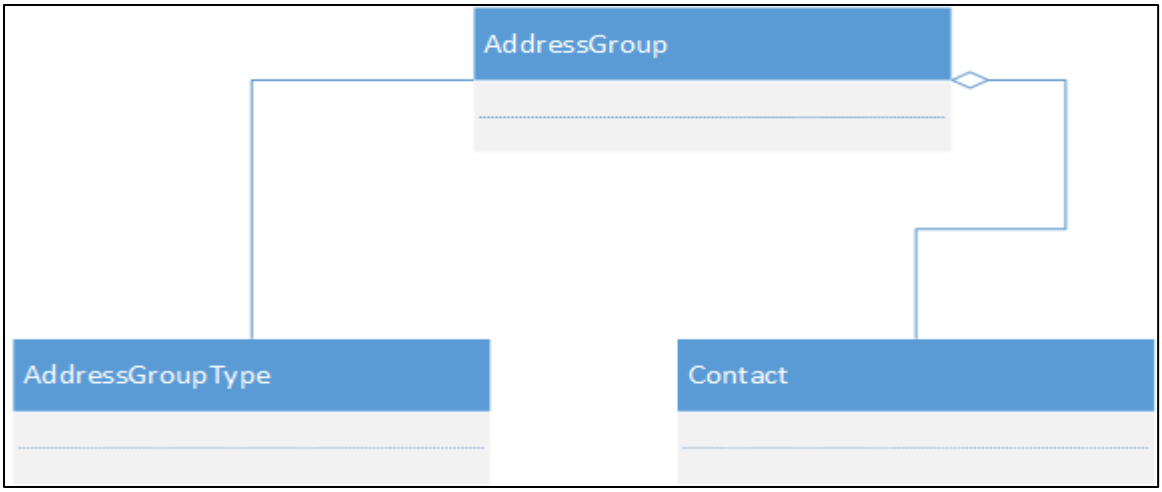

Figure 10: Domain model

2.7 Activity Diagram

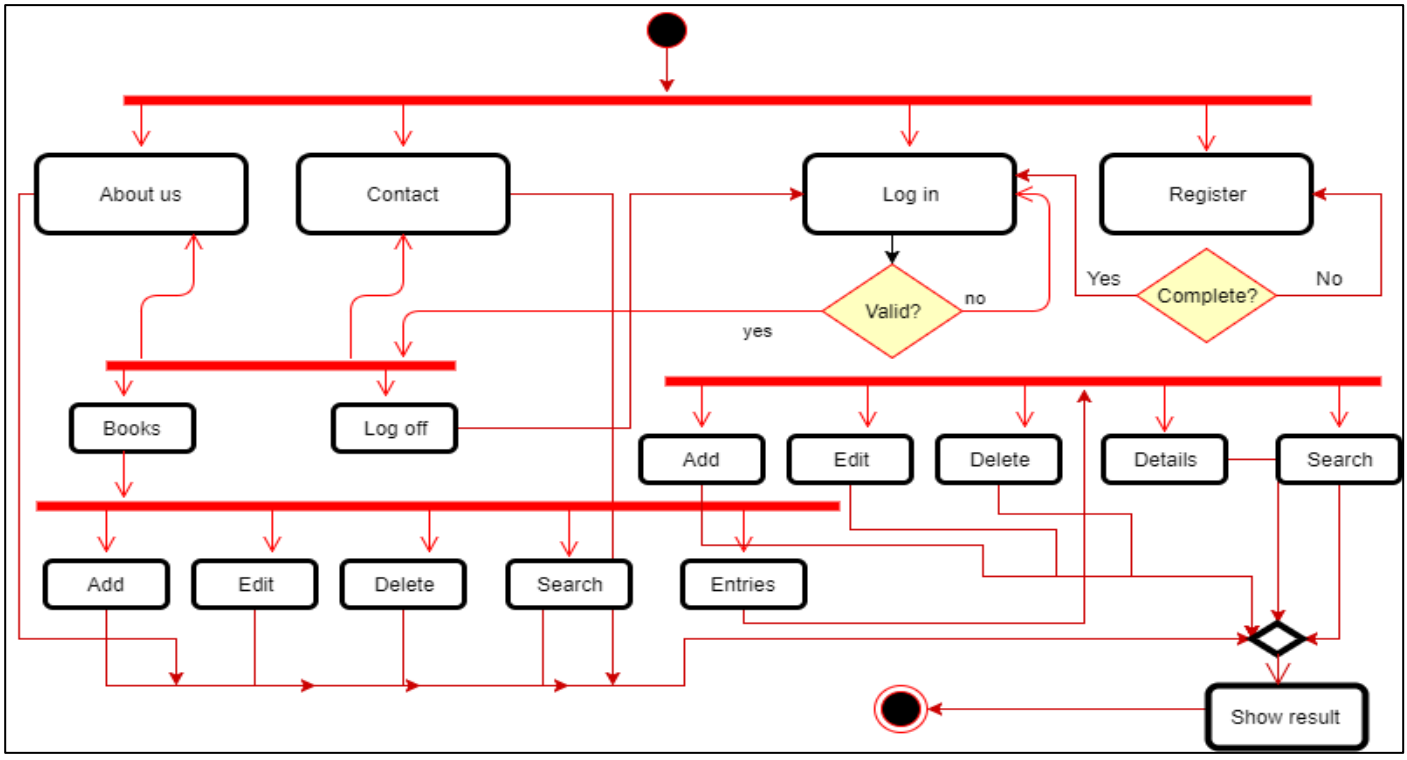

Figure 11: Activity Diagram

Chapter-03

3. System Design

3.1 Sequence Diagram

3.1.1 Add new book

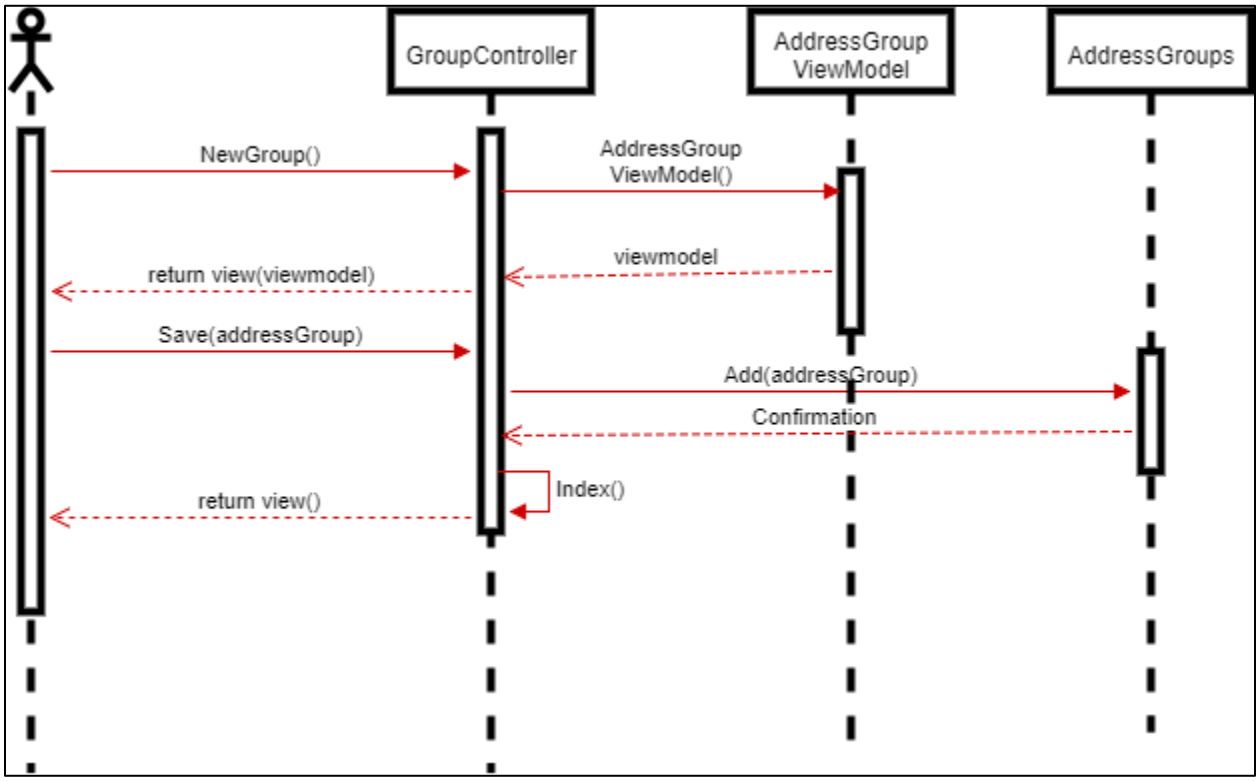

Figure 12: Add new book

3.1.2 Edit a book

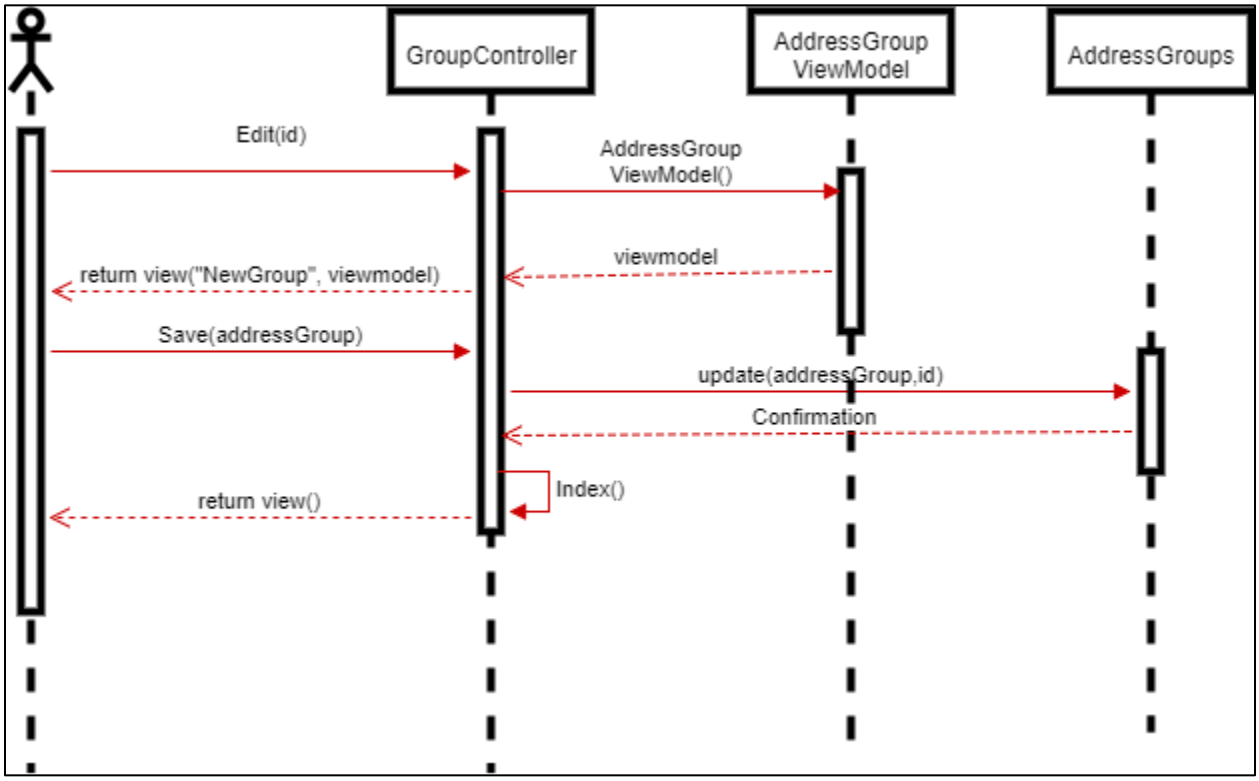

Figure 13: Edit a book

3.1.3 Delete a book

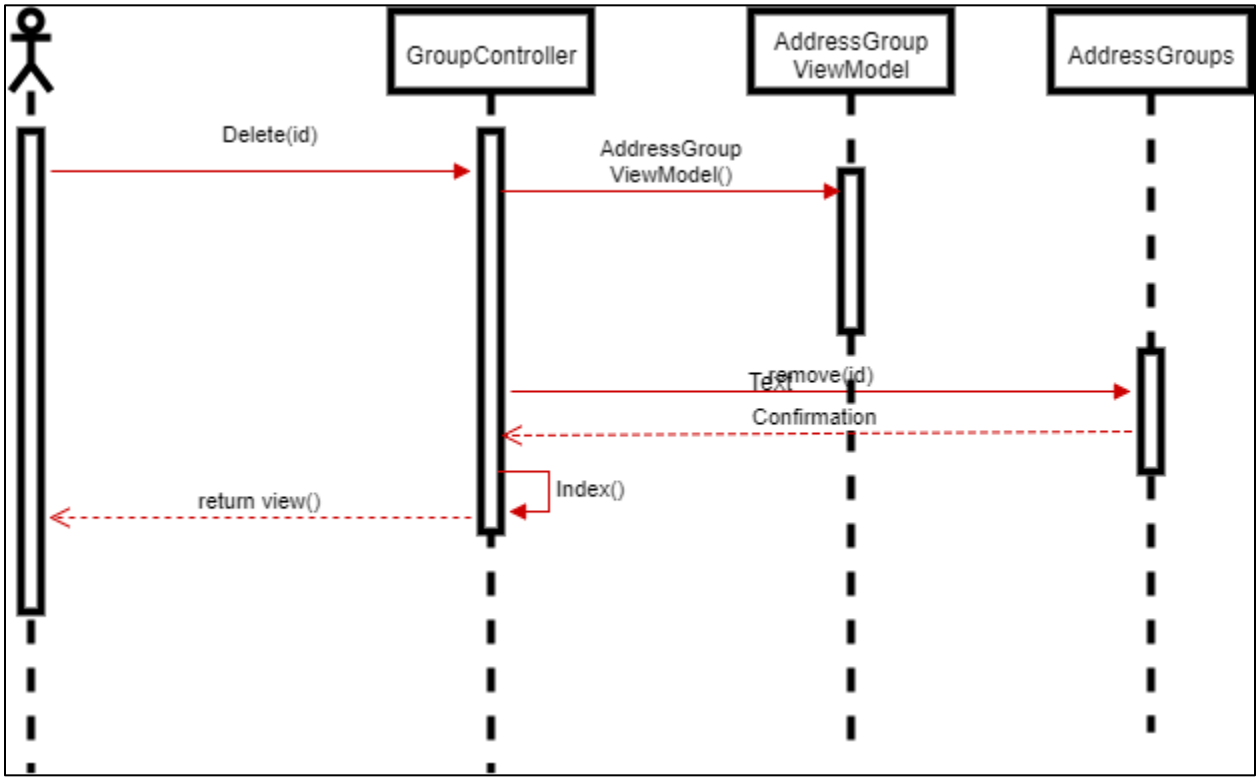

Figure 14: Delete a book

3.2 Class Diagram

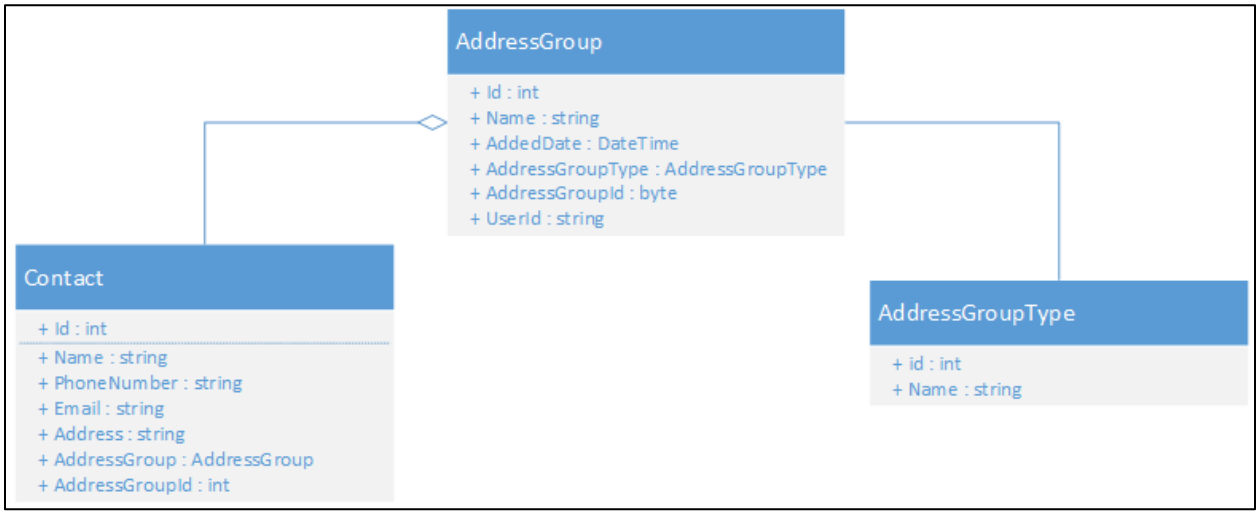

Figure 15: Class Diagram

3.3 Entity Relationship Diagram

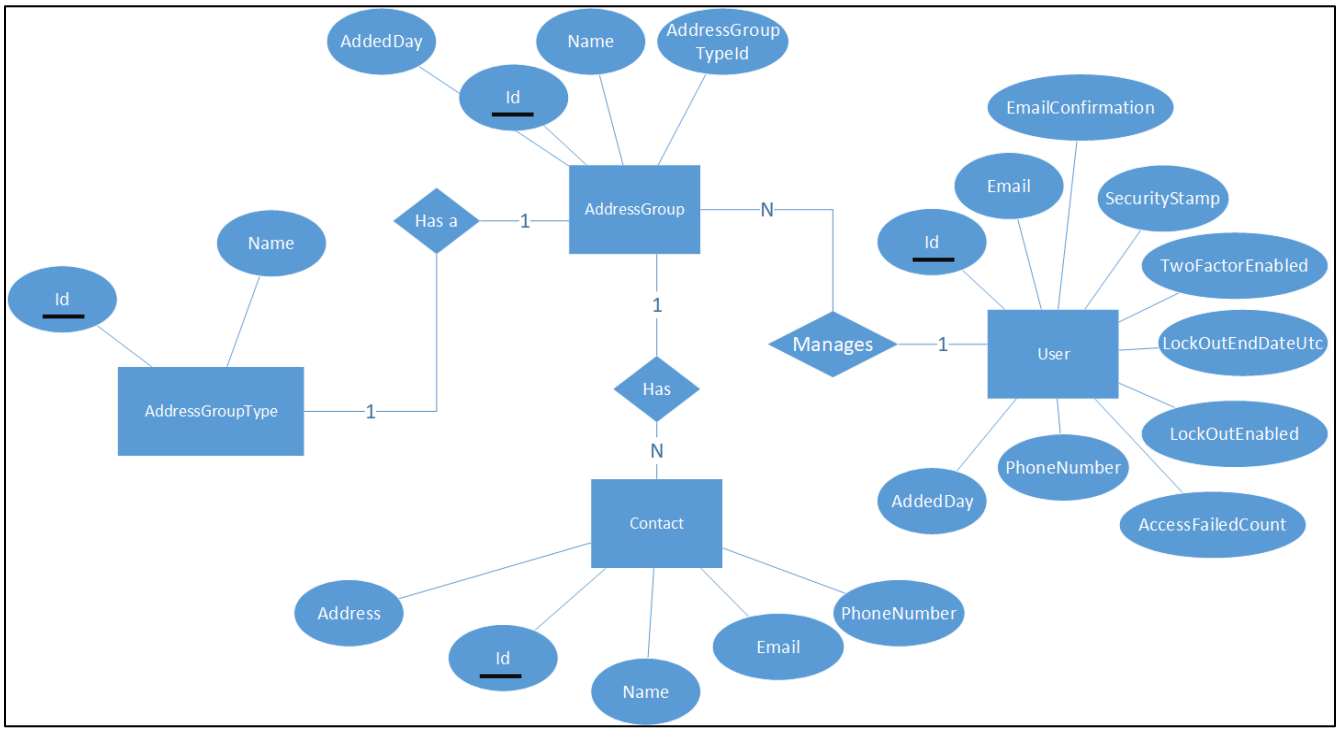

Figure 16: Entity relationship(ER diagram)

Chapter 04

4. Implementation

4.1 Platform and technology

- Visual studio 2015
- ASP.NET MVC
- C#
- API

4.2 Reference

https://github.com/trktuhin/TheAddressBook