

Vision

Revision History

Version	Date	Description	Author
1 st Iteration	Feb 22,2017	First draft.	Hussan Alsaylani

Introduction

The project is to develop a software that can be used to maintain an Electronic Address Book. An address book contains details of persons like a person's first and last names, address, city, state, zip and phone number.

more?

Product Features

On an individual address book, this product support features like

- Addition of a new person to the address book.
- Edit a person's details
- Delete a person

in detail please

Feature Overview

- **Add a new person**

This feature allows the user to add a new person to an address book. This person details include his first and last names, address, city, state, zip and phone number. Each entry in the address book should have a unique first name and last name combination. Duplicate entries won't be allowed.

- **Edit person**

This feature refers to editing a particular person's details. User selects a person from the entries of the address book either by highlighting the entry. All the details of the selected person are editable except his name.

- **Delete person**

This allows the user to delete a particular person's entry from the address book. In this case user selects a person from the entries of the address book by highlighting the entry. On clicking Delete button this entry will be deleted from the address book.

GLOSSARY

User

Refers to the end user of the system.

Address Book

The final product i.e. the software being developed.

} more ↑

Supplementary Specification Document

This contains details of all those requirements not listed in the use cases or main feature list. These requirements are essential for the main requirements to function properly.

- **List entries of address book**

When the main window of address book opens, all the entries currently present in the address book are listed in a tabular manner. For editing and deleting an entry, user selects an entry from the listed entries. Selection is done by highlighting an entry. This list is updated if any new entry is added or deleted.

Use Case UC1: Adding a person to address book

Scope: Electronic Address Book

Level: System feature

Primary Actor: User

Stakeholders and Interests:

- User: Wants a fast entry

Preconditions: Entry is identified and authenticated.

Success Guarantee (or Post conditions): Entry (Person details) is added to the address book.

Success Scenario (or Basic Flow):

1. Person details to be entered is made ready.
2. User clicks on the Add button.
3. User enters the details like name, address, zip, phone number etc.
4. Users clicks on Ok.
5. System adds the entry to the address book.
6. A success message is displayed.

} in detail
with two columns
for test too

Failure Scenario (or Alternate Flow):

1. Person details to be entered is made ready.
2. User clicks on the Add button.
3. User enters the details like name, address, zip, phone number etc.
4. Users clicks on Ok.
5. System detects a duplicate entry is present based on the name.
6. A failure message is displayed and entered details are cleared.

Frequency of Occurrence: Could be nearly continuous.

Use Case UC2: Editing a person details from address book

Scope: Electronic Address Book

Level: System feature

Primary Actor: User

Stakeholders and Interests:

- User: Wants a fast search result and complete operation

Preconditions: Entry for editing is identified and authenticated.

Success Guarantee (or Post conditions): Required fields of the particular entry is edited.

Success Scenario (or Basic Flow):

1. Person details to be edited is made ready.
2. User double clicks on the person details whose details has to be edited.
3. User edits the required fields and clicks OK.
4. System updates the entry in the address book.
5. A success message is displayed.

Failure Scenario (or Alternate Flow):

1. Person details to be edited is made ready.
2. User highlights the person whose details has to be edited.
3. User clicks on Edit.
4. User updates the required fields and clicks OK.
5. Updating fails due to some wrong entry like alphabetical data in phone number.
6. A failure message is displayed.

Frequency of Occurrence: Could be nearly continuous.

Use Case UC3: Deleting a person details from address book

Scope: Electronic Address Book

Level: System feature

Primary Actor: User

Stakeholders and Interests:

- User: Wants a fast search result and complete operation

Preconditions: Entry for deletion is identified and authenticated.

Success Guarantee (or Post conditions): Required entry is deleted.

Success Scenario (or Basic Flow):

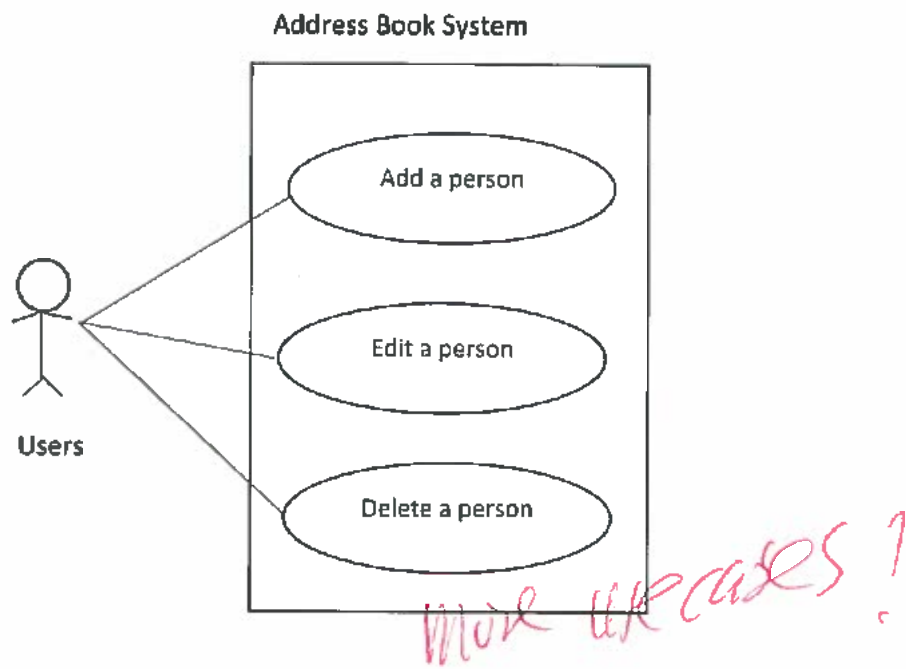
1. Person details to be deleted is made ready.
2. User highlights the person whose details has to be deleted.
3. User clicks delete.
4. System deletes the entry from the address book.
5. A success message is displayed.

Failure Scenario (or Alternate Flow):

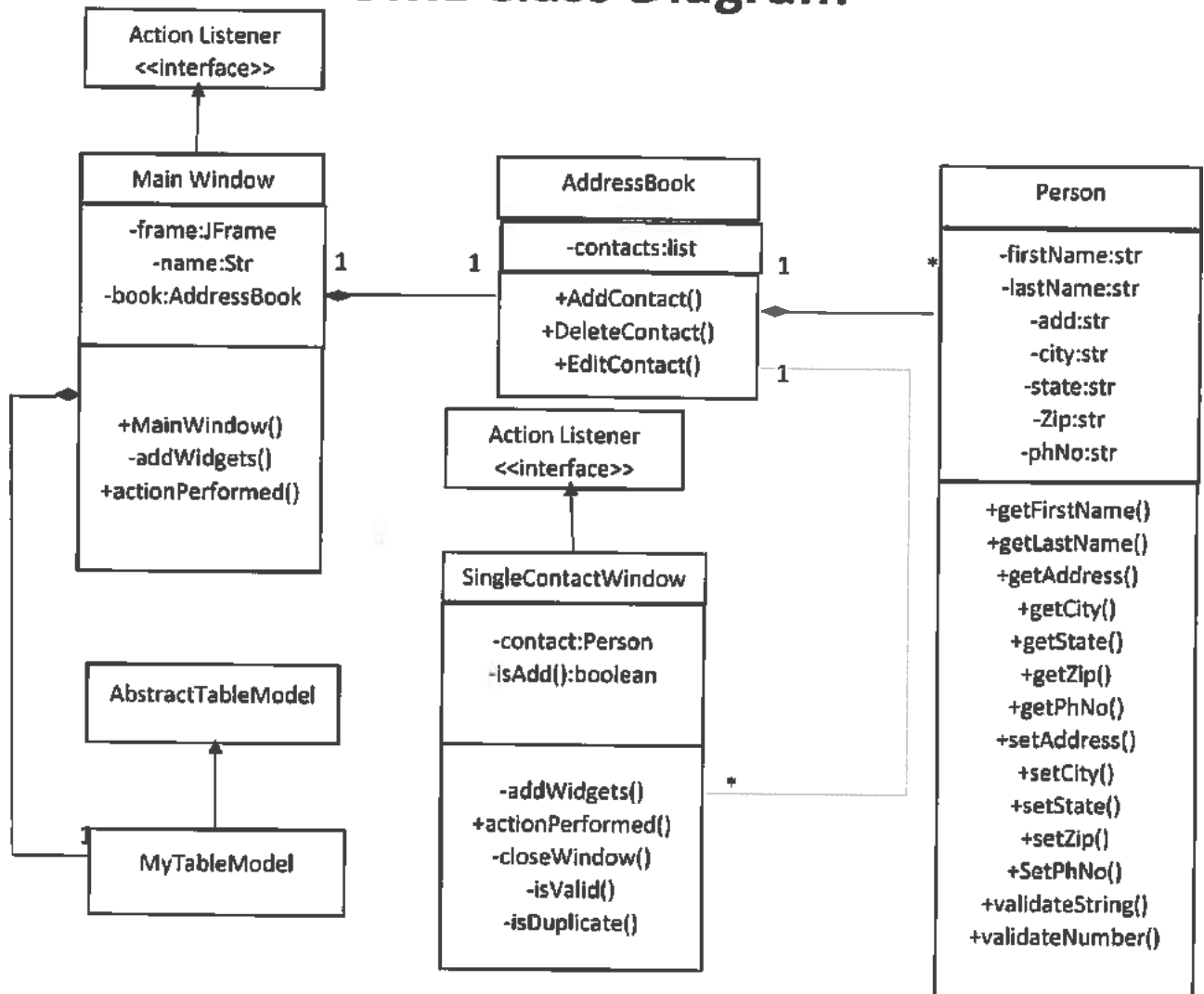
1. Person details to be deleted is made ready.
2. User highlights the person whose details has to be deleted.
3. User clicks delete.
4. System cannot delete the entry from the address book.
5. A failure message is displayed.

Frequency of Occurrence: Could be nearly continuous.

USE CASE DIAGRAM

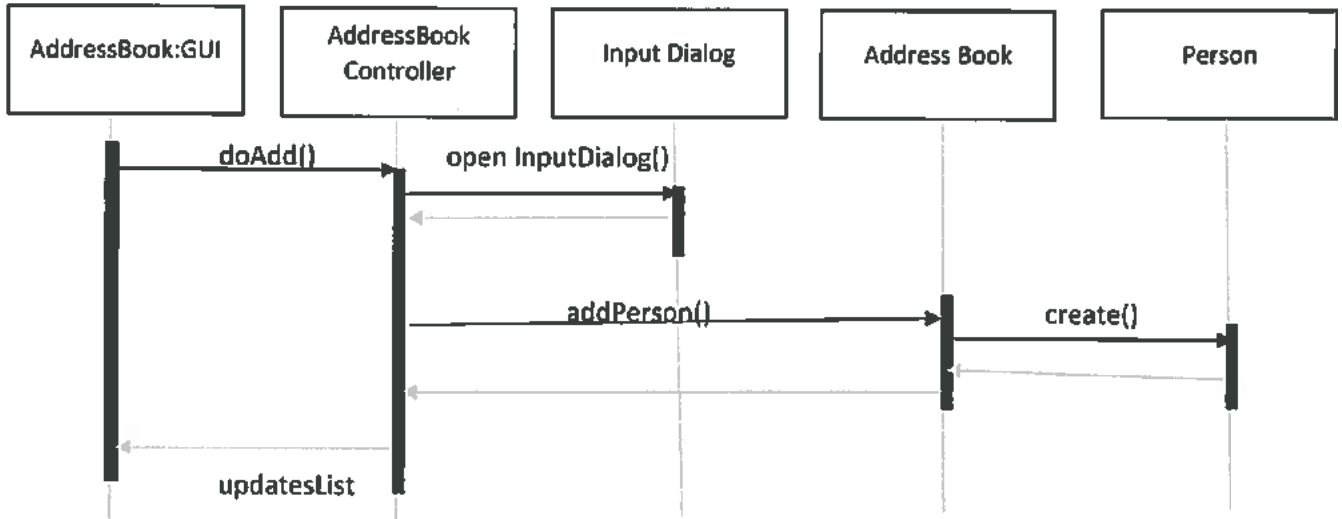


UML Class Diagram

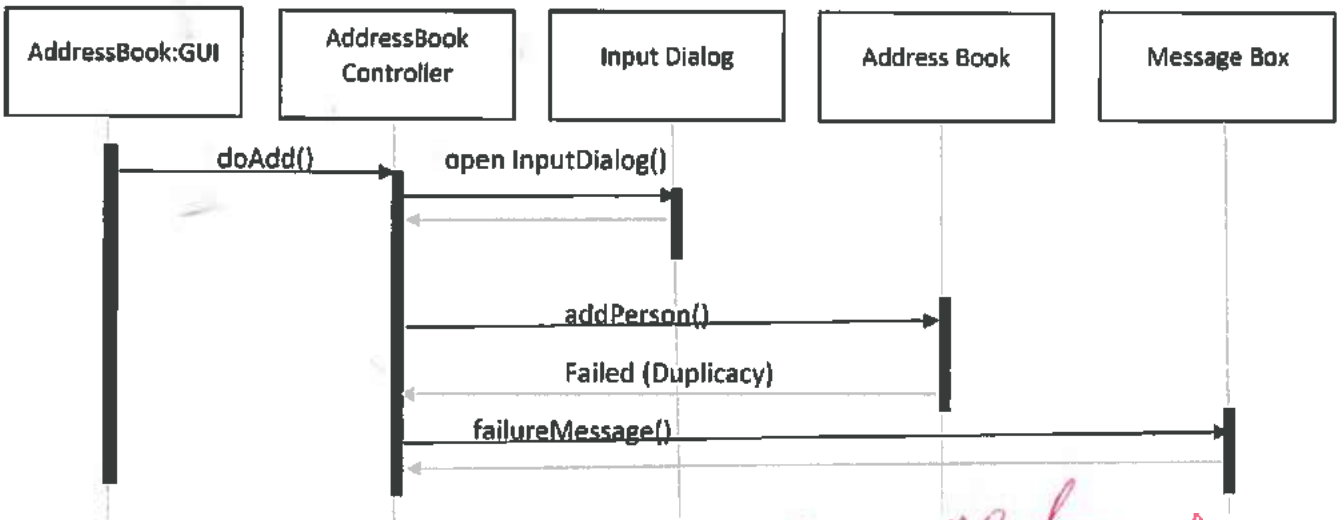


SYSTEM SEQUENCE DIAGRAM

Add a person:Success Scenario

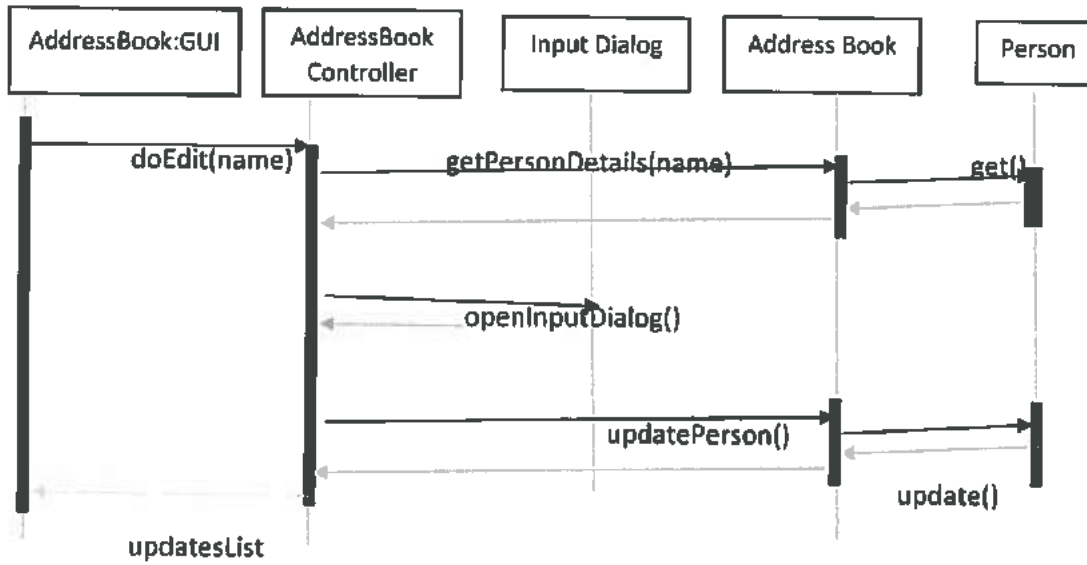


Add a person:Failure Scenario

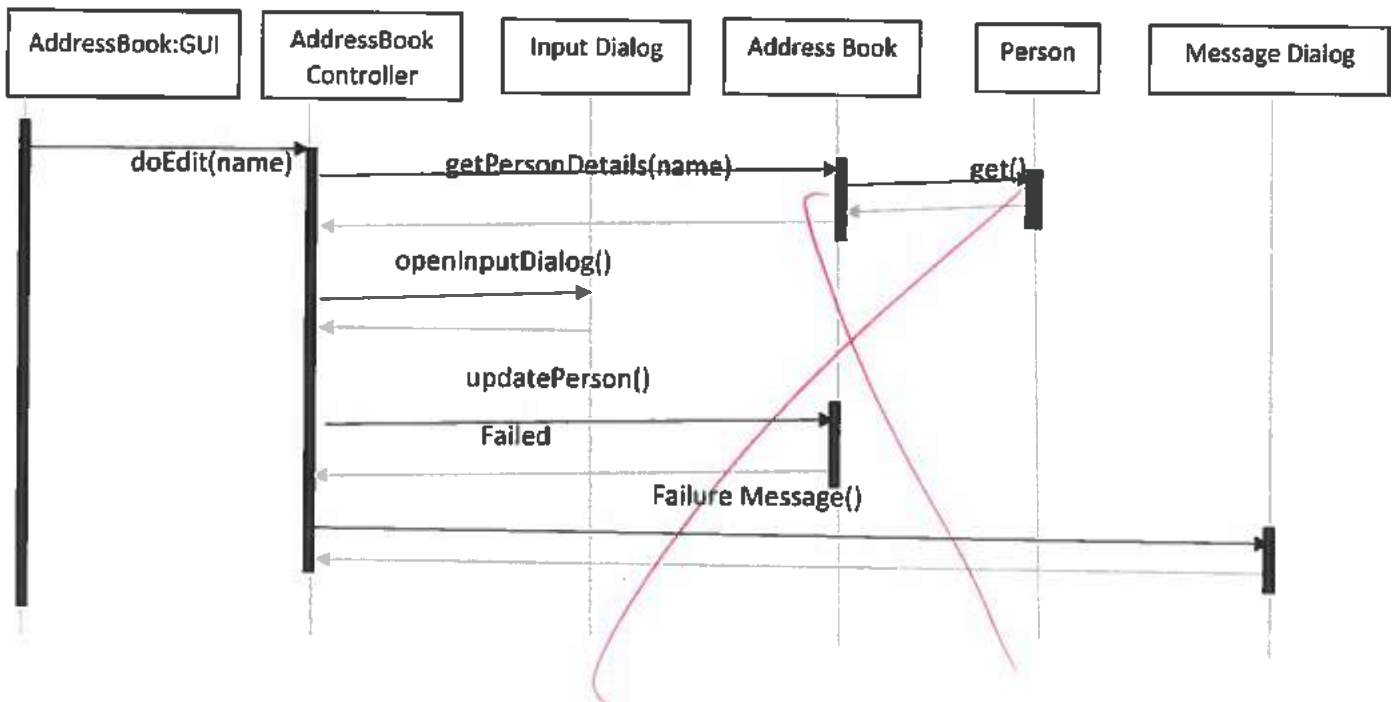


not correct !

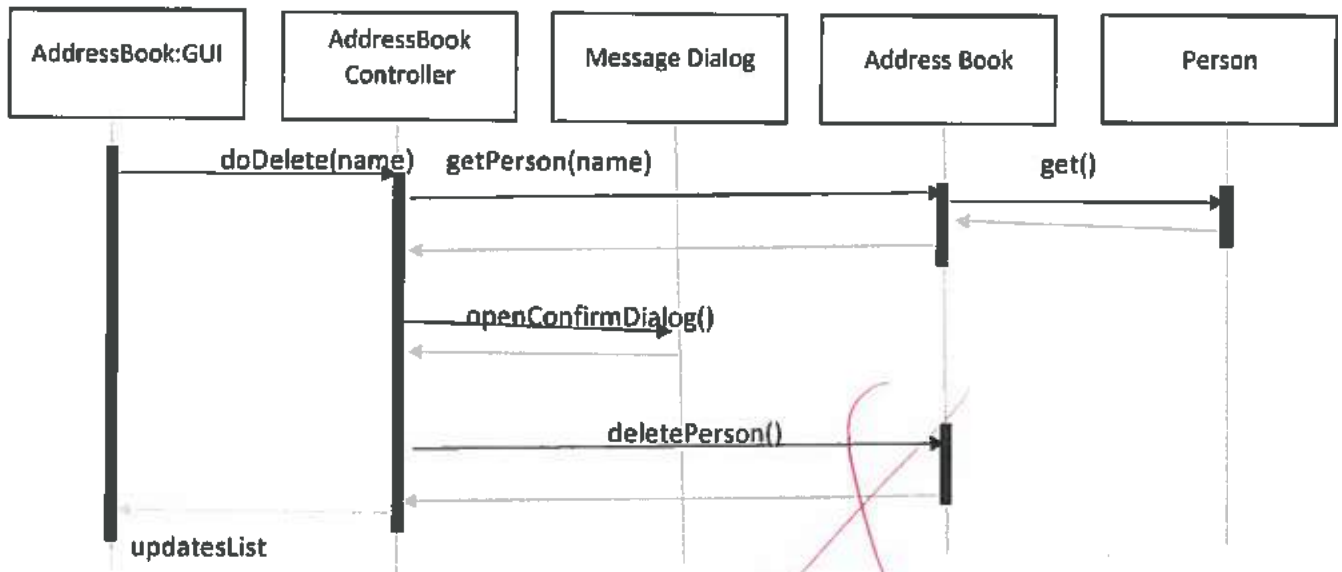
Edit a person: Success scenario



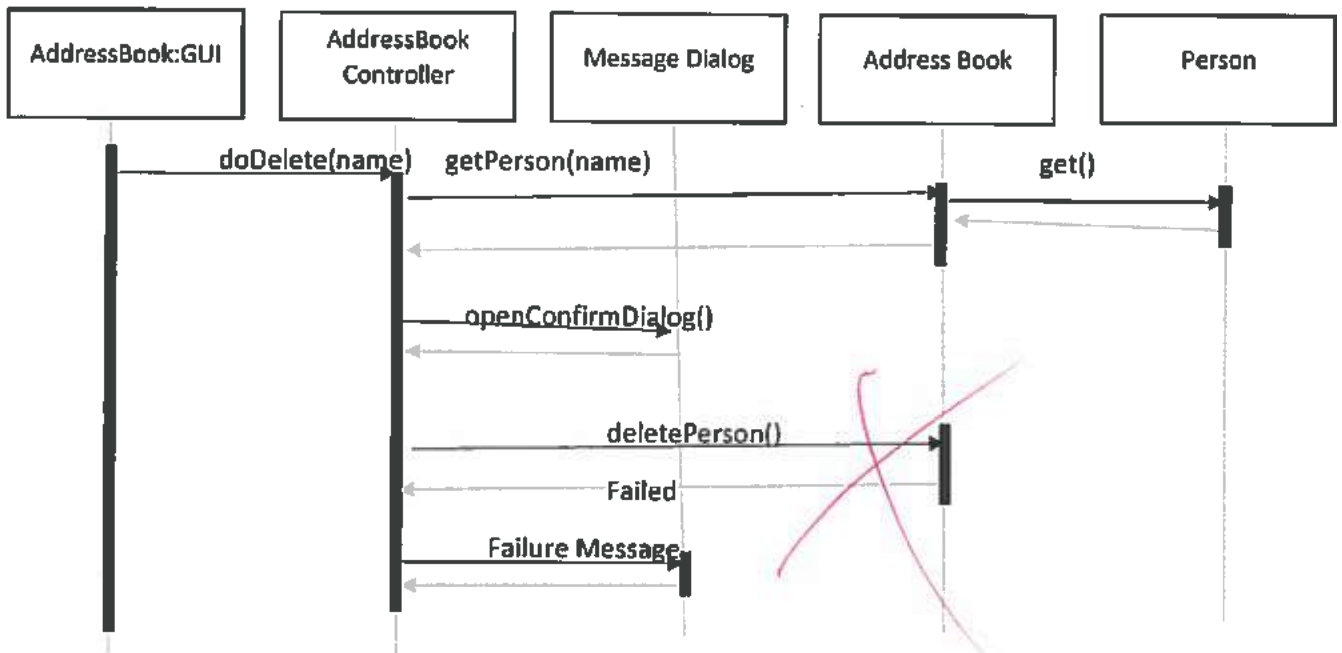
Edit a person: Failure scenario



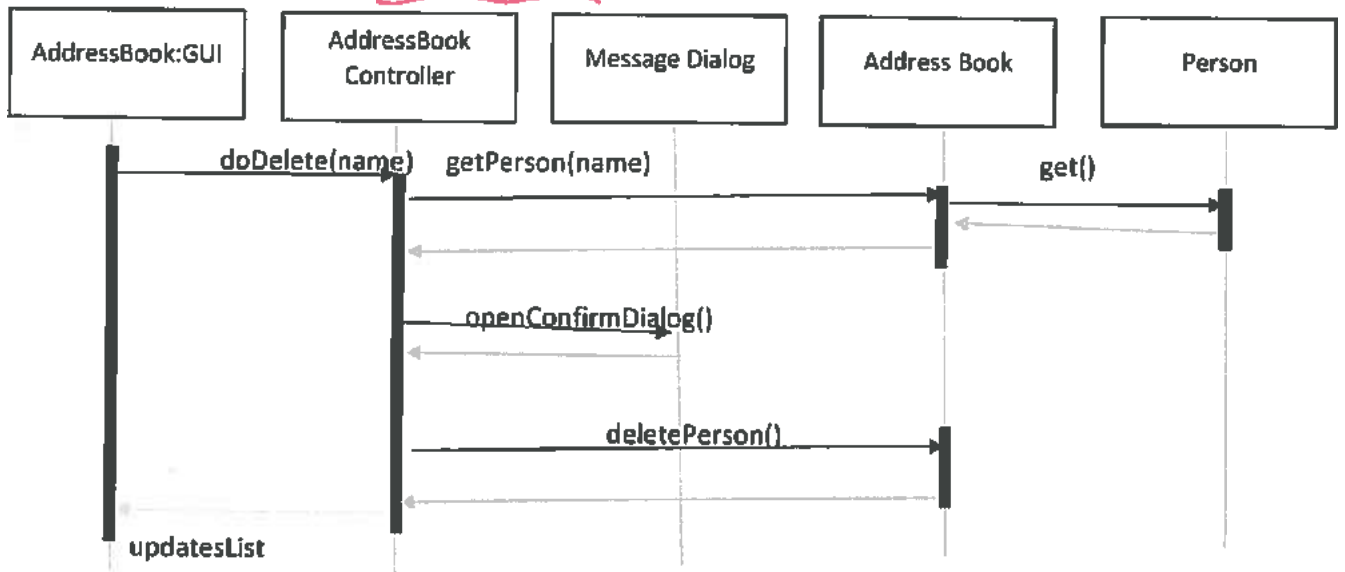
Delete a person:Success Scenario



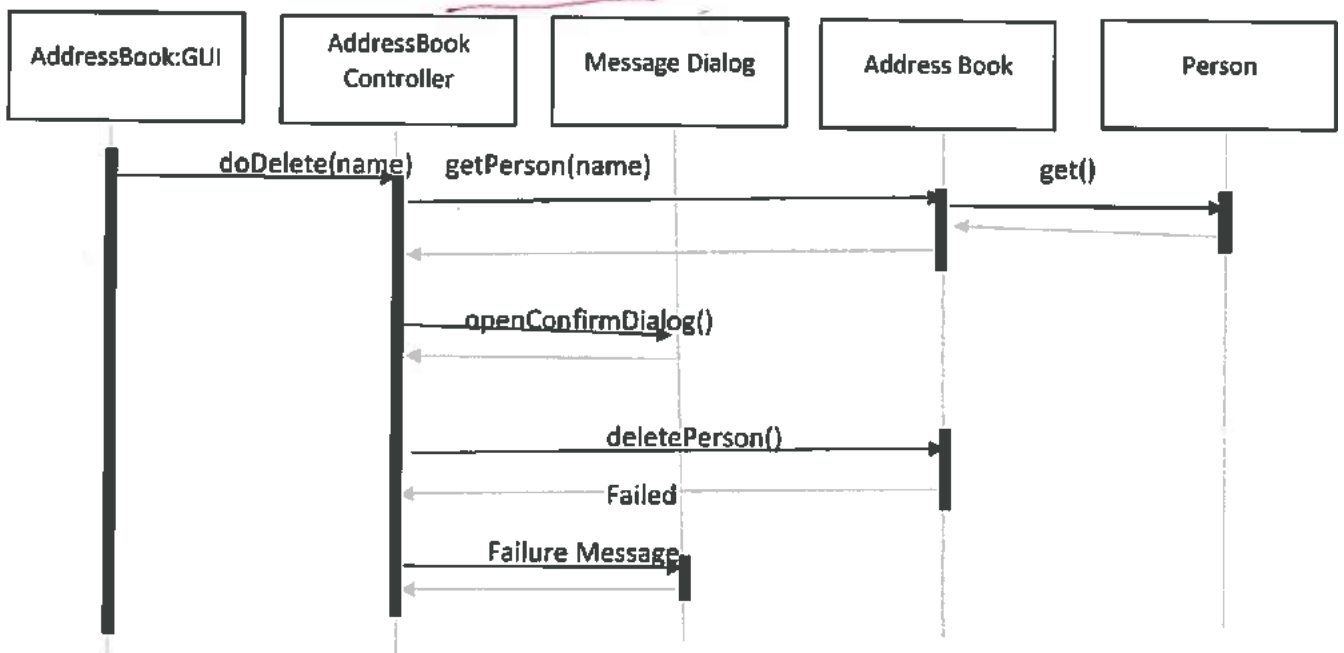
Delete a person:Failure Scenario



Delete a person: Success Scenario



Delete a person: Failure Scenario



operation contracts