

Future University in Egypt

Faculty of computers and information
Technology

Department of computer Science



Graduation Project
Social Real Estate System



SRES

Team Work

Omnia Ahmed Fathy

Shaimaa Siraj Ezzat

Abdelaziz Fayed Abdelaziz

Ali Ahmed Mohamed

Supervised by

Dr. Mahmoud Sami Othman

TA. Hadeer Khaled Tawfik

Academic Year

2020-2021

Acknowledgement

Contents

Chapter 1: Introduction

1.1. Introduction

Real estate and Booking websites have become popular but despite that, customers still search for properties through brokers.

Brokers are not always reliable since their work is mostly disorganized, therefore both customer and owner of property face difficulty finding what they are looking for which causes cooperation to lose customers and owners due to dissatisfaction. Owners also face difficulty managing their properties due to outdated and disorganized data recording. So the advantage of Real state website is maximizing reservations, You get paid quicker, You can effortlessly up-sell add-ons, It's easy to manage your calendar and You get valuable insight about your business.

On the other hand social media is online communication that allows you to interact with your customers and share information in real time. This can help you to reach your customers better, create online networks and sell and promote your products and services.

There are many different social media sites for different purposes, for example, LinkedIn is a social network for professionals, while Snapchat is a social network targeted mostly at teenagers (Based on this data).

Social media can help you engage with your customers and find out what people are saying about your business. You can also use social media for advertising, promotional giveaways and mobile applications.

Our aim is to produce a new genre of booking website that takes advantage of social web applications “marketing, attracting customers and getting customer feedback and build customer loyalty also to increase traffic to your website” to add more advantage to the real estate web applications. In our web application this will serve both the owner and the customer.

What are strength points about such app ?

A unique experience to the customer and owners to use a social real estate web application unlike traditional real estate web applications. Our web application will attract users by its friendly user interface and will help customers to find easily a property with their requirements and the system will market directly owner's property by customers interactions reviews , posts and the organized display of showing owner's property categories and details.

1.2. Problem Definition

Renting a property is a process that require many features like rules , security , organization , payment , reviews and marketing. Owners and customers both are facing kind of problems. Owners are facing a problem of organizing and manage their properties. The difficulty of finding a renter for the owner's property and this problem leads to gaining zero benefit from this property. According to customers they are facing a problem of founding properties in the place they want. Customers face difficulties of knowing the detail of the property like the schedule , price and number of rooms. The difficulty of knowing if a favorite property for the customer is available at a specific date or not. Properties are displayed in an unattractive traditional way that does not assure customer of the property level. The problem of that the customer need to see other people reviews and experience with this property. The poor financial management is also a problem that owner and customer both are facing. Difficulties in money transactions. The difficulties of finding a property with customer's requirements. Information given to customers by brokers are not always reliable. Difficulty of owners to have a history of all reservations on all his properties. Owners are facing difficulties finding customers of their properties.

1.3.Functional Requirements

1- Log in

System allow user to log in.

2- Register

System allow user to register.

3- Log out

System allow user to log out.

4- Forget Password

System allow user to make a new password.

5- Search by place

Customer can search for the state to rent a property.

6- Search by place and date

Customer can search for the state and the available property to rent by schedule.

7- Item Profile

System show item profile for all customers.

8- Item Galleries

System shows item galleries for all customers.

9- Item Location

System shows item location for all customers.

10- Item Reservations

System shows item reservations for owner.

11- Item Schedule

System shows item schedules for all customers.

12- Item reviews

System shows item reviews for all customers.

13- Item Owner's Profile

System shows item owner's profile for all customers.

14- Customer Comment

System allow customers to write a comment on a post.

15- Customer Replay

System allow customers to replay on a comment in a post.

16- Customer Edit Comment

System allow customers to edit their comment on a post.

17- Customer Edit Replay

System allow customers to edit their replay on a post.

18- Customer Delete Comment

System allow customers to delete their comment in a post.

19- Customer Delete Replay

System allow customers to delete their replay in a post.

20- Owner Comment

System allow owners to write a comment on a post .

21- Owner Replay

System allow owners to replay on a comment in a post.

22- Owner Edit Comment

System allow owners to edit their comment on a post.

23- Owner Edit Replay

System allow owners to edit their replay on a post.

24- Owner Delete Comment

System allow owners to delete their comment on a post.

25- Owner Delete Replay

System allow owners to delete their replay on a post.

26- Customer Comment

System allow customers to write a comment on a review.

27- Customer Replay

System allow customers to replay on a comment in a review.

28- Customer Edit Comment

System allow customers to edit their comment on a review.

29- Customer Edit Replay

System allow customers to edit their replay on a review.

30- Customer Delete Comment

System allow customers to delete their comment in a review.

31- Customer Delete Replay

System allow customers to delete their replay in a review.

32- Owner Comment

System allow owners to write a comment on a review.

33- Owner Replay

System allow owners to replay on a comment in a review.

34- Owner Edit Comment

System allow owners to edit their comment on a review.

35- Owner Edit Replay

System allow owners to edit their replay on a review.

36- Owner Delete Comment

System allow owners to delete their comment on a review.

37- Owner Delete Replay

System allow owners to delete their replay on a review.

38- Owner Edit Review

System allow customers to edit their review on a post.

39- Show Customer Reservations

System allow customer to view his previous reservations.

40- Rent

Customer can rent a property.

41- Show items reservations

System display for the owner all reservations on all his items.

42- Customer edit profile

Customer can edit his information in the profile.

43- Owner edit profile

Owner can edit his information in the profile.

44- Be Owner

System allow customer to be owner.

45- Add Property

Owner can add his property to the system.

46- Delete Property

Owner can delete his property from the system.

47- Payment

System allow customer to pay online by paypal.

48- Customer Review

System allow customer to rate a property.

49- Cancel rent

Owner can cancel his property from rent.

50- Edit property details

Owner can edit property details.

51- Item Schedule by owner

Owner can enter the available days of his item.

52- Update Item schedule

Owner can update item schedule

53- Show owner items

Customer can find owner's available items from his profile.

- 54- Send Notification to owner**
System can send notification to owner .
- 55- Send Notification to customer**
System can send notification to customer.
- 56- Add Profile Photo**
Owner can add a profile photo.
- 57- Update Profile Photo**
Owner can update his profile photo.
- 58- Delete Profile Photo**
Owner can delete his profile photo.
- 59- Add Profile Photo**
Customer can add a profile photo.
- 60- Update Profile Photo**
Customer can update his profile photo.
- 61- Delete Profile Photo**
Customer can delete his profile photo.
- 62- Add Cover Photo**
Owner can add a cover photo.
- 63- Update Cover Photo**
Owner can update his cover photo.
- 64- Delete Cover Photo**
Owner can delete his cover photo.
- 65- Add Cover Photo**
Customer can add a cover photo.
- 66- Update Cover Photo**
Customer can update his cover photo.
- 67- Delete Cover Photo**
Customer can delete his cover photo.
- 68- Customer Add Post**
Customer can add a post.
- 69- Customer Edit Post**
Customer can edit post.
- 70- Customer Delete Post**
Customer can delete post.
- 71- Owner Add Post**
Owner can add a new post.

72- Owner Edit Post

Owner can edit post.

73- Owner Delete Post

Owner can delete post.

74- Add User

Admin can add a new user.

75- Edit User

Admin can edit user information.

76- Delete User

Admin can delete user.

77- Show Users

System can display users for the admin.

78- Add User Type

Admin can add a new user type.

79- Edit User Type

Admin can edit user type name.

80- Delete User Type

Admin can delete user type name.

81- Show User Types

System can display user types for the admin.

82- Add Data Type

Admin can add a new data type.

83- Edit Data Type

Admin can edit data type name.

84- Delete Data Type

Admin can delete data type name.

85- Show Data Types

System can display data types for the admin.

86- Add Data Type

Admin can add a new data type.

87- Edit Data Type

Admin can edit data type name.

88- Delete Data Type

Admin can delete data type name.

89- Show Data Types

System can display data types for the admin

90- Add Country

Admin can add a new country.

91- Edit Country

Admin can edit country.

92- Delete Country

Admin can delete country.

93- Show Countries

System can display countries for the admin.

94- Add State

Admin can add a new state.

95- Edit State

Admin can edit state.

96- Delete State

Admin can delete state.

97- Show States

System can display states for the admin.

98- Add City

Admin can add a new city.

99- Edit City

Admin can edit city.

100- Delete City

Admin can delete city.

101- Show Cities

System can display cities for the admin.

102- Add Region

Admin can add a new region.

103- Edit Region

Admin can edit region.

104- Delete Region

Admin can delete region.

105- Show Regions

System can display regions for the admin.

106- Add Street

Admin can add a new street.

107- Edit Street

Admin can edit street.

108- Delete Street

Admin can delete street.

109- Show Streets

System can display streets for the admin.

110- Add Main Type

Admin can add a new main type.

111- Edit Main Type

Admin can edit main type.

112- Delete Main Type

Admin can delete main type.

113- Show Main Types

System can display main types for the admin.

114- Add Sub Type

Admin can add a new sub type.

115- Edit Sub Type

Admin can edit sub type.

116- Delete Sub Type

Admin can delete sub type.

117- Show Sub Types

System can display sub types for the admin.

118- Add Detail

Admin can add a new detail.

119- Edit Detail

Admin can edit detail.

120- Delete Detail

Admin can delete detail.

121- Show Details

System can display details for the admin.

122- Add Detail Value

Admin can add a new detail value.

123- Edit Detail Value

Admin can edit detail value.

124- Delete Detail Value

Admin can delete detail value.

125- Show Detail Values

System can display detail values for the admin.

126- Add Operation

Admin can add a new operation.

127- Edit Operation

Admin can edit operation.

128- Delete Operation

Admin can delete operation.

129- Show Operations

System can display operations for the admin.

130- Add Operation Type

Admin can add a new operation type.

131- Edit Operation Type

Admin can edit operation type.

132- Delete Operation Type

Admin can delete operation type.

133- Show Operation Type

System can display operation types for the admin.

134- Add Operation Type Name

Admin can add a new operation type name.

135- Edit Operation Type Name

Admin can edit operation type name.

136- Delete Operation Type Name

Admin can delete operation type name.

137- Show Operation Type Name

System can display operation type names for the admin.

138- Add Operation Type Value

Admin can add a new operation type value.

139- Edit Operation Type Value

Admin can edit operation type value.

140- Delete Operation Type Value

Admin can delete operation type value.

141- Show Operation Type Value

System can display operation type values for the admin.

142- Add Item

Admin can add a new item.

143- Edit Item

Admin can edit item.

144- Delete Item

Admin can delete item.

145- Show Items

System can display items for the admin.

146- Follow

System allow customer to follow the property

147- Unfollow

System allow customer to un follow the property

148- Follow

System allow customer to follow the customer

149- Unfollow

System allow customer to un follow the customer

150- Newest items

System display the newest item to the customer.

1.4. Non Functional Requirements

1.4.1 Product requirements

- **Efficiency:** Application should provide accurate location and clear information about each property with ease.
- **Usability requirements:** User should be able to report any difficulties they faced in the system for developers to fix.
- **Security requirements:** No user should be given owner privileges without completing their information. Any type of user cannot access other type privileges

1.4.2 Additional requirements

- **User interface:** User interface should be user friendly and suitable for socializing. User interface should be simple and easy to use. Owner cannot show item with uncompleted details.

Chapter 2: Background

2.1. System Tools

- **Laravel Framework**

Laravel is primarily used for building custom web apps using PHP. It's a web framework that handles many things that are annoying to build yourself, such as routing, templating HTML, and authentication.

Laravel is entirely server-side, due to running on PHP, and focuses heavily on data manipulation and sticking to a Model-View-Controller design. A framework like React might put most of its attention on user interaction and shiny features, but Laravel simply presents a solid foundation for you to build off—and does it right.

- **PHP**

PHP is a recursive acronym for “PHP: Hypertext Preprocessor”. PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites. It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.

- **PhpMyAdmin**

phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. phpMyAdmin supports a wide range of operations on MySQL and MariaDB. Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc) can be performed via the user interface, while you still have the ability to directly execute any SQL statement.

- **MySQL**

MySQL is an open-source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

- **Xampp**

XAMPP is a software distribution which provides the Apache web server, MySQL database (actually MariaDB), PHP and Perl (as command-line executables and Apache modules) all in one package. ... Also provided is PhpMyAdmin which gives a GUI tool for managing your MySQL databases.

- **Composer**

Composer is a tool for dependency management in PHP. It allows you to declare the libraries your project depends on and it will manage (install/update) them for you.

Suppose:

1. You have a project that depends on a number of libraries.
2. Some of those libraries depend on other libraries.

Composer:

1. Enables you to declare the libraries you depend on.
2. Finds out which versions of which packages can and need to be installed , and installs them (meaning it downloads them into your project).
3. You can update all your dependencies in one command.

- **Git**

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

- **Git hub**

GitHub is a web-based version-control and collaboration platform for software developers, Git is used to store the source code for a project and track the complete history of all changes to that code. It allows developers to collaborate on a project more effectively by providing tools for managing possibly conflicting changes from multiple developers.

- **JavaScript**

JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

- **JQuery**

JQuery is an open-sourced JavaScript library that simplifies creation and navigation of web applications. Specifically, jQuery simplifies HTML Document Object Model (DOM) manipulation, Asynchronous JavaScript and XML (Ajax) and event handling. Additionally, jQuery incorporates JavaScript functionalities by manipulating CSS properties to add effects such as fade-ins and outs for website elements. jQuery is a widely used JavaScript library and is supported by thousands of user-created plug-ins.

- **AJAX**

Asynchronous JavaScript and XML. AJAX is a technique for creating fast and dynamic web pages. AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

- **Html**

The Hyper Text Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript.

- **CSS**

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

- **Bootstrap**

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

- **PhpStorm**

PhpStorm is an innovative, Java-based integrated development environment (IDE) engineered by JetBrains for PHP and web developers. It supports PHP 5.3/5.4/5.5/5.6/7.0/7.1/7.2/8.0, provides on-the-fly error prevention, best auto completion and code refactoring, zero-configuration debugging, and an extended HTML, CSS and JavaScript editor.

- **Visual Studio Code**

Visual Studio Code is a free and open-source code editor developed by Microsoft. Visual Studio Code is often called VS Code, VS Code works across platforms including Windows, Linux, and macOS.

- **draw.io**

draw.io is completely free online diagram editor built around Google Drive(TM), that enables you to create flowcharts, UML, entity relation, network diagrams, mockups and more.

- **Lucidchart**

Lucidchart, created by Lucid Software, is a collaborative web-based diagramming and flowchart application. It is built on web standards such as HTML5 and Javascript and as such can be accessed from any device with a browser. It can be used to create flowcharts, iPhone app mockups, mind maps, organizational charts, website wireframes, UML diagrams, network diagrams, and many other diagram types.

Chapter 3: System Analysis

3.2. Use Case Models

3.2.1 Admin Use Case



Figure 1: Admin Use Case

3.2.2 User Login Use Case

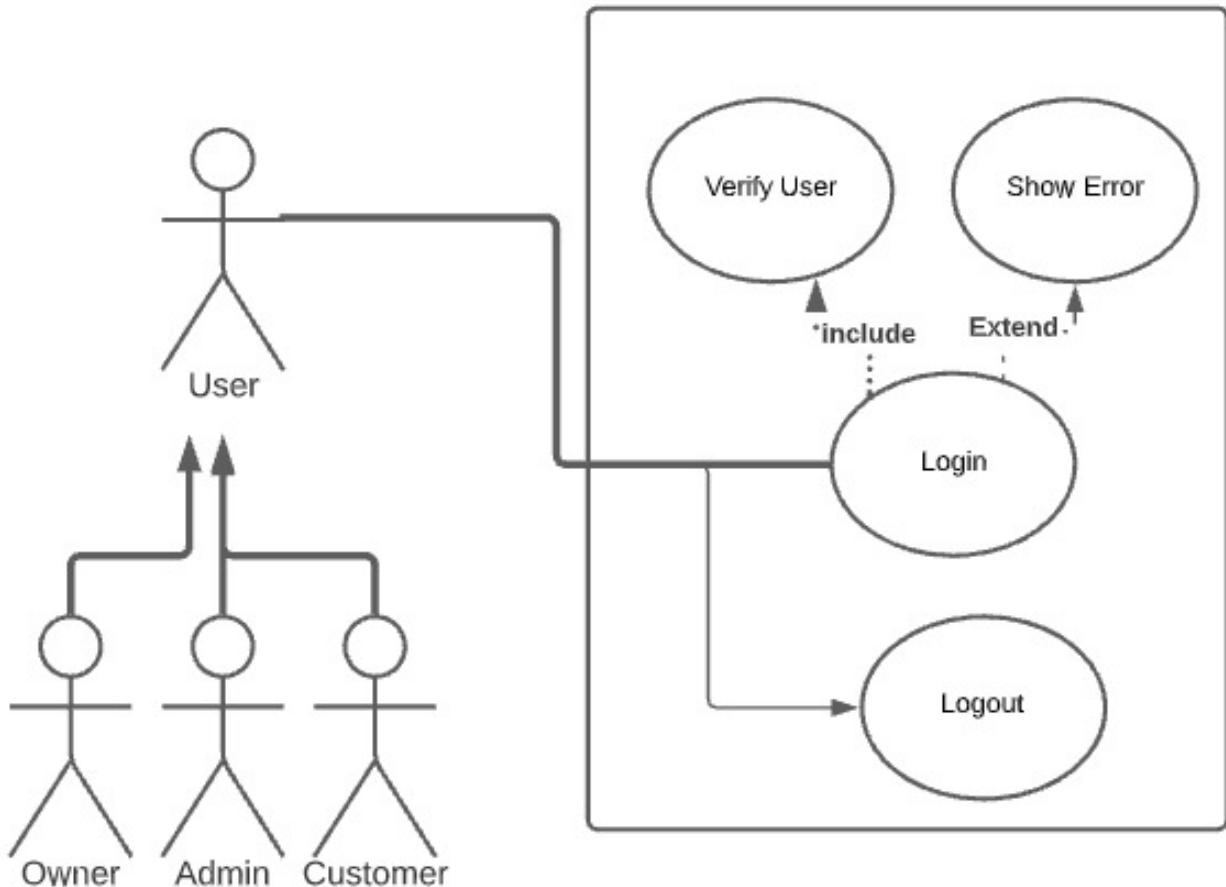


Figure 2:User Login Use Case

3.2.3 Admin and Owner Use Case

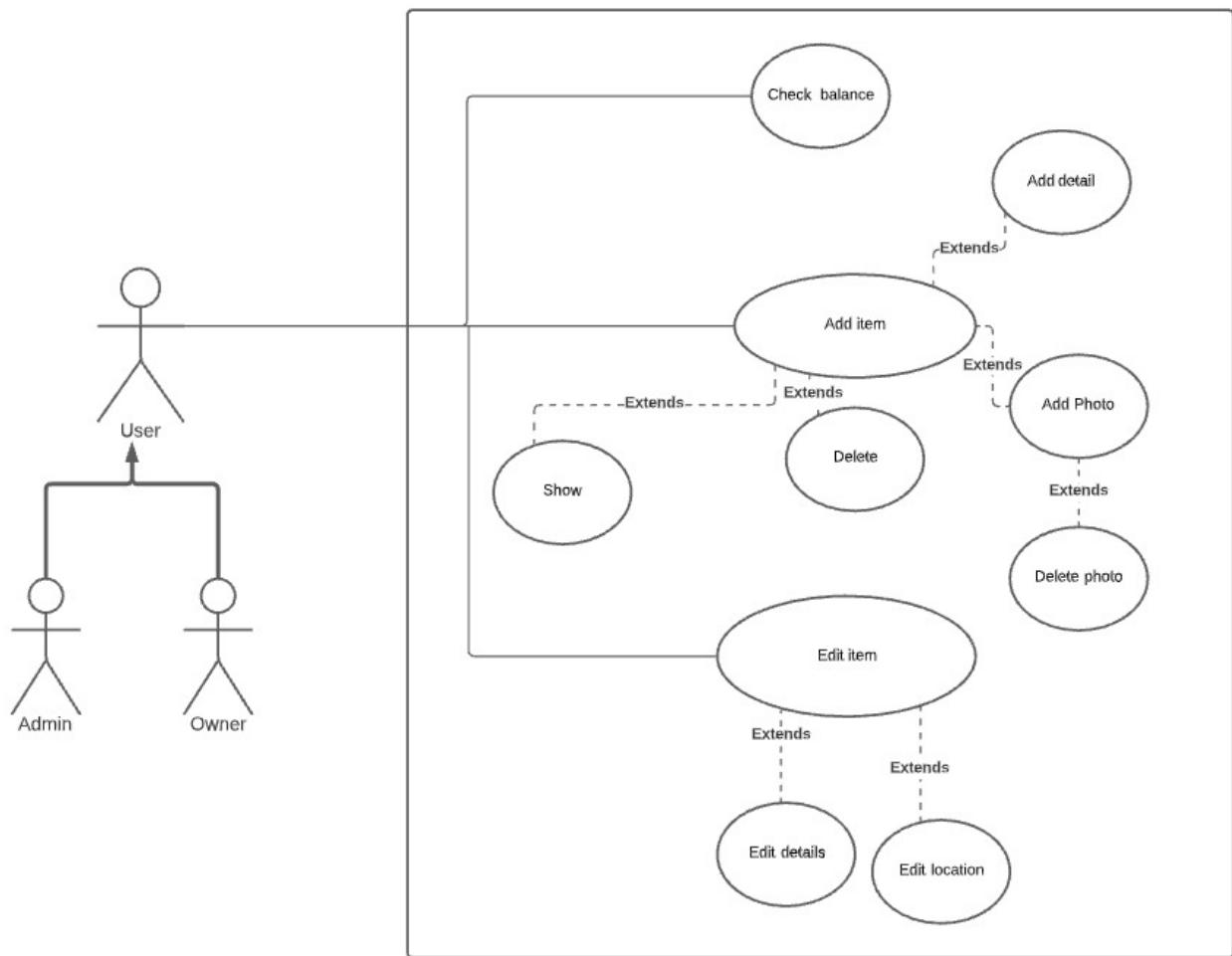


Figure 3:Admin and Owner Use Case

3.2.4 Owner and Customer Use Case

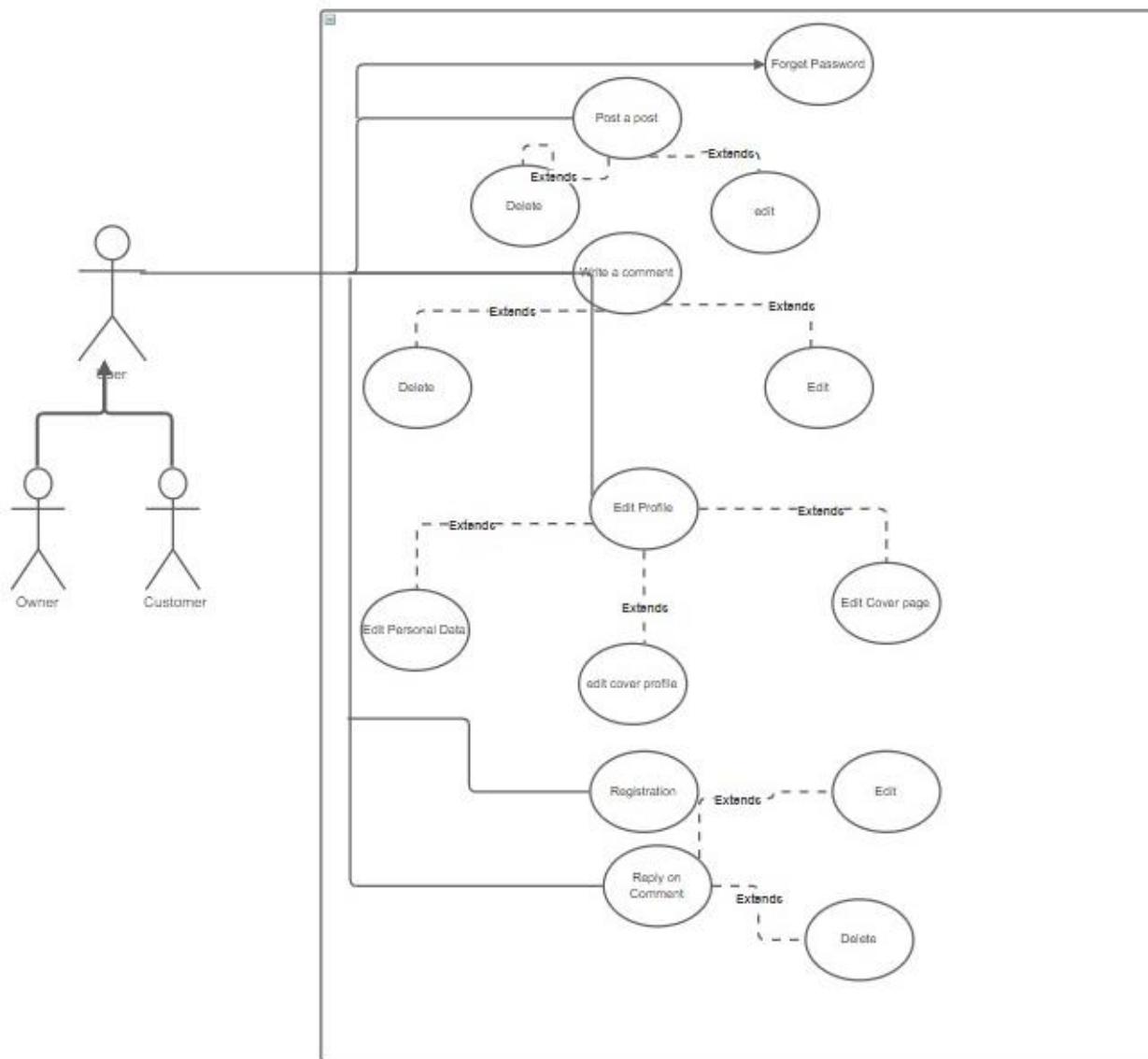


Figure 4: Owner and Customer Use Case

3.2.5 Customer Use Case

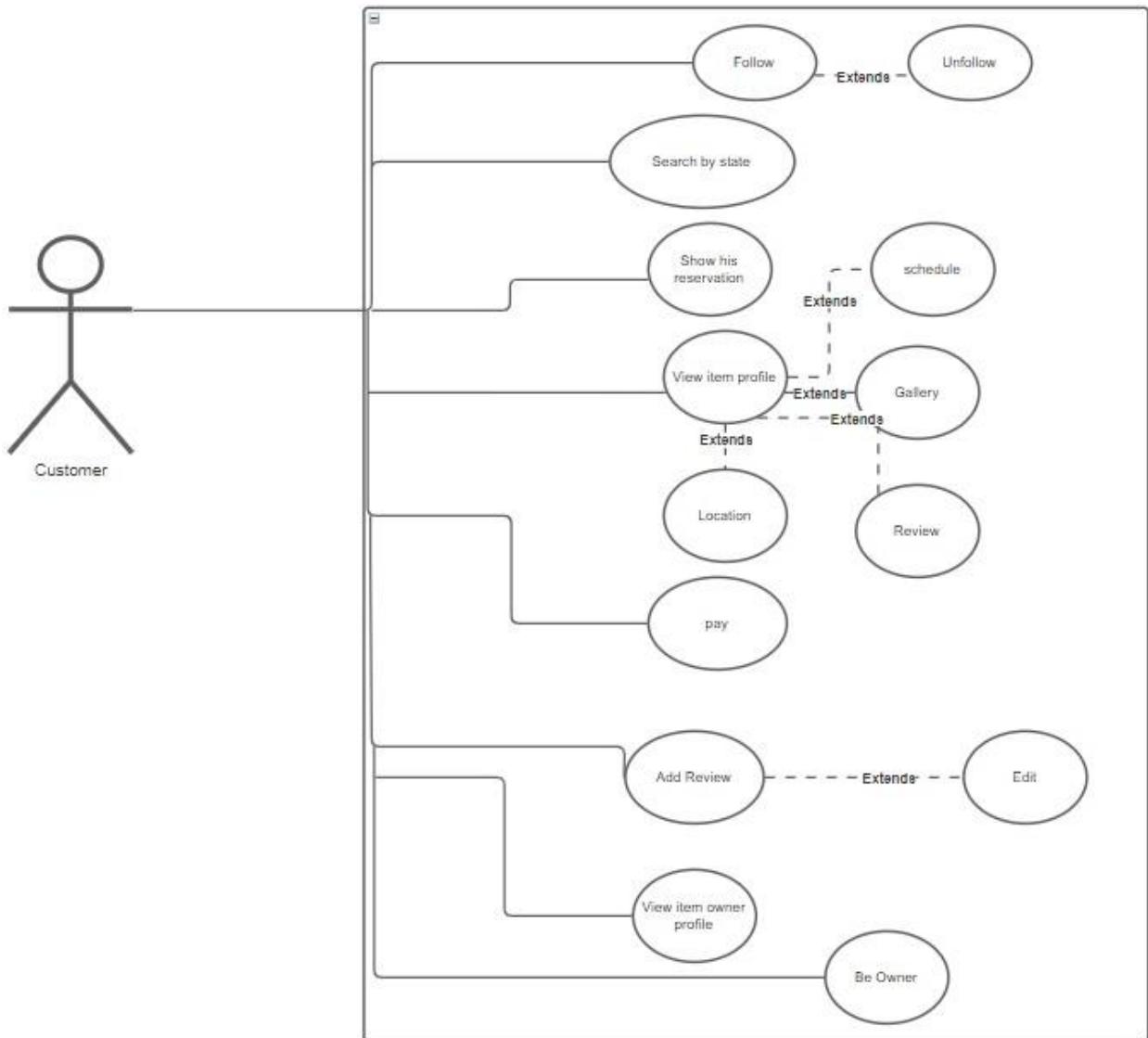


Figure 5:Customer Use Case

3.2.6 Owner Use Case

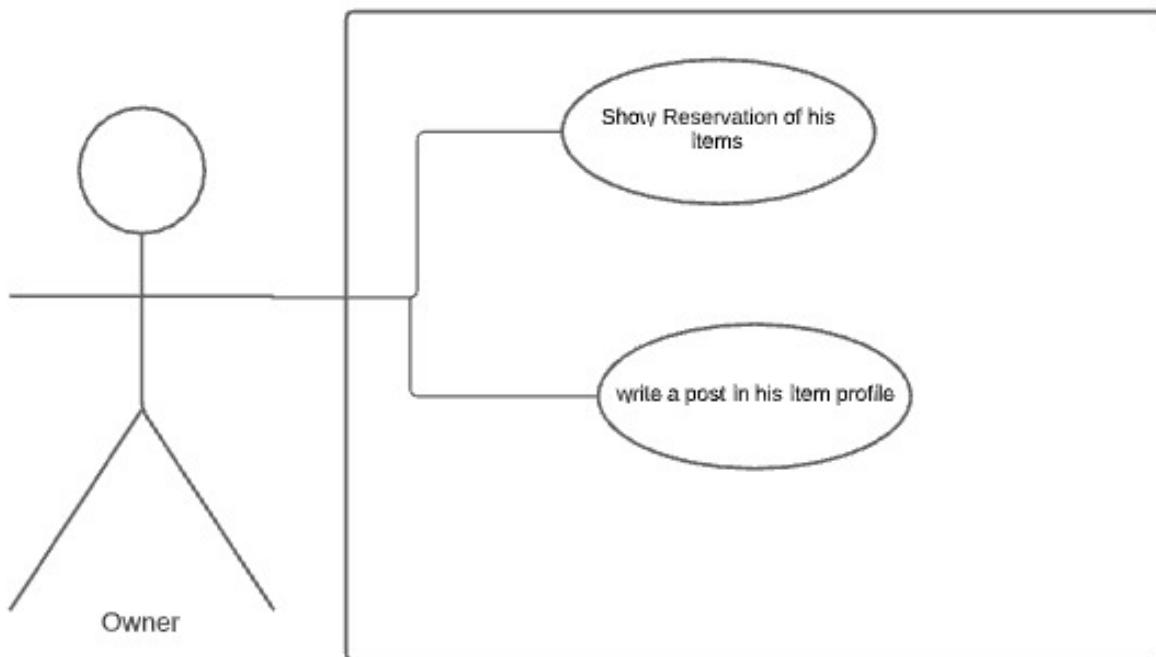


Figure 6: Owner Use Case

3.3 Use Case Specification Tables

Use Case Id	SRES-US-1	
Use Case Name	Registration	
Actor	Customer ,Owner	
Preconditions	<ul style="list-style-type: none"> • None 	
Postconditions	<ul style="list-style-type: none"> • Verify his account 	
Exceptions	<ul style="list-style-type: none"> • If user entered wrong information • If user miss fill required fields 	
Flow of Events	User Steps	System Steps
	1- The Actor will fill Sign up fields	2- The system will display verifying your profile

Table 1: Registration

Use Case Id	SRES-US-2	
Use Case Name	Log In	
Actor	Admin, Customer ,Owner	
Preconditions	<ul style="list-style-type: none"> • The Actor must have account 	
Postconditions	<ul style="list-style-type: none"> • None 	
Exceptions	<ul style="list-style-type: none"> • If user entered wrong information • If user miss fill required fields 	
Flow of Events	User Steps	System Steps
	1-The Actor will fill Log In fields	2-The system will display dashboard

Table 2: Log In

Use Case Id	SRES-US-3	
Use Case Name	Forget password	
Actor	Customer ,Owner	
Preconditions	<ul style="list-style-type: none"> Sign in page 	
Postconditions	<ul style="list-style-type: none"> Change password 	
Exceptions	<ul style="list-style-type: none"> If user entered wrong information If user miss fill required fields 	
Flow of Events	User Steps	System Steps
	1-The Actor will fill Log In fields	2-The system will display dashboard

Table 3: Forget password

Use Case Id	SRES-US-4	
Use Case Name	Add User	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> The admin must have account 	
Postconditions	<ul style="list-style-type: none"> A success message will be displayed and a new user is added 	
Exceptions	<ul style="list-style-type: none"> If admin entered wrong information If admin miss fill required fields 	
Flow of Events	User Steps	System Steps
	1- The admin will select type of the new user 2- The admin will fill new user form	3-The system will take new data and create new user 4-The system will display success message

Table 4: Add User

Use Case Id	SRES-US-5					
Use Case Name	Edit User					
Actor	User					
Preconditions	<ul style="list-style-type: none"> The user must be logged in 					
Postconditions	<ul style="list-style-type: none"> A success message will be displayed and data is updated 					
Exceptions	<ul style="list-style-type: none"> If user entered wrong information 					
Flow of Events	<table border="1"> <thead> <tr> <th>User Steps</th> <th>System Steps</th> </tr> </thead> <tbody> <tr> <td> 1- The user will click on edit 3- The user will enter the new data </td> <td> 2- The system will display a pop up form for editing 4- The System will return the page with the new data and display a success message </td> </tr> </tbody> </table>	User Steps	System Steps	1- The user will click on edit 3- The user will enter the new data	2- The system will display a pop up form for editing 4- The System will return the page with the new data and display a success message	
User Steps	System Steps					
1- The user will click on edit 3- The user will enter the new data	2- The system will display a pop up form for editing 4- The System will return the page with the new data and display a success message					

Table 5: Edit User

Use Case Id	SRES-US-6					
Use Case Name	Delete User					
Actor	Admin					
Preconditions	<ul style="list-style-type: none"> The admin must be logged in 					
Postconditions	<ul style="list-style-type: none"> A success message will be displayed with user is deleted 					
Exceptions	<ul style="list-style-type: none"> None 					
Flow of Events	<table border="1"> <thead> <tr> <th>User Steps</th> <th>System Steps</th> </tr> </thead> <tbody> <tr> <td> 1- The admin will select the type of user 3- The admin will select the user or group of users to delete </td> <td> 2- The system will show all users with type selected 4- The system will return the page without the deleted users with success message </td> </tr> </tbody> </table>	User Steps	System Steps	1- The admin will select the type of user 3- The admin will select the user or group of users to delete	2- The system will show all users with type selected 4- The system will return the page without the deleted users with success message	
User Steps	System Steps					
1- The admin will select the type of user 3- The admin will select the user or group of users to delete	2- The system will show all users with type selected 4- The system will return the page without the deleted users with success message					

Table 6: Delete User

Use Case Id	SRES-US-7	
Use Case Name	Show Users	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Users will be displayed 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1- The admin will select the type of users	2- The system will show all users with type selected

Table 7: Show Users

Use Case Id	SRES-US-8	
Use Case Name	Add User Type	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • A success message will be displayed 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1- The admin will enter the new type of users	3- The system will receive and save it in data base

Table 8: Add User Type

Use Case Id	SRES-US-9	
Use Case Name	Edit User Type	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • A success message will be displayed 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1- The admin will click on edit 3- The admin will enter the new user type name	2- The system will display a pop up form for editing 4- The System will return the page with the new user type and display a success message

Table 9: Edit User Type

Use Case Id	SRES-US-10	
Use Case Name	Delete User Type	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • A success message will be displayed with user type is deleted 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1- The admin will select the type of user to delete	2- The system will return the page without the deleted user type with success message

Table 10: Delete User Type

Use Case Id	SRES-US-11	
Use Case Name	Show User Types	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> User types will be displayed 	
Exceptions	<ul style="list-style-type: none"> None 	
Flow of Events	User Steps	System Steps
	1- The admin will click on show user types	2- The system will display all user types

Table 11: Show User Type

Use Case Id	SRES-US-12	
Use Case Name	Add Data Type	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> Success message will be displayed 	
Exceptions	<ul style="list-style-type: none"> If admin entered data already exists If the entered data does not match the required format 	
Flow of Events	User Steps	System Steps
	1- The admin will fill required field	2- The system will receive the data type and save it 3- The system will display success message

Table 12: Add Data Type

Use Case Id	SRES-US-13	
Use Case Name	Edit Data Type	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed and the data type is updated 	
Exceptions	<ul style="list-style-type: none"> • If the entered data does not match the required format 	
Flow of Events	User Steps	System Steps
	1- The admin will click on edit 3- The admin will enter the new data type name	2- The system will display a pop up form for editing with the current data 4- The System will return the page with the new data type and display a success message

Table 13: Edit Data Type

Use Case Id	SRES-US-14	
Use Case Name	Delete Data Type	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • A success message will be displayed with data type is deleted 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1- The admin will select the data type to delete	2- The system will return the page without the deleted data type with success message

Table 14: Delete Data Type

Use Case Id	SRES-US-15	
Use Case Name	Show Data Types	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> Data types will be displayed 	
Exceptions	<ul style="list-style-type: none"> None 	
Flow of Events	User Steps	System Steps
	1- The admin will click on show data types	2- The system will display all data types

Table 15: Show Data Types

Use Case Id	SRES-US-16	
Use Case Name	Add Country	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> Success message will be displayed 	
Exceptions	<ul style="list-style-type: none"> If admin entered data already exists If the entered data does not match the required format 	
Flow of Events	User Steps	System Steps
	1- The admin will fill required field	2- The system will receive the country name and save it 3- The system will display success message

Table 16: Add Country

Use Case Id	SRES-US-17					
Use Case Name	Edit Country					
Actor	Admin					
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 					
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed and the new country name 					
Exceptions	<ul style="list-style-type: none"> • If the entered data does not match the required format 					
Flow of Events	<table border="1"> <thead> <tr> <th>User Steps</th> <th>System Steps</th> </tr> </thead> <tbody> <tr> <td> 1- The admin will click on edit 3- The admin will enter the new country name </td> <td> 2- The system will display a pop up form for editing with the current data 4- The System will return the page with the new country name and display a success message </td> </tr> </tbody> </table>	User Steps	System Steps	1- The admin will click on edit 3- The admin will enter the new country name	2- The system will display a pop up form for editing with the current data 4- The System will return the page with the new country name and display a success message	
User Steps	System Steps					
1- The admin will click on edit 3- The admin will enter the new country name	2- The system will display a pop up form for editing with the current data 4- The System will return the page with the new country name and display a success message					

Table 17: Edit Country

Use Case Id	SRES-US-18					
Use Case Name	Delete Country					
Actor	Admin					
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 					
Postconditions	<ul style="list-style-type: none"> • A success message will be displayed with country is deleted 					
Exceptions	<ul style="list-style-type: none"> • None 					
Flow of Events	<table border="1"> <thead> <tr> <th>User Steps</th> <th>System Steps</th> </tr> </thead> <tbody> <tr> <td> 1- The admin will select the country to delete </td> <td> 2- The system will return the page without the deleted country with success message </td> </tr> </tbody> </table>	User Steps	System Steps	1- The admin will select the country to delete	2- The system will return the page without the deleted country with success message	
User Steps	System Steps					
1- The admin will select the country to delete	2- The system will return the page without the deleted country with success message					

Table 18: Delete Country

Use Case Id	SRES-US-19	
Use Case Name	Show Countries	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> All Countries will be displayed 	
Exceptions	<ul style="list-style-type: none"> None 	
Flow of Events	User Steps	System Steps
	1- The admin will click on show countries	2- The system will display all countries

Table 19: Show Countries

Use Case Id	SRES-US-20	
Use Case Name	Add State	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> Success message will be displayed 	
Exceptions	<ul style="list-style-type: none"> If admin entered data already exists If the entered data does not match the required format If admin didn't select which country he will add a new state to. 	
Flow of Events	User Steps	System Steps
	1- The admin will fill required fields	2- The system will receive the state name and save it 3- The system will display success message

Table 20: Add State

Use Case Id	SRES-US-21	
Use Case Name	Edit State	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> Success message will be displayed and the new country name 	
Exceptions	<ul style="list-style-type: none"> If the entered data does not match the required format 	
Flow of Events	User Steps	System Steps
	<p>1- <i>The admin will click on edit</i></p> <p>3- <i>The admin will enter the new state name</i></p>	<p>2- <i>The system will display a pop up form for editing with the current data</i></p> <p>4- <i>The System will return the page with the new state name and display a success message</i></p>

Table 21: Edit State

Use Case Id	SRES-US-22	
Use Case Name	Delete State	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> A success message will be displayed with state is deleted 	
Exceptions	<ul style="list-style-type: none"> None 	
Flow of Events	User Steps	System Steps
	<p>1- The admin will select the state to delete</p>	<p>2- The system will return the page without the deleted state with success message</p>

Table 22: Delete State

Use Case Id	SRES-US-23	
Use Case Name	Show States	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • All States will be displayed 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1- The admin will click on show states	2- The system will display all states

Table 23: Show States

Use Case Id	SRES-US-24	
Use Case Name	Add City	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed 	
Exceptions	<ul style="list-style-type: none"> • If admin entered data already exists • If the entered data does not match the required format • If admin didn't select which state he will add a new city to. 	
Flow of Events	User Steps	System Steps
	1- The admin will fill required fields	2- The system will receive the city name and save it 3- The system will display success message

Table 24: Add City

Use Case Id	SRES-US-25					
Use Case Name	Edit City					
Actor	Admin					
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 					
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed and the new country name 					
Exceptions	<ul style="list-style-type: none"> • If the entered data does not match the required format 					
Flow of Events	<table border="1"> <thead> <tr> <th>User Steps</th> <th>System Steps</th> </tr> </thead> <tbody> <tr> <td> 1- The admin will click on edit 3- The admin will enter the new city name </td> <td> 2- The system will display a pop up form for editing with the current data 4- The System will return the page with the new city name and display a success message </td> </tr> </tbody> </table>	User Steps	System Steps	1- The admin will click on edit 3- The admin will enter the new city name	2- The system will display a pop up form for editing with the current data 4- The System will return the page with the new city name and display a success message	
User Steps	System Steps					
1- The admin will click on edit 3- The admin will enter the new city name	2- The system will display a pop up form for editing with the current data 4- The System will return the page with the new city name and display a success message					

Table 25: Edit City

Use Case Id	SRES-US-26					
Use Case Name	Delete city					
Actor	Admin					
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 					
Postconditions	<ul style="list-style-type: none"> • A success message will be displayed with city is deleted 					
Exceptions	<ul style="list-style-type: none"> • None 					
Flow of Events	<table border="1"> <thead> <tr> <th>User Steps</th> <th>System Steps</th> </tr> </thead> <tbody> <tr> <td> 1- The admin will select the city to delete </td> <td> 2- The system will return the page without the deleted city with success message </td> </tr> </tbody> </table>	User Steps	System Steps	1- The admin will select the city to delete	2- The system will return the page without the deleted city with success message	
User Steps	System Steps					
1- The admin will select the city to delete	2- The system will return the page without the deleted city with success message					

Table 26: Delete city

Use Case Id	SRES-US-27	
Use Case Name	Show cities	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • All cities will be displayed 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1- The admin will click on show cities	2- The system will display all cities

Table 27: Show cities

Use Case Id	SRES-US-28	
Use Case Name	Add Region	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed 	
Exceptions	<ul style="list-style-type: none"> • If admin entered data already exists • If the entered data does not match the required format • If admin didn't select which city he will add a new region to. 	
Flow of Events	User Steps	System Steps
	1- The admin will fill required fields	2- The system will receive the region name and save it 3- The system will display success message

Table 28: Add Region

Use Case Id	SRES-US-29	
Use Case Name	Edit region	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed and the new region name 	
Exceptions	<ul style="list-style-type: none"> • If the entered data does not match the required format 	
Flow of Events	User Steps	System Steps
	1- The admin will click on edit 3- The admin will enter the new region name	2- The system will display a pop up form for editing with the current data 4- The System will return the page with the new region name and display a success message

Table 29: Edit region

Use Case Id	SRES-US-30	
Use Case Name	Delete Region	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • A success message will be displayed with region is deleted 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1- The admin will select the region to delete	2- The system will return the page without the deleted region with success message

Table 30: Delete Region

Use Case Id	SRES-US-31	
Use Case Name	Show Regions	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> All regions will be displayed 	
Exceptions	<ul style="list-style-type: none"> None 	
Flow of Events	User Steps	System Steps
	1- The admin will click on show regions	2- The system will display all regions

Table 31: Show Regions

Use Case Id	SRES-US-32	
Use Case Name	Add Street	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> Success message will be displayed A new location will be added 	
Exceptions	<ul style="list-style-type: none"> If admin entered data already exists If the entered data does not match the required format If admin didn't select which region he will add a new street to. 	
Flow of Events	User Steps	System Steps
	1- The admin will fill required fields	2- The system will receive the street name and save it 3- The system will display success message

Table 32: Add Street

Use Case Id	SRES-US-33	
Use Case Name	Edit street	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed and the new street name 	
Exceptions	<ul style="list-style-type: none"> • If the entered data does not match the required format 	
Flow of Events	User Steps	System Steps
	1- The admin will click on edit 3- The admin will enter the new street name	2- The system will display a pop up form for editing with the current data 4- The System will return the page with the new street name and display a success message

Table 33: Edit street

Use Case Id	SRES-US-34	
Use Case Name	Delete Street	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • A success message will be displayed with street is deleted 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1- The admin will select the street to delete	2- The system will return the page without the deleted street with success message

Table 34: Delete Street

Use Case Id	SRES-US-35	
Use Case Name	Show Streets	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • All streets will be displayed 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	3- The admin will click on show streets	4- The system will display all streets

Table 35: Show Streets

Use Case Id	SRES-US-36	
Use Case Name	Add Main Type	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed 	
Exceptions	<ul style="list-style-type: none"> • If admin entered data already exists • If the entered data does not match the required format 	
Flow of Events	User Steps	System Steps
	1- The admin will fill required fields	2- The system will receive the main type name and save it 3- The system will display success message

Table 36: Add Main Type

Use Case Id	SRES-US-37	
Use Case Name	Edit Main Type	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed and the new Main Type name 	
Exceptions	<ul style="list-style-type: none"> • If the entered data does not match the required format 	
Flow of Events	User Steps 1- The admin will click on edit 3- The admin will enter the new main type name	System Steps 2- The system will display a pop up form for editing with the current data 4- The System will return the page with the new main type name and display a success message

Table 37: Edit Main Type

Use Case Id	SRES-US-38	
Use Case Name	Delete Main Type	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • A success message will be displayed with main type is deleted 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps 1- The admin will select the main type to delete	System Steps 2- The system will return the page without the deleted main type with success message

Table 38: Delete Main Type

Use Case Id	SRES-US-39	
Use Case Name	Show Main Types	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • All main types will be displayed 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1- The admin will click on show main types	2- The system will display all main types

Table 39: Show Main Types

Use Case Id	SRES-US-40	
Use Case Name	Add Sub Type	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed 	
Exceptions	<ul style="list-style-type: none"> • If admin entered data already exists • If the entered data does not match the required format • If admin didn't select which main type he will add a new sub type to. 	
Flow of Events	User Steps	System Steps
	1- The admin will fill required fields	2- The system will receive the main type name and save it 3- The system will display success message

Table 40: Add Sub Type

Use Case Id	SRES-US-41	
Use Case Name	Edit Sub Type	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed and the new Sub Type name 	
Exceptions	<ul style="list-style-type: none"> • If the entered data does not match the required format 	
Flow of Events	User Steps 1- The admin will click on edit 3- The admin will enter the new sub type name	System Steps 2- The system will display a pop up form for editing with the current data 4- The System will return the page with the new sub type name and display a success message

Table 41: Edit Sub Type

Use Case Id	SRES-US-42	
Use Case Name	Delete Sub Type	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • A success message will be displayed with sub type is deleted 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps 1- The admin will select the sub type to delete	System Steps 2- The system will return the page without the deleted sub type with success message

Table 42: Delete Sub Type

Use Case Id	SRES-US-43	
Use Case Name	Show Sub Types	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • All sub types will be displayed 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1- The admin will click on show sub types	2- The system will display all sub types

Table 43: Show Sub Types

Use Case Id	SRES-US-44	
Use Case Name	Add Details Name	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed 	
Exceptions	<ul style="list-style-type: none"> • If admin entered data already exists • If the entered data does not match the required format • If admin didn't select which sub type he will add a new detail name to. 	
Flow of Events	User Steps	System Steps
	1- The admin will fill required fields	2- The system will receive the detail name and save it 3- The system will display success message

Table 44: Add Details Name

Use Case Id	SRES-US-45	
Use Case Name	Edit Detail Name	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed and the new detail name 	
Exceptions	<ul style="list-style-type: none"> • If the entered data does not match the required format 	
Flow of Events	User Steps 1- The admin will click on edit 3- The admin will enter the new detail name	System Steps 2- The system will display a pop up form for editing with the current data 4- The System will return the page with the new detail name and display a success message

Table 45: Edit Detail Name

Use Case Id	SRES-US-46	
Use Case Name	Delete Detail Name	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • A success message will be displayed with detail name is deleted 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps 1- The admin will select the detail name to delete	System Steps 2- The system will return the page without the deleted detail name with success message

Table 46: Delete Detail Name

Use Case Id	SRES-US-47	
Use Case Name	Show Detail Names	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • All detail names will be displayed 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1- The admin will click on show detail names	2- The system will display all detail names

Table 47: Show Detail Names

Use Case Id	SRES-US-48	
Use Case Name	Add Details Value	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed 	
Exceptions	<ul style="list-style-type: none"> • If admin entered data already exists • If the entered data does not match the required format • If admin didn't select which detail name he will add a new detail value to. 	
Flow of Events	User Steps	System Steps
	1- The admin will fill required fields	2- The system will receive the detail value and save it 3- The system will display success message

Table 48: Add Details Value

Use Case Id	SRES-US-49	
Use Case Name	Edit Detail Value	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed and the new detail value 	
Exceptions	<ul style="list-style-type: none"> • If the entered data does not match the required format 	
Flow of Events	User Steps	System Steps
	1- The admin will click on edit 3- The admin will enter the new detail value	2- The system will display a pop up form for editing with the current data 4- The System will return the page with the new detail value and display a success message

Table 49: Edit Detail Value

Use Case Id	SRES-US-50	
Use Case Name	Delete Detail Value	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • A success message will be displayed with detail value is deleted 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1- The admin will select the detail value to delete	2- The system will return the page without the deleted detail value with success message

Table 50: Delete Detail Value

Use Case Id	SRES-US-51					
Use Case Name	Show Detail Values					
Actor	Admin					
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 					
Postconditions	<ul style="list-style-type: none"> • All detail values will be displayed 					
Exceptions	<ul style="list-style-type: none"> • None 					
Flow of Events	<table border="1"> <thead> <tr> <th>User Steps</th> <th>System Steps</th> </tr> </thead> <tbody> <tr> <td>1- The admin will click on show detail values</td> <td>2- The system will display all detail values</td> </tr> </tbody> </table>	User Steps	System Steps	1- The admin will click on show detail values	2- The system will display all detail values	
User Steps	System Steps					
1- The admin will click on show detail values	2- The system will display all detail values					

Table 51: Show Detail Values

Use Case Id	SRES-US-52					
Use Case Name	Add Operation					
Actor	Admin					
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 					
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed 					
Exceptions	<ul style="list-style-type: none"> • If admin entered data already exists • If the entered data does not match the required format 					
Flow of Events	<table border="1"> <thead> <tr> <th>User Steps</th> <th>System Steps</th> </tr> </thead> <tbody> <tr> <td>1- The admin will fill required fields</td> <td> <ul style="list-style-type: none"> 2- The system will receive the operation name and save it 3- The system will display success message </td> </tr> </tbody> </table>	User Steps	System Steps	1- The admin will fill required fields	<ul style="list-style-type: none"> 2- The system will receive the operation name and save it 3- The system will display success message 	
User Steps	System Steps					
1- The admin will fill required fields	<ul style="list-style-type: none"> 2- The system will receive the operation name and save it 3- The system will display success message 					

Table 52: Add Operation

Use Case Id	SRES-US-53	
Use Case Name	Edit Operation	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed and the operation name is updated 	
Exceptions	<ul style="list-style-type: none"> • If the entered data does not match the required format 	
Flow of Events	User Steps	System Steps
	1- The admin will click on edit 3- The admin will enter the new operation name	2- The system will display a pop up form for editing with the current data 4- The System will return the page with the new operation and display a success message

Table 53: Edit Operation

Use Case Id	SRES-US-54	
Use Case Name	Delete Operation	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • A success message will be displayed with operation is deleted 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1- The admin will select the operation to delete	2- The system will return the page without the deleted operation with success message

Table 54: Edit Operation

Use Case Id	SRES-US-55	
Use Case Name	Show Operations	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • All operations will be displayed 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1- The admin will click on show operations	2- The system will display all operation

Table 55: Show Operations

Use Case Id	SRES-US-56	
Use Case Name	Add Operation Type	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed 	
Exceptions	<ul style="list-style-type: none"> • If admin entered data already exists • If the entered data does not match the required format 	
Flow of Events	User Steps	System Steps
	4- The admin will fill required fields	5- The system will receive the operation type and save it 6- The system will display success message

Table 56: Add Operation Type

Use Case Id	SRES-US-57					
Use Case Name	Edit Operation Type					
Actor	Admin					
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 					
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed and the new operation type 					
Exceptions	<ul style="list-style-type: none"> • If the entered data does not match the required format 					
Flow of Events	<table border="1"> <thead> <tr> <th>User Steps</th> <th>System Steps</th> </tr> </thead> <tbody> <tr> <td> 3- The admin will click on edit 3- The admin will enter the new operation type </td> <td> 4- The system will display a pop up form for editing with the current data 4- The System will return the page with the new operation type and display a success message </td> </tr> </tbody> </table>	User Steps	System Steps	3- The admin will click on edit 3- The admin will enter the new operation type	4- The system will display a pop up form for editing with the current data 4- The System will return the page with the new operation type and display a success message	
User Steps	System Steps					
3- The admin will click on edit 3- The admin will enter the new operation type	4- The system will display a pop up form for editing with the current data 4- The System will return the page with the new operation type and display a success message					

Table 57: Edit Operation Type

Use Case Id	SRES-US-58					
Use Case Name	Delete Operation Type					
Actor	Admin					
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 					
Postconditions	<ul style="list-style-type: none"> • A success message will be displayed with operation type is deleted 					
Exceptions	<ul style="list-style-type: none"> • None 					
Flow of Events	<table border="1"> <thead> <tr> <th>User Steps</th> <th>System Steps</th> </tr> </thead> <tbody> <tr> <td> 3- The admin will select the operation type to delete </td> <td> 4- The system will return the page without the deleted operation type with success message </td> </tr> </tbody> </table>	User Steps	System Steps	3- The admin will select the operation type to delete	4- The system will return the page without the deleted operation type with success message	
User Steps	System Steps					
3- The admin will select the operation type to delete	4- The system will return the page without the deleted operation type with success message					

Table 58: Delete Operation Type

Use Case Id	SRES-US-59					
Use Case Name	Show Operation Types					
Actor	Admin					
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 					
Postconditions	<ul style="list-style-type: none"> • All operation types will be displayed 					
Exceptions	<ul style="list-style-type: none"> • None 					
Flow of Events	<table border="1"> <thead> <tr> <th>User Steps</th> <th>System Steps</th> </tr> </thead> <tbody> <tr> <td>3- The admin will click on show operation types</td> <td>4- The system will display all operation types</td> </tr> </tbody> </table>	User Steps	System Steps	3- The admin will click on show operation types	4- The system will display all operation types	
User Steps	System Steps					
3- The admin will click on show operation types	4- The system will display all operation types					

Table 59: Show Operation Types

Use Case Id	SRES-US-60					
Use Case Name	Add Operation Type Name					
Actor	Admin					
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 					
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed 					
Exceptions	<ul style="list-style-type: none"> • If admin entered data already exists • If the entered data does not match the required format • If admin didn't select which operation type he will add a new operation type name to. 					
Flow of Events	<table border="1"> <thead> <tr> <th>User Steps</th> <th>System Steps</th> </tr> </thead> <tbody> <tr> <td>1- The admin will fill required fields</td> <td> <ul style="list-style-type: none"> 2- The system will receive the operation type name and save it 3- The system will display success message </td> </tr> </tbody> </table>	User Steps	System Steps	1- The admin will fill required fields	<ul style="list-style-type: none"> 2- The system will receive the operation type name and save it 3- The system will display success message 	
User Steps	System Steps					
1- The admin will fill required fields	<ul style="list-style-type: none"> 2- The system will receive the operation type name and save it 3- The system will display success message 					

Table 60: Add Operation Type Name

Use Case Id	SRES-US-61	
Use Case Name	Edit Operation Type Name	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed and the new operation type name 	
Exceptions	<ul style="list-style-type: none"> • If the entered data does not match the required format 	
Flow of Events	User Steps	System Steps
	1- The admin will click on edit 3- The admin will enter the new operation type name	2- The system will display a pop up form for editing with the current data 4- The System will return the page with the new operation type name and display a success message

Table 61: Edit Operation Type Name

Use Case Id	SRES-US-62	
Use Case Name	Delete Operation Type Name	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • A success message will be displayed with operation type name is deleted 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1- The admin will select the operation type name to delete	2- The system will return the page without the deleted operation type name with success message

Table 62: Delete Operation Type Name

Use Case Id	SRES-US-63					
Use Case Name	Show Operation Type Names					
Actor	Admin					
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 					
Postconditions	<ul style="list-style-type: none"> • All operation type names will be displayed 					
Exceptions	<ul style="list-style-type: none"> • None 					
Flow of Events	<table border="1"> <thead> <tr> <th>User Steps</th> <th>System Steps</th> </tr> </thead> <tbody> <tr> <td>1- The admin will click on show operation type names</td> <td>2- The system will display all operation type names</td> </tr> </tbody> </table>	User Steps	System Steps	1- The admin will click on show operation type names	2- The system will display all operation type names	
User Steps	System Steps					
1- The admin will click on show operation type names	2- The system will display all operation type names					

Table 63: Show Operation Type Names

Use Case Id	SRES-US-64					
Use Case Name	Add Operation Type Value					
Actor	Admin					
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 					
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed 					
Exceptions	<ul style="list-style-type: none"> • If admin entered data already exists • If the entered data does not match the required format • If admin didn't select which operation type name he will add a new operation type value to. 					
Flow of Events	<table border="1"> <thead> <tr> <th>User Steps</th> <th>System Steps</th> </tr> </thead> <tbody> <tr> <td>1- The admin will fill required fields</td> <td> <ul style="list-style-type: none"> 2- The system will receive the operation type value and save it 3- The system will display success message </td> </tr> </tbody> </table>	User Steps	System Steps	1- The admin will fill required fields	<ul style="list-style-type: none"> 2- The system will receive the operation type value and save it 3- The system will display success message 	
User Steps	System Steps					
1- The admin will fill required fields	<ul style="list-style-type: none"> 2- The system will receive the operation type value and save it 3- The system will display success message 					

Table 64: Add Operation Type Value

Use Case Id	SRES-US-65	
Use Case Name	Edit Operation Type Value	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed and the new operation type value 	
Exceptions	<ul style="list-style-type: none"> • If the entered data does not match the required format 	
Flow of Events	User Steps	System Steps
	1- The admin will click on edit 3- The admin will enter the new operation type value	2- The system will display a pop up form for editing with the current data 4- The System will return the page with the new operation type value and display a success message

Table 65: Edit Operation Type Value

Use Case Id	SRES-US-66	
Use Case Name	Delete Operation Type Value	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • A success message will be displayed with operation type value is deleted 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1- The admin will select the operation type value to delete	2- The system will return the page without the deleted operation type value with success message

Table 66: Delete Operation Type Value

Use Case Id	SRES-US-67	
Use Case Name	Show Operation Type Values	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • All operation type values will be displayed 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1- The admin will click on show operation type values	2- The system will display all operation type values

Table 67: Show Operation Type Values

Use Case Id	SRES-US-68	
Use Case Name	Add Item	
Actor	Admin	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed 	
Exceptions	<ul style="list-style-type: none"> • If the entered data does not match the required format 	
Flow of Events	User Steps	System Steps
	1- The admin will add the owner of the item 2- The admin will add the location of the item 3- The admin will fill the required fields in the steps of adding item	4- The system will receive the information and save it 5- The system will display success message

Table 68: Add Item

Use Case Id	SRES-US-69	
Use Case Name	Add Item	
Actor	Owner	
Preconditions	<ul style="list-style-type: none"> • The admin must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed 	
Exceptions	<ul style="list-style-type: none"> • If the entered data does not match the required format 	
Flow of Events	User Steps	System Steps
	1-The owner will add the location of the item 2-The owner will fill the required fields in the steps of adding item	3-The system will receive the information and save it 4-The system will display success message

Table 69: Add Item

Use Case Id	SRES-US-70	
Use Case Name	Edit Item(Detail)	
Actor	Admin, owner	
Preconditions	<ul style="list-style-type: none"> • The actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed and the item is updated 	
Exceptions	<ul style="list-style-type: none"> • If the entered data does not match the required format 	
Flow of Events	User Steps	System Steps
	1- The actor will click on edit 3- The actor will enter the new details	2- The system will display a pop up form for editing with the current data 4- The System will return the page with the new details and display a success message

Table 70: Edit Item(Detail)

Use Case Id	SRES-US-71	
Use Case Name	Edit Item(Location)	
Actor	Admin, owner	
Preconditions	<ul style="list-style-type: none"> • The actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed and the location is updated 	
Exceptions	<ul style="list-style-type: none"> • none 	
Flow of Events	User Steps	System Steps
	1- The actor will click on edit 2- The actor will enter the new location	3- The System will return the page with the new location and display a success message

Table 71: Edit Item(Location)

Use Case Id	SRES-US-72	
Use Case Name	Show Items	
Actor	Owner	
Preconditions	<ul style="list-style-type: none"> • The actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • All items will be displayed 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1- The actor will click on show items	2- The system will display all items

Table 72: Show Items

Use Case Id	SRES-US-73	
Use Case Name	Reserve	
Actor	Customer	
Preconditions	<ul style="list-style-type: none"> • The (admin, owner) must be logged in 	
Post-conditions	<ul style="list-style-type: none"> • Success message will be displayed 	
Exceptions	None	
Flow of Events	User Steps	System Steps
	1- The Actor must choose an item then choose date, press reserve	2- The system will receive new reservation and save it 3- The system will display success message

Table 73: Reserve

Use Case Id	SRES-US-74	
Use Case Name	Delete Reserve	
Actor	Customer	
Preconditions	<ul style="list-style-type: none"> • The (admin, owner) must be logged in 	
Postconditions	<ul style="list-style-type: none"> • A success message will be displayed with reservation is deleted 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1- The actor will select the reservation to delete	2- The system will return the page without the deleted reservation with success message

Table 74: Delete Reserve

Use Case Id	SRES-US-75	
Use Case Name	Show Reservation	
Actor	Customer	
Preconditions	<ul style="list-style-type: none"> The actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> The reservation will appear. 	
Exceptions	<ul style="list-style-type: none"> None 	
Flow of Events	User Steps	System Steps
	1-The actor will click on show Reservation	2-The system will display the reservation

Table 75: Show Reservation

Use Case Id	SRES-US-76	
Use Case Name	Search by state	
Actor	Customer	
Preconditions	<ul style="list-style-type: none"> The customer must be logged in 	
Post-conditions	<ul style="list-style-type: none"> All Item in this places will appear 	
Exceptions	<ul style="list-style-type: none"> If the entered data does not match the required format 	
Flow of Events	User Steps	System Steps
	1-The customer will write the state and press search	2-The system will display all items in state.

Table 76: Search by state

Use Case Id	SRES-US-77	
Use Case Name	View item profile	
Actor	Customer	
Preconditions	<ul style="list-style-type: none"> The customer must be logged in 	
Postconditions	<ul style="list-style-type: none"> The reservation will appear. 	
Exceptions	<ul style="list-style-type: none"> None 	
Flow of Events	User Steps	System Steps
	1-The customer will click on show Reservation	2-The system will display the reservation

Table 77: View item profile

Use Case Id	SRES-US-78	
Use Case Name	View the reservation of the owner item	
Actor	Owner	
Preconditions	<ul style="list-style-type: none"> The owner must be logged in 	
Postconditions	<ul style="list-style-type: none"> The reservation will appear. 	
Exceptions	<ul style="list-style-type: none"> None 	
Flow of Events	User Steps	System Steps
	1-The owner will click on show Reservation	2-The system will display the reservation

Table 78: View the reservation of the owner item

Use Case Id	SRES-US-79	
Use Case Name	Post a post	
Actor	Customer ,Owner	
Preconditions	<ul style="list-style-type: none"> • The Actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • His profile will have a new post 	
Exceptions	None	
Flow of Events	User Steps	System Steps
	1-The Actor will click write a post	2-The system will display the post in his profile

Table 79: Post a post

Use Case Id	SRES-US-80	
Use Case Name	Delete post	
Actor	<ul style="list-style-type: none"> • Customer , Owner 	
Preconditions	<ul style="list-style-type: none"> • The actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Actor profile will not have the deleted post 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1-The actor will click trash icon in post place to delete	2-The system will return the page without the deleted post value

Table 80: Delete post

Use Case Id	SRES-US-81	
Use Case Name	Edit post	
Actor	<ul style="list-style-type: none"> • Customer , Owner 	
Preconditions	<ul style="list-style-type: none"> • The actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed and the new post 	
Exceptions		
Flow of Events	User Steps	System Steps
	1-The actor will click on edit 2-The actor will enter the new post	3-The system will display a pop up form for editing with the current data 4- The System will return the page with the new post and display a success message

Table 81: Post a post

Use Case Id	SRES-US-82	
Use Case Name	Write a post in item	
Actor	Owner	
Preconditions	<ul style="list-style-type: none"> • The Actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • His item profile will have a new post 	
Exceptions		
Flow of Events	User Steps	System Steps
	1-The Actor will click write a post	2-The system will display the post in his item profile

Table 82: Write a post in item

Use Case Id	SRES-US-83	
Use Case Name	Write a comment	
Actor	Owner, customer	
Preconditions	<ul style="list-style-type: none"> The Actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> The post will have a new comment 	
Exceptions		
Flow of Events	User Steps	System Steps
	1-The Actor will click comment	2-The system will display the comment after the post

Table 83: Write a comment

Use Case Id	SRES-US-84	
Use Case Name	Delete Comment	
Actor	<ul style="list-style-type: none"> Customer , Owner 	
Preconditions	<ul style="list-style-type: none"> The actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> Actor profile will not have the deleted comment 	
Exceptions	<ul style="list-style-type: none"> None 	
Flow of Events	User Steps	System Steps
	1-The actor will select click trash in the comment place	2-The system will return the page without the deleted Comment

Table 84: Delete Comment

Use Case Id	SRES-US-85	
Use Case Name	Edit comment	
Actor	<ul style="list-style-type: none"> • Customer , Owner 	
Preconditions	<ul style="list-style-type: none"> • The actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed and the new comment 	
Exceptions		
Flow of Events	User Steps	System Steps
	1-The actor will click on edit 2-The actor will enter the new comment	3-The system will display a pop up form for editing with the current data 4- The System will return the page with the new comment and display a success message

Table 85: Edit comment

Use Case Id	SRES-US-86	
Use Case Name	Edit profile (personal data)	
Actor	<ul style="list-style-type: none"> • Customer , Owner 	
Preconditions	<ul style="list-style-type: none"> • The actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed and the new edit 	
Exceptions		
Flow of Events	User Steps	System Steps
	1-The actor will click on edit 2-The actor will enter the new data	3-The system will display a pop up form for editing with the current data 4- The System will return the page with the new data and display a success message

Table 86: Edit profile (personal data)

Use Case Id	SRES-US-87	
Use Case Name	Edit profile (Cover photo)	
Actor	<ul style="list-style-type: none"> • Customer , Owner 	
Preconditions	<ul style="list-style-type: none"> • The actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • The System will redirect same page with the new edit 	
Exceptions		
Flow of Events	User Steps	System Steps
	1-The actor will click on edit 2-The actor will choose new photo	3 - The System will return the page with the new photo and display a success message

Table 87: Edit profile (Cover photo)

Use Case Id	SRES-US-88	
Use Case Name	Edit profile (profile photo)	
Actor	<ul style="list-style-type: none"> • Customer , Owner 	
Preconditions	<ul style="list-style-type: none"> • The actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed and the new edit 	
Exceptions	<ul style="list-style-type: none"> • none 	
Flow of Events	User Steps	System Steps
	1-The actor will click on edit 2-The actor will choose new photo	3 - The System will return the page with the new photo and display a success message

Table 88: Edit profile (profile photo)

Use Case Id	SRES-US-89	
Use Case Name	Reply a comment	
Actor	<ul style="list-style-type: none"> • Owner, customer 	
Preconditions	<ul style="list-style-type: none"> • The Actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • The comment will have a reply 	
Exceptions		
Flow of Events	User Steps	System Steps
	1-The Actor will click reply comment	2-The system will display the reply comment after the comment

Table 89: Reply a comment

Use Case Id	SRES-US-90	
Use Case Name	Delete reply	
Actor	<ul style="list-style-type: none"> • Customer , Owner 	
Preconditions	<ul style="list-style-type: none"> • The actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Actor profile will not have the deleted reply of comment 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1-The actor will select click trash in the reply of comment place	2-The system will return the page without the deleted reply

Table 90: Delete reply

Use Case Id	SRES-US-91					
Use Case Name	Edit reply					
Actor	<ul style="list-style-type: none"> • Customer , Owner 					
Preconditions	<ul style="list-style-type: none"> • The actor must be logged in 					
Postconditions	<ul style="list-style-type: none"> • Success message will be displayed and the new reply 					
Exceptions						
Flow of Events	<table border="1"> <thead> <tr> <th>User Steps</th> <th>System Steps</th> </tr> </thead> <tbody> <tr> <td> 1-The actor will click on edit 2-The actor will enter the new reply </td> <td> 3-The system will display a pop up form for editing with the current data 4- The System will return the page with the new reply and display a success message </td> </tr> </tbody> </table>	User Steps	System Steps	1-The actor will click on edit 2-The actor will enter the new reply	3-The system will display a pop up form for editing with the current data 4- The System will return the page with the new reply and display a success message	
User Steps	System Steps					
1-The actor will click on edit 2-The actor will enter the new reply	3-The system will display a pop up form for editing with the current data 4- The System will return the page with the new reply and display a success message					

Table 91: Edit reply

Use Case Id	SRES-US-92					
Use Case Name	Follow Item					
Actor	<ul style="list-style-type: none"> • Customer 					
Preconditions	<ul style="list-style-type: none"> • The actor must be logged in 					
Postconditions	<ul style="list-style-type: none"> • The actor will see any updates to this item on the actor home 					
Exceptions						
Flow of Events	<table border="1"> <thead> <tr> <th>User Steps</th> <th>System Steps</th> </tr> </thead> <tbody> <tr> <td> 1-The actor will click on this item 2-The actor will click follow </td> <td> 3- The System will show the posts in this item to home page of the actor </td> </tr> </tbody> </table>	User Steps	System Steps	1-The actor will click on this item 2-The actor will click follow	3- The System will show the posts in this item to home page of the actor	
User Steps	System Steps					
1-The actor will click on this item 2-The actor will click follow	3- The System will show the posts in this item to home page of the actor					

Table 92: Follow Item

Use Case Id	SRES-US-93	
Use Case Name	Unfollow Item	
Actor	<ul style="list-style-type: none"> • Customer 	
Preconditions	<ul style="list-style-type: none"> • The actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • The actor will not see any updates to this item on the actor home 	
Exceptions		
Flow of Events	User Steps	System Steps
	1-The actor will click on this item 2-The actor will click unfollow	3- The System will erase the posts in this item to home page of the actor

Table 93: Unfollow Item

Use Case Id	SRES-US-94	
Use Case Name	View reservation of Item	
Actor	<ul style="list-style-type: none"> • Owner 	
Preconditions	<ul style="list-style-type: none"> • The actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • All the reservation of the items the owner has will be appear 	
Exceptions	<ul style="list-style-type: none"> • none 	
Flow of Events	User Steps	System Steps
	1-The actor will click on show reservation	2- The System will show all reservation of owner's items

Table 94: View reservation of Item

Use Case Id	SRES-US-95	
Use Case Name	Check Balance	
Actor	Admin ,Owner	
Preconditions	<ul style="list-style-type: none"> • The Actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Balance of owner will appear 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1-The Actor will click Check balance	2-The system will display the balance of owner

Table 95: *Check Balance*

Use Case Id	SRES-US-96	
Use Case Name	Follow user	
Actor	<ul style="list-style-type: none"> • Customer 	
Preconditions	<ul style="list-style-type: none"> • The actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Be able to see the posts of this user 	
Exceptions		
Flow of Events	User Steps	System Steps
	1-The actor will click on this item actor name that write the post 2-The actor will click follow	3- The System will show the posts in this actor page

Table 96: *Unfollow user*

Use Case Id	SRES-US-97	
Use Case Name	Unfollow user	
Actor	<ul style="list-style-type: none"> • Customer 	
Preconditions	<ul style="list-style-type: none"> • The actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Will not be able to see this user posts 	
Exceptions		
Flow of Events	User Steps	System Steps
	1-The actor will click on this items actor name 2-The actor will click unfollow	3- The System will erase the posts of the user from the actor homepage

Table 97: Unfollow user

Use Case Id	SRES-US-98	
Use Case Name	View Item Profile	
Actor	<ul style="list-style-type: none"> • Customer 	
Preconditions	<ul style="list-style-type: none"> • The actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • know everything about this item 	
Exceptions		
Flow of Events	User Steps	System Steps
	1-The actor will click on this item	2- The System will get all data of this item

Table 98: View Item Profile

Use Case Id	SRES-US-99	
Use Case Name	Review Item	
Actor	<ul style="list-style-type: none"> • Customer 	
Preconditions	<ul style="list-style-type: none"> • The actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Change the rate of the item 	
Exceptions		
Flow of Events	User Steps	System Steps
	1-The actor will click on this item review 2-The actor will choose number of stars	3- The system will change number of stars appear to users

Table 99: Review Item

Use Case Id	SRES-US-100	
Use Case Name	Edit Review	
Actor	Customer	
Preconditions	<ul style="list-style-type: none"> • The (admin, owner) must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Number of stars have been edited 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1-The actor will select number of stars that want to change	2-The system will return the page with the edited number of stars .

Table 100: Edit Review

Use Case Id	SRES-US-101	
Use Case Name	Be Owner	
Actor	Customer	
Preconditions	<ul style="list-style-type: none"> The Customer must be logged in 	
Postconditions	<ul style="list-style-type: none"> Add item form appear 	
Exceptions	<ul style="list-style-type: none"> None 	
Flow of Events	User Steps	System Steps
	1-The actor will continue as owner	2-The system will redirect same page with popup to complete his data.

Table 101: Be Owner

Use Case Id	SRES-US-102	
Use Case Name	pay	
Actor	Customer	
Preconditions	<ul style="list-style-type: none"> The Customer must be logged in 	
Postconditions	<ul style="list-style-type: none"> PayPal form appear 	
Exceptions	<ul style="list-style-type: none"> None 	
Flow of Events	User Steps	System Steps
	1-The actor will choose the item and number of days 2-The actor will click pay	3-The system calculates the total price and paypal page will be viewed

Table 102: pay

Use Case Id	SRES-US-103	
Use Case Name	View his reservations	
Actor	Customer	
Preconditions	<ul style="list-style-type: none"> • The Customer must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Know number of reservation he had 	
Exceptions	<ul style="list-style-type: none"> • None 	
Flow of Events	User Steps	System Steps
	1-The Customer will click my reservations from slide bar	2-the system will get the data and return it in show reservations

Table 103: View his reservations

Use Case Id	SRES-US-104	
Use Case Name	Delete(cover, profile)	
Actor	<ul style="list-style-type: none"> • Customer , Owner 	
Preconditions	<ul style="list-style-type: none"> • The actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • The actor profile will appear with the default photo in cover photo or profile 	
Exceptions	<ul style="list-style-type: none"> • none 	
Flow of Events	User Steps	System Steps
	1-The actor will click on delete	2-The System will redirect same page without picture.

Table 104: Delete(cover, profile)

Use Case Id	SRES-US-105	
Use Case Name	Show Item	
Actor	<ul style="list-style-type: none"> • Owner 	
Preconditions	<ul style="list-style-type: none"> • The actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • The actor will know all data of a specific item 	
Exceptions	<ul style="list-style-type: none"> • none 	
Flow of Events	User Steps	System Steps
	1-The actor will click on item name	2-The System will view the page of item profile

Table 105: Show Item

Use Case Id	SRES-US-106	
Use Case Name	Delete Item	
Actor	<ul style="list-style-type: none"> • Owner 	
Preconditions	<ul style="list-style-type: none"> • The actor must be logged in 	
Postconditions	<ul style="list-style-type: none"> • Actor items will decrease one 	
Exceptions	<ul style="list-style-type: none"> • none 	
Flow of Events	User Steps	System Steps
	1-The actor will click on item name 2-Then clicks delete icon	3-The System will return show item page without the deleted item

Table 106: Delete Item

3.4. Scenarios

- Add Datatype

SCENARIO ID	SRES-SC-1
SCENARIO NAME	Add Datatype
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system, then from the slide bar, admin will click “other”, click “add data type”, so the system will display to the index page of data type, the admin will enter new datatype and click “add”, the system will redirect to the same page with success message: “Data type created successfully”.</p> <p>If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.</p> <p>If the admin writes an existing datatype the system will show error message that “Datatype already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- Edit Datatype

SCENARIO ID	SRES-SC-2
SCENARIO NAME	Edit Datatype
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “other” then will click “datatype for details”, the system will display the index page of data type with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this datatype, the system will redirect the same page with the edit of the data type name.</p> <p>If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.</p> <p>If the admin writes in the field of the pop-up an existing datatype the system will show error message that “Datatype already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- **delete Datatype**

SCENARIO ID	SRES-SC-3
SCENARIO NAME	delete Datatype
ACTOR	Admin
NORMAL	The Admin will log in to the system. From the slide bar, admin will click “other” then will click “datatype for details”, so the system will display the index page of data type and show all data types, the admin has to select the data type that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted data type.

- Show Datatype

SCENARIO ID	SRES-SC-4
SCENARIO NAME	Show Datatype
ACTOR	Admin
NORMAL	The Admin will log in to the system. From the slide bar, admin will click “other” then will click “datatype for details”, so the system will display the index page of data type and show all data types.

- Add User

SCENARIO ID	SRES-SC-5
SCENARIO NAME	Add User
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system, then from the slide bar, admin will click “User”, click “add user”, so the system will display to the index page of user, the admin will follow the steps to add new user, the system will redirect to the same page with success message: “User created successfully”.</p> <p>If the admin leaves any field empty and clicks “add” the system will show error by changing the field color.</p> <p>If the admin writes an existing email the system will show error message that “user already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- **Edit User**

SCENARIO ID	SRES-SC-6
SCENARIO NAME	Edit User
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “User” then will click “Show users”, the system will display the index page of user with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this user, new email, new Phone number, the system will redirect the same page with the edit of the user.</p> <p>If the admin leaves the fields of the pop-up and click add the system will redirect the same page without change in data.</p> <p>If the admin writes in the field of the pop-up an existing user e-mail the system will show error message that “User already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- **delete User**

SCENARIO ID	SRES-SC-7
SCENARIO NAME	delete User
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “User” then will click “Show users”, so the system will display the index page of user and show all users, the admin has to select the user that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted user.</p>

- **Show User**

SCENARIO ID	SRES-SC-8
SCENARIO NAME	Show User
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “User” then will click “Show users”, so the system will display the index page of user and show all users.</p>

- Add User type

SCENARIO ID	SRES-SC-9
SCENARIO NAME	Add User type
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system, then from the slide bar, admin will click “User”, click “Add User Type”, so the system will display to the index page of user type, the admin will enter new User type and click “add”, the system will redirect to the same page with success message: “User type created successfully”.</p> <p>If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.</p> <p>If the admin writes an existing datatype the system will show error message that “User type already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- **Edit User Type**

SCENARIO ID	SRES-SC-10
SCENARIO NAME	Edit User Type
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “User” then will click “Show User Type”, the system will display the index page of user type with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this user type, the system will redirect the same page with the edit of the user type name.</p> <p>If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.</p> <p>If the admin writes in the field of the pop-up an existing user type the system will show error message that “user type already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- **Delete User Type**

SCENARIO ID	SRES-SC-11
SCENARIO NAME	Delete User Type
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “User” then will click “Show user type”, so the system will display the index page of user and show all user types, the admin has to select the user type that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted user type.</p>

- Show User Type

SCENARIO ID	SRES-SC-12
SCENARIO NAME	Show User Type
ACTOR	Admin
NORMAL	The Admin will log in to the system. From the slide bar, admin will click “User” then will click “Show user Type”, so the system will display the index page of user type and show all user type.

- Add Item

SCENARIO ID	SRES-SC-13
SCENARIO NAME	Add Item
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system, then from the slide bar, admin will click “Item”, click “Add Item”, so the system will display to the index page of add item, the admin will insert the email or any part of it of the owner, after that the admin will click “search” button, the system will redirect the emails that contain these words will be displayed in list, the admin has to select one of these emails then he must fill all the fields in the wizard form , after the admin clicks “add” a success message appears “Item created successfully”.</p> <p>If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.</p> <p>If the admin writes an existing item name the system will show error message that “item already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- **Edit Item**

SCENARIO ID	SRES-SC-14
SCENARIO NAME	Edit Item
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “Item Details” then will click “detail of items”, the system will display the index page of choosing item, the admin will choose the item the admin wants to edit then click “see item”, the system will redirect to the page of item details.</p> <p>If the admin wants to edit owner’s information, the admin must click the edit icon that is in User, the system will display the page of search to choose the owner, the admin will insert the email or any part of it of the owner, after that the admin will click “search” button, the system will redirect the emails that contain these words will be displayed in list, the admin has to select one of these emails to be the owner of the item .</p> <p>If the admin wants to edit location, the admin must click edit icon that is in Location, the system will display location form, the admin must insert all fields in the wizard form then the admin clicks “edit location”, the system will display the page of item details.</p>

- **Delete Item**

SCENARIO ID	SRES-SC-15
SCENARIO NAME	Delete Item
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “Item Details” then will click “Detail of Items”, so the admin will choose the item the admin wants to delete it, the admin must click “see item”, the system will display the index of the item detail page, the admin will click trash icon to delete this item.</p>

- Show Item

SCENARIO ID	SRES-SC-16
SCENARIO NAME	Show Item
ACTOR	Admin
NORMAL	The Admin will log in to the system. From the slide bar, admin will click “Item Details” then will click “Detail of Items”, so the admin will choose the item the admin wants to show it, the admin must click “see item”, the system will display item details that is chosen.

- Show Reservation

SCENARIO ID	SRES-SC-17
SCENARIO NAME	Show Reservation
ACTOR	Admin
NORMAL	The Admin will log in to the system. From the slide bar, admin will click “Item Details” then will click “Detail of Items”, so the admin will choose the item the admin wants to show it, the admin must click “see item”, the system will display item details that is chosen, the admin will click reservation icon, the system will display the index of detail of item .

- Add Operation

SCENARIO ID	SRES-SC-18
SCENARIO NAME	Add Operation
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system, then from the slide bar, admin will click "", click "", so the system will display to the index page of operation, the admin will enter new operation and click "add", the system will redirect to the same page with success message: "operation created successfully".</p> <p>If the admin leaves the field empty and clicks "add" the system will show error message: "Please fill out this field".</p> <p>If the admin writes an existing operation the system will show error message that "operation already exists".</p> <p>If the entered data does not match the required format, error message will occur "".</p>

- Edit Operation

SCENARIO ID	SRES-SC-19
SCENARIO NAME	Edit Operation
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click "" then will click "", the system will display the index page of operation with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this operation, the system will redirect the same page with the edit of the operation name.</p> <p>If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.</p> <p>If the admin writes in the field of the pop-up an existing operation, the system will show error message that "operation already exists".</p> <p>If the entered data does not match the required format, error message will occur "".</p>

- Delete Operation

SCENARIO ID	SRES-SC-20
SCENARIO NAME	Delete Operation
ACTOR	Admin
NORMAL	The Admin will log in to the system. From the slide bar, admin will click “” then will click “”, so the system will display the index page of operation and show all operations, the admin has to select the operation that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted operation.

- Show Operation

SCENARIO ID	SRES-SC-21
SCENARIO NAME	Show Operation
ACTOR	Admin
NORMAL	The Admin will log in to the system. From the slide bar, admin will click “” then will click “”, so the system will display the index page of Operations and show all operations.

- Add country

SCENARIO ID	SRES-SC-22
SCENARIO NAME	Add country
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system, then from the slide bar, admin will click “location”, click “country”, so the system will display to the index page of country, the admin will enter new country and click “add”, the system will redirect to the same page with success message: “country created successfully”.</p> <p>If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.</p> <p>If the admin writes an existing country the system will show error message that “country already exists”.</p>

If the entered data does not match the required format, error message will occur "".

- **Edit Country**

SCENARIO ID	SRES-SC-23
SCENARIO NAME	Edit Country
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “location” then will click “country”, the system will display the index page of country with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this country, the system will redirect the same page with the edit of the country name.</p> <p>If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.</p> <p>If the admin writes in the field of the pop-up an existing country the system will show error message that “country already exists”.</p> <p>If the entered data does not match the required format, error message will occur "".</p>

- **delete Country**

SCENARIO ID	SRES-SC-24
SCENARIO NAME	delete Country
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “Location” then will click “Country”, so the system will display the index page of country and show all countries, the admin has to select the country that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted country.</p>

- Show country

SCENARIO ID	SRES-SC-25
SCENARIO NAME	Show country
ACTOR	Admin
NORMAL	The Admin will log in to the system. From the slide bar, admin will click “location” then will click “country”, so the system will display the index page of country and show all countries.

- Add state

SCENARIO ID	SRES-SC-26
SCENARIO NAME	Add state
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system, then from the slide bar, admin will click “location”, click “state”, so the system will display to the index page of state, the admin will choose the country before he inserts the state name and click “add”, the system will redirect to the same page with success message: “state created successfully”.</p> <p>If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.</p> <p>If the admin writes an existing state the system will show error message that “state already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- **Edit state**

SCENARIO ID	SRES-SC-27
SCENARIO NAME	Edit state
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “location” then will click “state”, the system will display the index page of state with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this state, the system will redirect the same page with the edit of the state name.</p> <p>If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.</p> <p>If the admin writes in the field of the pop-up an existing state the system will show error message that “state already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- Add City

SCENARIO ID	SRES-SC-28
SCENARIO NAME	Add City
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system, then from the slide bar, admin will click “location”, click “City”, so the system will display to the index page of City, the admin will choose the country after that he will choose the state before he inserts the city name and click “add”, the system will redirect to the same page with success message: “city created successfully”.</p> <p>If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.</p> <p>If the admin writes an existing city the system will show error message that “city already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- Edit city

SCENARIO ID	SRES-SC-29
SCENARIO NAME	Edit city
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “location” then will click “city”, the system will display the index page of city with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this city, the system will redirect the same page with the edit of the city name.</p> <p>If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.</p> <p>If the admin writes in the field of the pop-up an existing city the system will show error message that “city already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- **delete city**

SCENARIO ID		SRES-SC-30
SCENARIO NAME		delete city
ACTOR		Admin
NORMAL		The Admin will log in to the system. From the slide bar, admin will click “Location” then will click “City”, so the system will display the index page of city and show all cities, the admin has to select the state that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted city.

- **Show City**

SCENARIO ID		SRES-SC-31
SCENARIO NAME		Show City
ACTOR		Admin
NORMAL		The Admin will log in to the system. From the slide bar, admin will click “location” then will click “City”, so the system will display the index page of state and show all cities.

- Show City

SCENARIO ID	SRES-SC-32
SCENARIO NAME	Add region
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system, then from the slide bar, admin will click “location”, click “region”, so the system will display to the index page of region, the admin will choose the country after that he will choose the state, choose city before he inserts the region name and click “add”, the system will redirect to the same page with success message: “region created successfully”.</p> <p>If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.</p> <p>If the admin writes an existing region the system will show error message that “region already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- Edit Region

SCENARIO ID	SRES-SC-33
SCENARIO NAME	Edit Region
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “location” then will click “region”, the system will display the index page of region with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this region, the system will redirect the same page with the edit of the region name.</p> <p>If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.</p> <p>If the admin writes in the field of the pop-up an existing region the system will show error message that “region already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- **delete Region**

SCENARIO ID		SRES-SC-34
SCENARIO NAME	delete Region	
ACTOR	Admin	
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “Location” then will click “Region”, so the system will display the index page of region and show all regions, the admin has to select the region that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted region.</p>	

- **Show Region**

SCENARIO ID		SRES-SC-35
SCENARIO NAME	Show Region	
ACTOR	Admin	
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “location” then will click “Region”, so the system will display the index page of state and show all regions.</p>	

- Add street

SCENARIO ID		SRES-SC-36
SCENARIO NAME	Actor	Add street
ACTOR		Admin
NORMAL		<p>The Admin will log in to the system, then from the slide bar, admin will click “location”, click “street”, so the system will display to the index page of street, the admin will choose the country after that he will choose the state, choose city, choose region before he inserts the street name and click “add”, the system will redirect to the same page with success message: “street created successfully”.</p> <p>If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.</p> <p>If the admin writes an existing street the system will show error message that “street already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- Edit Street

SCENARIO ID		SRES-SC-37
SCENARIO NAME	Actor	Edit Street
ACTOR		Admin
NORMAL		<p>The Admin will log in to the system. From the slide bar, admin will click “location” then will click “Street”, the system will display the index page of street with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this street, the system will redirect the same page with the edit of the street name.</p> <p>If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.</p> <p>If the admin writes in the field of the pop-up an existing street the system will show error message that “street already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- **delete street**

SCENARIO ID	SRES-SC-38
SCENARIO NAME	delete street
ACTOR	Admin
NORMAL	The Admin will log in to the system. From the slide bar, admin will click “Location” then will click “Street”, so the system will display the index page of street and show all streets, the admin has to select the street that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted street.

- Show street

SCENARIO ID	SRES-SC-39
SCENARIO NAME	Show street
ACTOR	Admin
NORMAL	The Admin will log in to the system. From the slide bar, admin will click “location” then will click “street”, so the system will display the index page of state and show all streets.

- Add Operation type name

SCENARIO ID	SRES-SC-40
SCENARIO NAME	Add Operation type name
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system, then from the slide bar, admin will click “”, click “”, so the system will display to the index page of operation type name, the admin will choose operation type before the admin inserts operation type name and click “add”, the system will redirect to the same page with success message: “operation type created successfully”.</p> <p>If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.</p> <p>If the admin writes an existing operation type the system will show error message that “operation type already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- **Edit Operation type**

SCENARIO ID	SRES-SC-41
SCENARIO NAME	Edit Operation type
ACTOR	Admin
NORMAL	The Admin will log in to the system. From the slide bar, admin will click “” then will click “”, the system will display the index page of operation type with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this operation type, the system will redirect the same page with the edit of the operation type name.

- **Delete Operation type**

SCENARIO ID	SRES-SC-42
SCENARIO NAME	Delete Operation type
ACTOR	Admin
NORMAL	The Admin will log in to the system. From the slide bar, admin will click “” then will click “”, so the system will display the index page of operation type and show all operation types, the admin has to select the operation type that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted operation type.

- **Show Operation type**

SCENARIO ID	SRES-SC-43
SCENARIO NAME	Show Operation type
ACTOR	Admin
NORMAL	The Admin will log in to the system. From the slide bar, admin will click “” then will click “”, so the system will display the index page of operation types and show all operation types.

- Add Main type

SCENARIO ID	SRES-SC-44
SCENARIO NAME	Add Main type
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system, then from the slide bar, admin will click “item details”, click “main type”, so the system will display to the index page of main type, the admin will enter new main type and click “add”, the system will redirect to the same page with success message: “main type created successfully”.</p> <p>If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.</p> <p>If the admin writes an existing main type the system will show error message that “Main type already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- Edit Main type

SCENARIO ID	SRES-SC-45
SCENARIO NAME	Edit Main type
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “main type”, the system will display the index page of main type with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this main type, the system will redirect the same page with the edit of the main type name.</p> <p>If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.</p> <p>If the admin writes in the field of the pop-up an existing main type the system will show error message that “main type already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- **delete main type**

SCENARIO ID		SRES-SC-46
SCENARIO NAME		delete main type
ACTOR		Admin
NORMAL		The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “Main type”, so the system will display the index page of main type and show all main types, the admin has to select the main type that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted main type.

- **Show main type**

SCENARIO ID		SRES-SC-47
SCENARIO NAME		Show main type
ACTOR		Admin
NORMAL		The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “Main type”, so the system will display the index page of main type and show all main types.

- Add Sub type

SCENARIO ID	SRES-SC-48
SCENARIO NAME	Add Sub type
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system, then from the slide bar, admin will click “item details”, click “sub type”, so the system will display to the index page of sub type, the admin will choose new main type before the admin inserts sub type and click “add”, the system will redirect to the same page with success message: “sub type created successfully”.</p> <p>If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.</p> <p>If the admin writes an existing sub type the system will show error message that “Sub type already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- Edit sub type

SCENARIO ID	SRES-SC-49
SCENARIO NAME	Edit sub type
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “sub type”, the system will display the index page of sub type with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this sub type, the system will redirect the same page with the edit of the sub type name.</p> <p>If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.</p> <p>If the admin writes in the field of the pop-up an existing sub type the system will show error message that “Sub type already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- **delete sub type**

SCENARIO ID	SRES-SC-50
SCENARIO NAME	delete sub type
ACTOR	Admin
NORMAL	The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “Sub type”, so the system will display the index page of sub type and show all sub types, the admin has to select the sub type that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted sub type.

- **Show sub type**

SCENARIO ID	SRES-SC-51
SCENARIO NAME	Show sub type
ACTOR	Admin
NORMAL	The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “Sub type”, so the system will display the index page of sub type and show all sub types.

- Add Subtype property

SCENARIO ID	SRES-SC-52
SCENARIO NAME	Add Subtype property
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system, then from the slide bar, admin will click “item details”, click “sub type property”, so the system will display to the index page of sub type property, the admin will choose new main type, chooses sub type before the admin inserts sub type property and click “add”, the system will redirect to the same page with success message: “sub type property created successfully”.</p> <p>If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.</p> <p>If the admin writes an existing sub type property, the system will show error message that “Sub type property already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- Edit sub type property

SCENARIO ID	SRES-SC-53
SCENARIO NAME	Edit sub type property
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “sub type property”, the system will display the index page of sub type property with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this sub type property, the system will redirect the same page with the edit of the sub type property name.</p> <p>If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.</p> <p>If the admin writes in the field of the pop-up an existing sub type property, the system will show error message that “Sub type property already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- **delete sub type property**

SCENARIO ID	SRES-SC-54
SCENARIO NAME	delete sub type property
ACTOR	Admin
NORMAL	The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “Sub type property”, so the system will display the index page of sub type and show all sub type properties, the admin has to select the sub type property that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted sub type property.

- **Show sub type property**

SCENARIO ID	SRES-SC-55
SCENARIO NAME	Show sub type property
ACTOR	Admin
NORMAL	The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “Sub type property”, so the system will display the index page of sub type property and show all sub type properties.

- Add details of property

SCENARIO ID	SRES-SC-56
SCENARIO NAME	Add details of property
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system, then from the slide bar, admin will click “item details”, click “Details of property”, so the system will display to the index page of details of property, the admin will choose new main type, chooses sub type, chooses sub type property, before the admin inserts detail of the property and click “add”, the system will redirect to the same page with success message: “Detail of property created successfully”.</p> <p>If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.</p> <p>If the admin writes an existing details of property, the system will show error message that “Details of property already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- Edit details of property

SCENARIO ID	SRES-SC-57
SCENARIO NAME	Edit details of property
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “Details of property”, the system will display the index page of details of property with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new name of this details of property, the system will redirect the same page with the edit of the details of property name.</p> <p>If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.</p> <p>If the admin writes in the field of the pop-up an existing details of property, the system will show error message that “details of property already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- Show details of property

SCENARIO ID	SRES-SC-58
SCENARIO NAME	delete details of property
ACTOR	Admin
NORMAL	The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “Details of property”, so the system will display the index page of details of property and show all details of properties, the admin has to select the details of property that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted details of property.

- Show details of property

SCENARIO ID	SRES-SC-59
SCENARIO NAME	Show details of property
ACTOR	Admin
NORMAL	The Admin will log in to the system. From the slide bar, admin will click “item details” then will click “Details of property”, so the system will display the index page of details of property and show all details of properties.

- Add details of item

SCENARIO ID	SRES-SC-60
SCENARIO NAME	Add details of item
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system, then from the slide bar, admin will click “item details”, click “Details of item”, so the system will display to the index page of details of item, the admin will choose new main type, chooses sub type, chooses sub type property, chooses detail of property before the admin inserts detail of the item and click “add”, the system will redirect to the same page with success message: “Detail of item created successfully”.</p> <p>If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.</p> <p>If the admin writes an existing details of item, the system will show error message that “Details of item already exists”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- Add Location

SCENARIO ID	SRES-SC-61
SCENARIO NAME	Add Location
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system, then from the slide bar, admin will click “Item”, click “Add Item”, so the system will display to the index page of add item, the admin has two ways to enter the location the first complete the wizard, secondly the admin has to click on google maps to search about the street and click in right marker, the system saves the longitude and latitude, successful message appears “location saved successfully”;</p>

- Add Operation type value

SCENARIO ID	SRES-SC-62
SCENARIO NAME	Add Operation type value
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system, then from the slide bar, admin will click “”, click “”, so the system will display to the index page of operation type value, the admin will choose operation type, chooses operation type name before the admin inserts operation type value and click “add”, the system will redirect to the same page with success message: “operation type value created successfully”.</p> <p>If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- **Edit Operation type value**

SCENARIO ID	SRES-SC-63
SCENARIO NAME	Edit Operation type value
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “operation” then will click “”, the system will display the index page of operation type value with the data in the table, the admin will click edit icon of the index that the admin wants to edit, the system will display pop-up that the admin will enter the new input of this operation type value, the system will redirect the same page with the edit of the operation type value.</p> <p>If the admin leaves the field of the pop-up and click add the system will redirect the same page without change in data.</p> <p>If the entered data does not match the required format, error message will occur “”.</p>

- **Delete Operation type value**

SCENARIO ID	SRES-SC-64
SCENARIO NAME	Delete Operation type value
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “” then will click “”, so the system will display the index page of operation type value and show all operation type values, the admin has to select the operation type value that the admin wants to delete then the admin will click trash icon, the system will redirect the same page but without the deleted operation type value.</p>

- **Show Operation type value**

SCENARIO ID	SRES-SC-65
SCENARIO NAME	Show Operation type value
ACTOR	Admin
NORMAL	<p>The Admin will log in to the system. From the slide bar, admin will click “” then will click “”, so the system will display the index page of operation types and show all operation type values.</p>

- Add Item

SCENARIO ID	SRES-SC-66
SCENARIO NAME	Add Item
ACTOR	Owner
NORMAL	<p>The Owner will log in to the system, then from the slide bar, owner will click “Add Item”, so the system will display to the index page of add item, then the owner must fill all the fields in the wizard form, after the admin clicks “add” a success message appears “Item created successfully”.</p> <p>If the admin leaves the field empty and clicks “add” the system will show error message: “Please fill out this field”.</p> <p>If the admin writes an existing Item name, the system will show error message that “item already exists”.</p> <p>If the entered data does not match the required format, error message will occur “Please matches the requested format”.</p>

- Add photo

SCENARIO ID	SRES-SC-67
SCENARIO NAME	Add photo
ACTOR	Owner
NORMAL	<p>The Owner will log in to the system, then from the slide bar, owner will click “Add Item”, clicks file upload, so the system will display to the index page of add item, clicks file upload, then the owner must choose the photo the owner wants to upload, after the owner clicks “add”, the gallery will have new photo.</p> <p>If the entered data does not match the required format, error message will occur “Please matches the requested format”.</p>

- Edit Item

SCENARIO ID	SRES-SC-68
SCENARIO NAME	Edit Item
ACTOR	Owner
NORMAL	<p>The owner will log in to the system. From the slide bar, owner will click “My properties”, the system will display the index page of choosing item, the owner will choose the item the owner wants to edit then click “see item”, the system will redirect to the page of item details.</p> <p>If the owner wants to edit item’s details, the admin must click the edit icon that is in item details, the pop-up will appear with wizard of changing details then clicks edit the changing data will be appeared</p> <p>If the owner wants to edit location, the owner must click edit icon that is in Location, the system will display location form, the admin must insert all fields in the wizard form then the admin clicks “edit location”, the system will display the page of item details or choose from map the owner will write the name of street then clicks search, the map center will get screen of this street to choose the right building using marker.</p>

- Login

SCENARIO ID	SRES-SC-69
SCENARIO NAME	Login
ACTOR	Owner, admin, customer
NORMAL	<p>The actor will open the site, login page will appear, the actor must fill the email and password fields.</p> <p>The system will verify this account if email and password are right the system will go to home page.</p> <p>If email or password are wrong so the system will send error message “Incorrect email or password”.</p>

- **Sign Up**

SCENARIO ID	SRES-SC-70
SCENARIO NAME	Sign Up
ACTOR	Admin, Customer
NORMAL	<p>If the actor enters the site the first time, sign up page will appear, the actor must fill the email and password fields.</p> <p>The system wants to check this email using sending email to this email to verify this account to let the actor enters the same number that is send to him.</p>

- **Follow Item**

SCENARIO ID	SRES-SC-71
SCENARIO NAME	Follow Item
ACTOR	Customer
NORMAL	The customer will log in to the system, the customer enters name of state then clicks search then choose the item, the system will redirect the item profile, the customer clicks "follow".

- **Unfollow Item**

SCENARIO ID	SRES-SC-72
SCENARIO NAME	Unfollow Item
ACTOR	Customer
NORMAL	The customer will log in to the system the customer enters name of state then clicks search then choose the item, the system will redirect the item profile, the customer clicks "unfollow".

- **Search by state**

SCENARIO ID	SRES-SC-73
SCENARIO NAME	Search by state
ACTOR	Customer
NORMAL	The customer will log in to the system, the customer enters name of state then clicks search the system will get all the properties in this state.

- **Be owner**

SCENARIO ID	SRES-SC-74
SCENARIO NAME	Be owner
ACTOR	Customer
NORMAL	The customer will log in to the system, if the customer has never been owner and wants to be the customer can click "Be Owner" from the header, the system will redirect to be owner page to enter data of owner in the pop-up model.

- **View item profile**

SCENARIO ID	SRES-SC-75
SCENARIO NAME	View item profile
ACTOR	Customer
NORMAL	The customer will log in to the system, the customer enters name of state then clicks "search" then choose the item, the system will redirect the item profile.

- **View item profile (Gallery)**

SCENARIO ID	SRES-SC-76
SCENARIO NAME	View item profile (Gallery)
ACTOR	Customer
NORMAL	The customer will log in to the system, the customer enters name of state then clicks search then choose the item, the system will redirect the item profile, clicks "Gallery", the system will redirect gallery page.

- **View item profile (Location)**

SCENARIO ID	SRES-SC-77
SCENARIO NAME	View item profile (Location)
ACTOR	Customer
NORMAL	The customer will log in to the system, the customer enters name of state then clicks search then choose the item, the system will redirect the item profile, clicks “Location”, the system will redirect location page.

- **View item profile (Review)**

SCENARIO ID	SRES-SC-78
SCENARIO NAME	View item profile (Review)
ACTOR	Customer
NORMAL	The customer will log in to the system, the customer enters name of state then clicks search then choose the item, the system will redirect the item profile, clicks “Review” the system will redirect the review page.

- **View item profile (schedule)**

SCENARIO ID	SRES-SC-79
SCENARIO NAME	View item profile (schedule)
ACTOR	Customer
NORMAL	The customer will log in to the system, the customer enters name of state then clicks search then choose the item, the system will open the item profile, clicks “schedule” the system will redirect schedule page.

- **Add review**

SCENARIO ID	SRES-SC-80
SCENARIO NAME	Add review
ACTOR	Customer
NORMAL	The customer will log in to the system, the customer enters name of state then clicks “search” then choose the item, the system will redirect the item profile , clicks “Review”, the system will redirect the page to add the review, choose the rating.

- Edit review

SCENARIO ID	SRES-SC-81
SCENARIO NAME	Edit review
ACTOR	Customer
NORMAL	The customer will log in to the system, the customer enters name of state then clicks “search” then choose the item, the system will redirect the item profile , clicks “Review” , the system will redirect the page to add the review, clicks edit then rate it.

- Pay

SCENARIO ID	SRES-SC-82
SCENARIO NAME	Pay
ACTOR	Customer
NORMAL	The customer will log in to the system, the customer enters name of state then clicks “search” then choose the item, the system will redirect the item profile, the customer knows everything about the item if the customer likes this item customer will clicks “reserve”, paypal page will appear to enter his credit card number to take the money from it.

- View my reservation

SCENARIO ID	SRES-SC-83
SCENARIO NAME	View my reservation
ACTOR	Customer
NORMAL	The customer will log in to the system, the customer open the slide bar, clicks “my reservation”, the system will redirect my reservation page with the customer information and reservations.

- **View item owner profile**

SCENARIO ID	SRES-SC-84
SCENARIO NAME	View item owner profile
ACTOR	Customer
NORMAL	The customer will log in to the system, the customer enters name of state then clicks “search” then choose the item, the system will redirect the item profile, the customer will click at the name of owner , the system will redirect item owner profile.

- **Write a post**

SCENARIO ID	SRES-SC-85
SCENARIO NAME	Write a post
ACTOR	Owner
NORMAL	The owner will log in to the system, the owner will clicks profile icon, clicks “my profile”, the system will redirect the owner profile, clicks write a post then the text area will appear to write inside it then click “post”.

- **Write a comment**

SCENARIO ID	SRES-SC-86
SCENARIO NAME	Write a comment
ACTOR	Owner
NORMAL	The actor will log in to the system, the owner will clicks profile icon, clicks “my profile”, the system will redirect the owner profile, clicks “write a comment” then the text area will appear to write inside it then click “comment” to a post.

- **Forget password**

SCENARIO ID	SRES-SC-87
SCENARIO NAME	Forget password
ACTOR	Owner customer
NORMAL	The actor will click “forget password”, the system will show a popup to write his email, The system will send an email to this email to know his new password and the system will show page of “forgotPassword” to know that email has been sent

- **Logout**

SCENARIO ID	SRES-SC-88
SCENARIO NAME	Logout
ACTOR	Owner customer
NORMAL	After the actor login in the system, actor will click Profile icon the clicks logout the system will return him to login page to change the entered account

- **Search by state and date**

SCENARIO ID	SRES-SC-89
SCENARIO NAME	Search by state and date
ACTOR	Customer
NORMAL	The customer will log in to the system, the customer enters name of state, the arrival date and the departure date then clicks search the system will get all the properties in this state and allow in this period of time.

- **Follow user**

SCENARIO ID	SRES-SC-90
SCENARIO NAME	Follow user
ACTOR	Customer
NORMAL	The customer will log in to the system, the customer enters name of state then clicks search then choose the item, the system will redirect the item profile, the he customer will click owner name then , the customer clicks "follow" or the customer will find this user in the comments then clicks on user name and clicks follow.

- **Unfollow User**

SCENARIO ID	SRES-SC-91
SCENARIO NAME	Unfollow User
ACTOR	Customer
NORMAL	The customer will log in to the system, then the customer will get the followers then clicks the user wants to unfollow then clicks unfollow

- **Write a comment**

SCENARIO ID	SRES-SC-92
SCENARIO NAME	Write a comment
ACTOR	Customer
NORMAL	The actor will log in to the system, The actor will search for property, the system will get item profile then write comment under specific post then clicks “comment”, the system will redirect the same page with a comment under the post

- **Delete a comment**

SCENARIO ID	SRES-SC-93
SCENARIO NAME	Delete a comment
ACTOR	Customer
NORMAL	The actor will log in to the system, The actor will search for property, the system will get item profile then clicks delete icon of this comment , the system will redirect the same page without the comments and replies of this comments

- **Reply Comment**

SCENARIO ID	SRES-SC-94
SCENARIO NAME	Reply Comment
ACTOR	Customer
NORMAL	The actor will log in to the system, The actor will search for property, the system will get item profile then clicks edit that is inside comment under specific post, a popup will appear with a textbox to write comment. The system will redirect same page with the edited comment

- **Write a reply**

SCENARIO ID	SRES-SC-95
SCENARIO NAME	Write a reply
ACTOR	Customer
NORMAL	The actor will log in to the system, The actor will search for property, the system will get item profile then write reply under specific comment under specific post then clicks "reply" ,the system will redirect the same page with a reply under the Comment

- **Delete a reply**

SCENARIO ID	SRES-SC-96
SCENARIO NAME	Delete a reply
ACTOR	Customer
NORMAL	The actor will log in to the system, The actor will search for property, the system will get item profile then clicks delete icon of this reply , the system will redirect the same page without the reply of this comments

- **Review Item**

SCENARIO ID	SRES-SC-97
SCENARIO NAME	Review Item
ACTOR	Customer
NORMAL	The actor will log in to the system, The actor will search for property, the system will get item profile then clicks review , the system will appear the “review” page then choose number of stars the system will redirect the same page with change in number of stars

- **Edit Review Item**

SCENARIO ID	SRES-SC-98
SCENARIO NAME	Edit Review Item
ACTOR	Customer
NORMAL	The actor will log in to the system, the actor will search for property, the system will get item profile then clicks review, the system will appear the “review” page then I will change number of stars I already make. The system will redirect same page with change in number of stars

- **Add post**

SCENARIO ID	SRES-SC-99
SCENARIO NAME	Add post
ACTOR	Customer
NORMAL	The actor will log in to the system, the actor will search for property, the system will get item profile then clicks “write post” the text area will appear to write in it then clicks post The system will redirect same page with a new post.

- **Edit post**

SCENARIO ID	SRES-SC-100
SCENARIO NAME	Edit post
ACTOR	Customer
NORMAL	The actor will log in to the system, the actor will search for property, the system will get item profile then clicks “edit post” a popup will appear with text area to write in it then clicks edit The system will redirect same page with an edited post.

- **Delete post**

SCENARIO ID	SRES-SC-101
SCENARIO NAME	Delete post
ACTOR	Customer
NORMAL	<p>The actor will log in to the system, the actor will search for property, the system will get item profile then clicks “delete” to the specific post the system will redirect same page without post, comments and replies</p> <p>The system will redirect same page with an edited post.</p>

- **View his reservations**

SCENARIO ID	SRES-SC-102
SCENARIO NAME	View his reservations
ACTOR	Customer
NORMAL	<p>The actor will log in to the system, the actor will press “my reservations” the system will get Show_Reservation page with the reservation of the customer.</p>

- **Add cover photo**

SCENARIO ID	SRES-SC-103
SCENARIO NAME	Add cover photo
ACTOR	Customer
NORMAL	<p>The customer will log in to the system, clicks profile icon, clicks profile, the system will return customer profile, the user will click add photo the system will open the screen of choosing image from the memory of computer then clicks on this photo and click add, the system will redirect the same page with the new cover photo.</p> <p>If the entered data does not match the required format, error message will occur “Please matches the requested format”.</p>

- **Edit cover photo**

SCENARIO ID	SRES-SC-104
SCENARIO NAME	Edit cover photo
ACTOR	Customer
NORMAL	<p>The customer will log in to the system, clicks profile icon, clicks profile, the system will return customer profile, the user will click cover edit photo the system will open the screen of choosing image from the memory of computer then clicks on this photo and click edit, the system will redirect the same page with the edited photo.</p> <p>If the entered data does not match the required format, error message will occur "Please matches the requested format".</p>

- **Delete cover photo**

SCENARIO ID	SRES-SC-105
SCENARIO NAME	Delete cover photo
ACTOR	Customer
NORMAL	<p>The customer will log in to the system, clicks profile icon, clicks profile, the system will return customer profile, the user will click cover delete icon, the system will redirect same page with the default cover picture.</p>

- **Add profile photo**

SCENARIO ID	SRES-SC-106
SCENARIO NAME	Add profile photo
ACTOR	Customer
NORMAL	<p>The customer will log in to the system, clicks profile icon, clicks profile, the system will return customer profile picture, the user will click profile add photo the system will open the screen of choosing image from the memory of computer then clicks on this photo and click add, the system will redirect the same page with the new profile photo.</p> <p>If the entered data does not match the required format, error message will occur "Please matches the requested format".</p>

- **Edit profile photo**

SCENARIO ID	SRES-SC-107
SCENARIO NAME	Edit profile photo
ACTOR	Customer
NORMAL	<p>The customer will log in to the system, clicks profile icon, clicks profile, the system will return customer profile picture, the user will click edit profile photo the system will open the screen of choosing image from the memory of computer then clicks on this photo and click edit, the system will redirect the same page with the edited profile photo.</p> <p>If the entered data does not match the required format, error message will occur "Please matches the requested format".</p>

- **Delete profile photo**

SCENARIO ID	SRES-SC-108
SCENARIO NAME	Delete profile photo
ACTOR	Customer
NORMAL	<p>The customer will log in to the system, clicks profile icon, clicks profile, the system will return customer profile picture, the user will click profile picture delete icon, the system will redirect same page with the default profile picture.</p>

- **Edit profile detail**

SCENARIO ID	SRES-SC-109
SCENARIO NAME	Edit profile detail
ACTOR	Customer
NORMAL	<p>The customer will log in to the system, from the sidebar, the actor will click "edit profile" the system will get edit user profile, a form will appear to change the information the actor wants to change then clicks save, the system will redirect the same page with the edited data.</p>

- Add cover photo

SCENARIO ID	SRES-SC-110
SCENARIO NAME	Add cover photo
ACTOR	Owner
NORMAL	<p>The Owner will log in to the system, clicks profile icon, clicks profile, the system will return customer profile, the user will click cover add photo the system will open the screen of choosing image from the memory of computer then clicks on this photo and click add, the system will redirect the same page with the new cover photo.</p> <p>If the entered data does not match the required format, error message will occur “Please matches the requested format”.</p>

- Edit cover photo

SCENARIO ID	SRES-SC-111
SCENARIO NAME	Edit cover photo
ACTOR	Owner
NORMAL	<p>The Owner will log in to the system, clicks profile icon, clicks profile, the system will return customer profile, the user will click cover edit photo the system will open the screen of choosing image from the memory of computer then clicks on this photo and click edit, the system will redirect the same page with the edited photo.</p> <p>If the entered data does not match the required format, error message will occur “Please matches the requested format”.</p>

- Delete cover photo

SCENARIO ID	SRES-SC-112
SCENARIO NAME	Delete cover photo
ACTOR	Owner
NORMAL	<p>The Owner will log in to the system, clicks profile icon, clicks profile, the system will return customer profile, the user will click cover delete icon, the system will redirect same page with the default cover picture.</p>

- Add profile photo

SCENARIO ID	SRES-SC-113
SCENARIO NAME	Add profile photo
ACTOR	Owner
NORMAL	<p>The Owner will log in to the system, clicks profile icon, clicks profile, the system will return customer profile picture, the user will click profile add photo the system will open the screen of choosing image from the memory of computer then clicks on this photo and click add, the system will redirect the same page with the new profile photo.</p> <p>If the entered data does not match the required format, error message will occur "Please matches the requested format".</p>

- Edit profile photo

SCENARIO ID	SRES-SC-114
SCENARIO NAME	Edit profile photo
ACTOR	Owner
NORMAL	<p>The Owner will log in to the system, clicks profile icon, clicks profile, the system will return customer profile picture, the user will click edit profile photo the system will open the screen of choosing image from the memory of computer then clicks on this photo and click edit, the system will redirect the same page with the edited profile photo.</p> <p>If the entered data does not match the required format, error message will occur "Please matches the requested format".</p>

- Delete profile photo

SCENARIO ID	SRES-SC-115
SCENARIO NAME	Delete profile photo
ACTOR	Owner
NORMAL	<p>The Owner will log in to the system, clicks profile icon, clicks profile, the system will return customer profile picture, the user will click profile picture delete icon, the system will redirect same page with the default profile picture.</p>

- **Delete item photo**

SCENARIO ID	SRES-SC-116
SCENARIO NAME	Delete item photo
ACTOR	Owner
NORMAL	The Owner will log in to the system, From the sidebar, the actor will click my items, the system will view the show item page, clicks item name, the system will view item profile page then then choose gallery, the system will view item profile gallery, the actor chooses the photo wants to delete then clicks delete icon, the system will redirect the same page without this picture.

- **Delete Item**

SCENARIO ID	SRES-SC-117
SCENARIO NAME	Delete Item
ACTOR	Owner
NORMAL	The Owner will log in to the system, From the sidebar, the actor will click my items, the system will view the show item page, clicks the delete icon of this item , the system will redirect same page without this item.

- **Edit post**

SCENARIO ID	SRES-SC-118
SCENARIO NAME	Edit post
ACTOR	Owner
NORMAL	The actor will log in to the system, From the slide bar, the owner will press on my items, the system will redirect show item page, clicks on item name, the system will view item profile page, the owner will click edit icon of the post wants to be edited, a popup will be appeared to write the new post then clicks edit, the system will redirect the same page with the edited post.

- **Delete post**

SCENARIO ID	SRES-SC-119
SCENARIO NAME	Delete post
ACTOR	owner
NORMAL	The actor will log in to the system, From the slide bar, the owner will press on my items, the system will redirect show item page, clicks on item name, the system will view item profile page, the owner will click delete icon of the post want to be deleted the system will redirect same page without the post and its comments and replies.

- **Edit comment**

SCENARIO ID	SRES-SC-120
SCENARIO NAME	Edit comment
ACTOR	Owner
NORMAL	The actor will log in to the system, From the slide bar, the owner will press on my items, the system will redirect show item page, clicks on item name, the system will view item profile page, the owner will click edit icon of the comment of specific post wants to be edited, a popup will be appeared to write the new comment then clicks edit, the system will redirect the same page with the edited comment.

- **Delete comment**

SCENARIO ID	SRES-SC-121
SCENARIO NAME	Delete comment
ACTOR	owner
NORMAL	The actor will log in to the system, From the slide bar, the owner will press on my items, the system will redirect show item page, clicks on item name, the system will view item profile page, the owner will click delete icon of the comment of specific post want to be deleted the system will redirect same page without the post and its comments and replies.

- Write a reply

SCENARIO ID	SRES-SC-122
SCENARIO NAME	Write a reply
ACTOR	Owner
NORMAL	The actor will log in to the system, From the slide bar, the owner will press on my items, the system will redirect show item page, clicks on item name, the system will view item profile page, the owner will click reply of the comment of specific post, textarea will appear to write in it the reply want to be added, the system will redirect same page with new reply.

- Delete Reply

SCENARIO ID	SRES-SC-123
SCENARIO NAME	Delete Reply
ACTOR	Owner
NORMAL	The actor will log in to the system, From the slide bar, the owner will press on my items, the system will redirect show item page, clicks on item name, the system will view item profile page, the owner will click delete icon of the reply of the comment of specific post, the system will redirect same page without the chosen reply.

- Edit a reply

SCENARIO ID	SRES-SC-124
SCENARIO NAME	Edit a reply
ACTOR	Owner
NORMAL	The actor will log in to the system, From the slide bar, the owner will press on my items, the system will redirect show item page, clicks on item name, the system will view item profile page, the owner will click edit icon of the reply of the comment of specific post, textarea will appear to write in it the reply wants to be edited, the system will redirect same page with edited reply.

- **View items reservation**

SCENARIO ID	SRES-SC-125
SCENARIO NAME	View items reservation
ACTOR	Owner
NORMAL	The actor will log in to the system, From the slide bar, the owner will press on my items, the system will redirect show item page, clicks on item name, the system will view item profile page, the owner will click Item reservation, the system will get all data about the item reservations and put them in item profile reservation page.

- **Show Items**

SCENARIO ID	SRES-SC-126
SCENARIO NAME	Show Items
ACTOR	Owner
NORMAL	The actor will log in to the system, From the slide bar, the owner will press on my items, the system will redirect show item page.

Chapter 3: System Analysis

4.1 System Architecture

4.1.1 Application Architecture

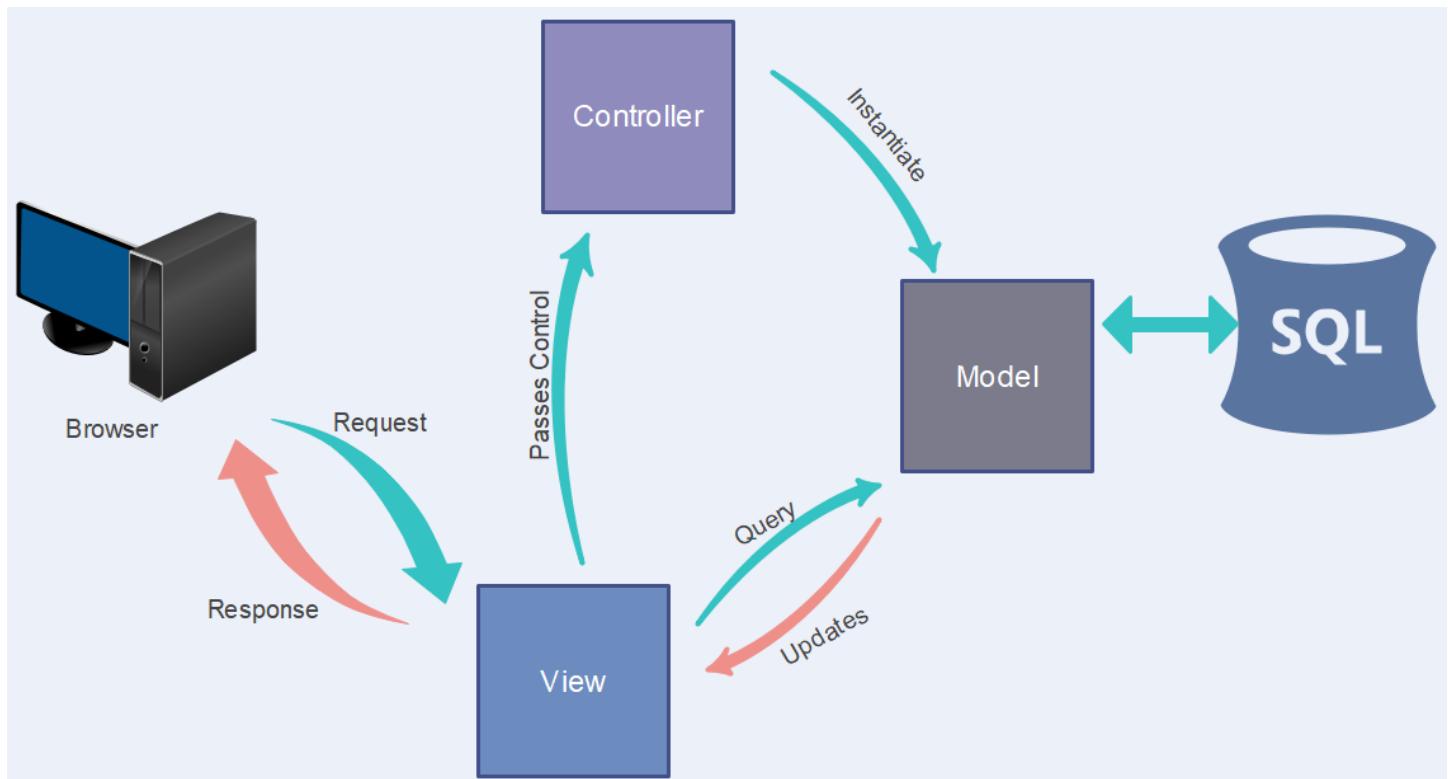


Figure 7: Application Architecture

4.2 class diagram

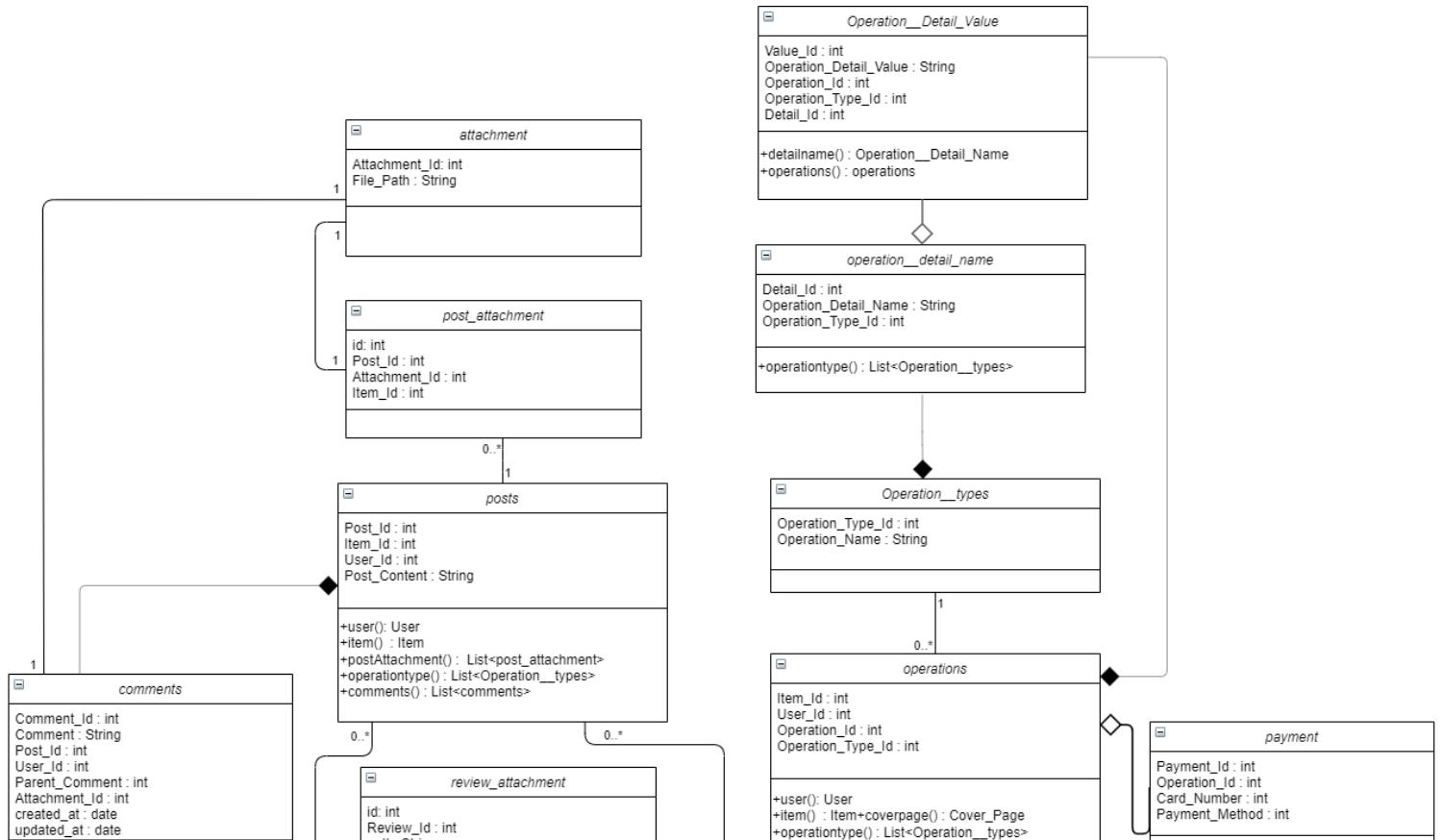


Figure 8: Class Diagram 1

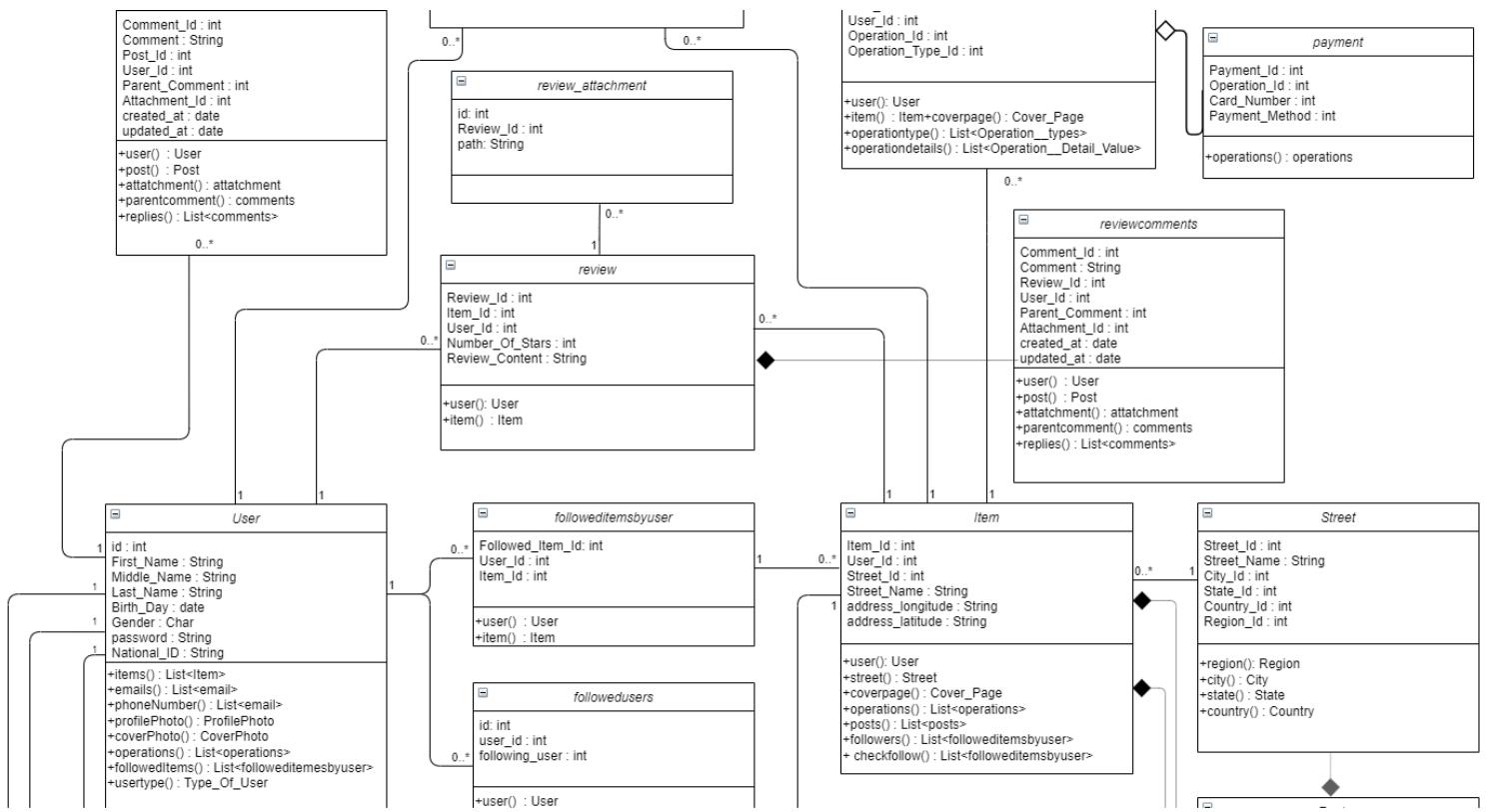


Figure 9: Class Diagram 2

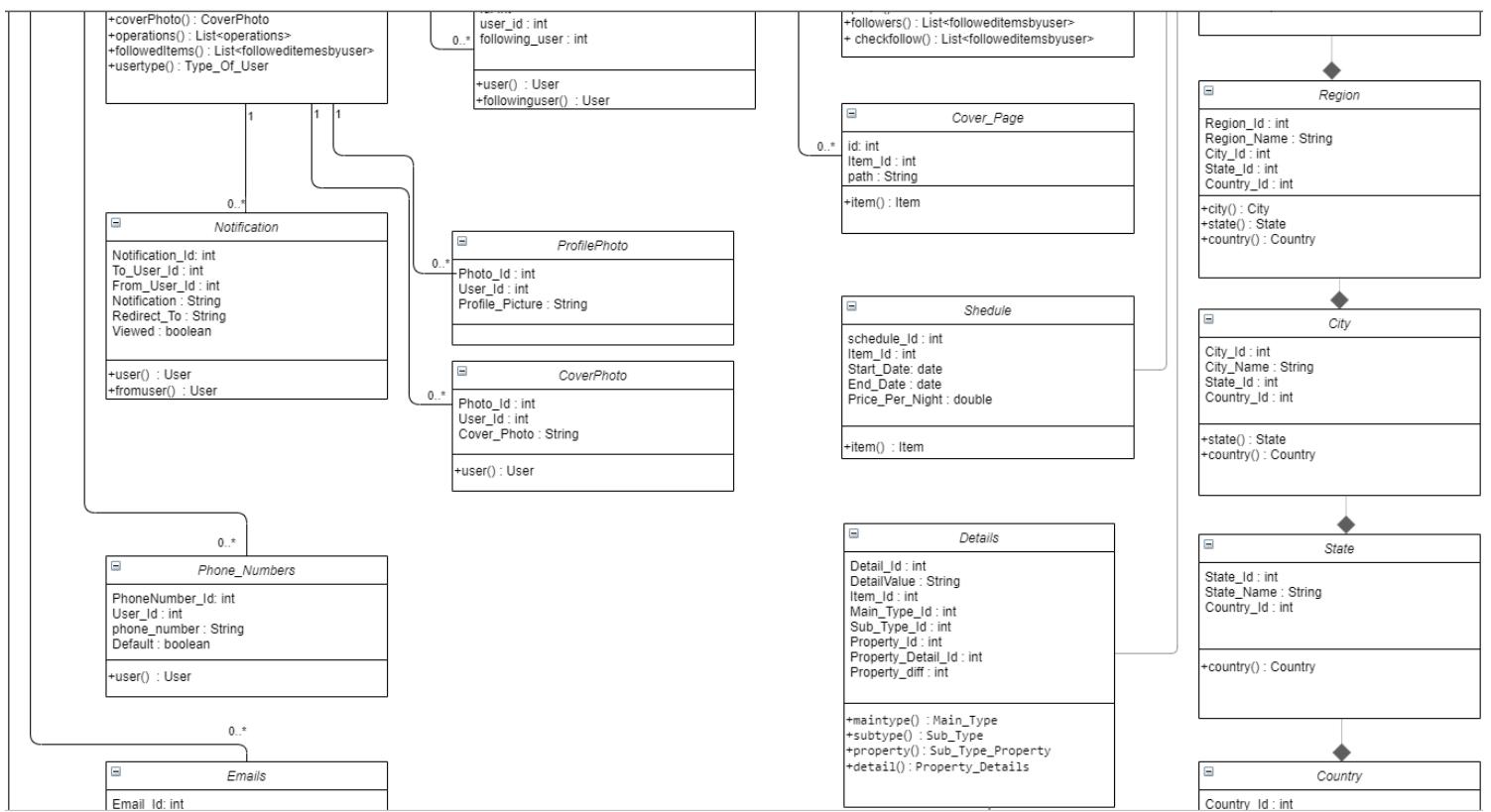


Figure 10: Class Diagram 3

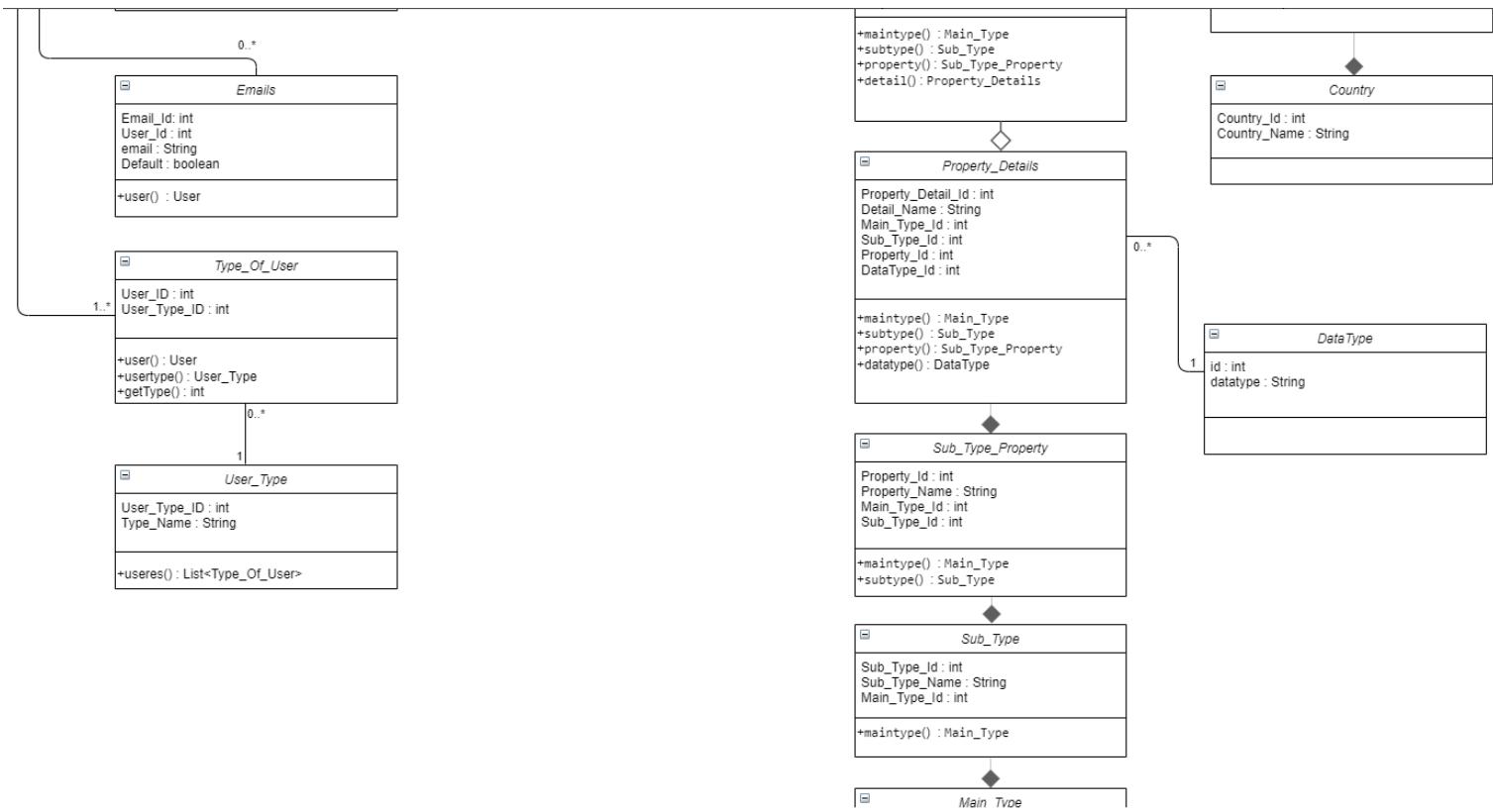


Figure 11: Class Diagram 4

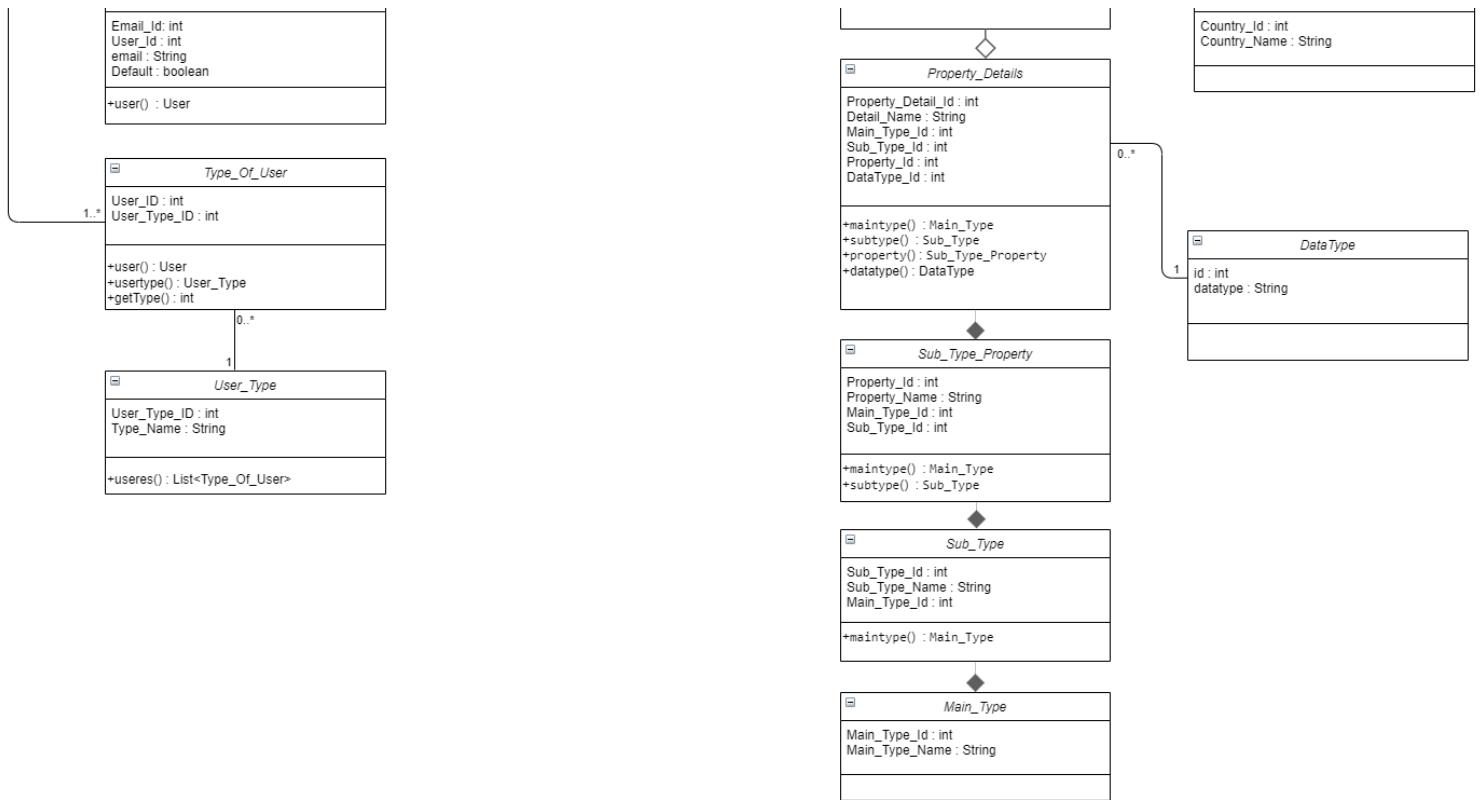


Figure 12: Class Diagram 5

4.3. Physical Model

1- attachments

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Attachment_Id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	File_Path	varchar(255)	utf8mb4_unicode_ci		No	None		
3	created_at	timestamp			Yes	NULL		
4	updated_at	timestamp			Yes	NULL		

2- cities:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	City_Id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Country_Id 	bigint(20)		UNSIGNED	No	None		
3	State_Id 	bigint(20)		UNSIGNED	No	None		
4	City_Name 	varchar(255)	utf8mb4_unicode_ci		No	None		
5	created_at	timestamp			Yes	NULL		
6	updated_at	timestamp			Yes	NULL		

3- comments:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Comment_Id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Post_Id 	bigint(20)		UNSIGNED	No	None		
3	User_Id 	bigint(20)		UNSIGNED	No	None		
4	Attachment_Id 	bigint(20)		UNSIGNED	Yes	NULL		
5	Comment	varchar(255)	utf8mb4_unicode_ci		No	None		
6	Parent_Comment	int(11)			Yes	NULL		
7	created_at	timestamp			Yes	NULL		
8	updated_at	timestamp			Yes	NULL		

4- counties:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Country_Id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Country_Name 	varchar(255)	utf8mb4_unicode_ci		No	None		
3	created_at	timestamp			Yes	NULL		
4	updated_at	timestamp			Yes	NULL		

5- cover_photos:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Photo_Id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	User_Id	bigint(20)		UNSIGNED	No	None		
3	Cover_Photo	varchar(100)	utf8mb4_unicode_ci		No	None		
4	created_at	timestamp			Yes	NULL		
5	updated_at	timestamp			Yes	NULL		

6- cover_pages:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Item_Id	bigint(20)		UNSIGNED	No	None		
3	path	varchar(255)	utf8mb4_unicode_ci		No	None		
4	created_at	timestamp			Yes	NULL		
5	updated_at	timestamp			Yes	NULL		

7- datatypes:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	datatype	varchar(255)	utf8mb4_unicode_ci		No	None		
3	created_at	timestamp			Yes	NULL		
4	updated_at	timestamp			Yes	NULL		

8- details:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Detail_Id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Item_Id	bigint(20)		UNSIGNED	No	None		
3	Main_Type_Id	bigint(20)		UNSIGNED	No	None		
4	Sub_Type_Id	bigint(20)		UNSIGNED	No	None		
5	Property_Id	bigint(20)		UNSIGNED	No	None		
6	Property_Detail_Id	bigint(20)		UNSIGNED	No	None		
7	Property_diff	bigint(20)			No	None		
8	DetailValue	varchar(255)	utf8mb4_unicode_ci		No	None		
9	created_at	timestamp			Yes	NULL		
10	updated_at	timestamp			Yes	NULL		

9- emails:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Email_Id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	User_ID 	bigint(20)		UNSIGNED	No	None		
3	email 	varchar(255)	utf8mb4_unicode_ci		No	None		
4	Default	tinyint(1)			Yes	NULL		
5	email_verified_at	timestamp			Yes	NULL		
6	created_at	timestamp			Yes	NULL		
7	updated_at	timestamp			Yes	NULL		

10- followeditemsbyusers:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Followed_Item_Id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	User_ID 	bigint(20)		UNSIGNED	No	None		
3	Item_Id 	bigint(20)		UNSIGNED	No	None		
4	created_at	timestamp			Yes	NULL		
5	updated_at	timestamp			Yes	NULL		

11- followedusers:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	user_id 	bigint(20)		UNSIGNED	No	None		
3	following_user 	bigint(20)		UNSIGNED	No	None		
4	created_at	timestamp			Yes	NULL		
5	updated_at	timestamp			Yes	NULL		

12- items:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Item_Id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Street_Id 	bigint(20)		UNSIGNED	No	None		
3	User_Id 	bigint(20)		UNSIGNED	No	None		
4	Item_Name	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
5	created_at	timestamp			Yes	NULL		
6	updated_at	timestamp			Yes	NULL		
7	address_longitude	varchar(100)	utf8mb4_unicode_ci		Yes	NULL		
8	address_latitude	varchar(100)	utf8mb4_unicode_ci		Yes	NULL		
9	visible	tinyint(1)			No	0		

13- main_types:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Main_Type_Id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Main_Type_Name 	varchar(255)	utf8mb4_unicode_ci		No	None		
3	created_at	timestamp			Yes	NULL		
4	updated_at	timestamp			Yes	NULL		

14- notifications:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Notification_Id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	To_User_Id 	bigint(20)		UNSIGNED	No	None		
3	From_User_Id 	bigint(20)		UNSIGNED	No	None		
4	Notification	varchar(255)	utf8mb4_unicode_ci		No	None		
5	Redirect_To	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
6	Viewed	tinyint(1)			No	0		
7	created_at	timestamp			Yes	NULL		
8	updated_at	timestamp			Yes	NULL		

15- operation_detail_name:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Detail_Id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Operation_Type_Id 	bigint(20)		UNSIGNED	No	None		
3	Operation_Detail_Name	varchar(255)	utf8mb4_unicode_ci		No	None		
4	created_at	timestamp			Yes	NULL		
5	updated_at	timestamp			Yes	NULL		

16- operation_types:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Operation_Type_Id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Operation_Name	varchar(255)	utf8mb4_unicode_ci		No	None		
3	created_at	timestamp			Yes	NULL		
4	updated_at	timestamp			Yes	NULL		

17- operation_detail_values:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Value_Id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Operation_Id	bigint(20)		UNSIGNED	No	None		
3	Operation_Type_Id	bigint(20)		UNSIGNED	No	None		
4	Detail_Id	bigint(20)		UNSIGNED	No	None		
5	Operation_Detail_Value	varchar(255)	utf8mb4_unicode_ci		No	None		
6	created_at	timestamp			Yes	NULL		
7	updated_at	timestamp			Yes	NULL		

18- operations:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Operation_Id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Item_Id	bigint(20)		UNSIGNED	No	None		
3	User_Id	bigint(20)		UNSIGNED	No	None		
4	created_at	timestamp			Yes	NULL		
5	updated_at	timestamp			Yes	NULL		

19- payments:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Payment_Id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Operation_Id	bigint(20)		UNSIGNED	No	None		
3	Payment_Method	varchar(255)	utf8mb4_unicode_ci		No	None		
4	Card_Number	varchar(255)	utf8mb4_unicode_ci		No	None		
5	Paid_Amount	double			No	None		
6	confirmed	tinyint(1)			No	None		
7	created_at	timestamp			Yes	NULL		
8	updated_at	timestamp			Yes	NULL		

21- phone_numbers:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	PhoneNumber_Id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	User_ID	bigint(20)		UNSIGNED	No	None		
3	phone_number	varchar(255)	utf8mb4_unicode_ci		No	None		
4	Default	tinyint(1)			No	None		
5	created_at	timestamp			Yes	NULL		
6	updated_at	timestamp			Yes	NULL		

22- posts:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Post_Id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Item_Id	bigint(20)		UNSIGNED	Yes	NULL		
3	User_Id	bigint(20)		UNSIGNED	No	None		
4	Post_Title	varchar(255)	utf8mb4_unicode_ci		No	None		
5	Post_Content	varchar(255)	utf8mb4_unicode_ci		No	None		
6	created_at	timestamp			Yes	NULL		
7	updated_at	timestamp			Yes	NULL		

23- post_attachments:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Post_Id	bigint(20)		UNSIGNED	Yes	NULL		
3	Attachment_Id	bigint(20)		UNSIGNED	No	None		
4	Item_Id	bigint(20)		UNSIGNED	Yes	NULL		
5	created_at	timestamp			Yes	NULL		
6	updated_at	timestamp			Yes	NULL		

24- profile_photos:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Photo_Id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Profile_Picture	varchar(100)	utf8mb4_unicode_ci		No	None		
3	User_Id	bigint(20)		UNSIGNED	No	None		
4	created_at	timestamp			Yes	NULL		
5	updated_at	timestamp			Yes	NULL		

25- property__details:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Property_Detail_Id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Main_Type_Id	bigint(20)		UNSIGNED	No	None		
3	Sub_Type_Id	bigint(20)		UNSIGNED	No	None		
4	Property_Id	bigint(20)		UNSIGNED	No	None		
5	DataType_Id	bigint(20)		UNSIGNED	No	None		
6	Detail_Name	varchar(255)	utf8mb4_unicode_ci		No	None		
7	created_at	timestamp			Yes	NULL		
8	updated_at	timestamp			Yes	NULL		

26- regions:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Region_Id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Country_Id	bigint(20)		UNSIGNED	No	None		
3	State_Id	bigint(20)		UNSIGNED	No	None		
4	City_Id	bigint(20)		UNSIGNED	No	None		
5	Region_Name	varchar(255)	utf8mb4_unicode_ci		No	None		
6	created_at	timestamp			Yes	NULL		
7	updated_at	timestamp			Yes	NULL		

27- reviews:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Review_Id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Item_Id	bigint(20)		UNSIGNED	No	None		
3	User_Id	bigint(20)		UNSIGNED	No	None		
4	Review_Title	varchar(255)	utf8mb4_unicode_ci		No	None		
5	Review_Content	varchar(255)	utf8mb4_unicode_ci		No	None		
6	Number_Of_Stars	int(11)			No	None		
7	created_at	timestamp			Yes	NULL		
8	updated_at	timestamp			Yes	NULL		

28- review_attachments:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Review_Id 	bigint(20)		UNSIGNED	No	None		
3	Item_Id 	bigint(20)		UNSIGNED	No	None		
4	path	varchar(255)	utf8mb4_unicode_ci		No	None		
5	created_at	timestamp			Yes	NULL		
6	updated_at	timestamp			Yes	NULL		

29- reviewcomments:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Comment_Id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Post_Id 	bigint(20)		UNSIGNED	No	None		
3	User_Id 	bigint(20)		UNSIGNED	No	None		
4	Attachment_Id 	bigint(20)		UNSIGNED	No	None		
5	Comment	varchar(255)	utf8mb4_unicode_ci		No	None		
6	Parent_Comment	int(11)			No	None		
7	created_at	timestamp			Yes	NULL		
8	updated_at	timestamp			Yes	NULL		

30- states:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	State_Id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Country_Id 	bigint(20)		UNSIGNED	No	None		
3	State_Name 	varchar(255)	utf8mb4_unicode_ci		No	None		
4	created_at	timestamp			Yes	NULL		
5	updated_at	timestamp			Yes	NULL		

31- schedules:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	schedule_Id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Item_Id	bigint(20)		UNSIGNED	No	None		
3	Start_Date	date			No	None		
4	End_Date	date			No	None		
5	Price_Per_Night	double			No	None		
6	created_at	timestamp			Yes	NULL		
7	updated_at	timestamp			Yes	NULL		

32- streets:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Street_Id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Country_Id	bigint(20)		UNSIGNED	No	None		
3	State_Id	bigint(20)		UNSIGNED	No	None		
4	City_Id	bigint(20)		UNSIGNED	No	None		
5	Region_Id	bigint(20)		UNSIGNED	No	None		
6	Street_Name	varchar(255)	utf8mb4_unicode_ci		No	None		
7	created_at	timestamp			Yes	NULL		
8	updated_at	timestamp			Yes	NULL		

33- syb_types:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Sub_Type_Id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Main_Type_Id	bigint(20)		UNSIGNED	No	None		
3	Sub_Type_Name	varchar(255)	utf8mb4_unicode_ci		No	None		
4	created_at	timestamp			Yes	NULL		
5	updated_at	timestamp			Yes	NULL		

34- syb_type_properties:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Property_Id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Main_Type_Id	bigint(20)		UNSIGNED	No	None		
3	Sub_Type_Id	bigint(20)		UNSIGNED	No	None		
4	Property_Name	varchar(255)	utf8mb4_unicode_ci		No	None		
5	created_at	timestamp			Yes	NULL		
6	updated_at	timestamp			Yes	NULL		

35- type_of_users:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	int(11)			No	None		AUTO_INCREMENT
2	User_ID 	bigint(20)		UNSIGNED	No	None		
3	User_Type_ID 	bigint(20)		UNSIGNED	No	None		
4	created_at	timestamp			Yes	NULL		
5	updated_at	timestamp			Yes	NULL		

36- users:

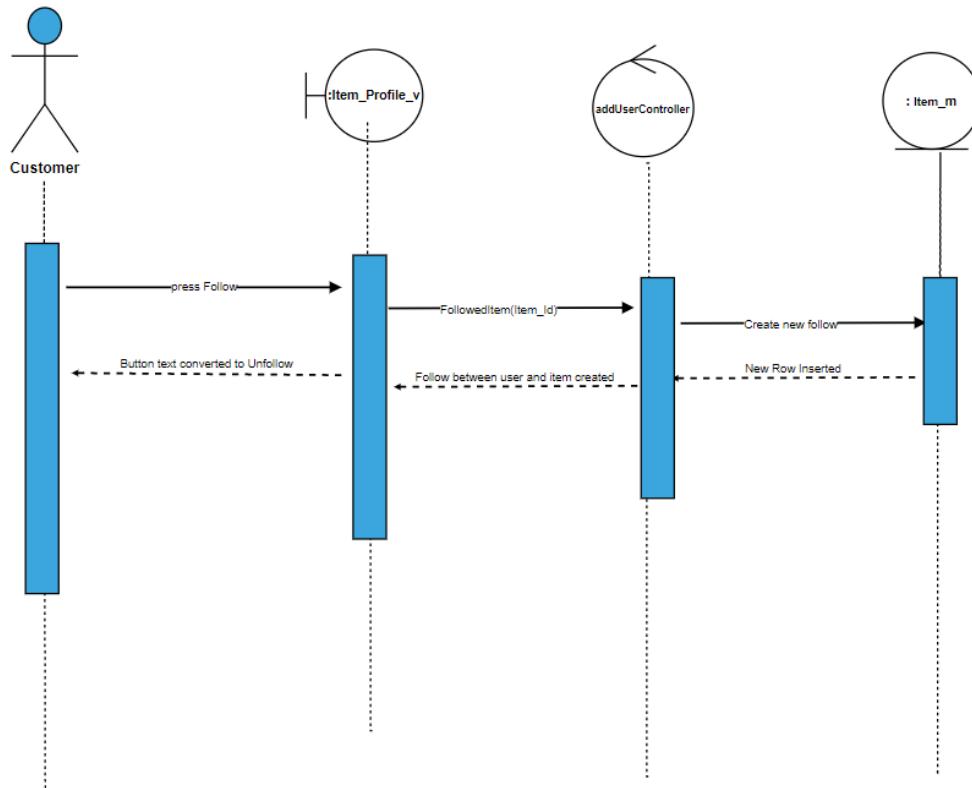
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Image	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
3	First_Name	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
4	Middle_Name	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
5	Last_Name	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
6	Birth_Day	date			Yes	NULL		
7	Gender	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
8	password	varchar(255)	utf8mb4_unicode_ci		No	None		
9	National_ID	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
10	remember_token	varchar(100)	utf8mb4_unicode_ci		Yes	NULL		
11	created_at	timestamp			Yes	NULL		
12	updated_at	timestamp			Yes	NULL		

37- user_types:

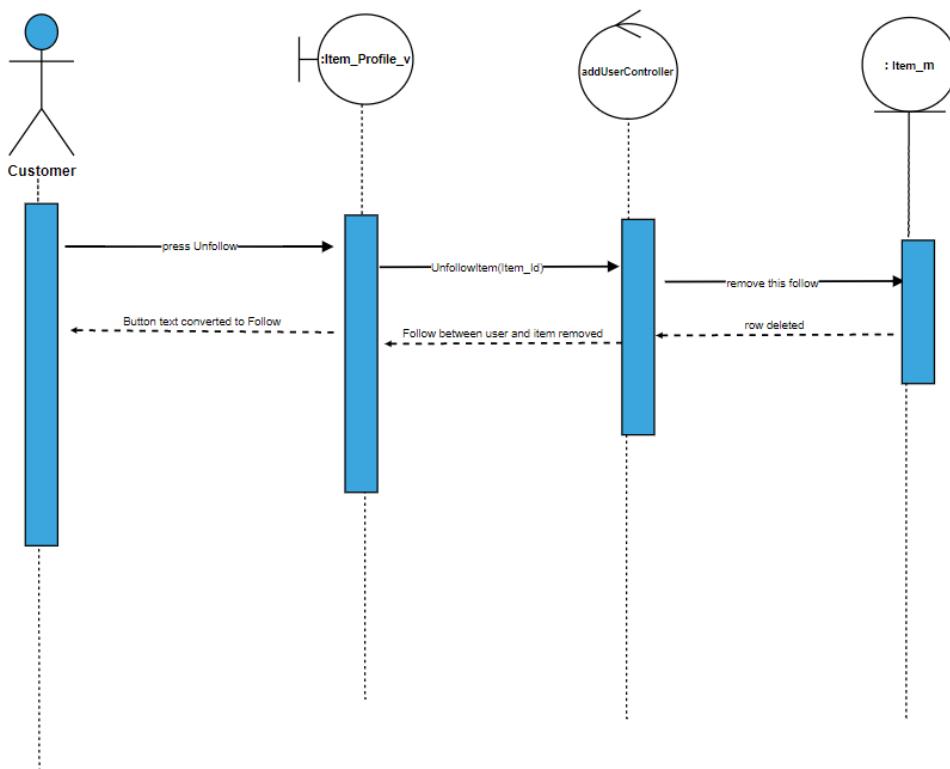
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	User_Type_ID 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	Type_Name 	varchar(255)	utf8mb4_unicode_ci		No	None		
3	created_at	timestamp			Yes	NULL		
4	updated_at	timestamp			Yes	NULL		

4.4 Sequence Diagram

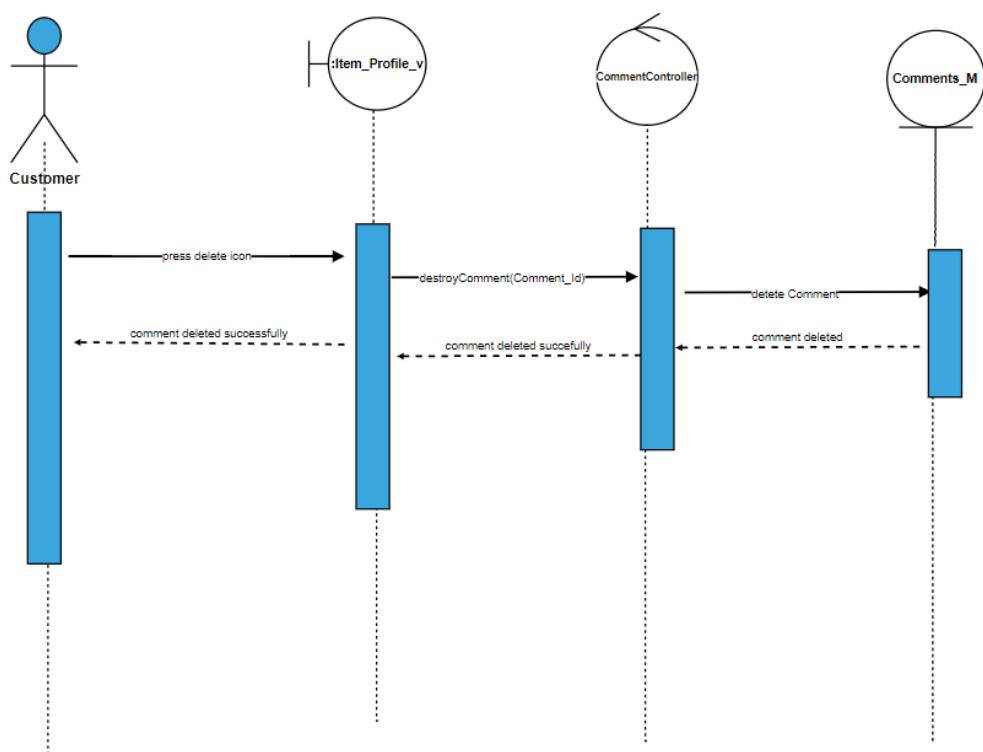
1) Follow Item



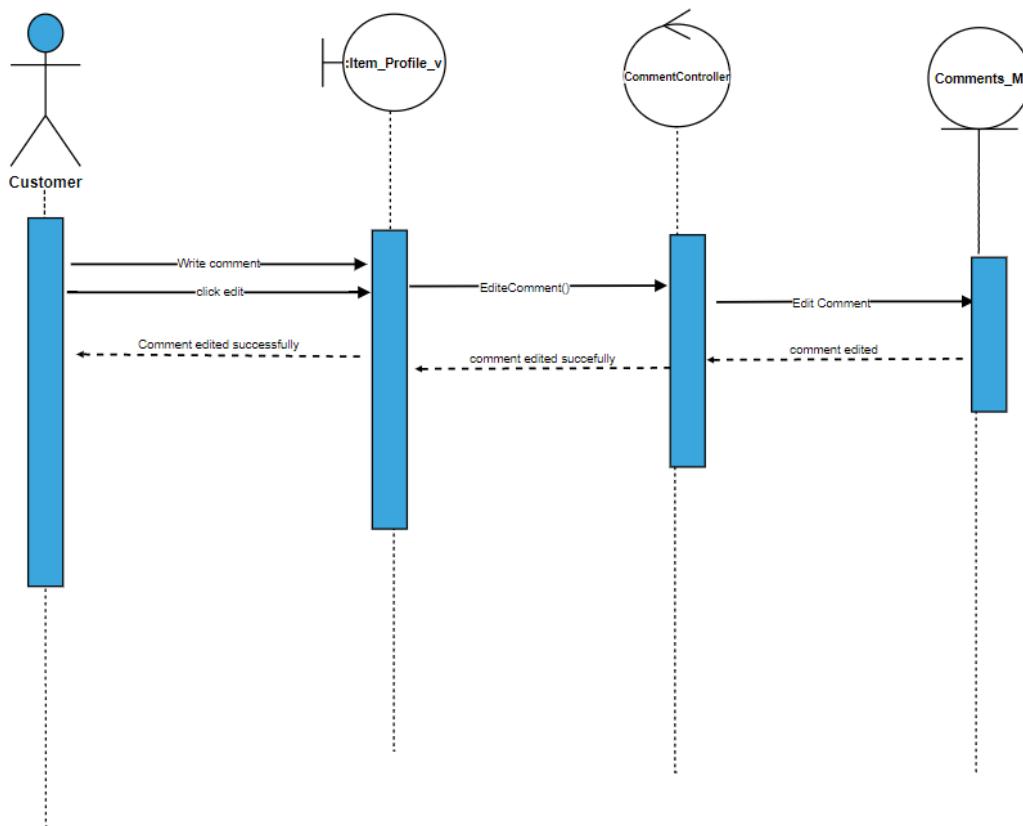
2) UnFollow Item



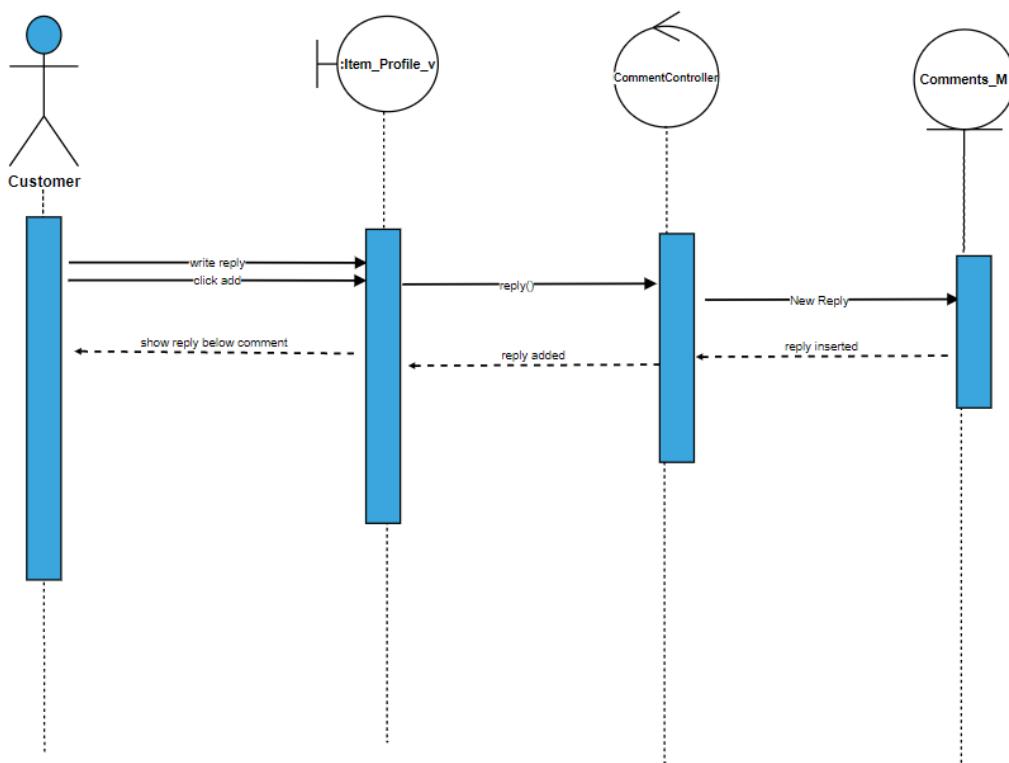
3)Delete Comment



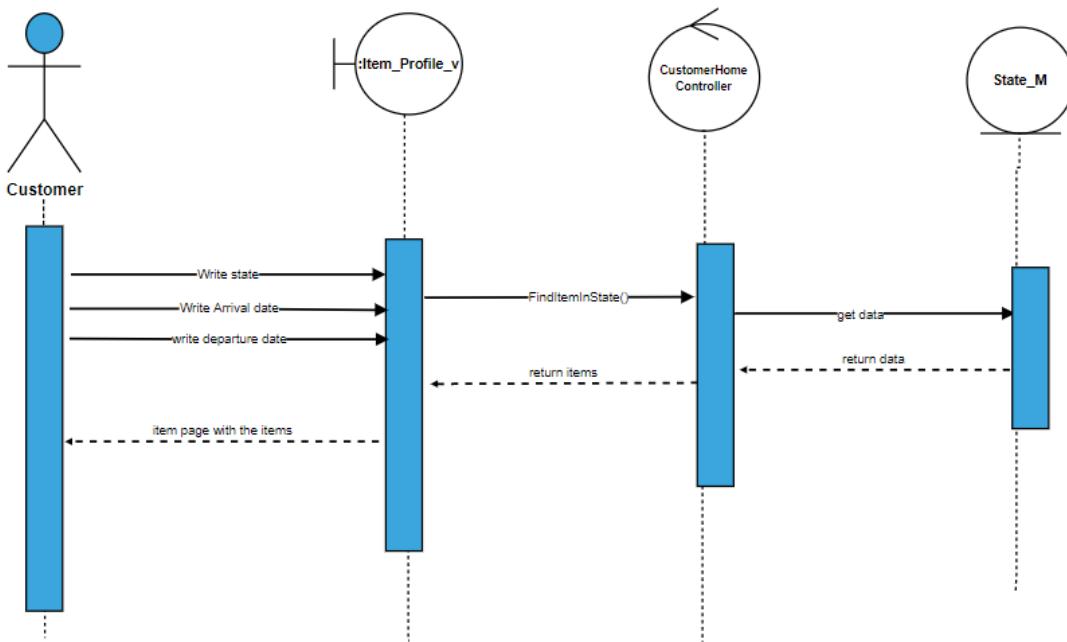
4) Comment edited



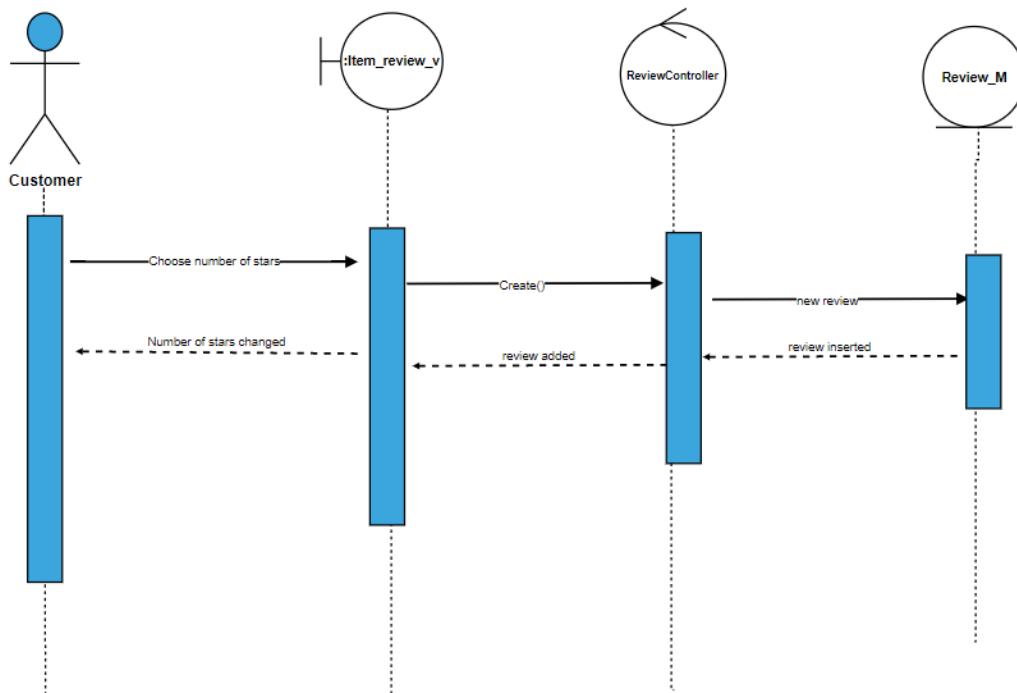
5) Reply Comment



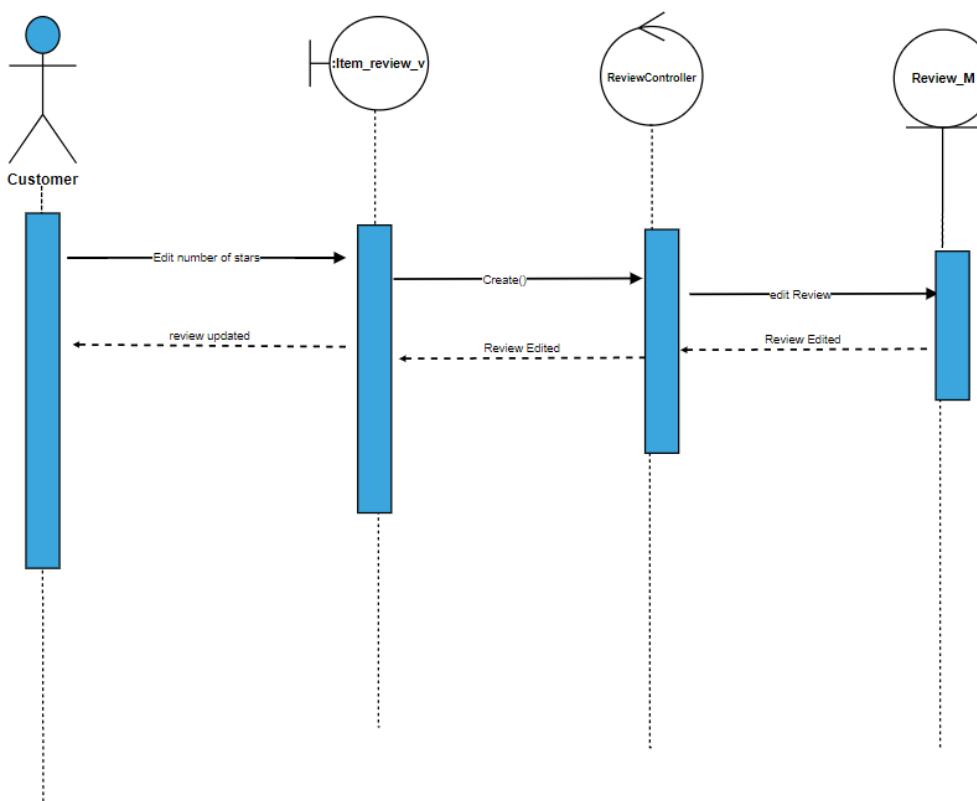
6) Search by State



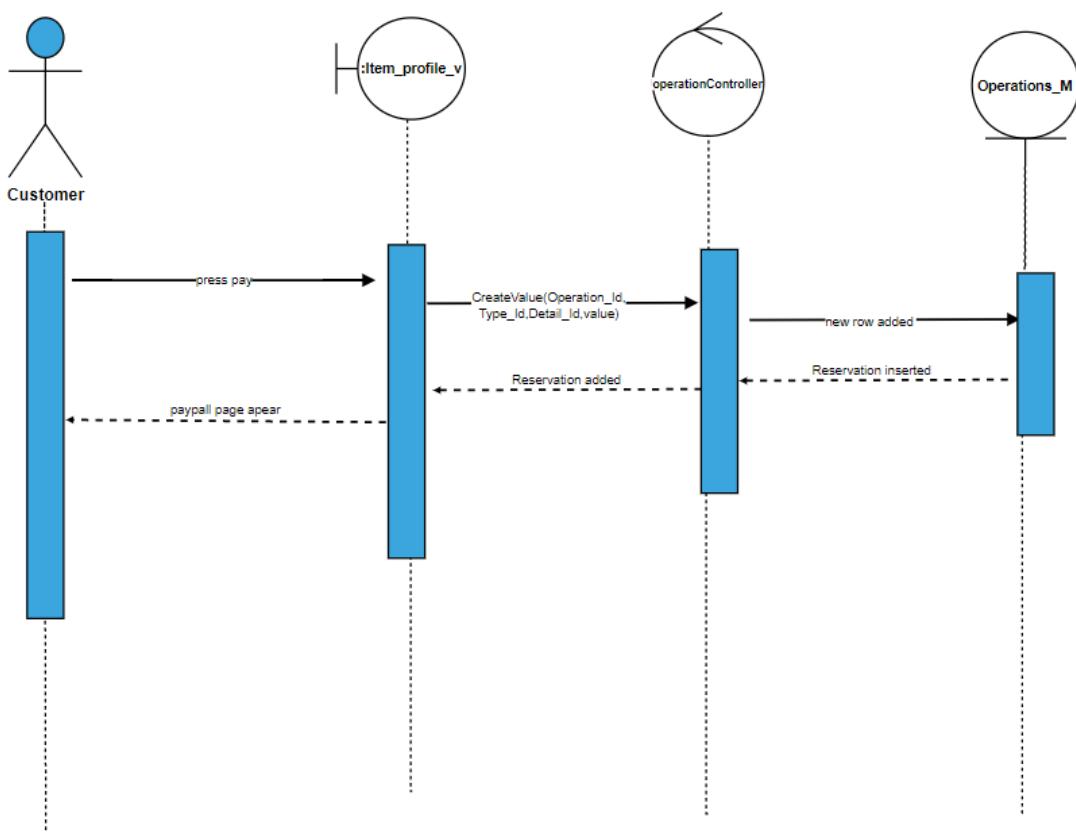
7) Review



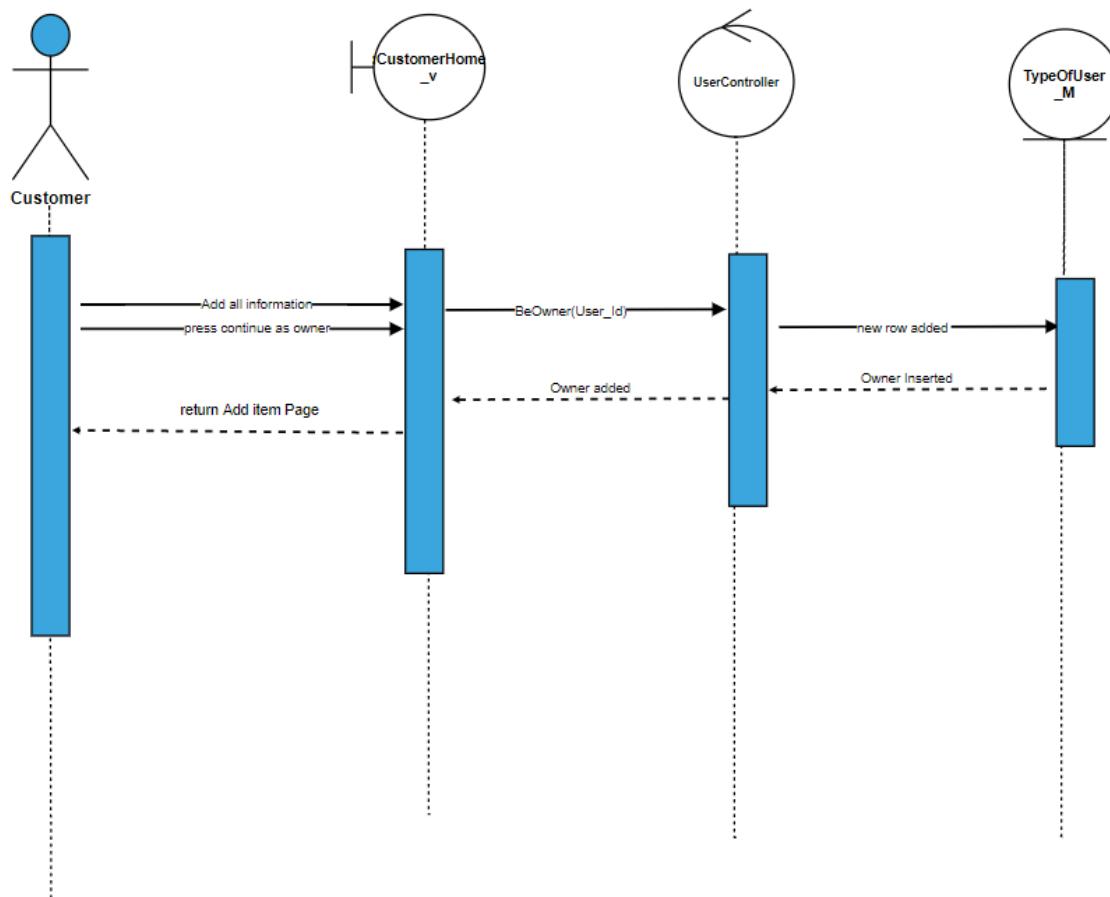
8)Edit Review



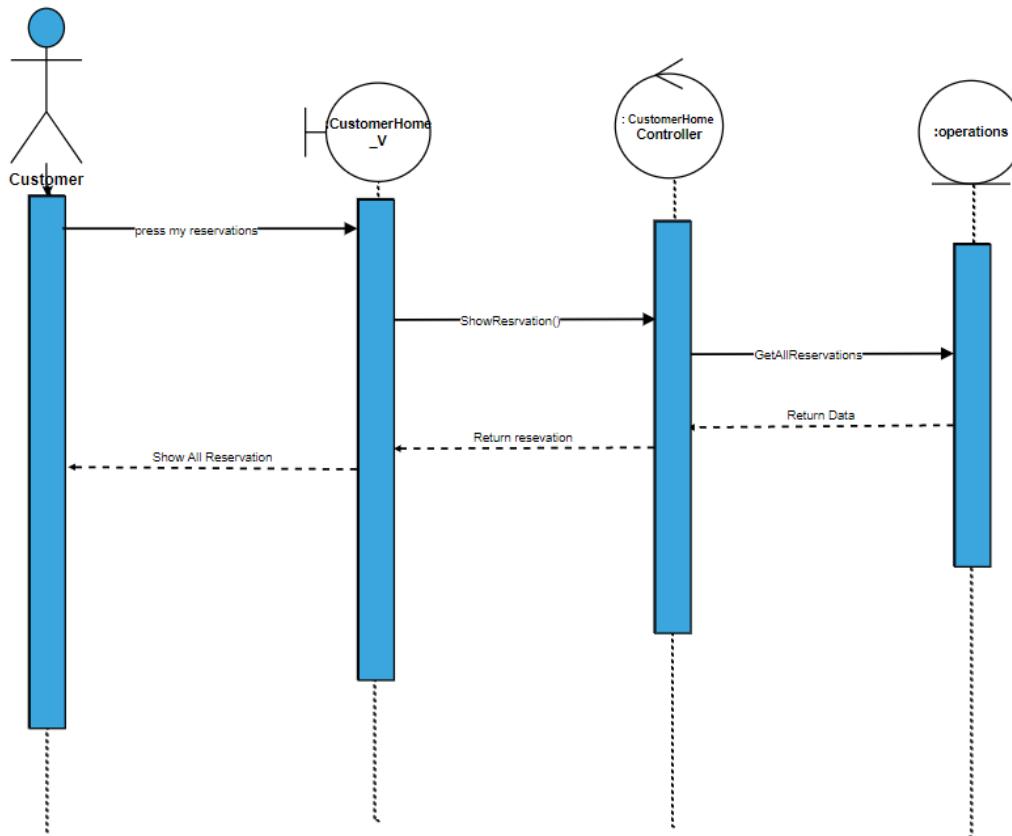
9)Pay



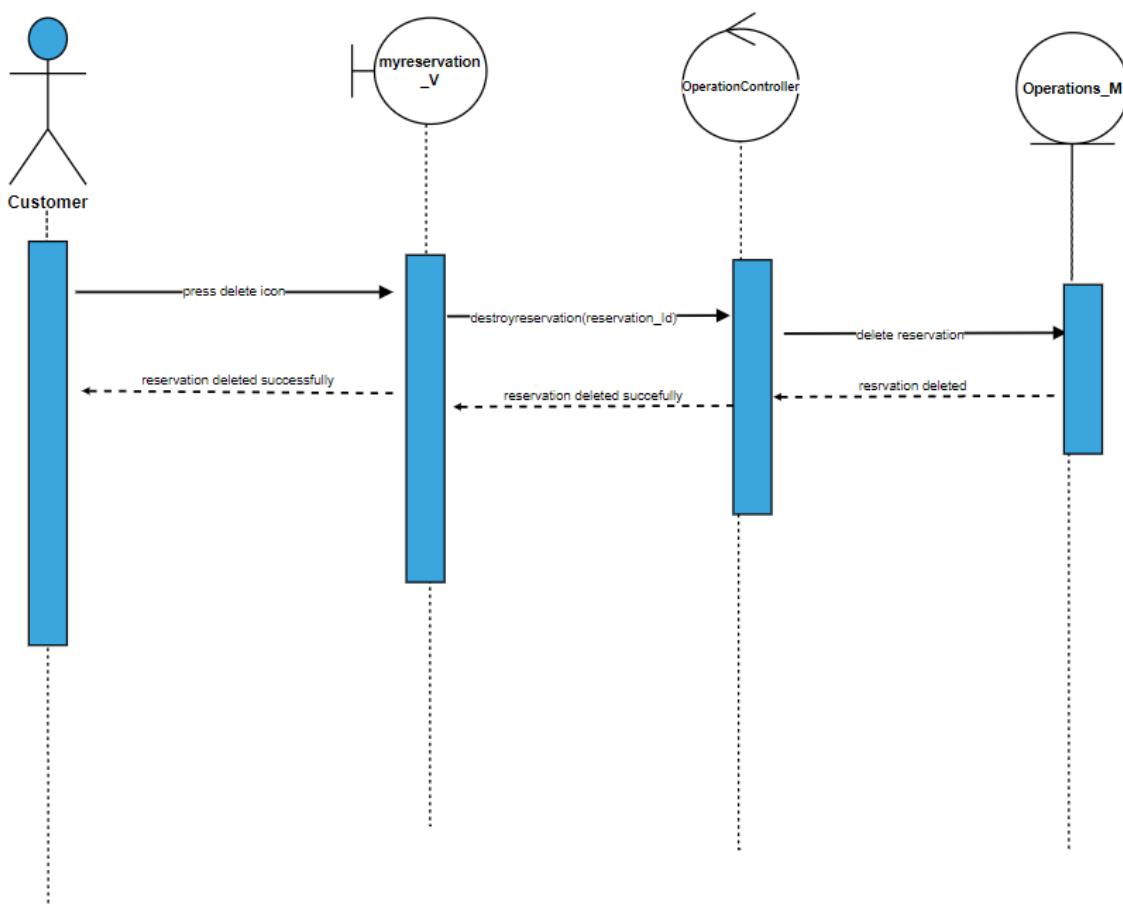
10) Be Owner



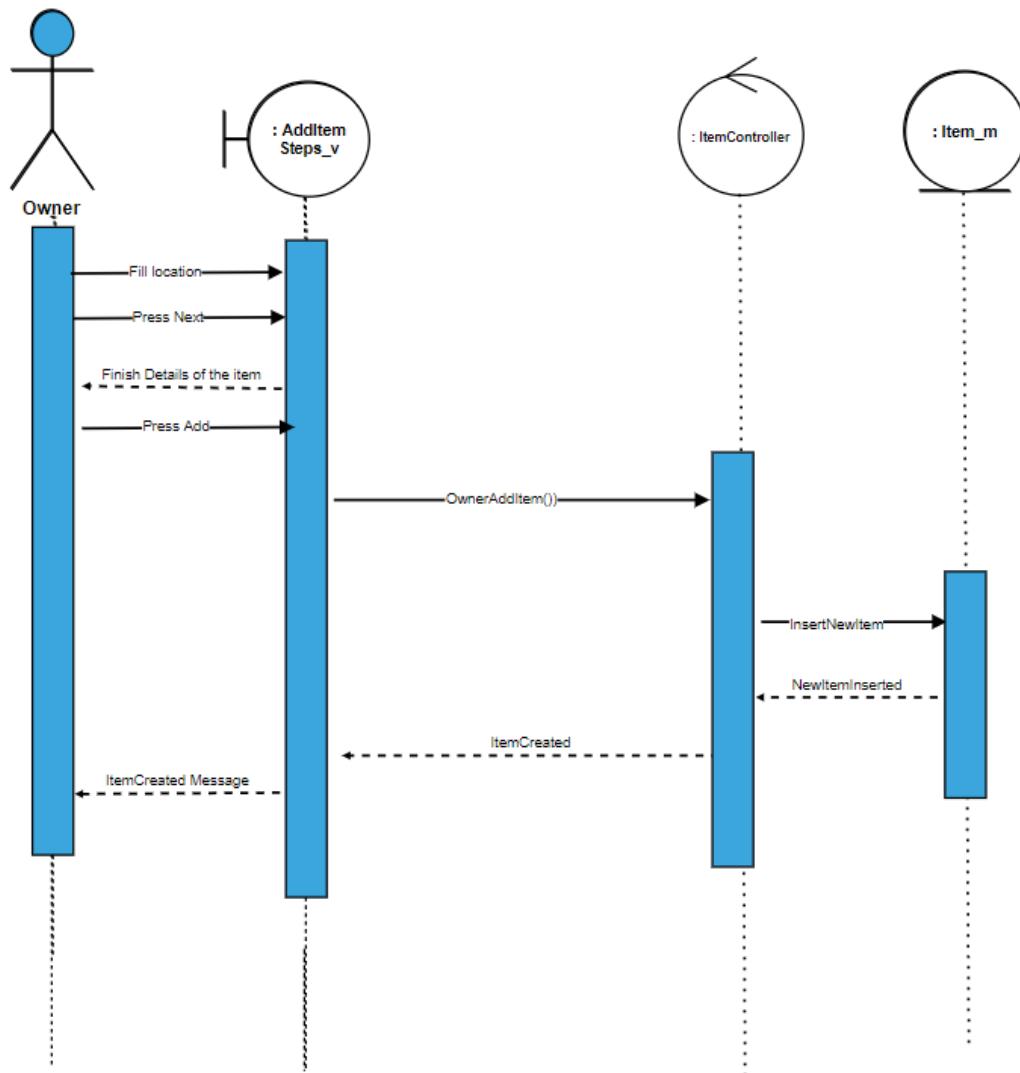
11) show my reservations



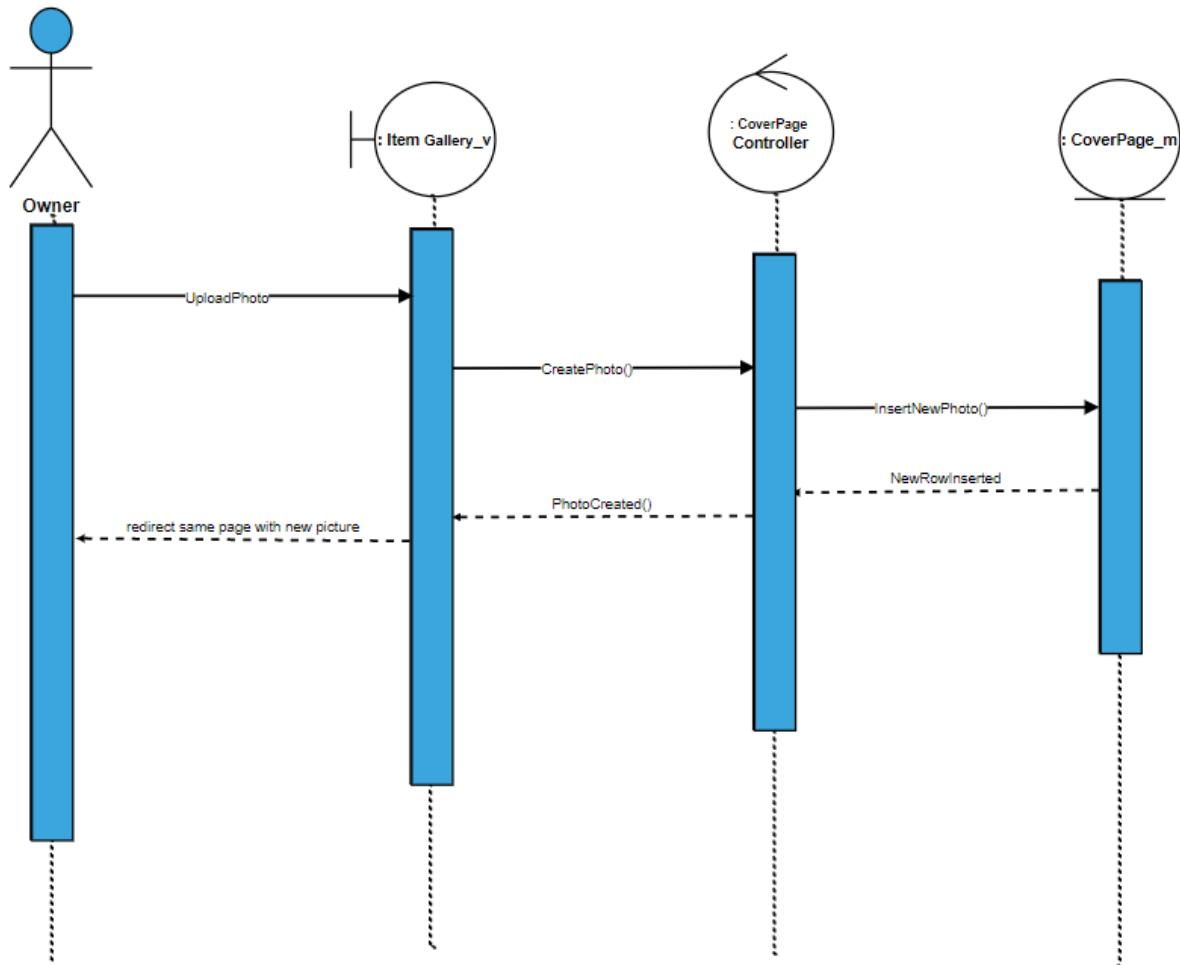
12)Delete reservation



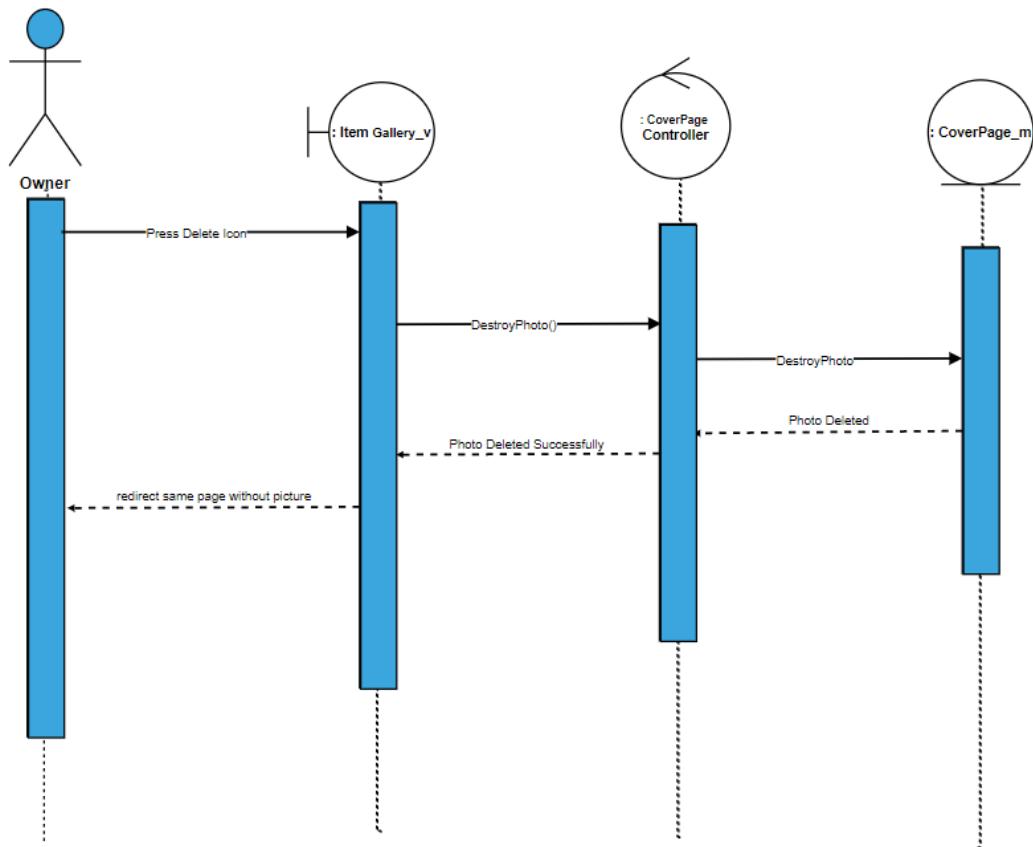
13)Add item



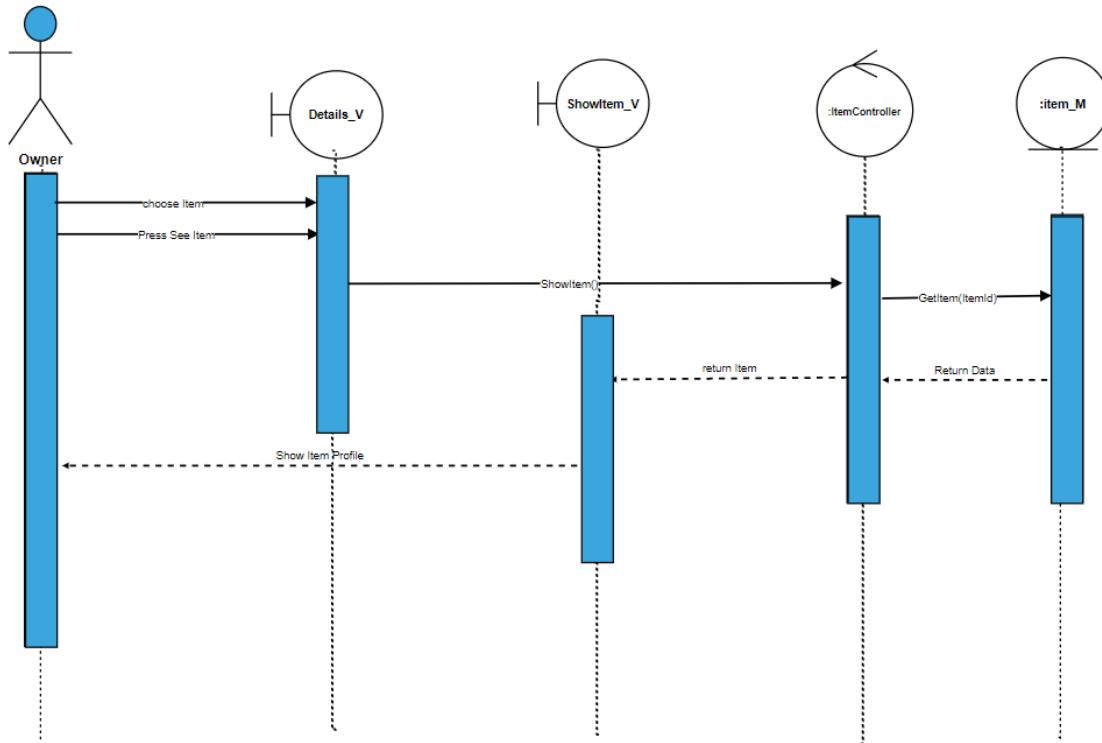
14)add Cover page picture



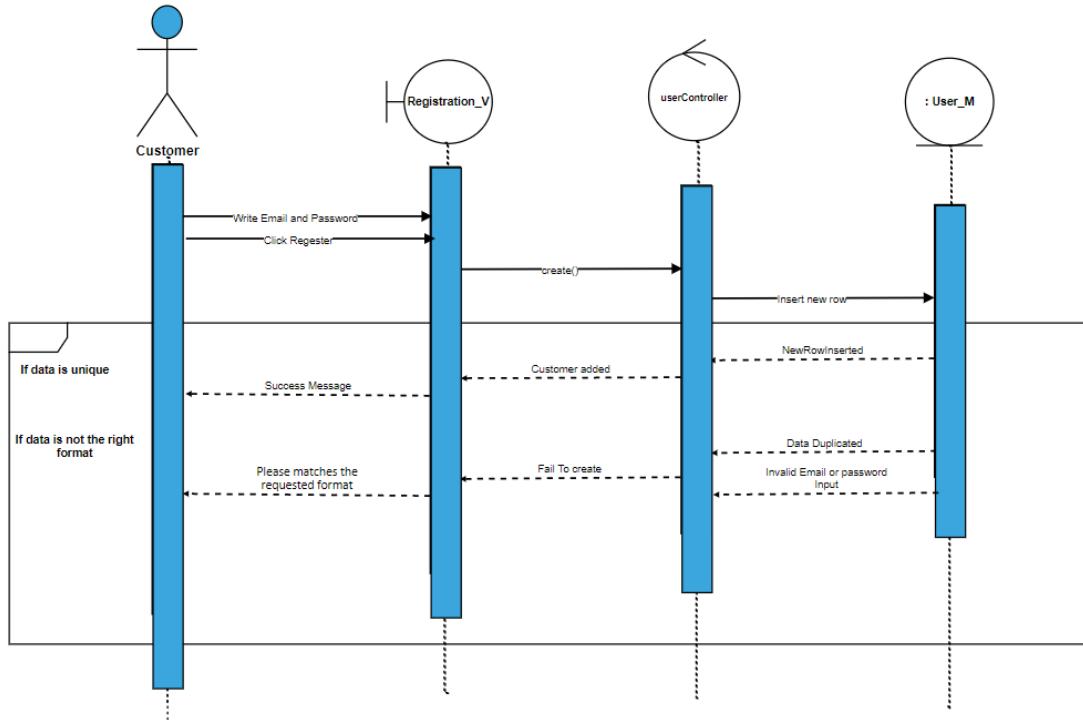
15)Delete cover page



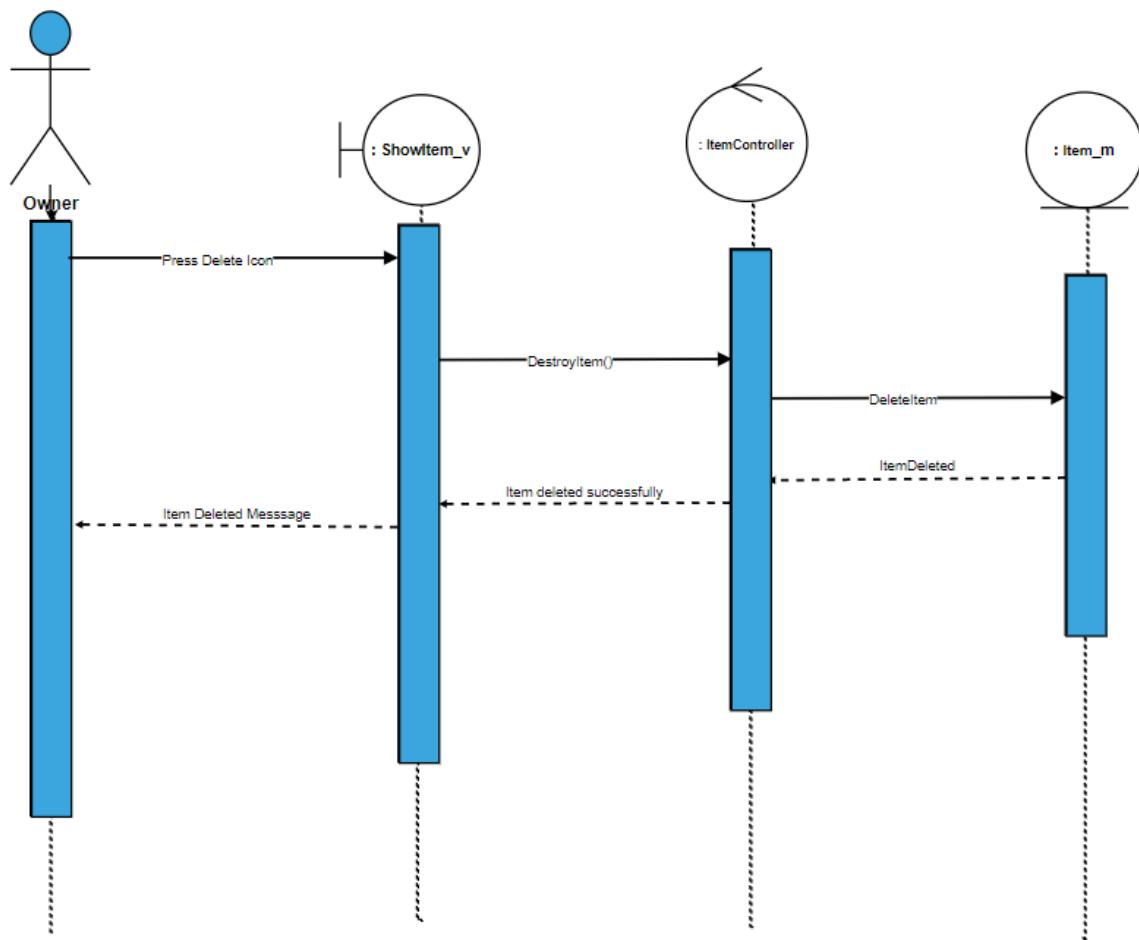
16) Show item



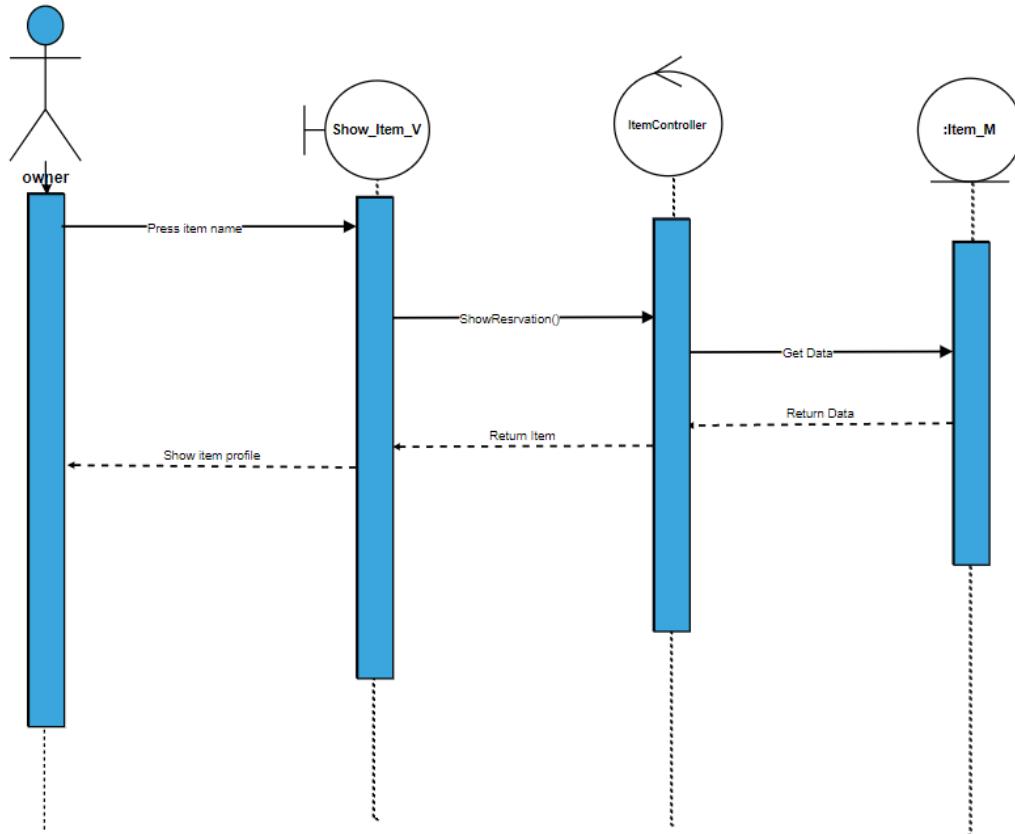
17)registration



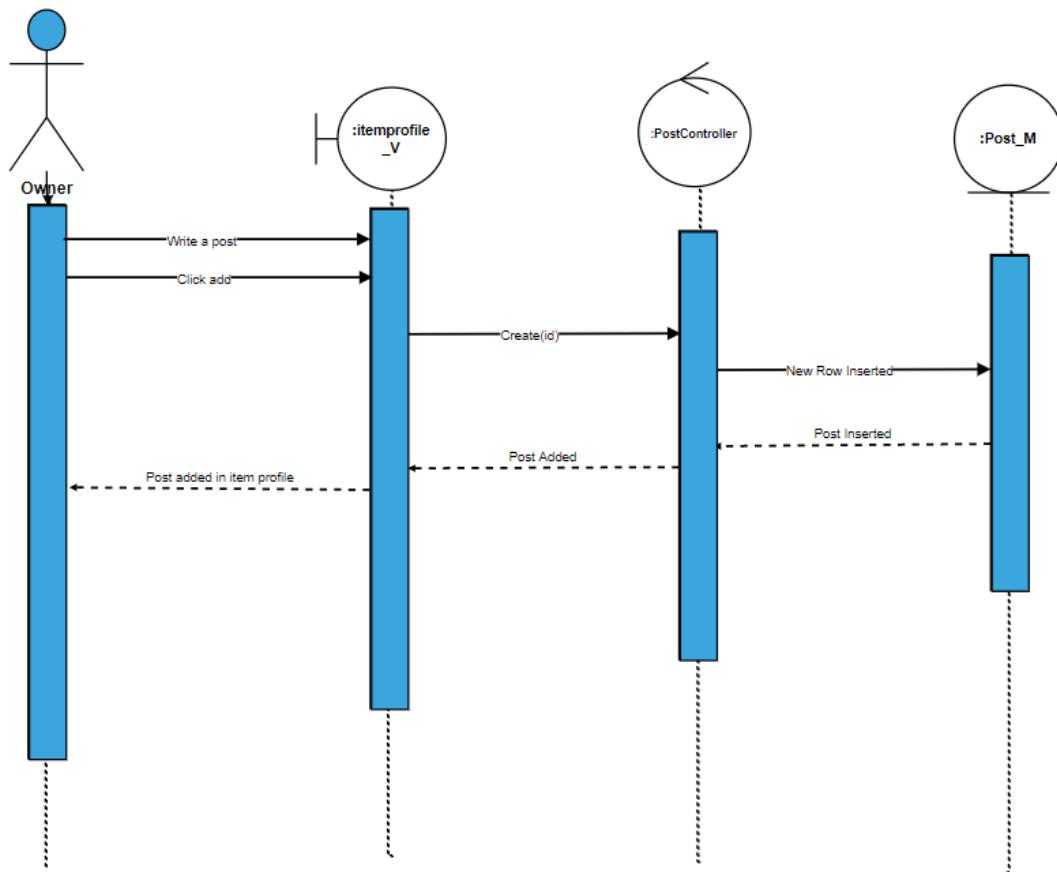
18)Delete item



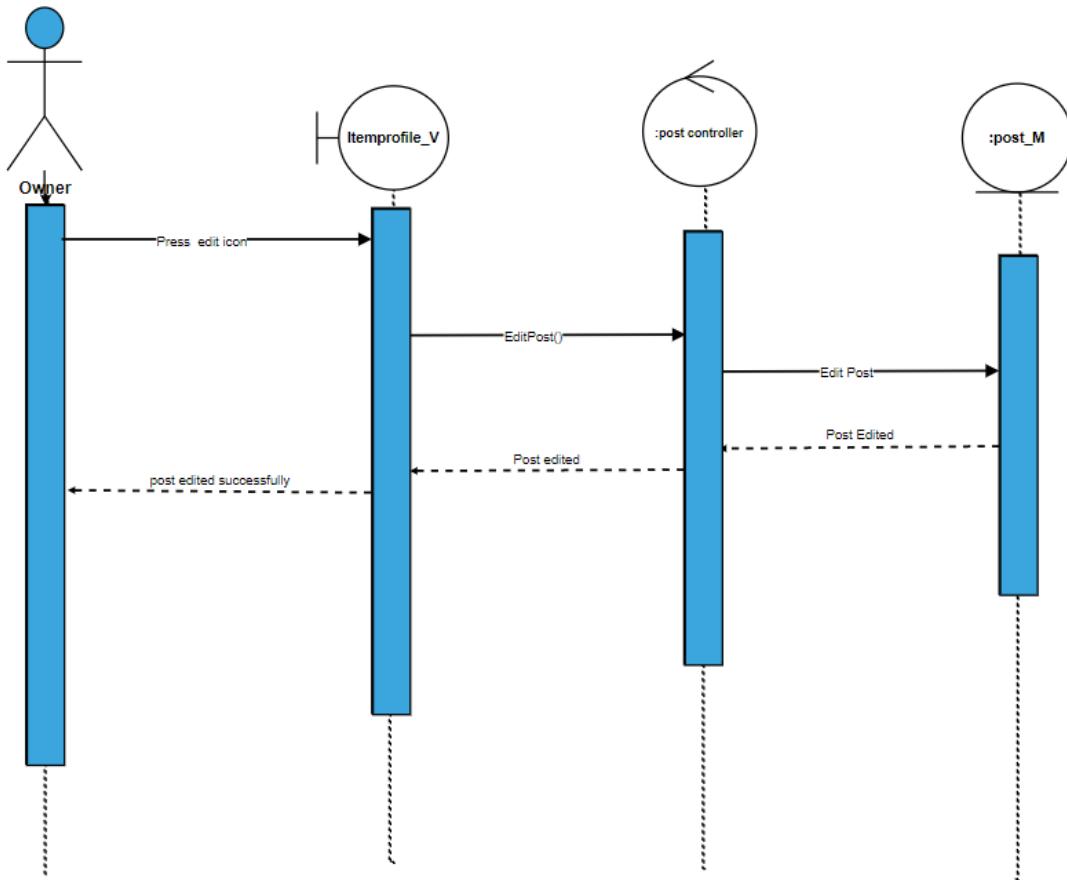
19) View item reservation



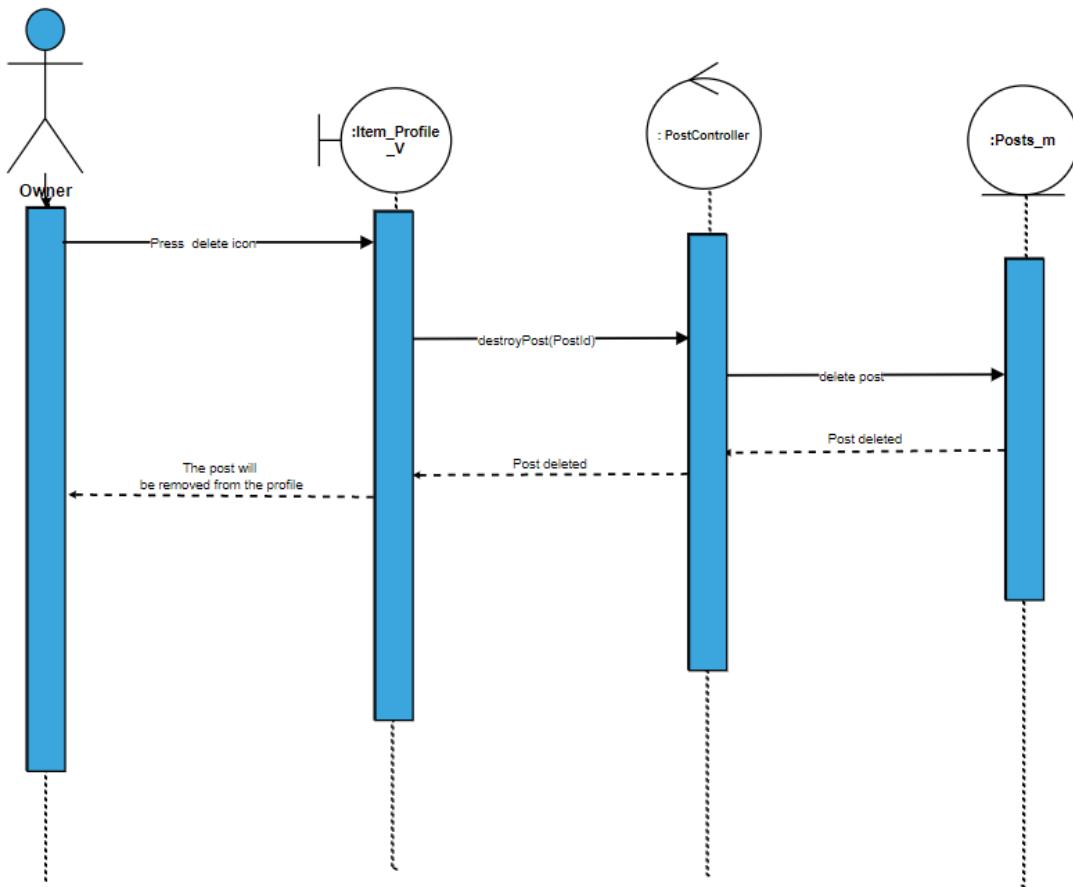
20) post a post



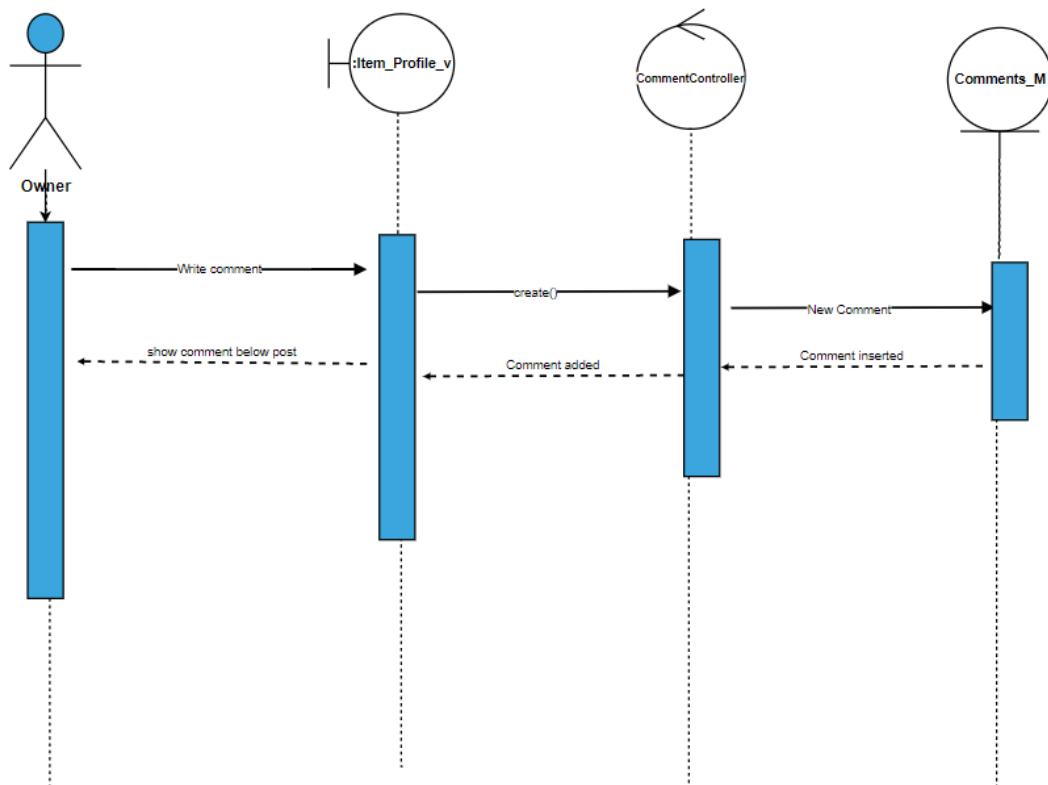
21) Post a post



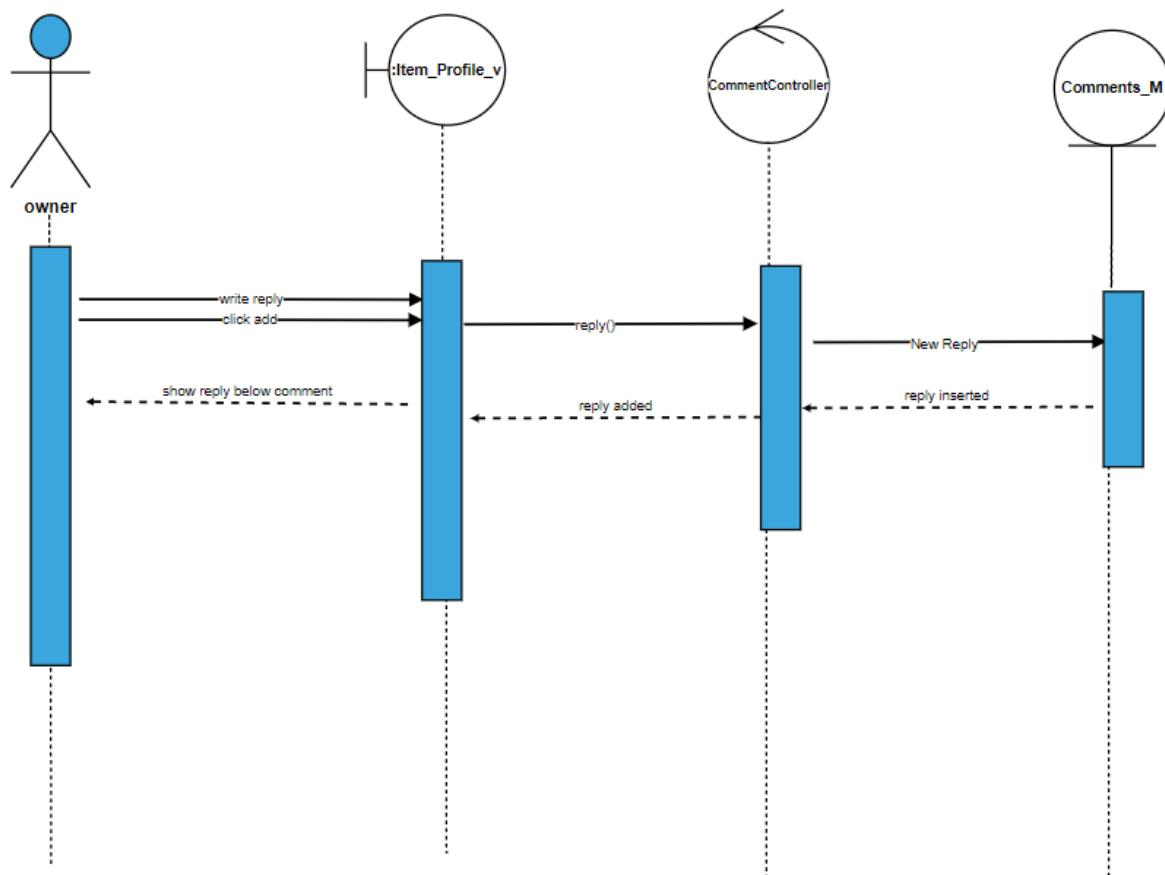
22)Delete post



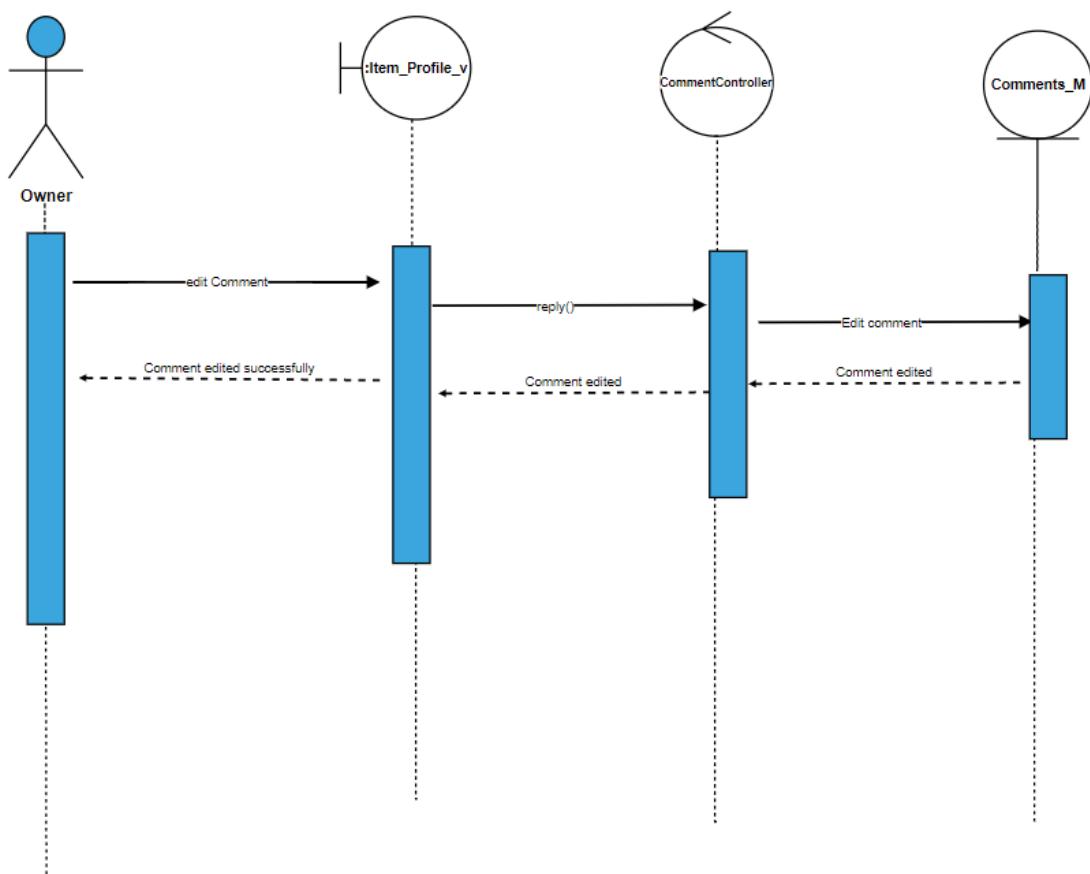
23) Write a comment



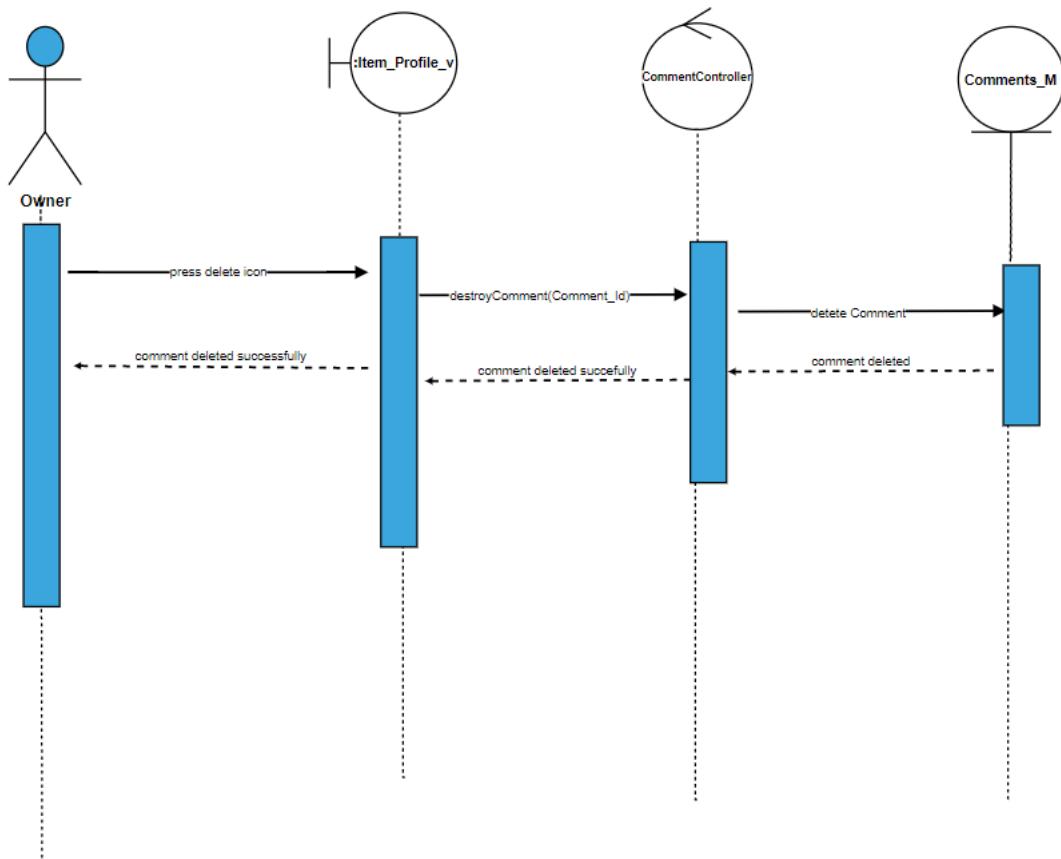
24)Reply comment



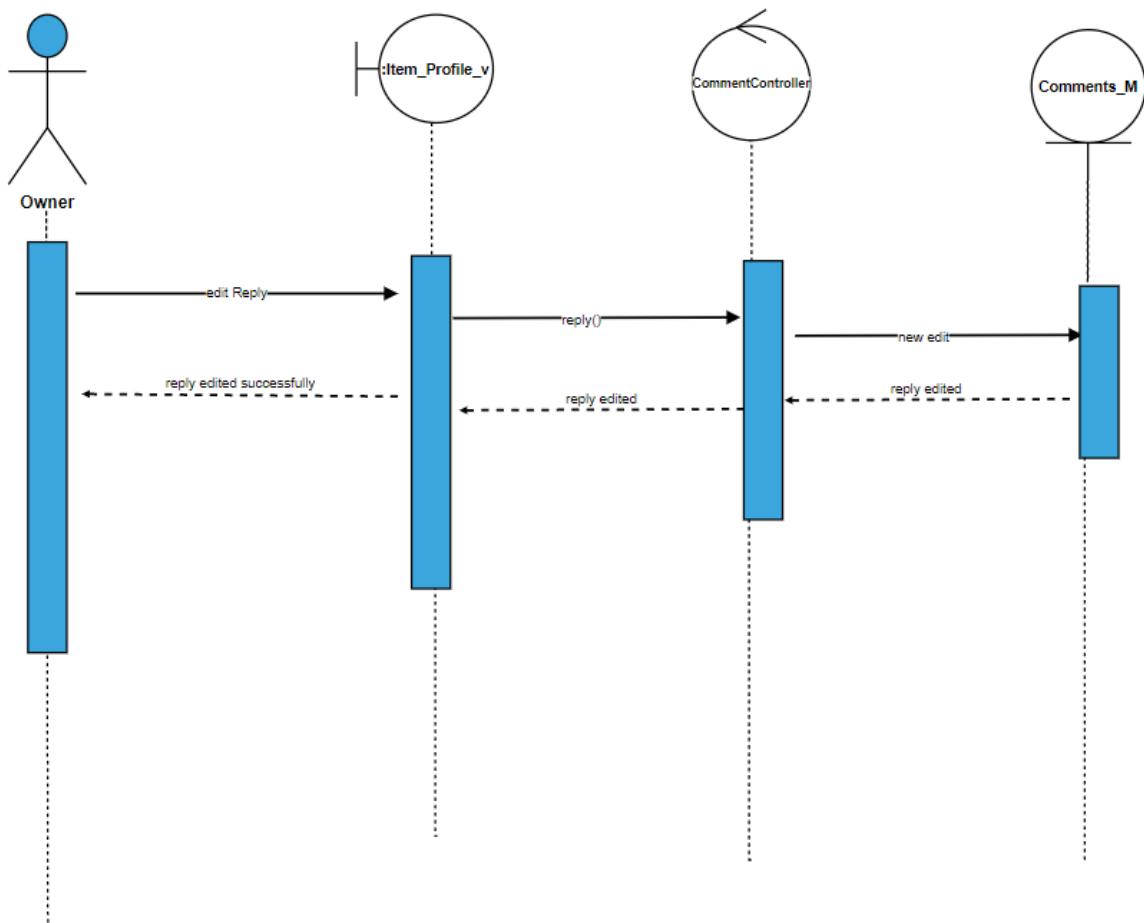
25)Edit comment



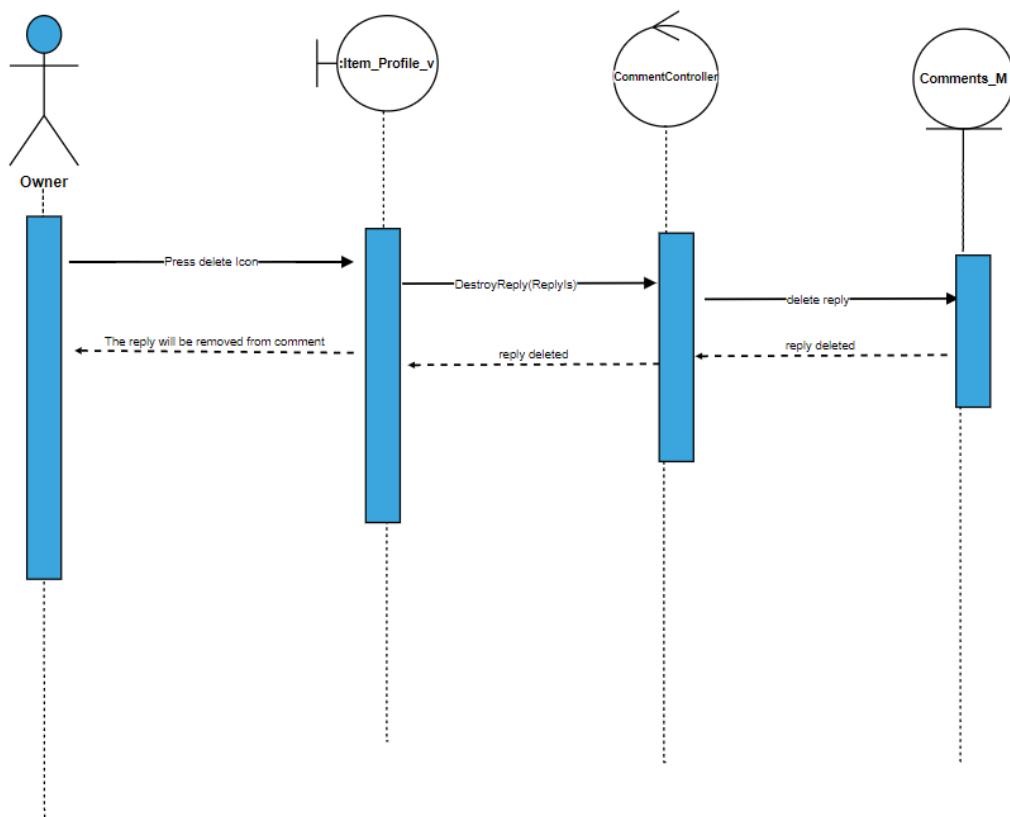
26)Delete comment



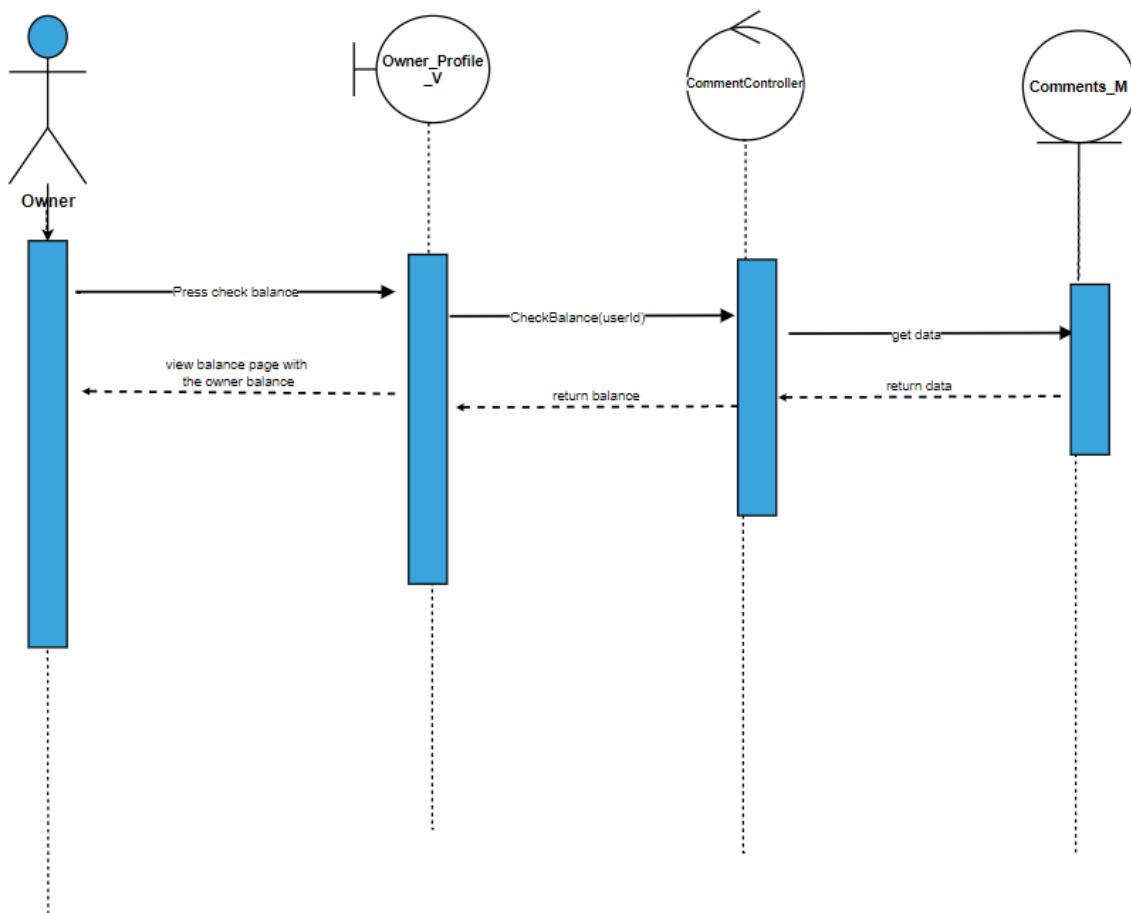
27)Edit reply



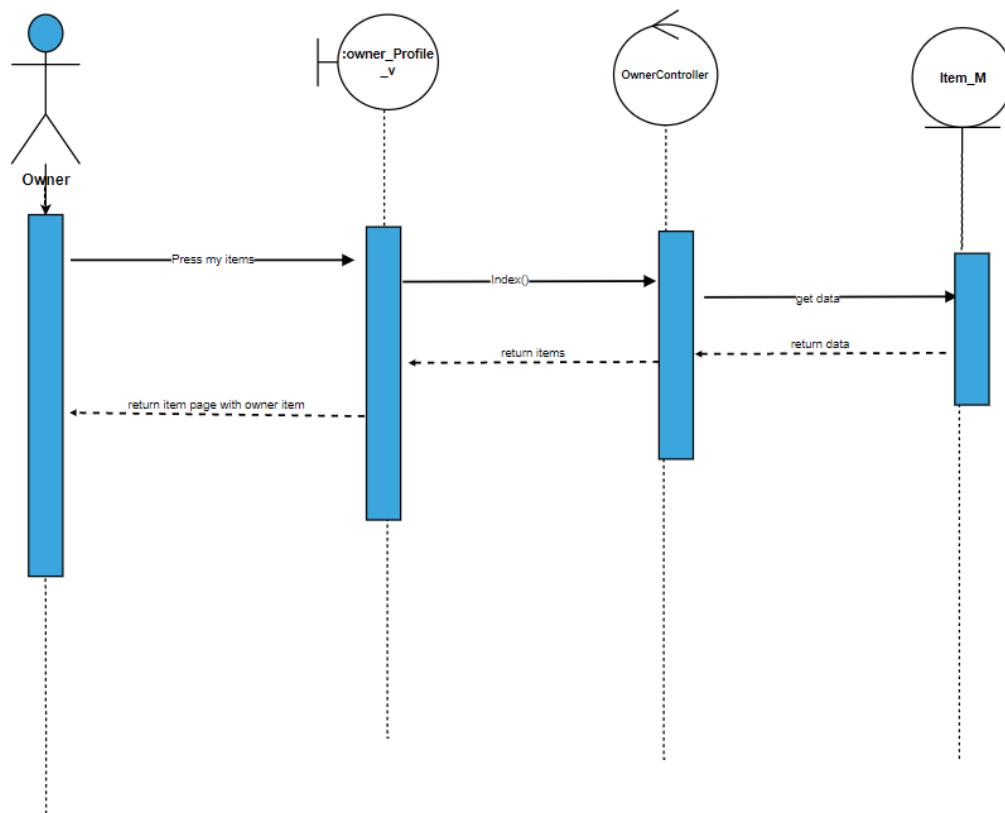
28)Delete reply



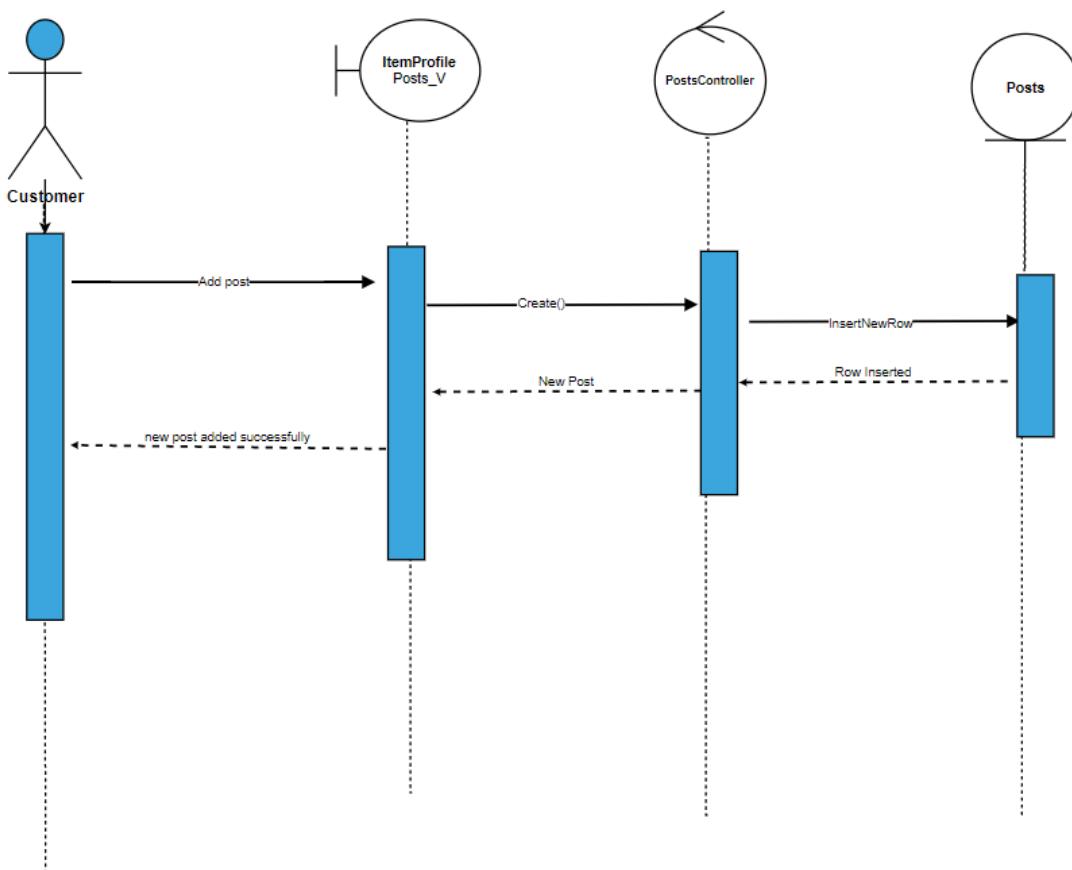
29)Check balance



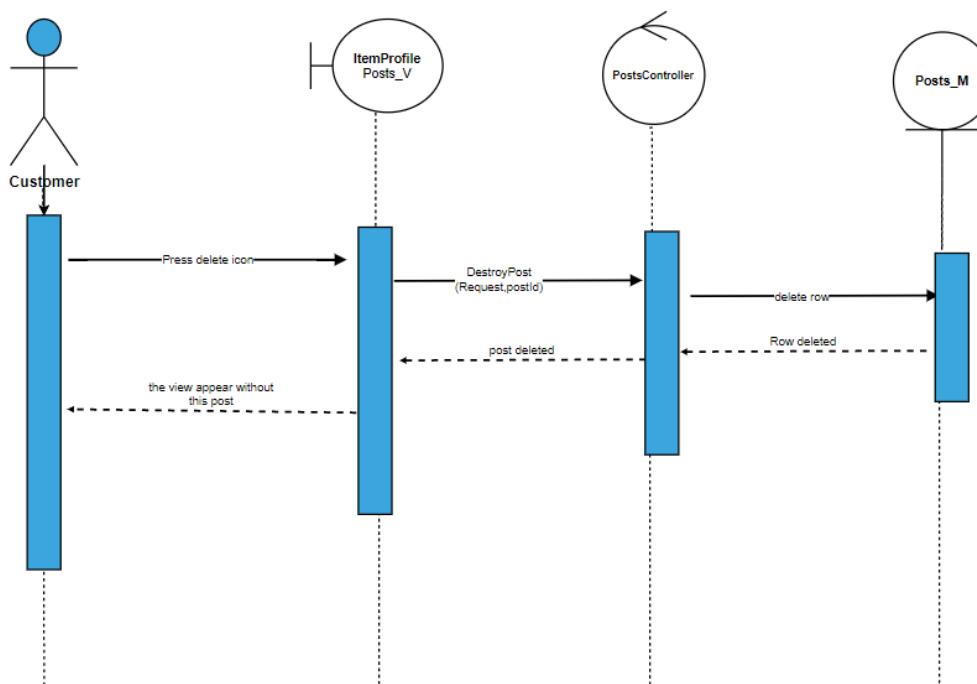
30) Show my items



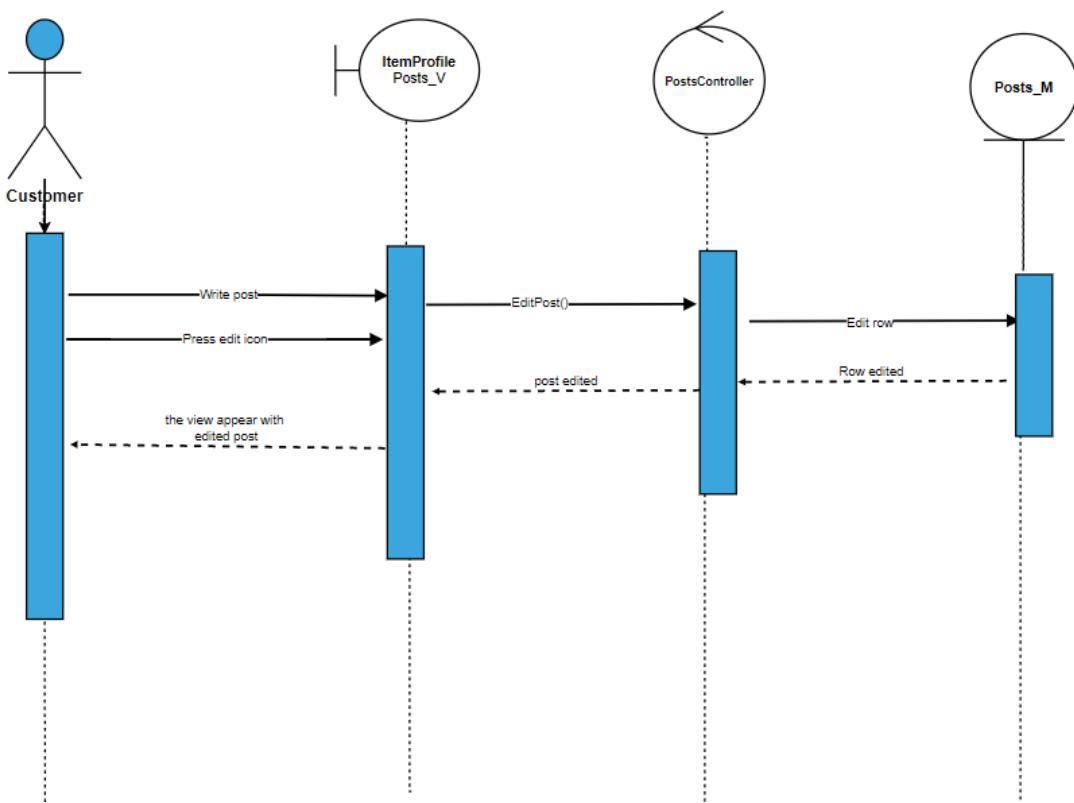
31)Add post



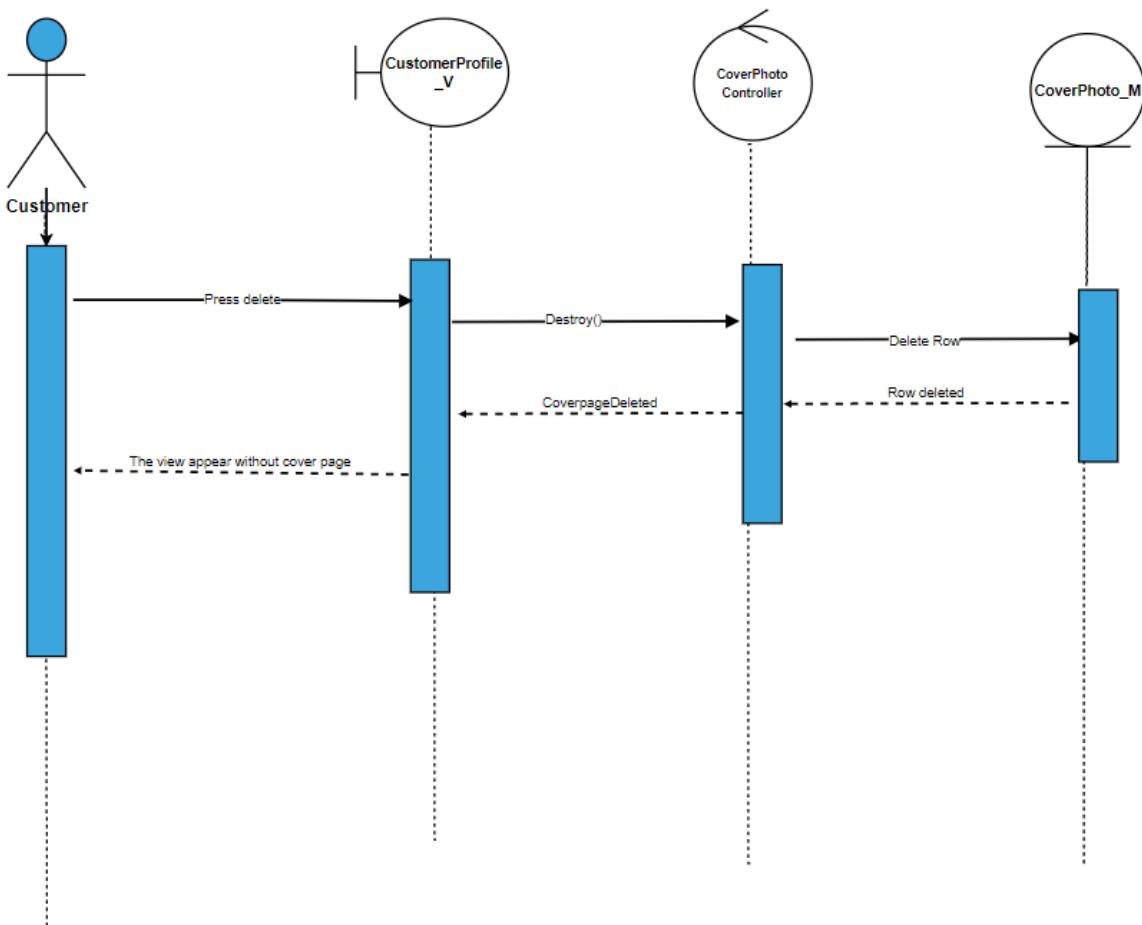
32)Delete post



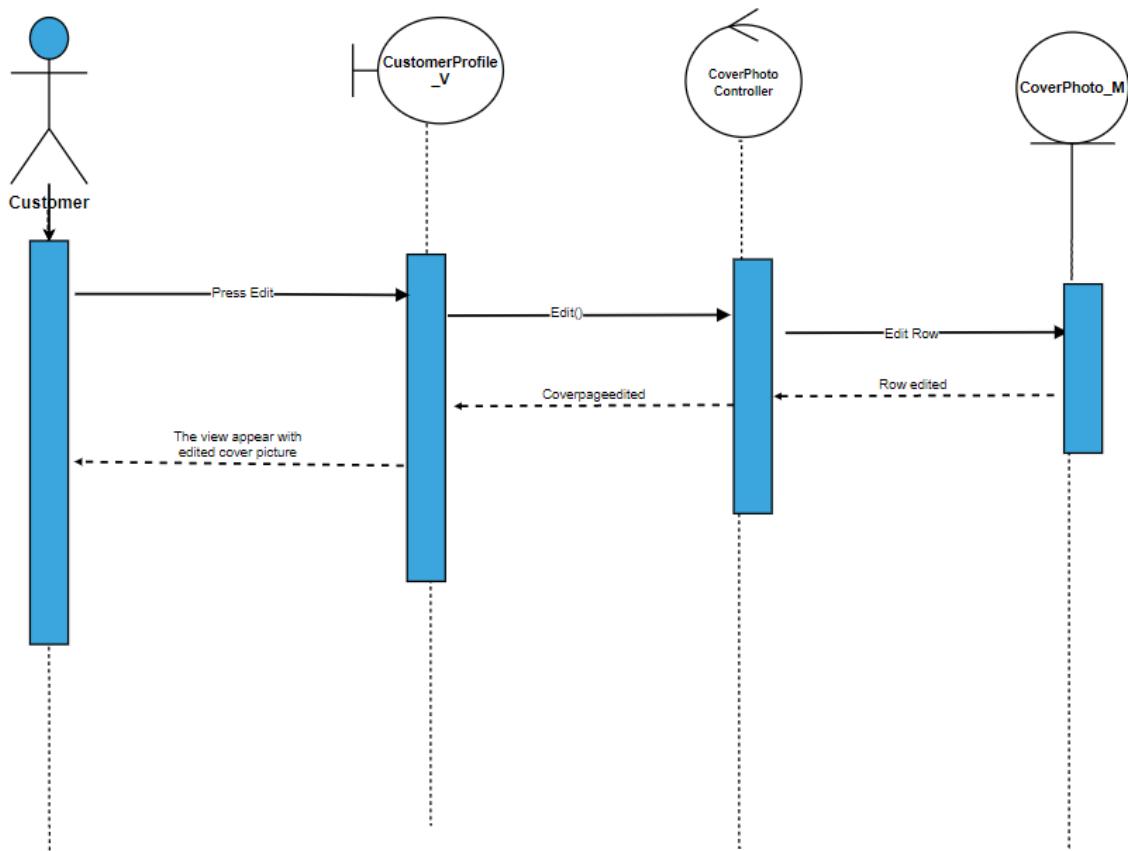
33)Edit post



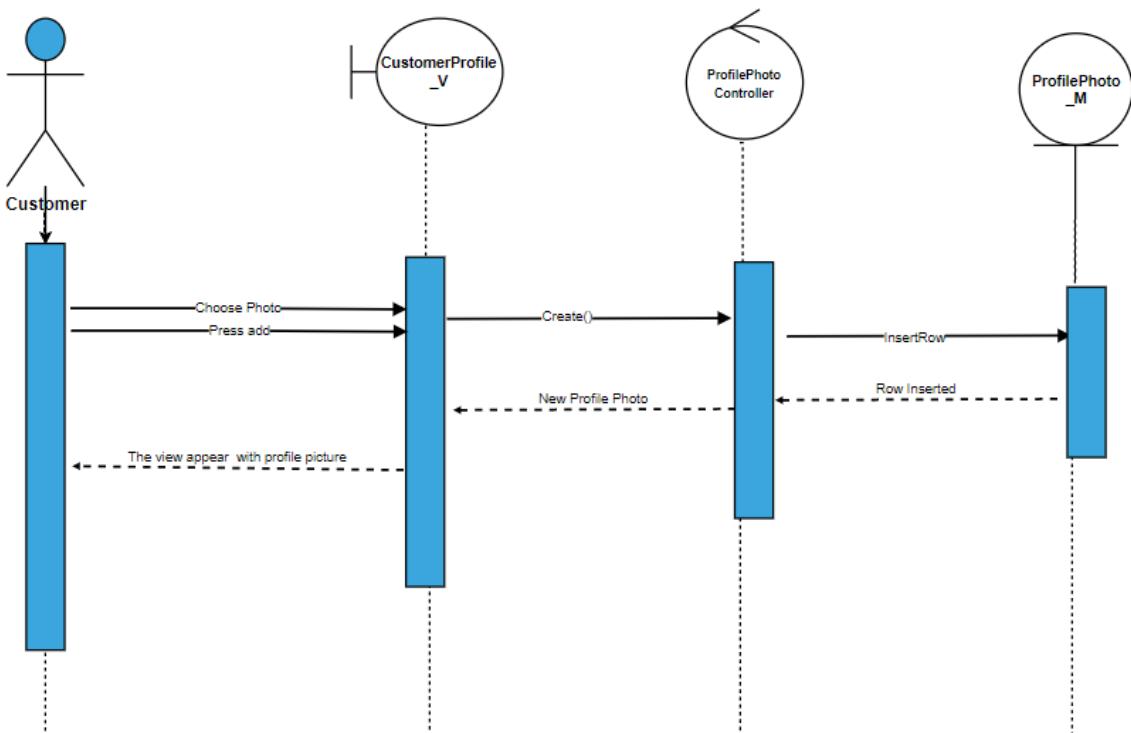
34)Delete Cover page



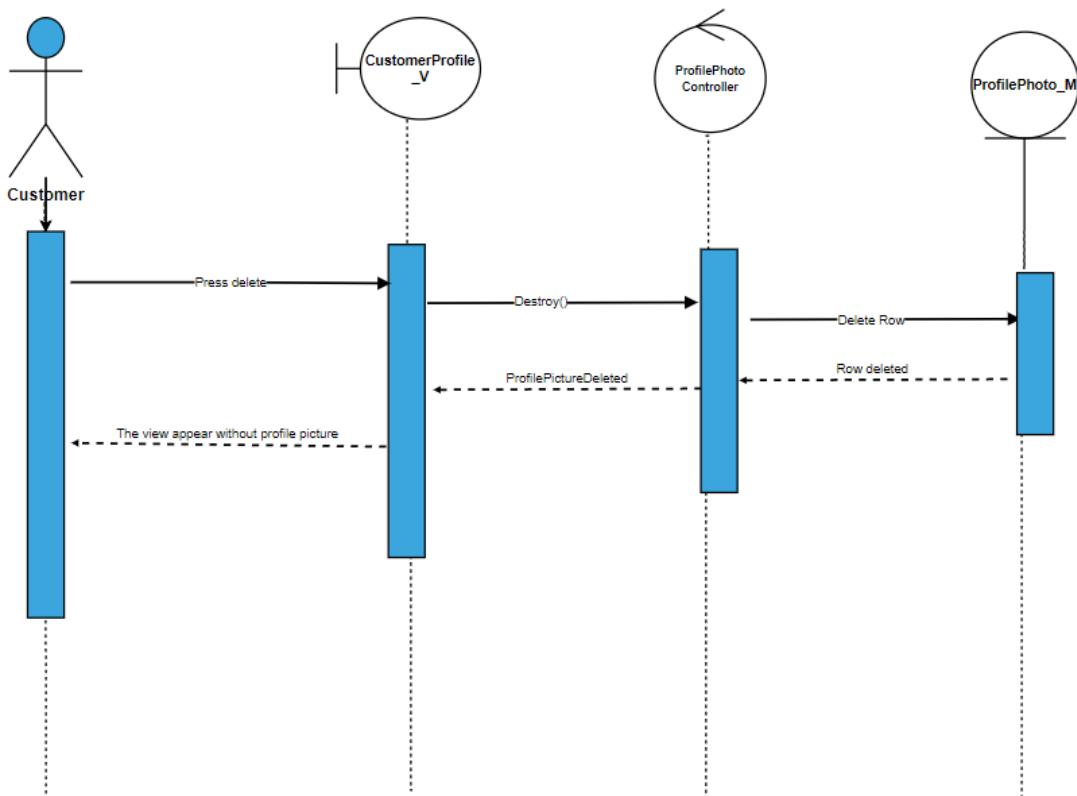
35)Edit cover page



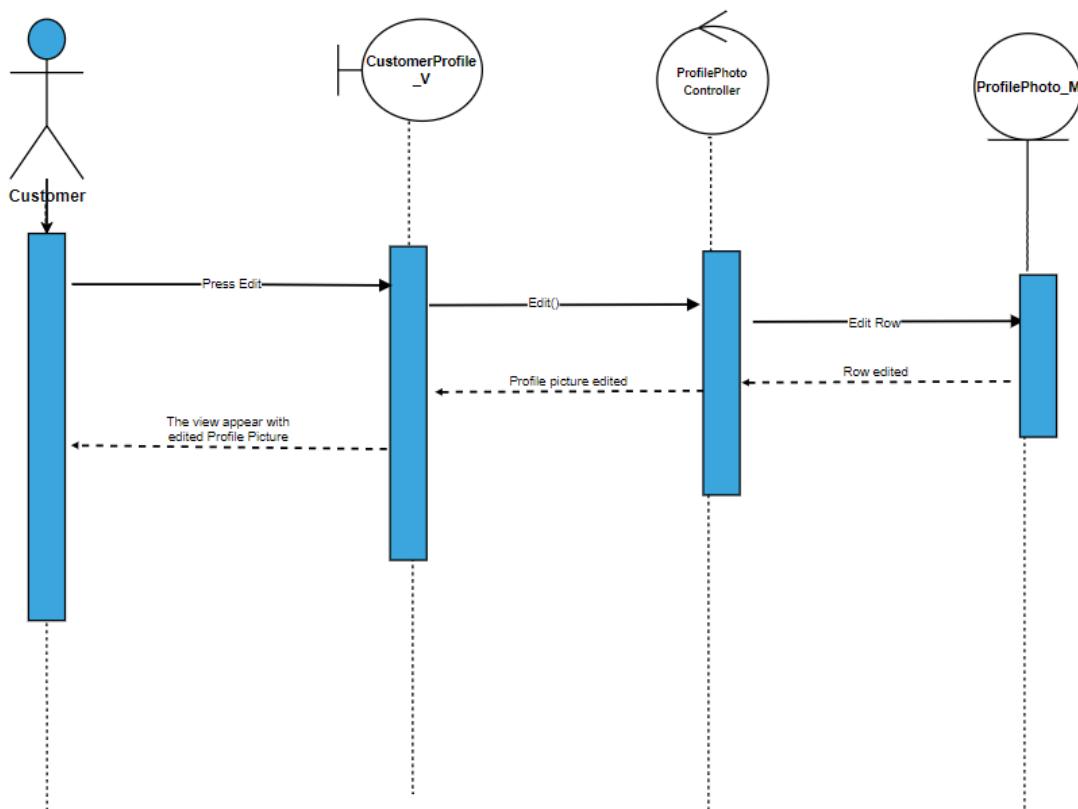
36)Add profile picture



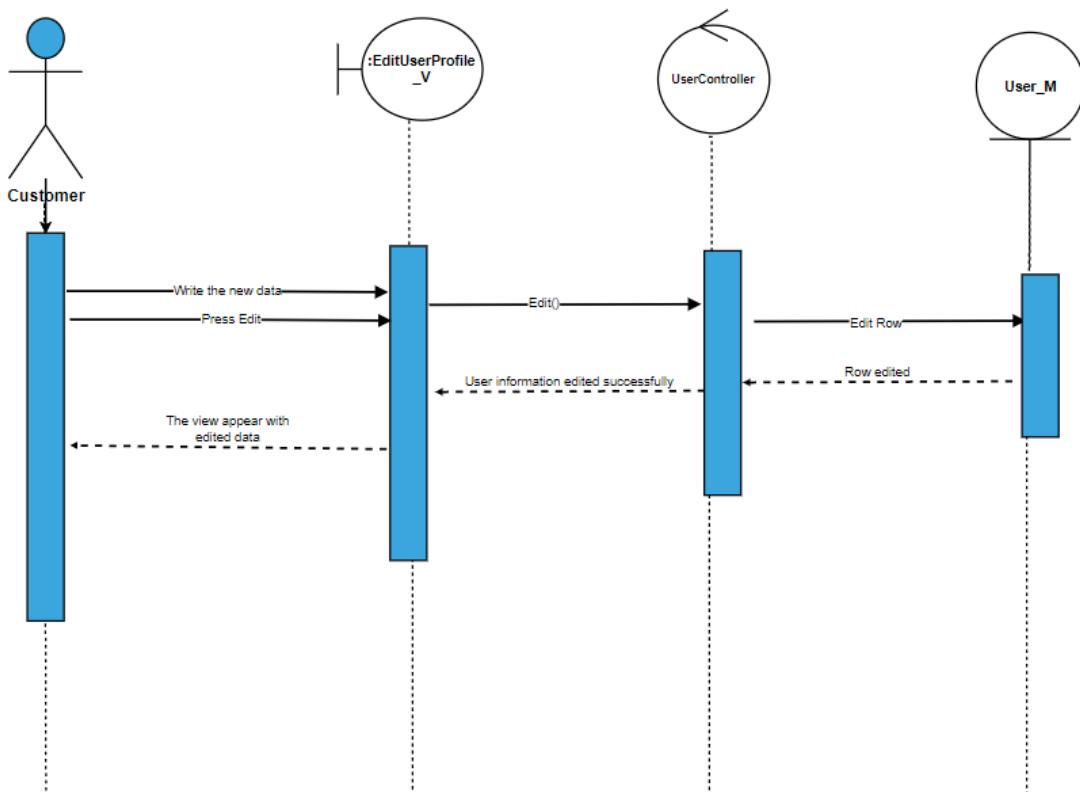
37) delete profile picture



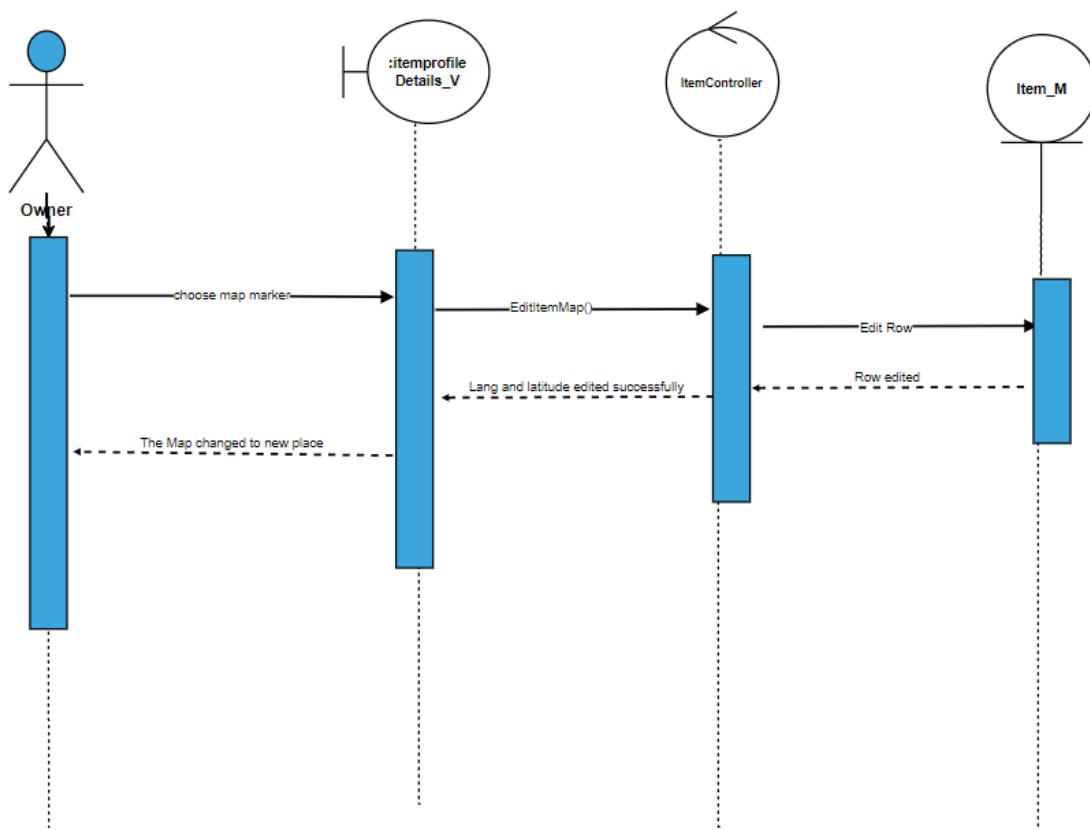
38)Edit profile picture



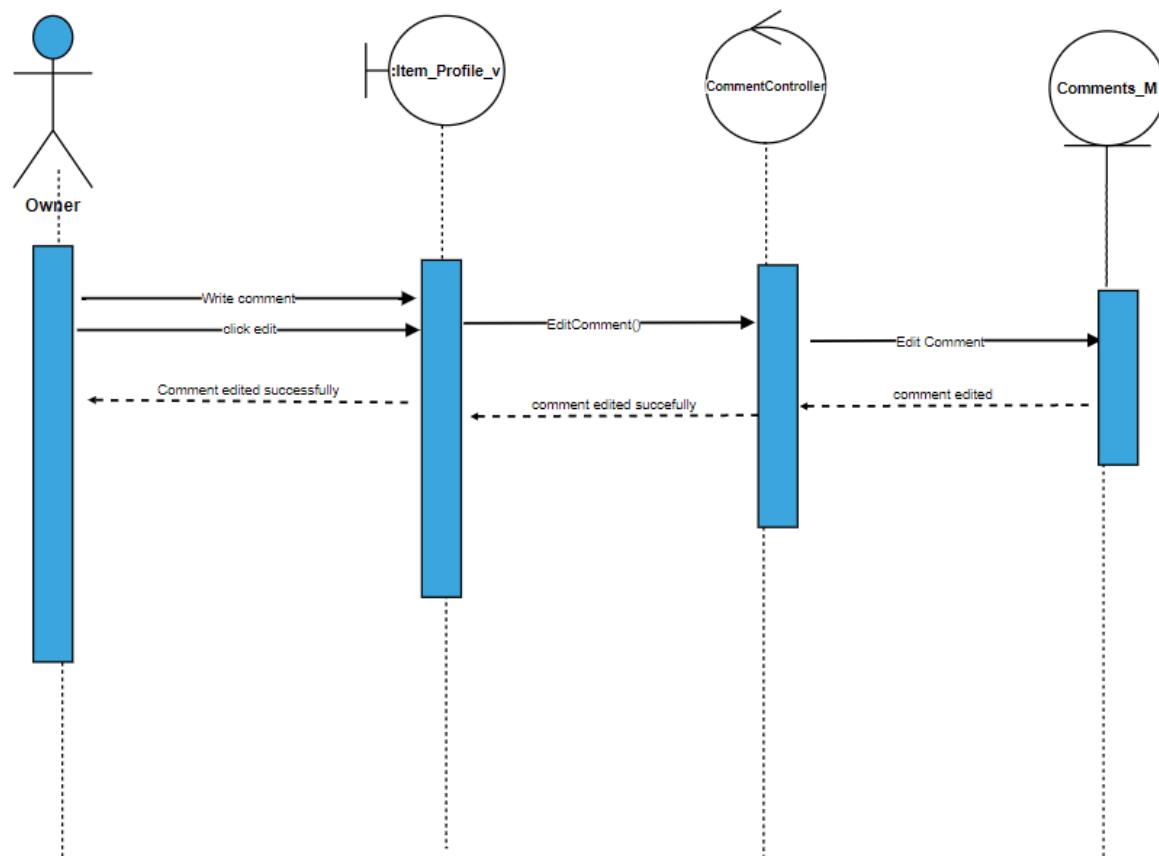
39)Edit user information



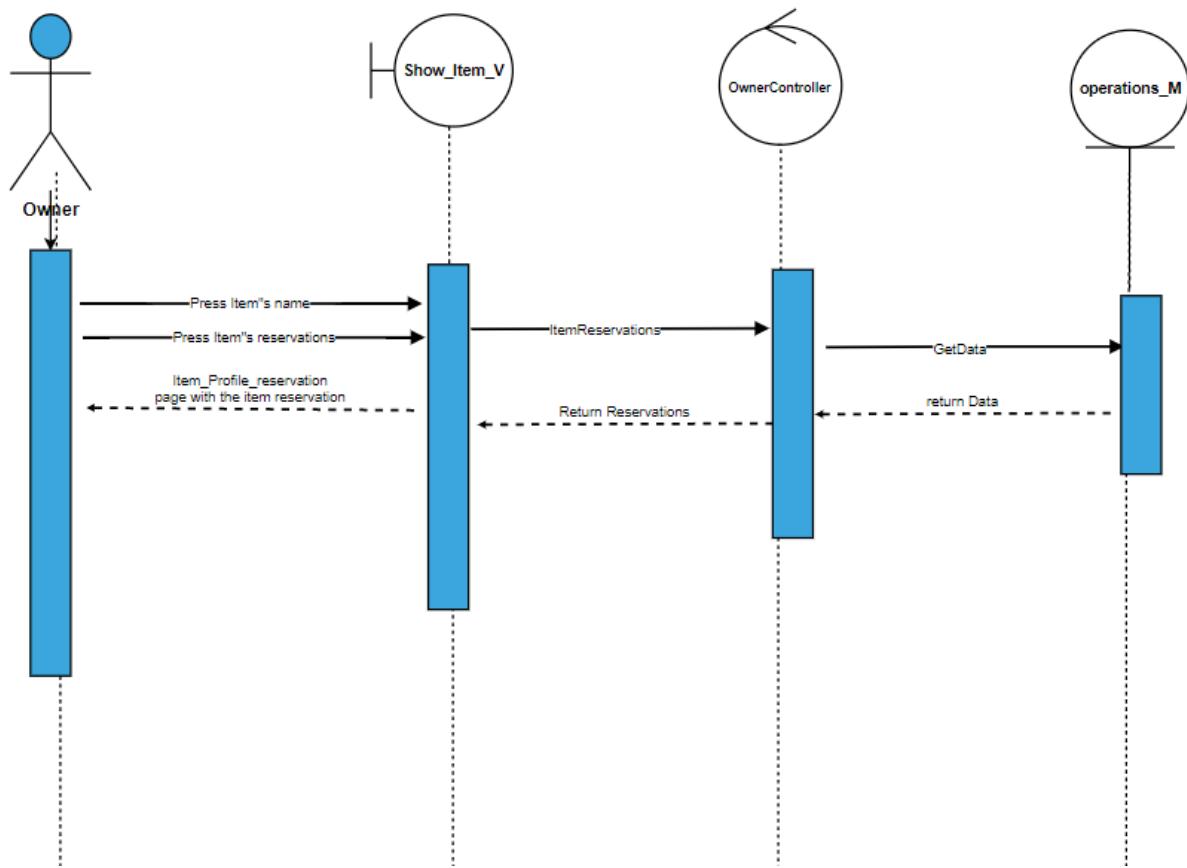
40)Edit Location



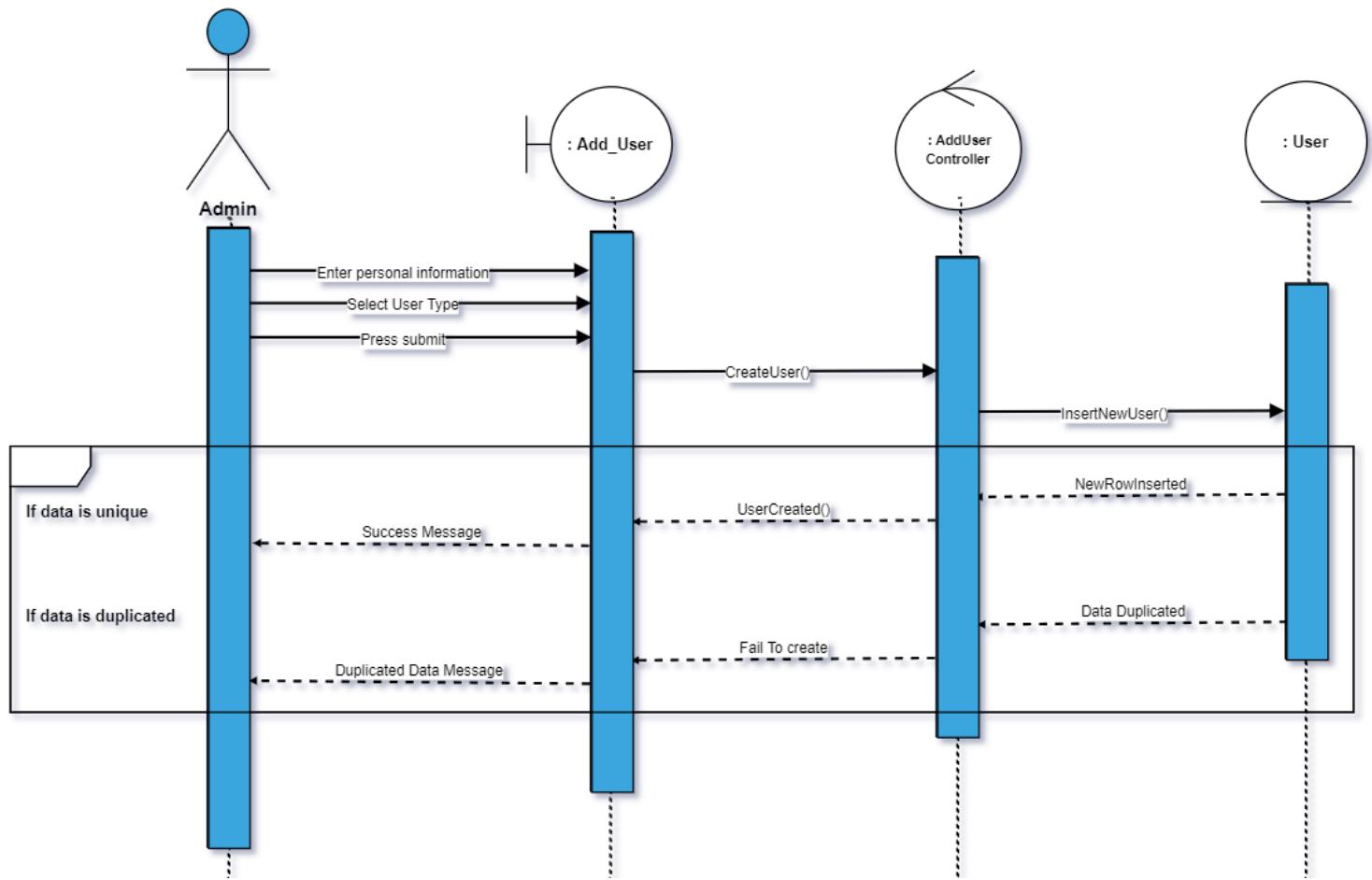
41)Edit Comment



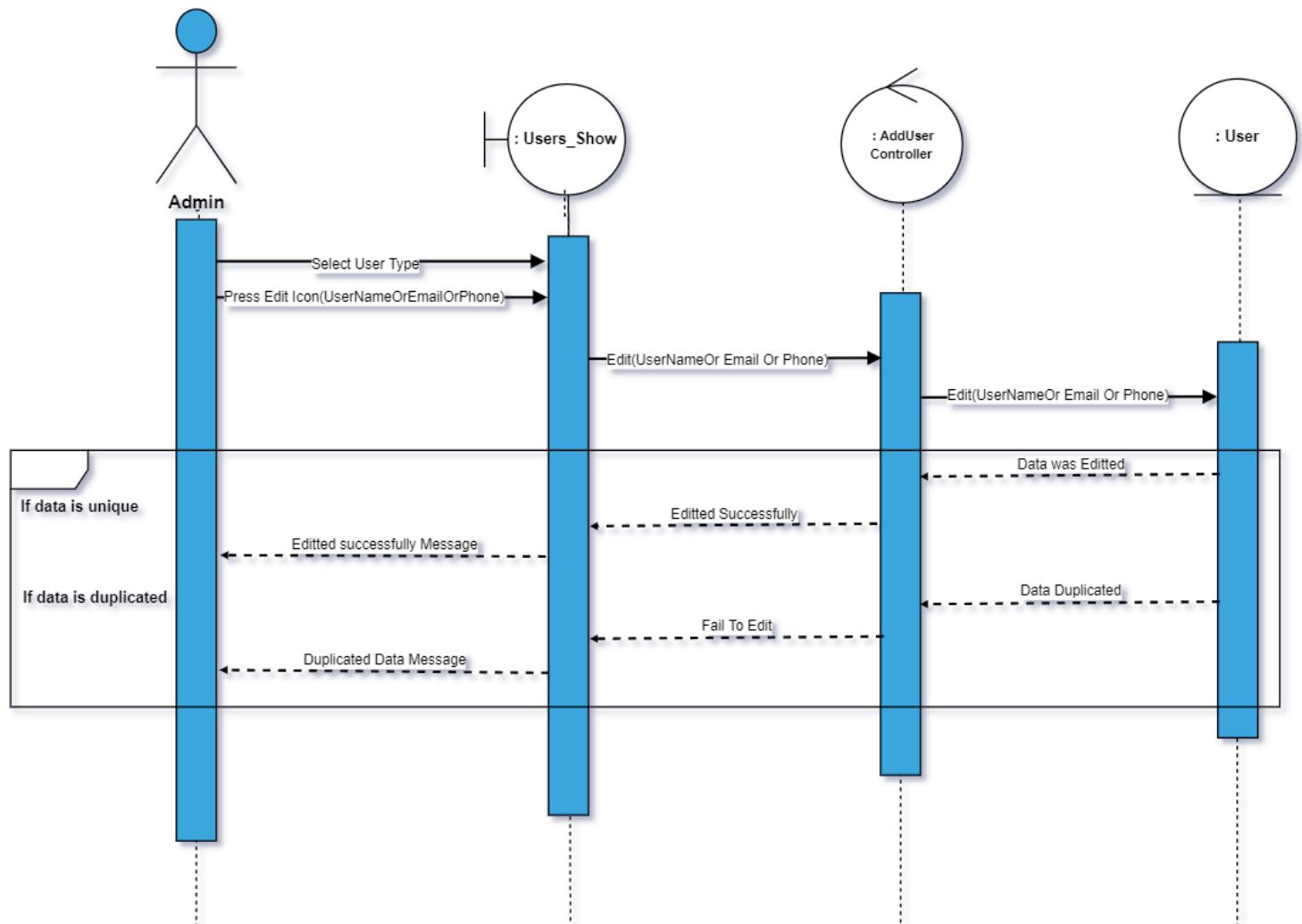
42) view item reservation



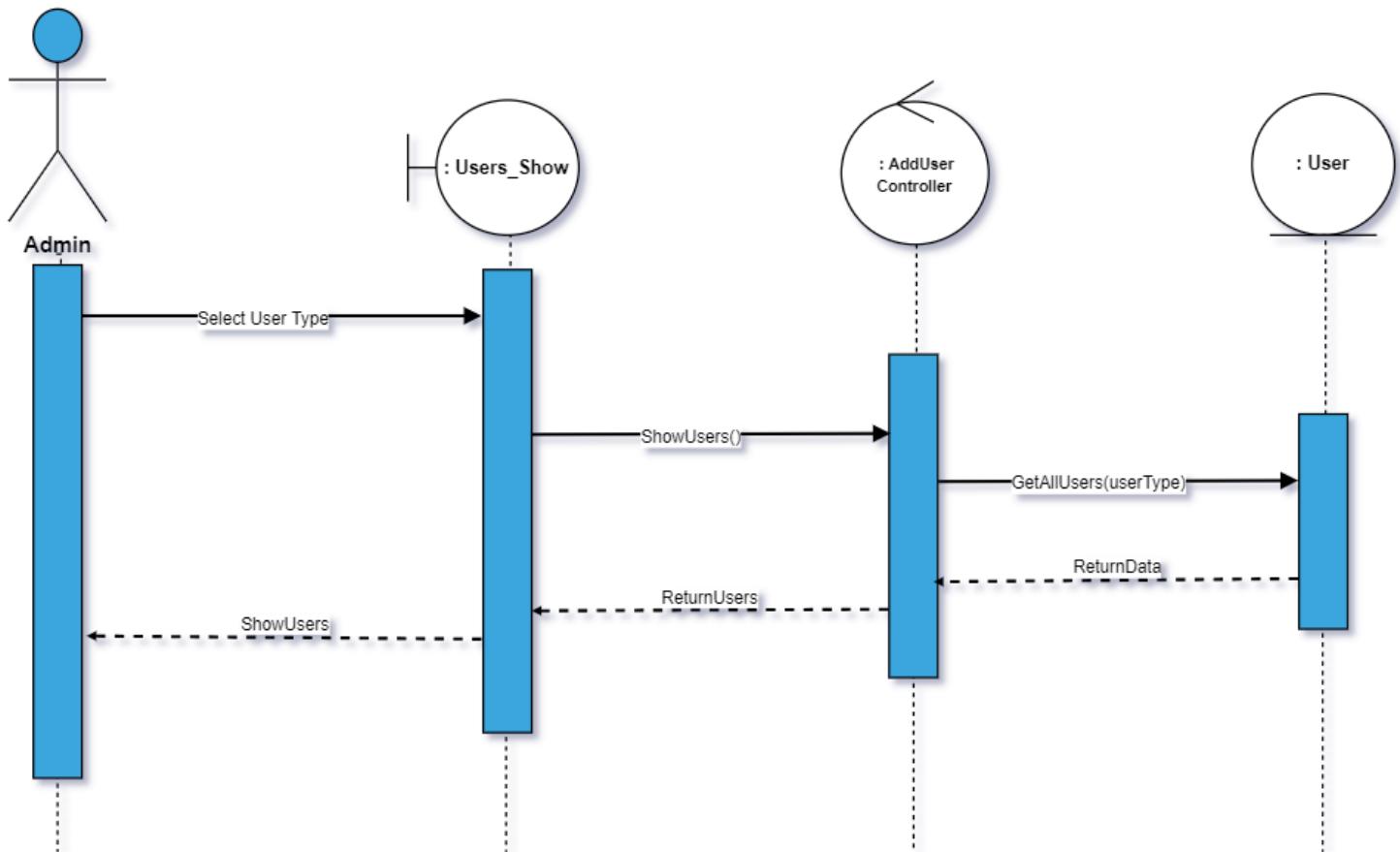
43) Registration



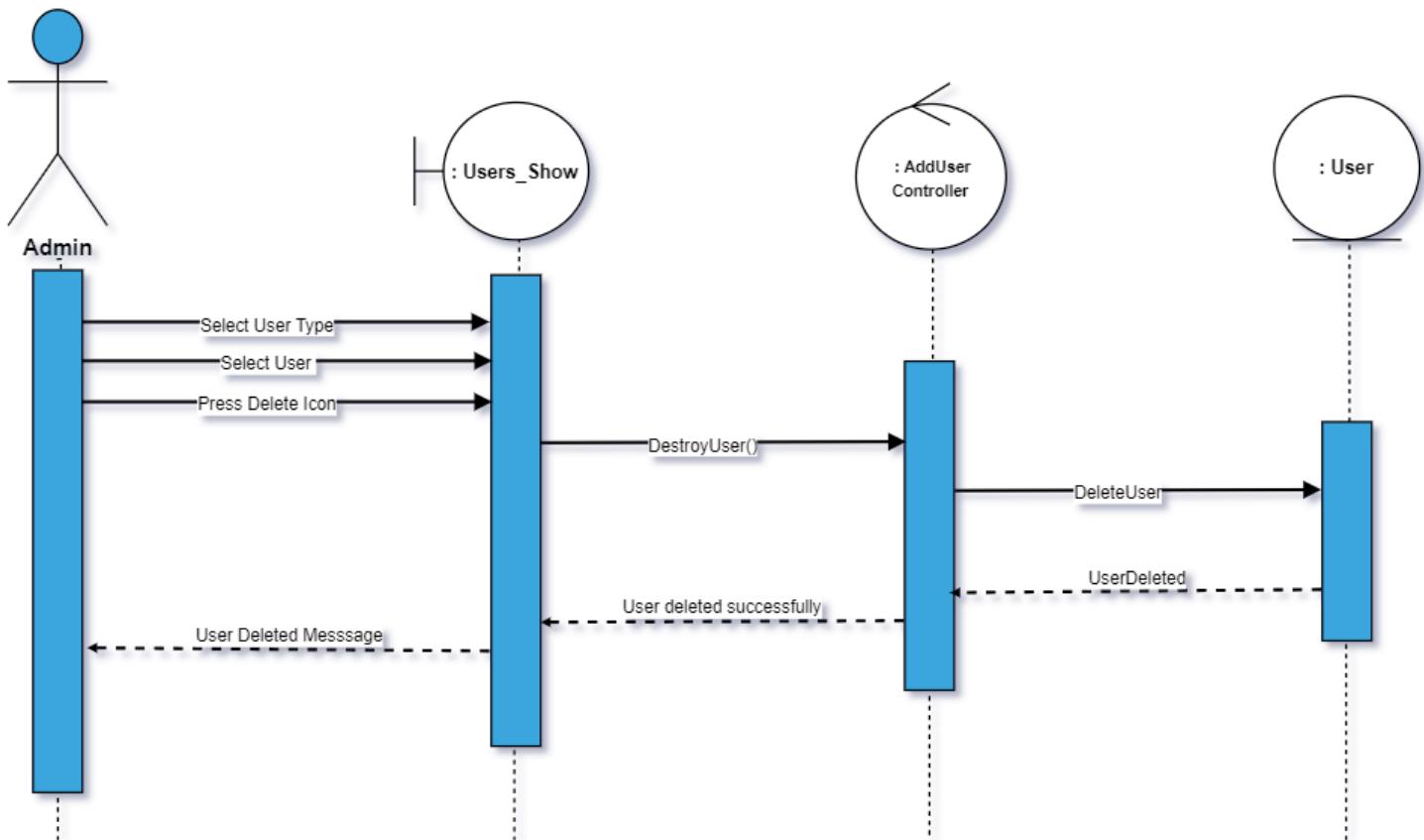
44)Edit account information



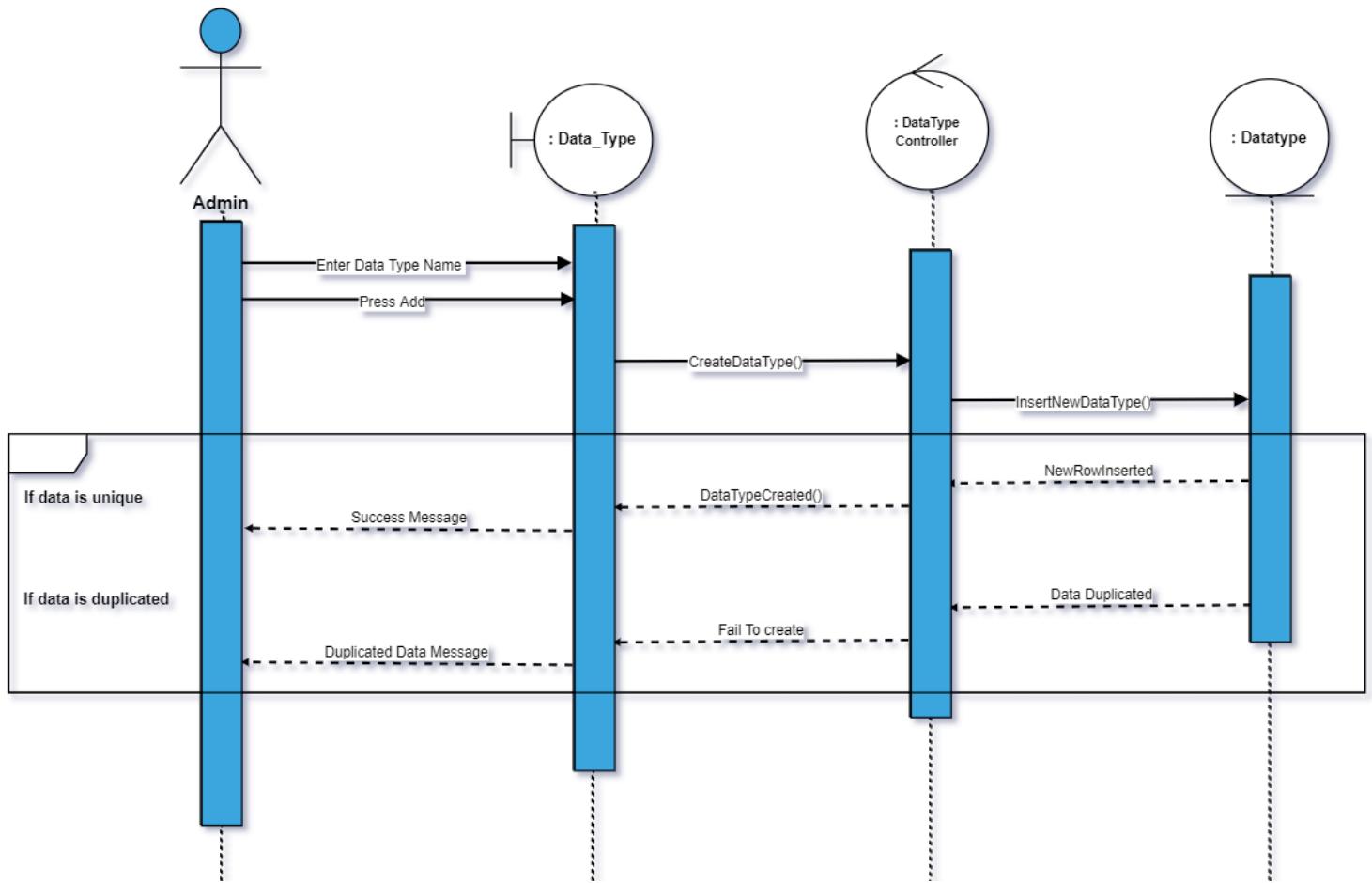
45) Show user types



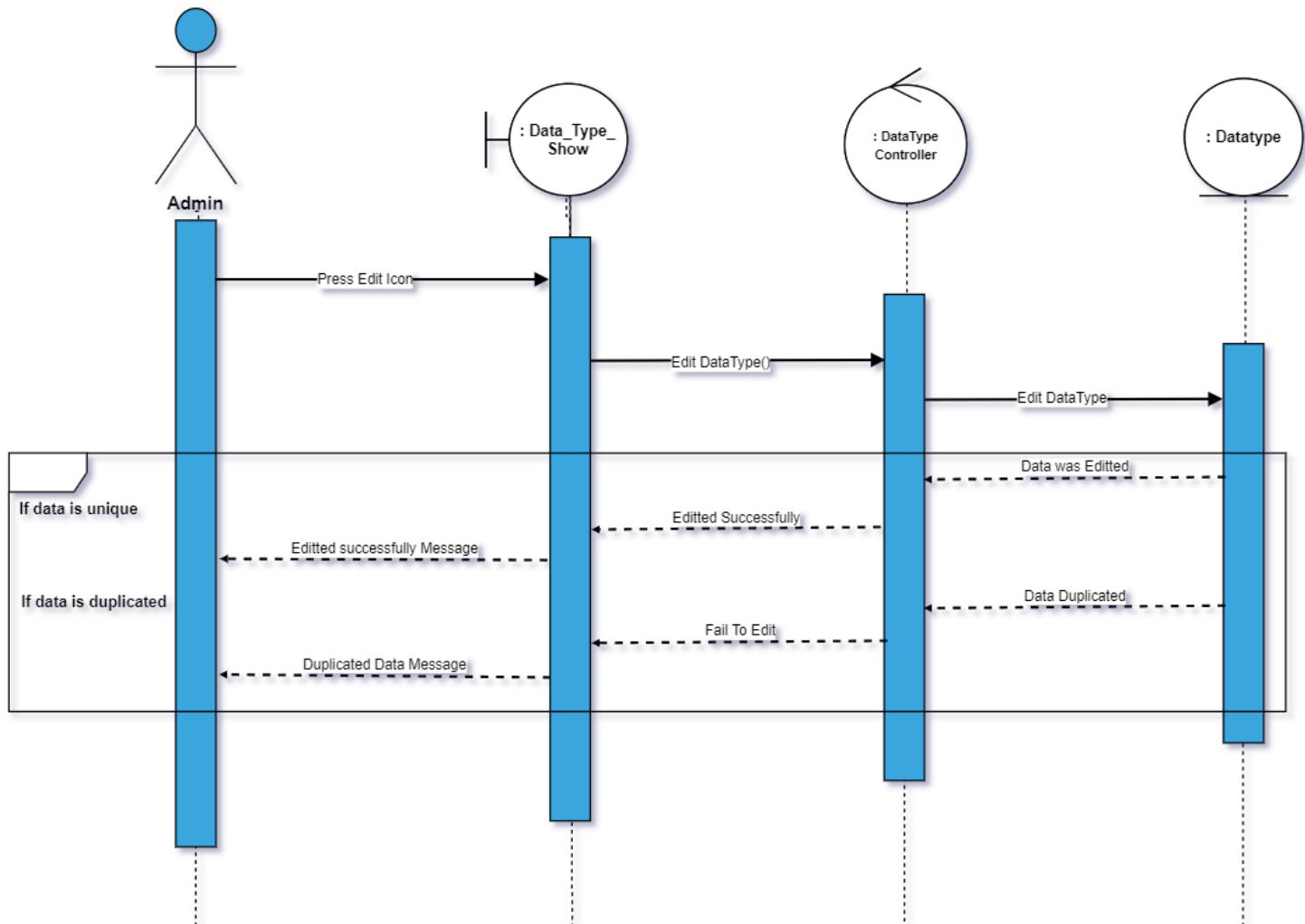
46) Delete user type



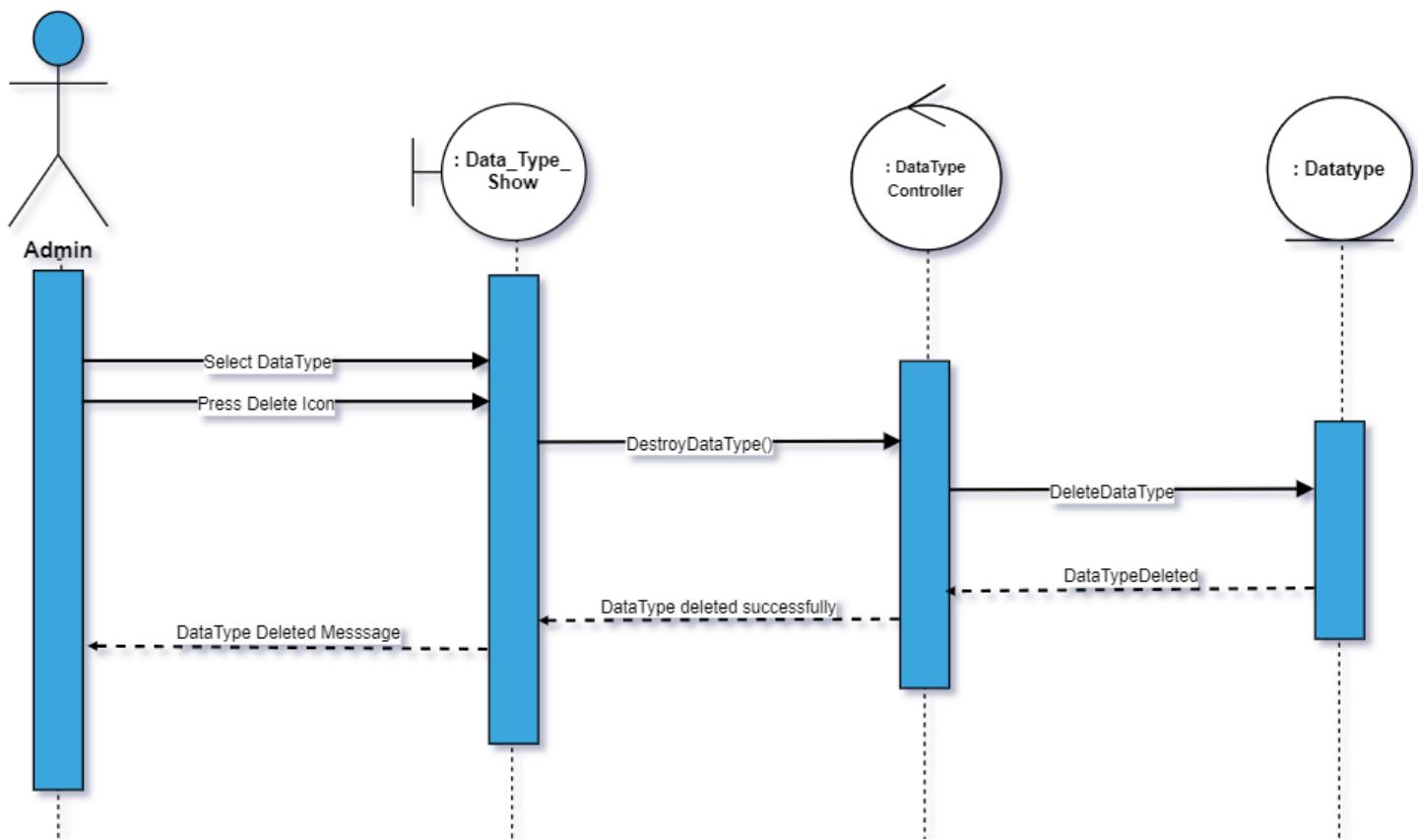
47) add data type



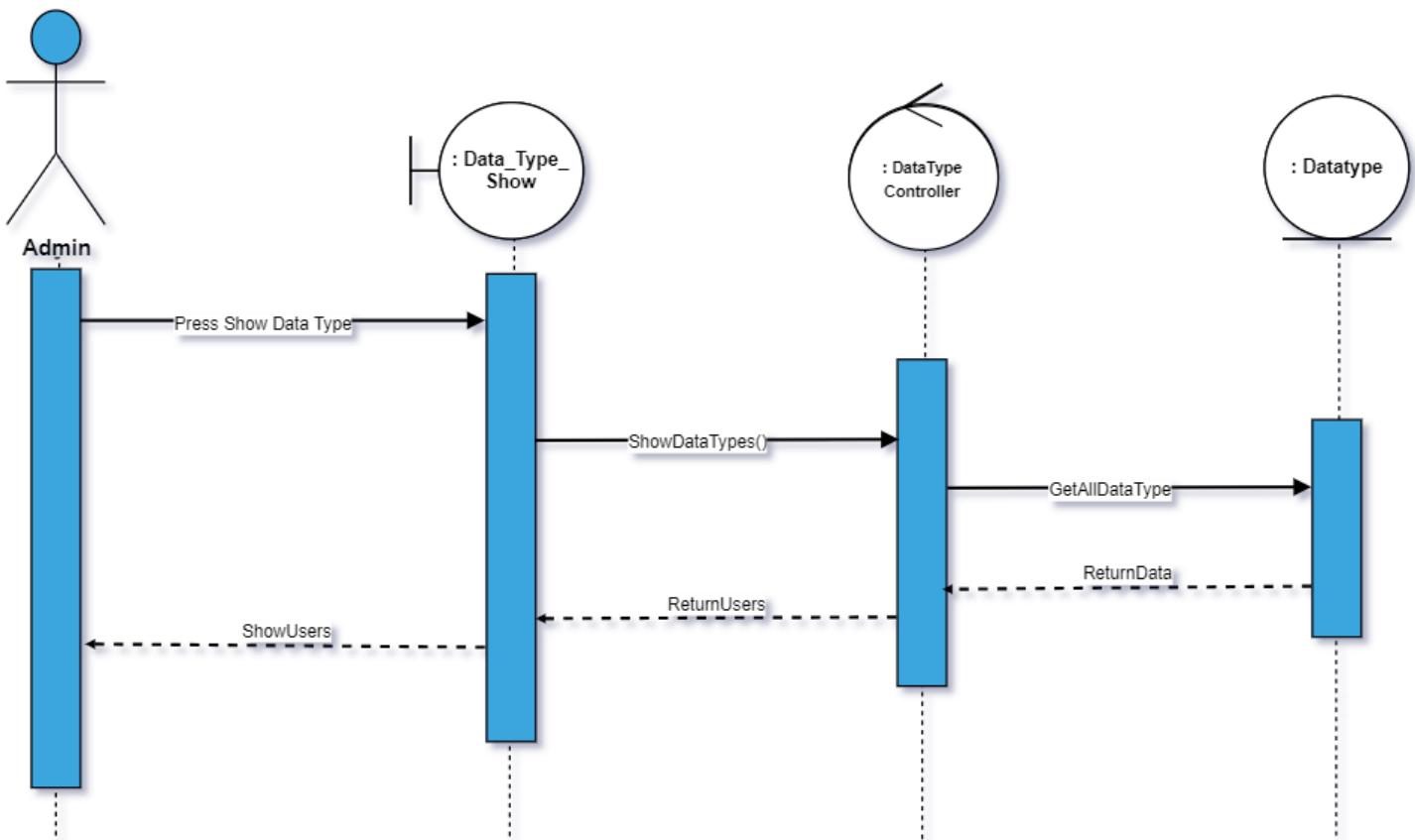
48)Edit data type



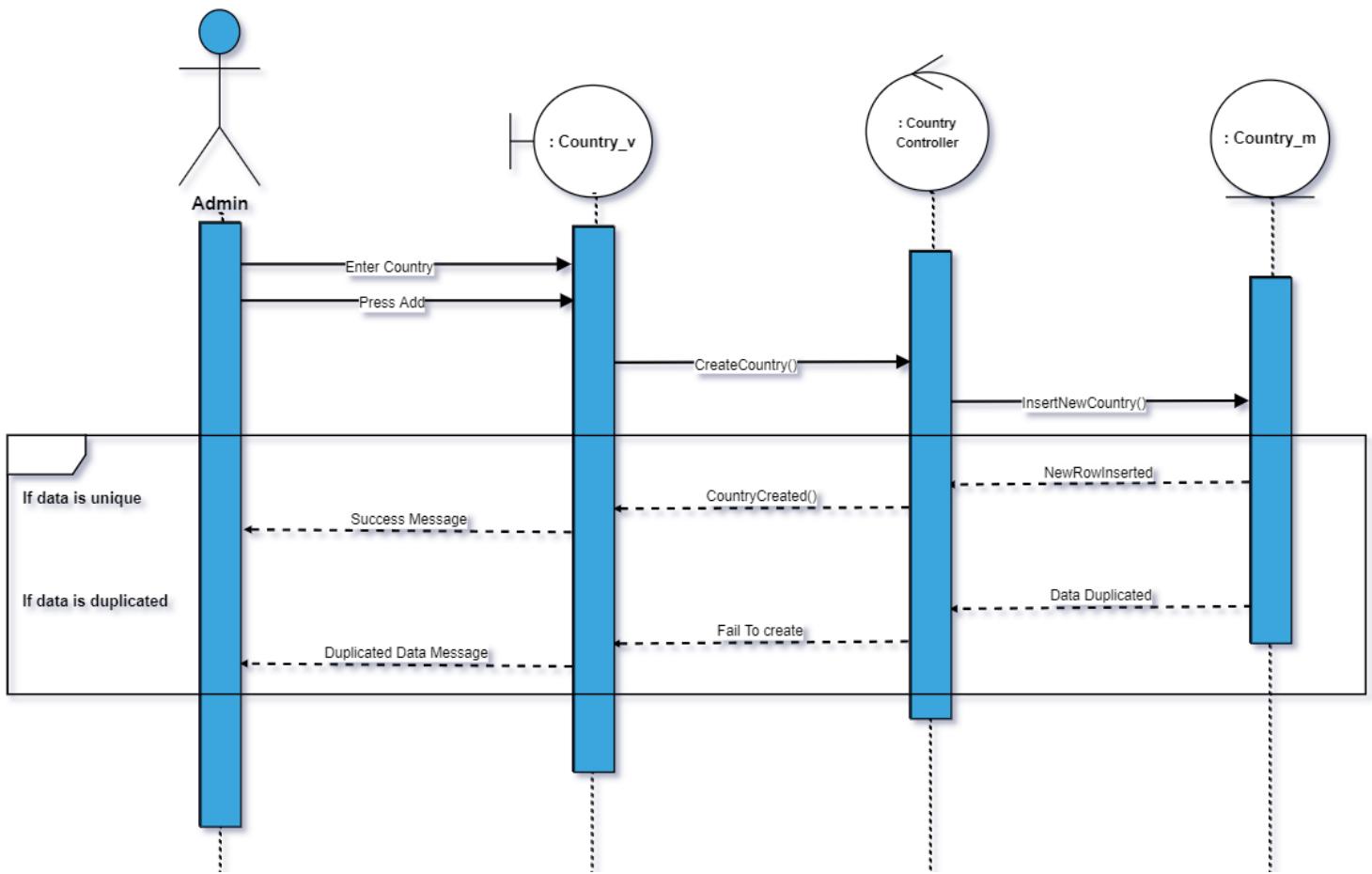
49) Delete data type



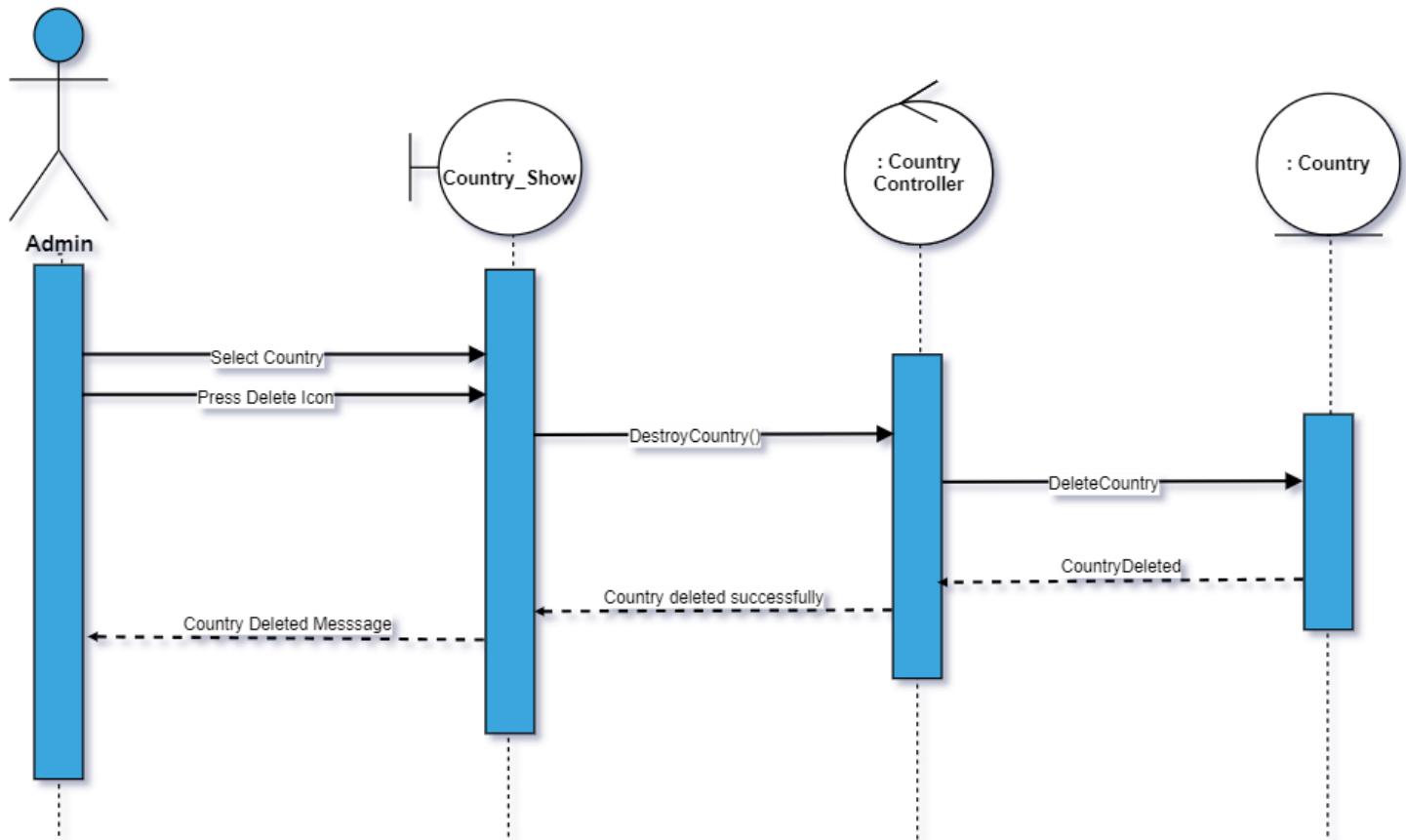
50) Show data type



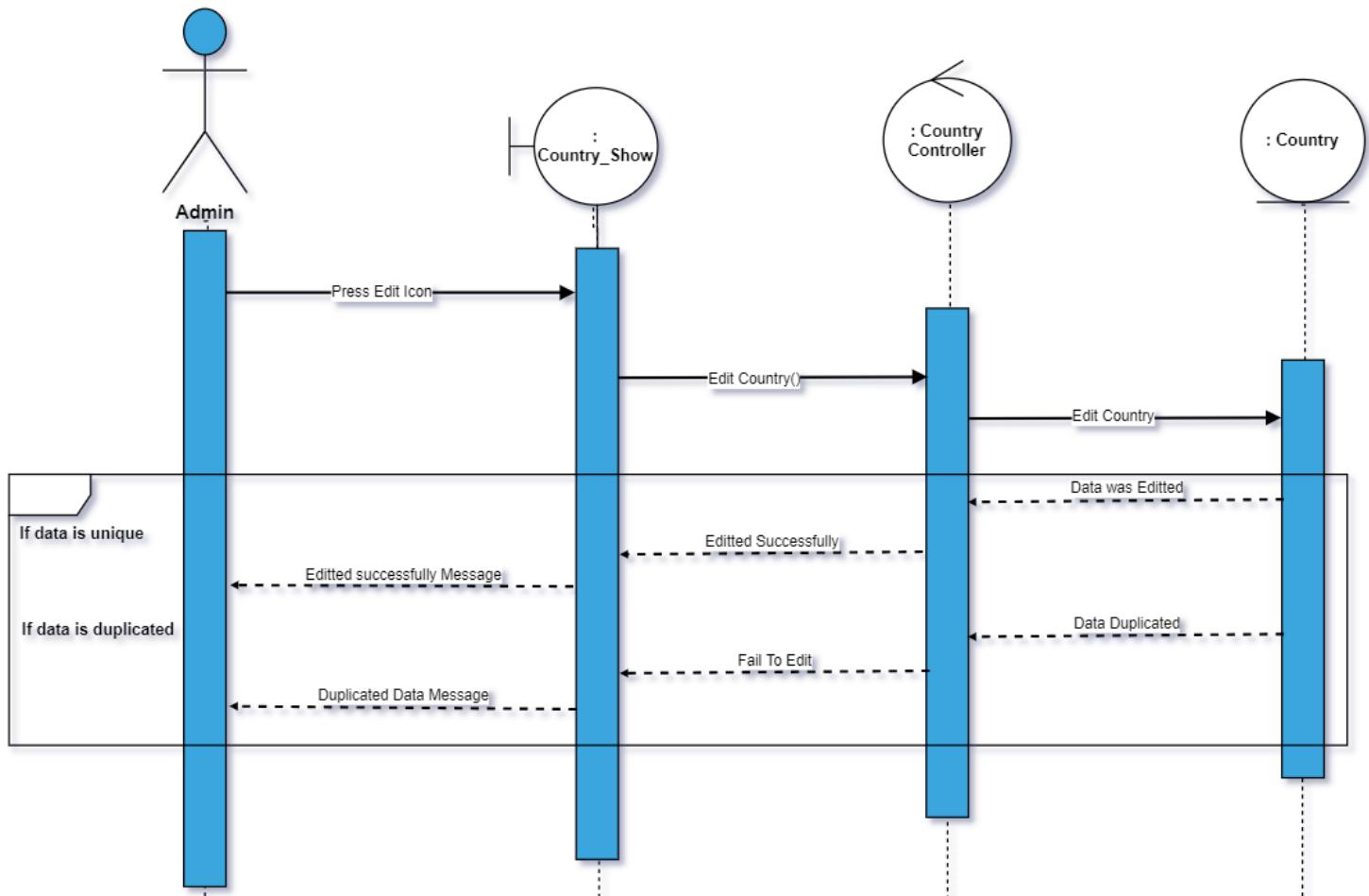
51) Add Country



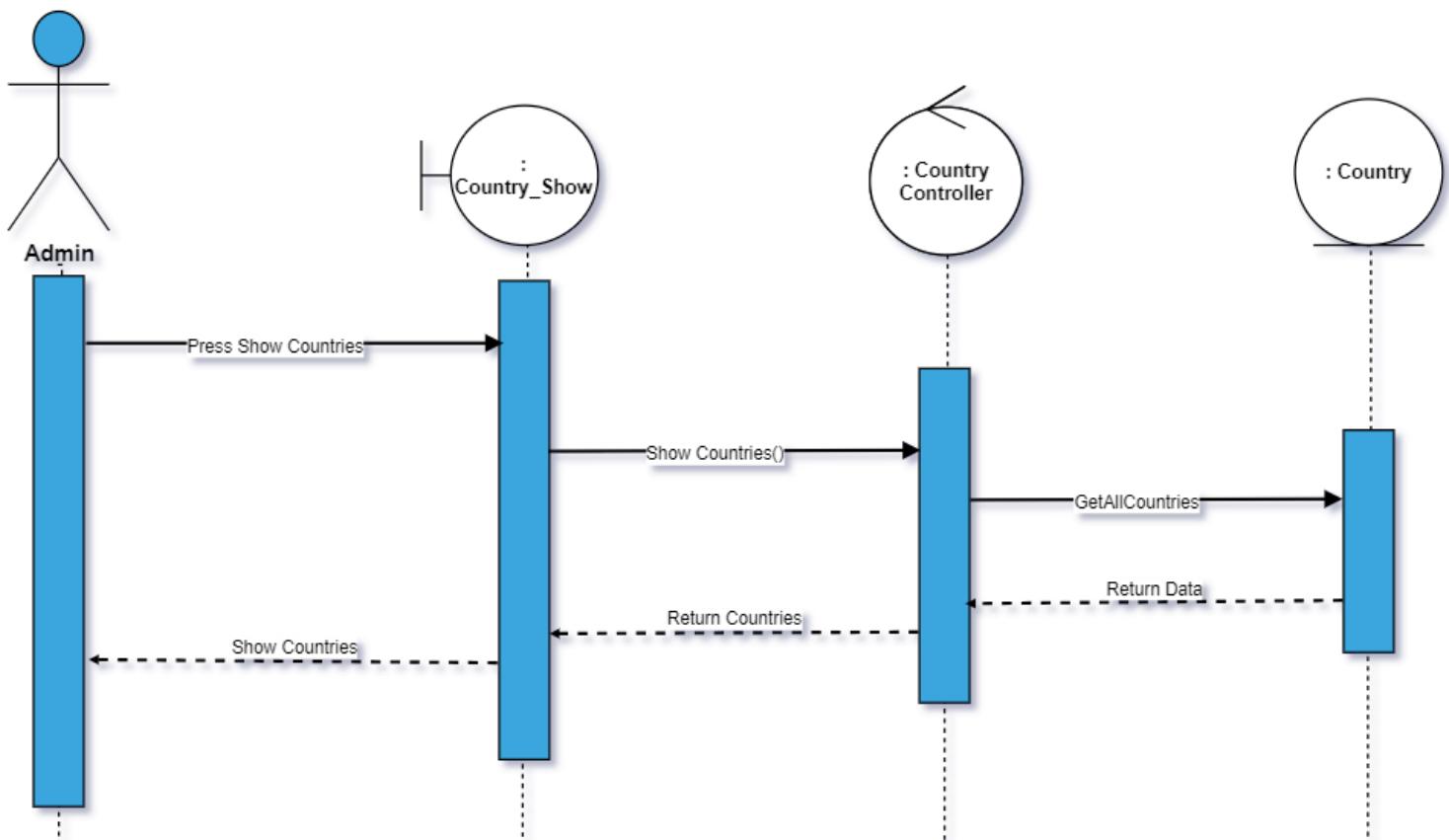
52)Delete country



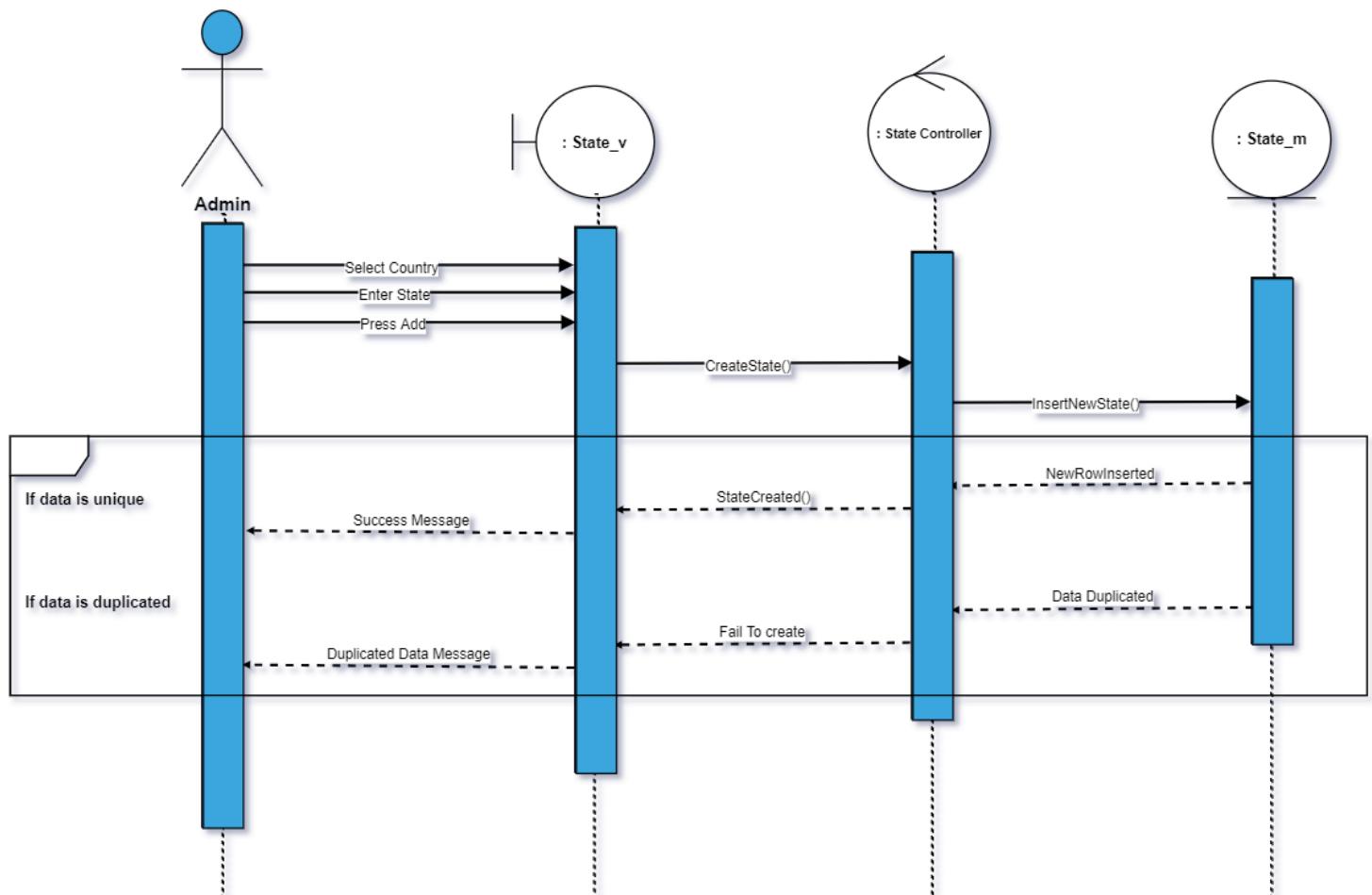
53)Edit Country



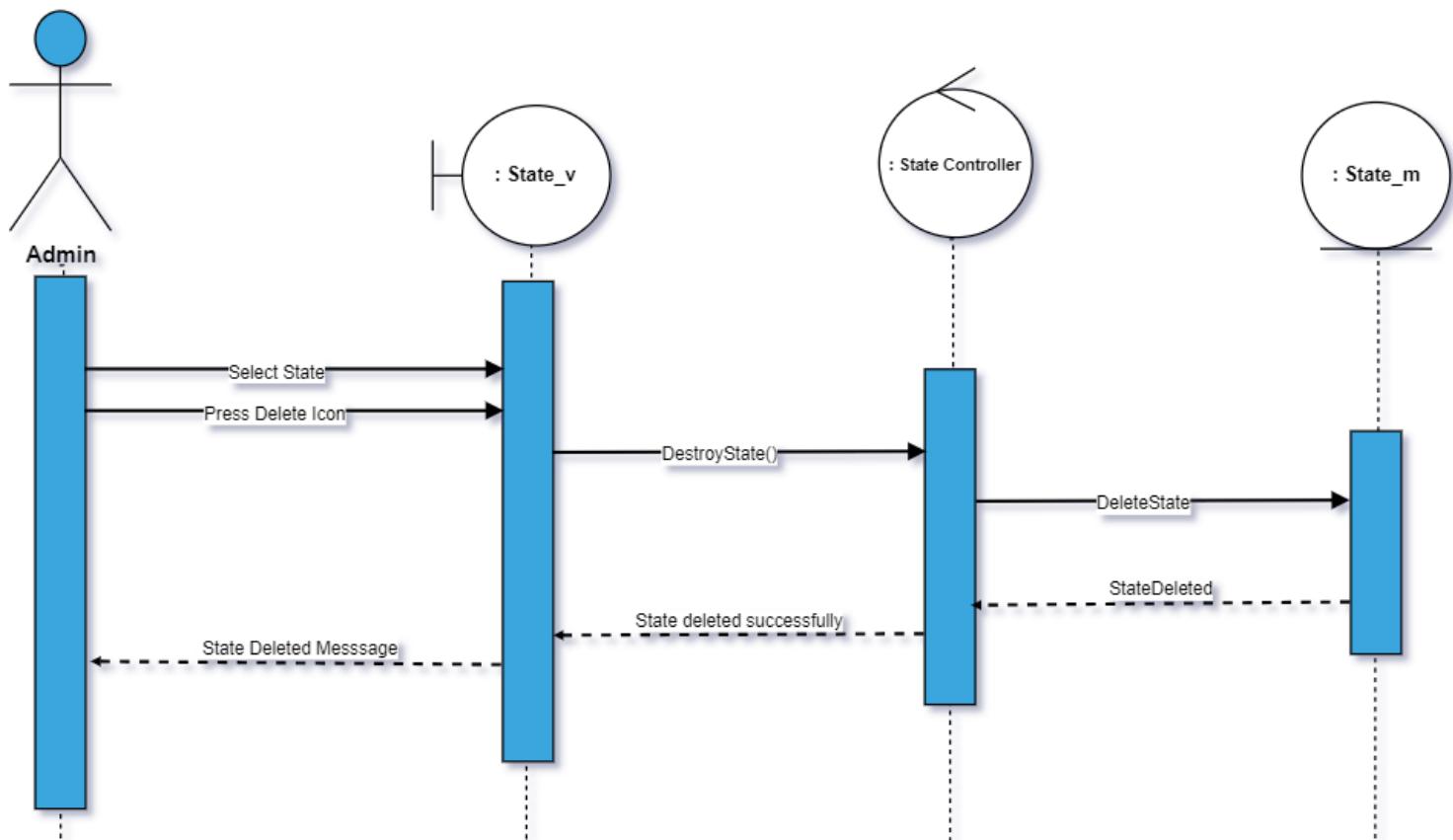
54) Show country



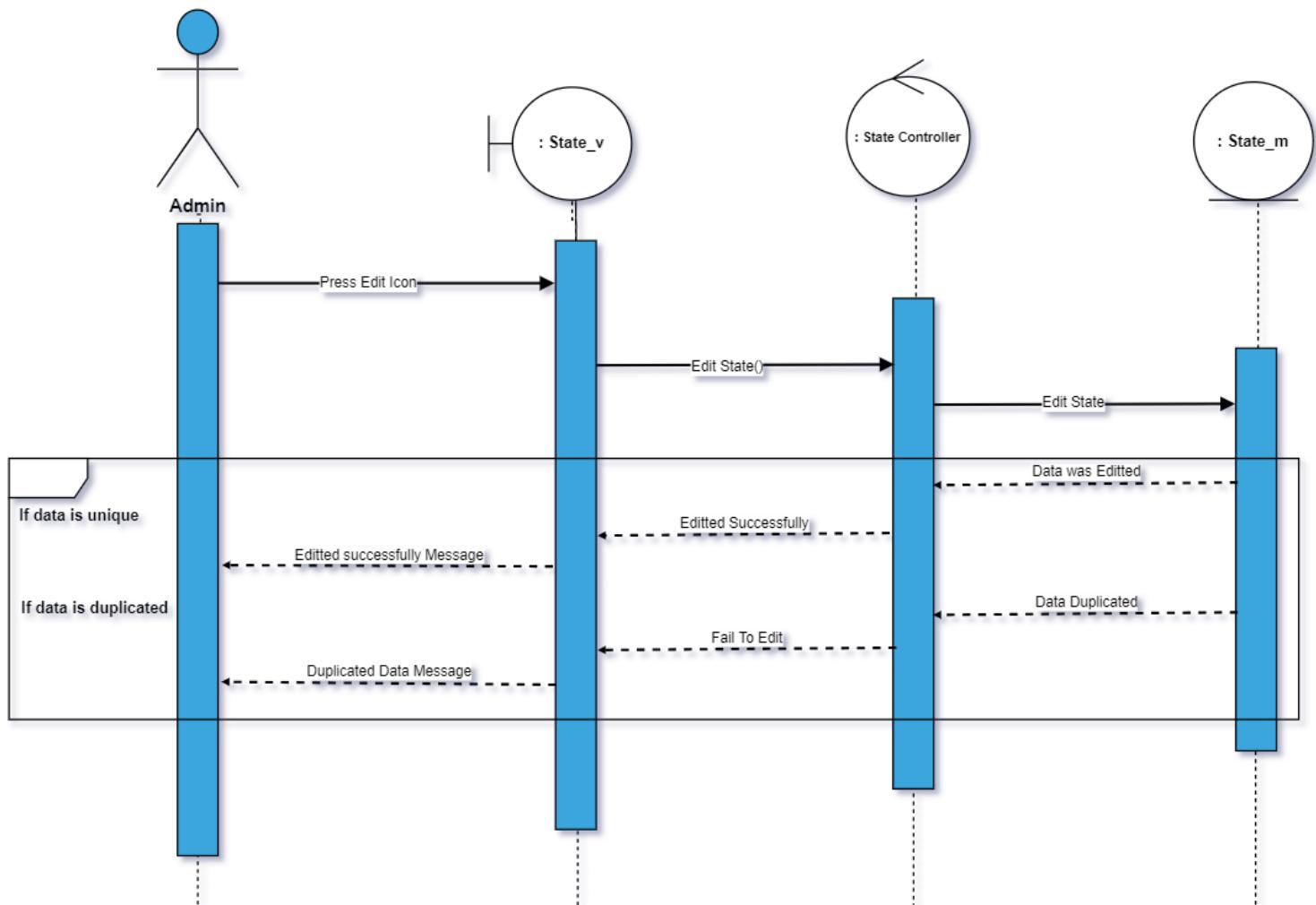
55)add state



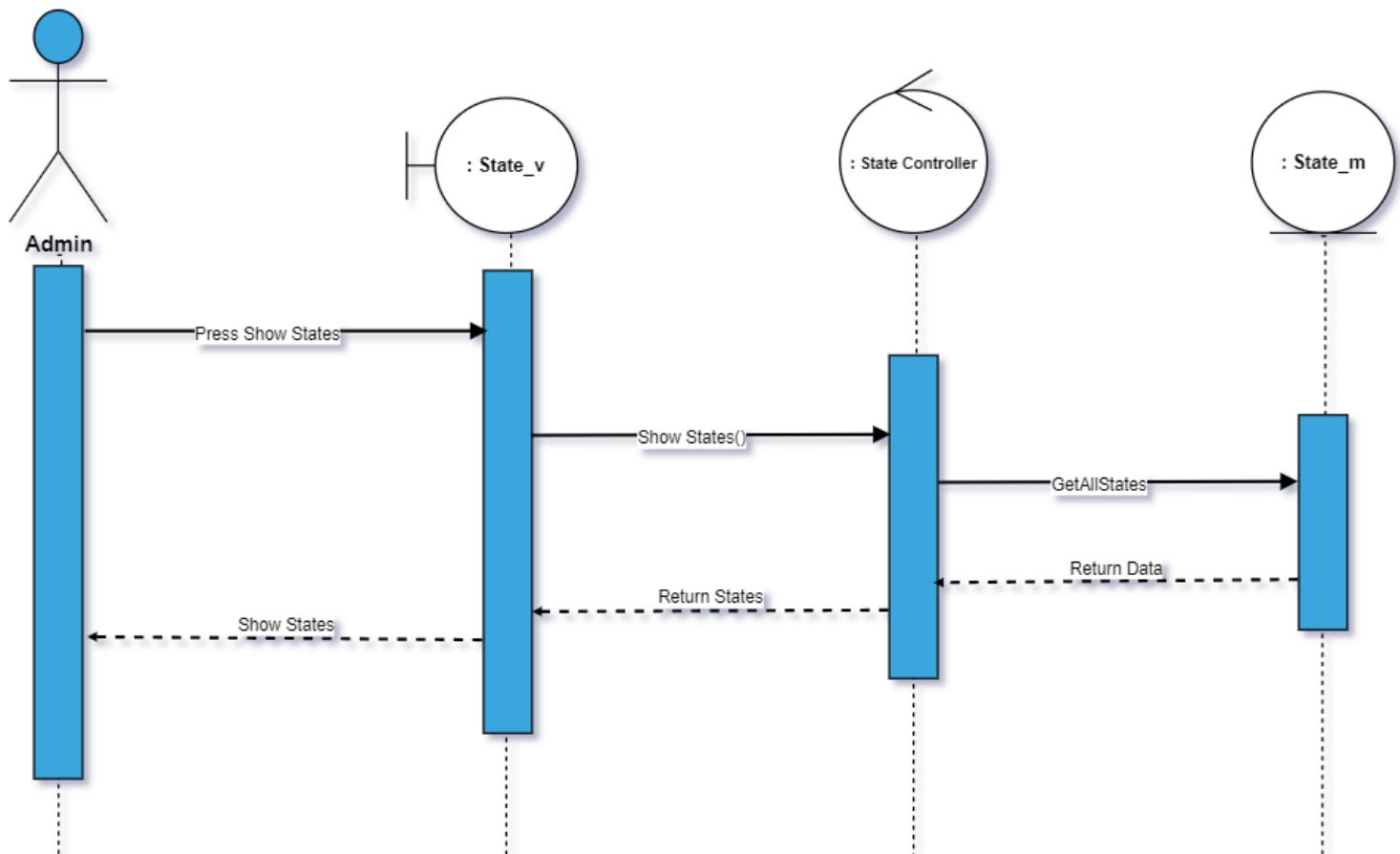
56) delete state



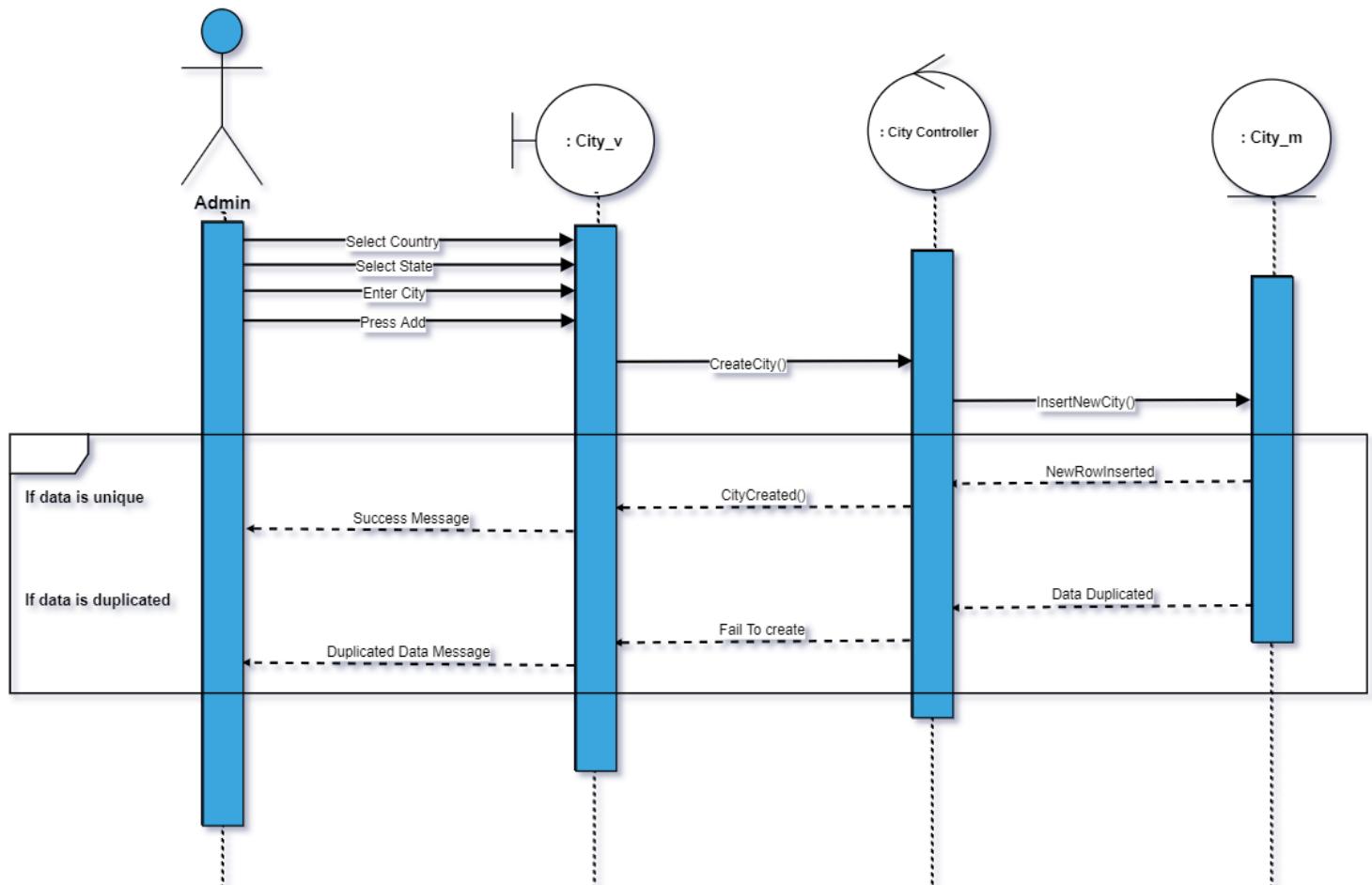
57) Edit State



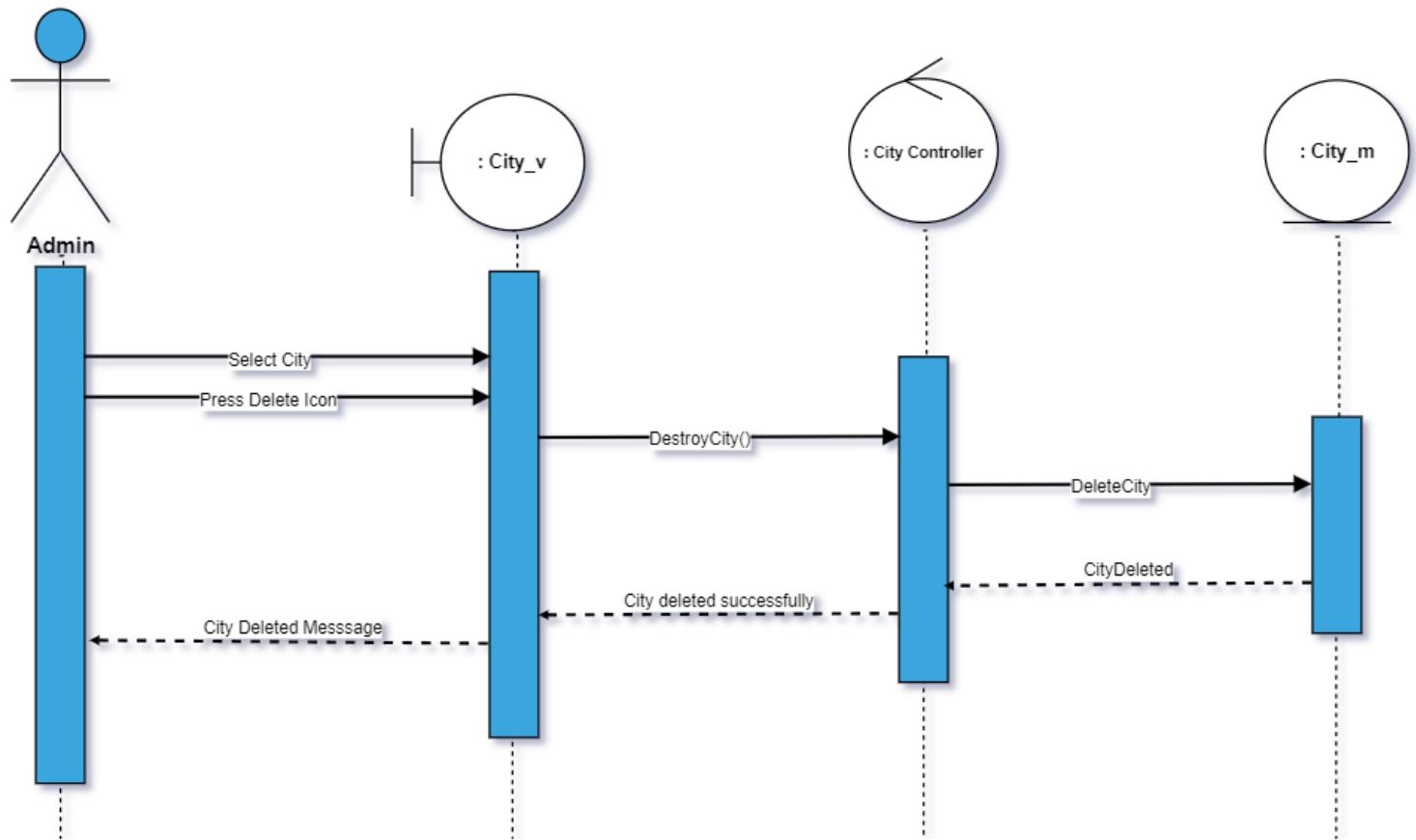
58) Show state



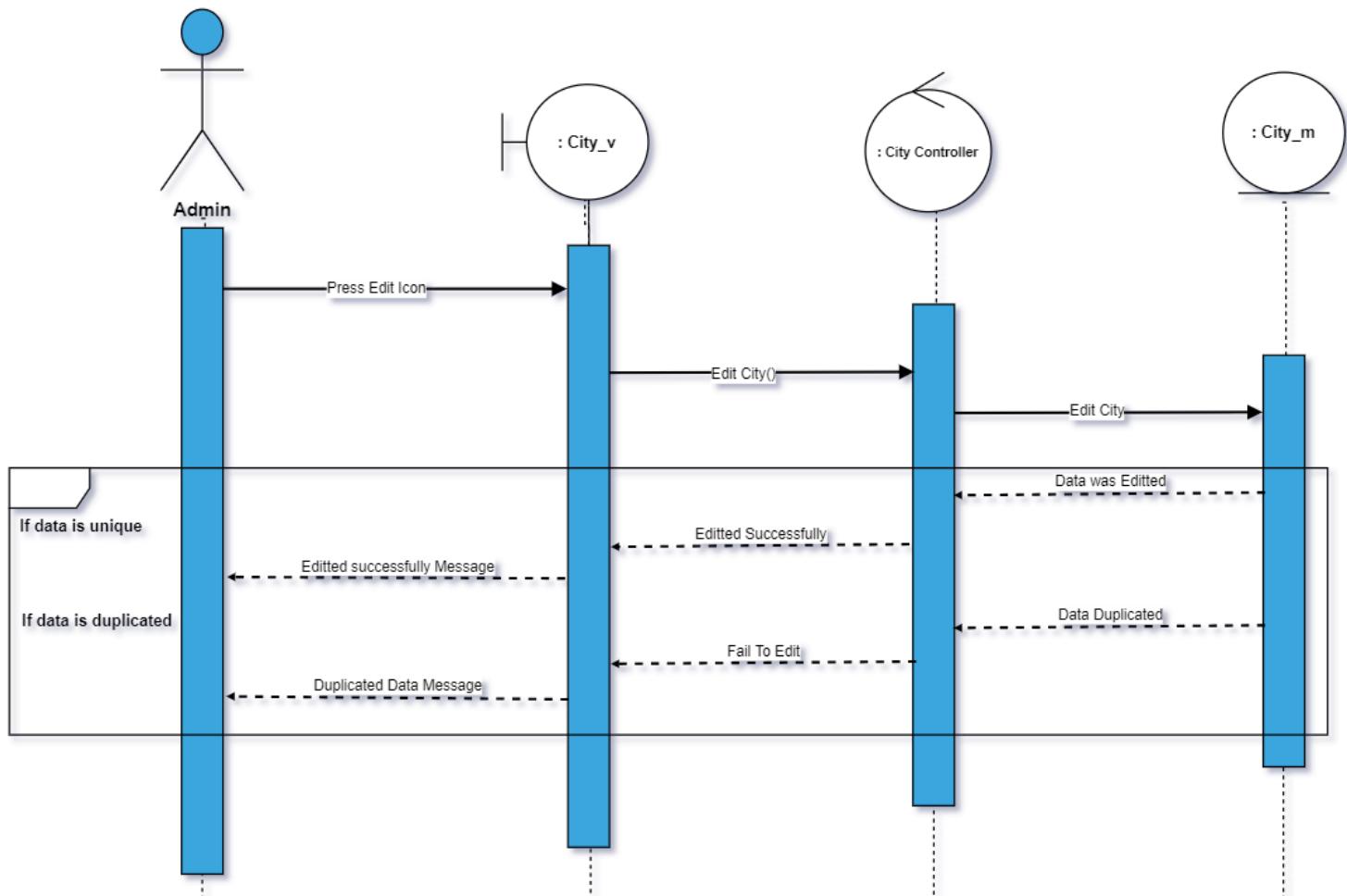
58)Add city



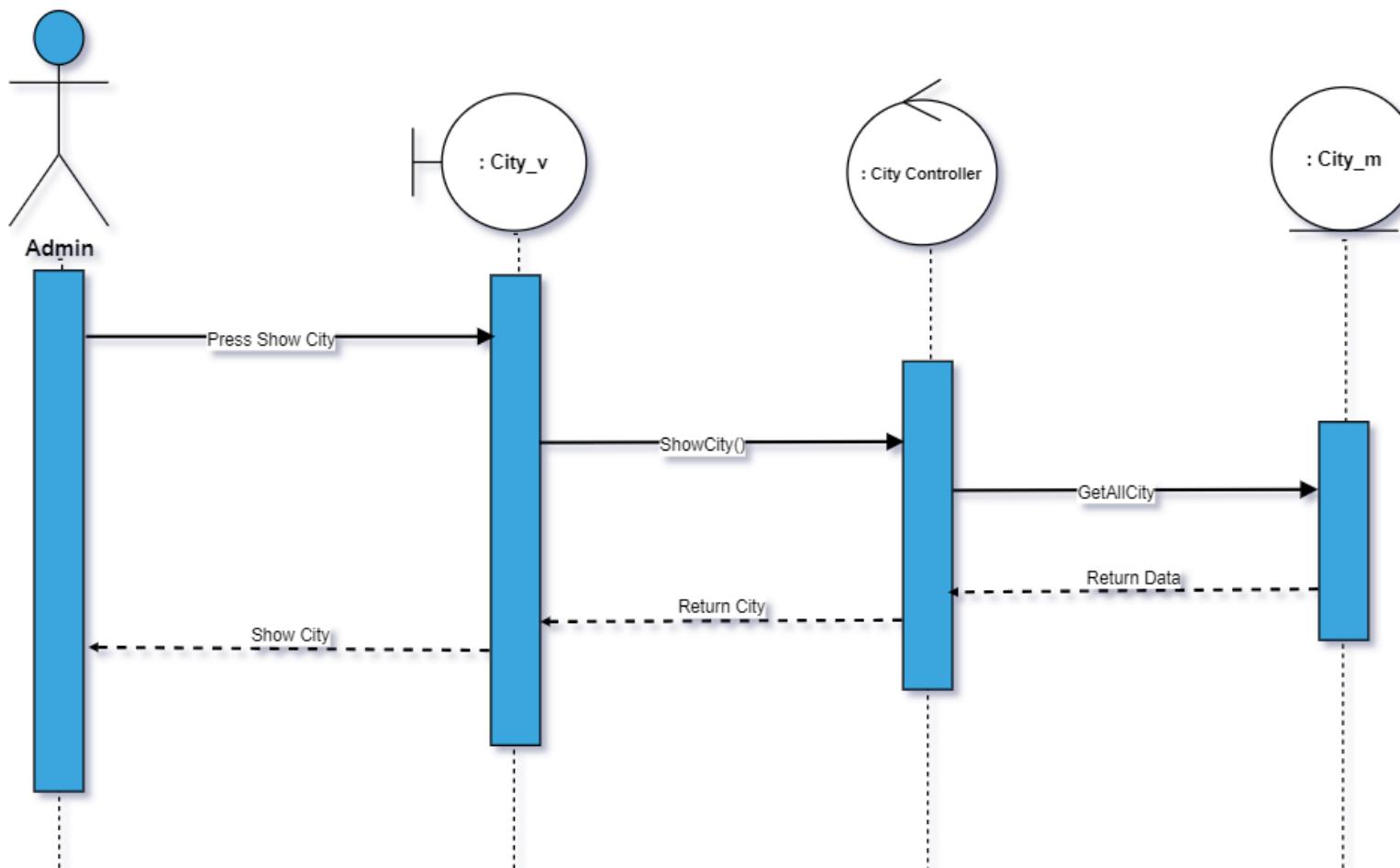
59)Delete City



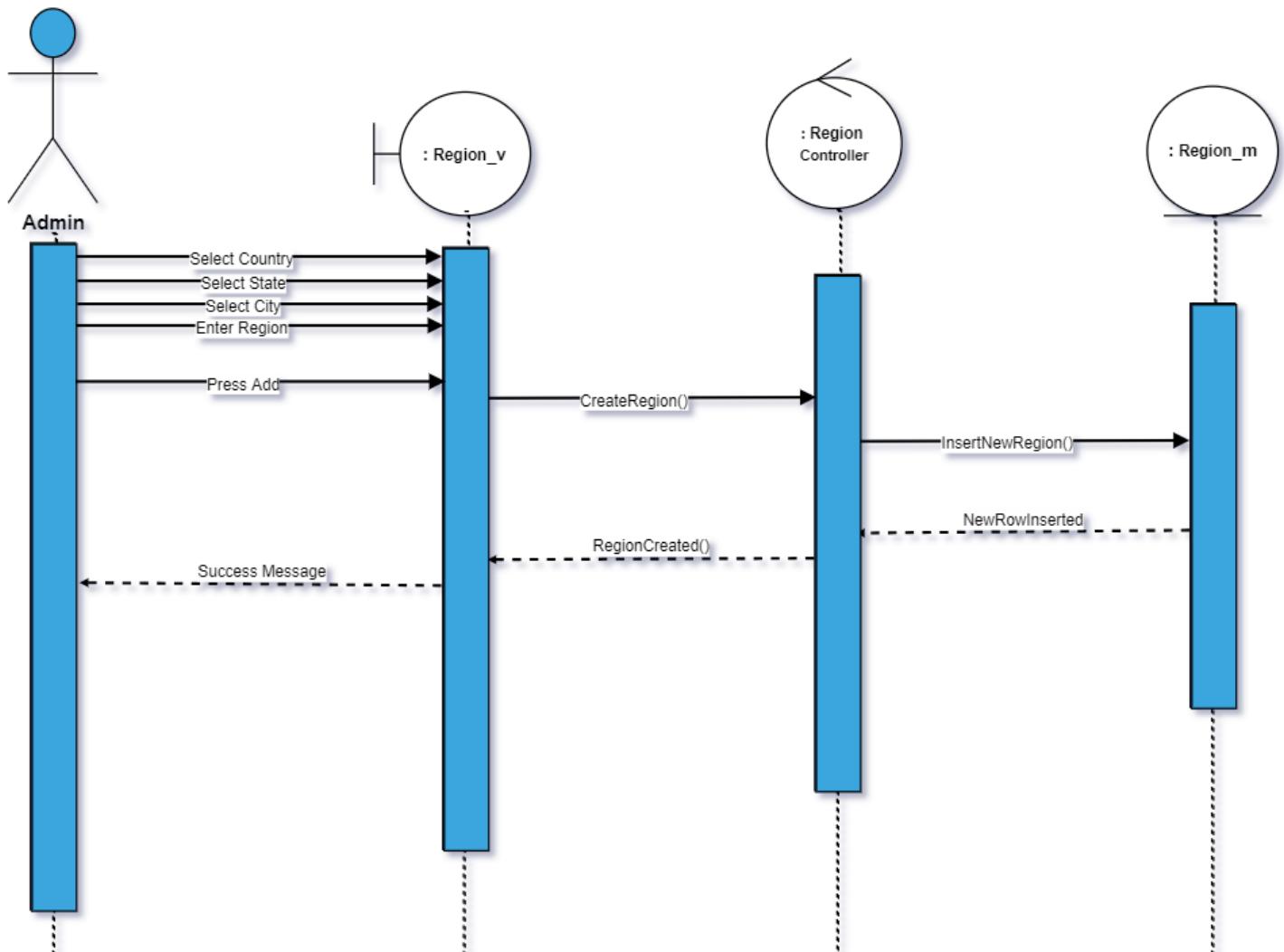
60)edit city



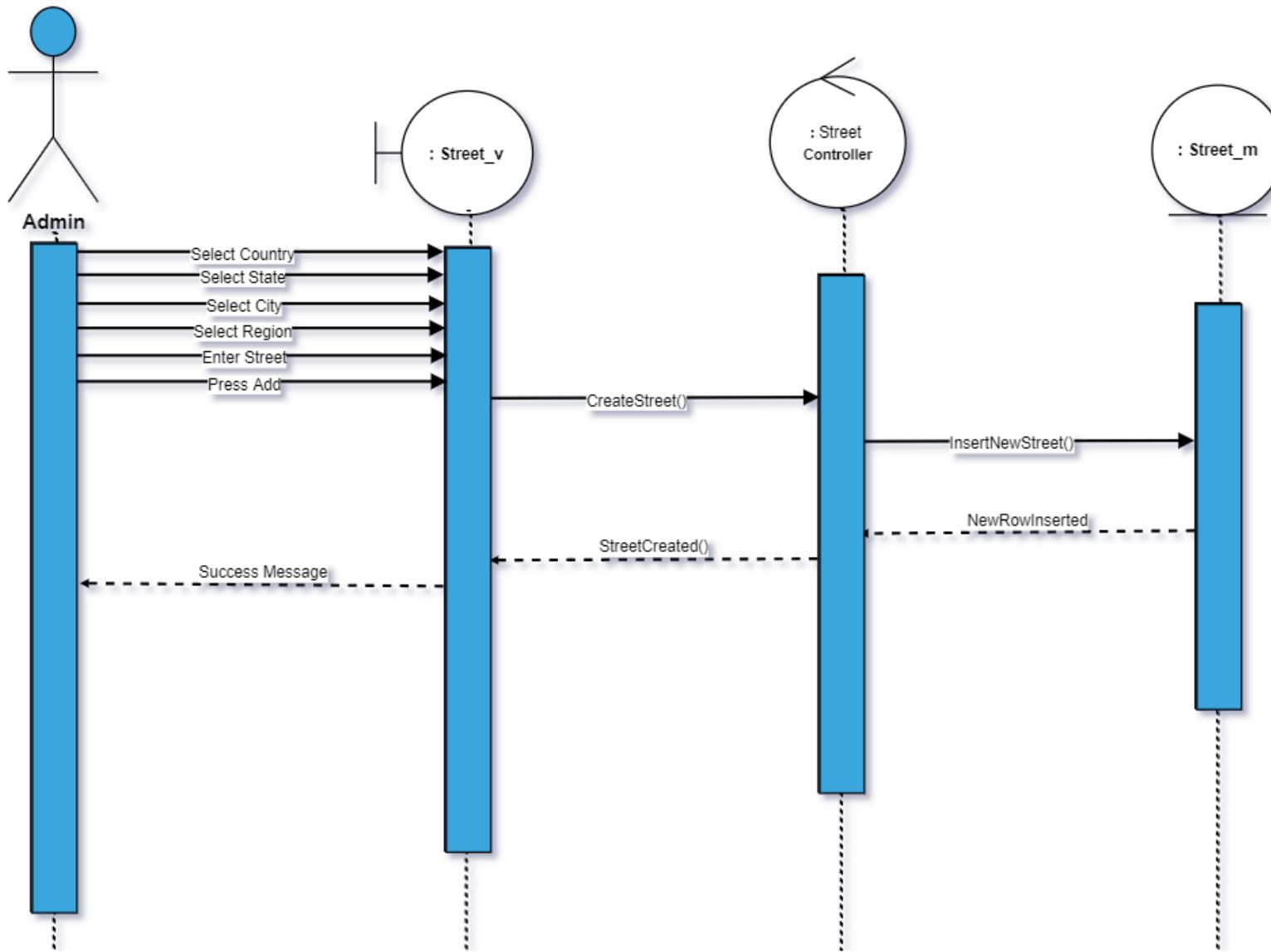
61) Show City



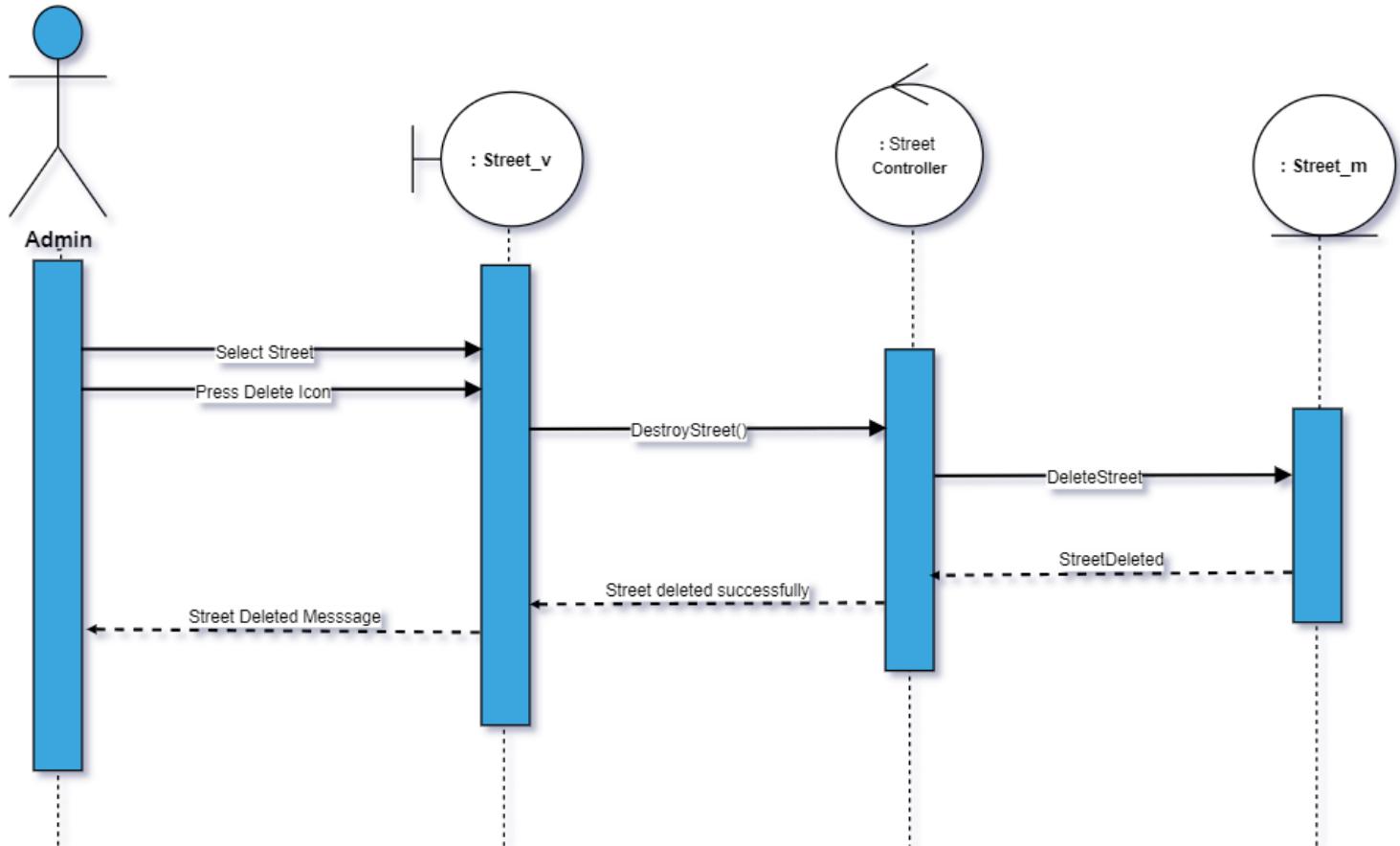
62)add region



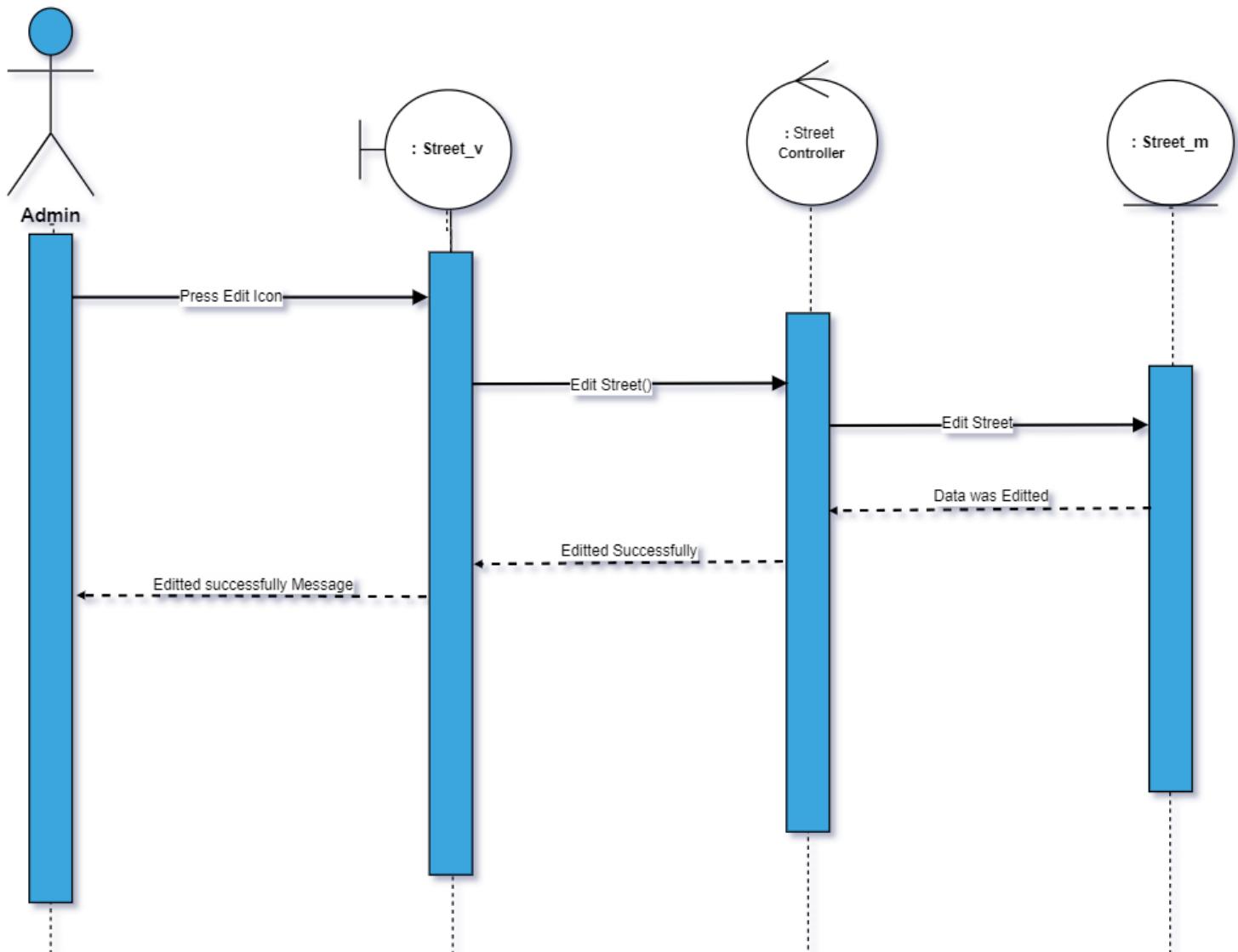
63)add street



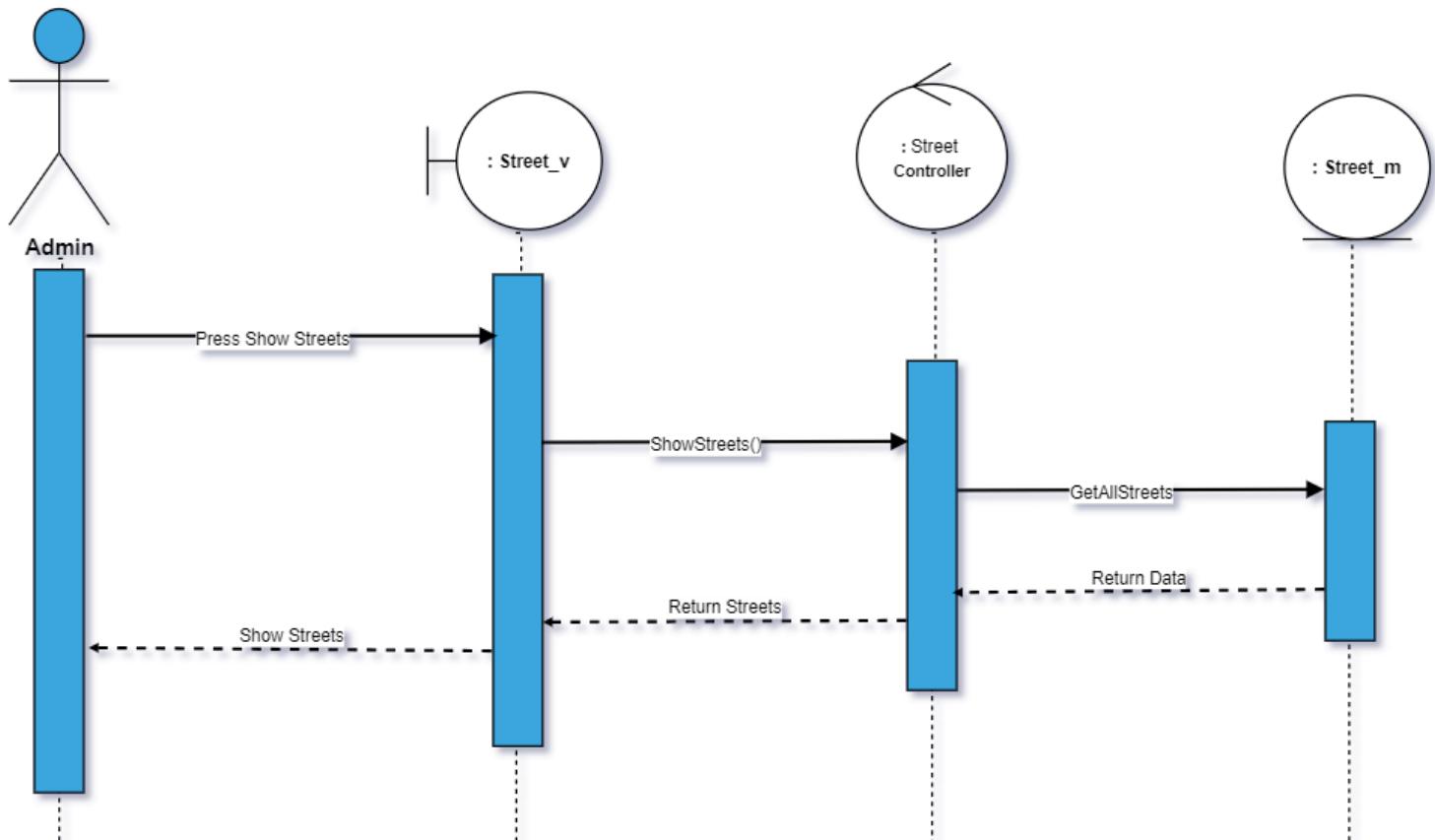
64)Delete street



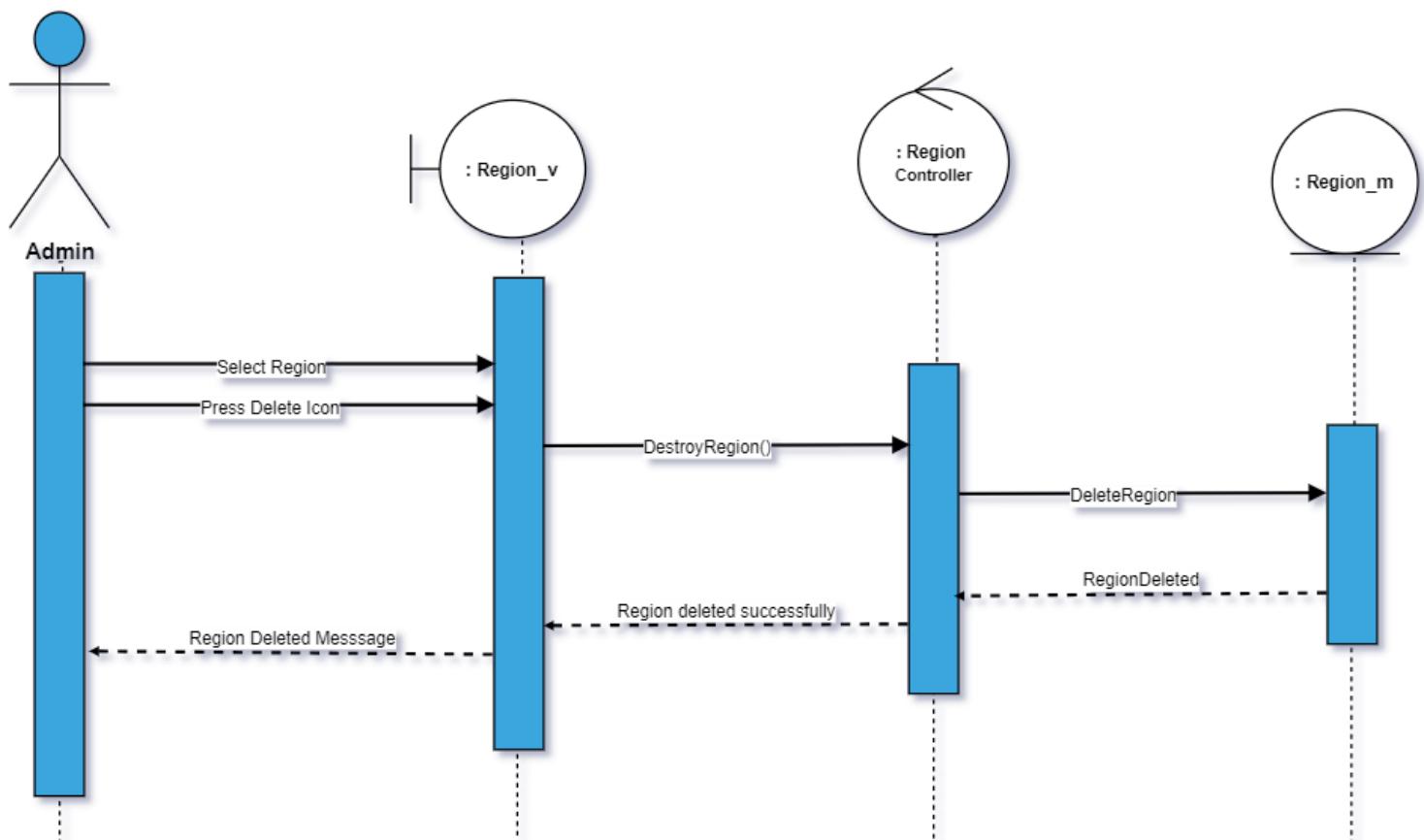
65)Edit Street



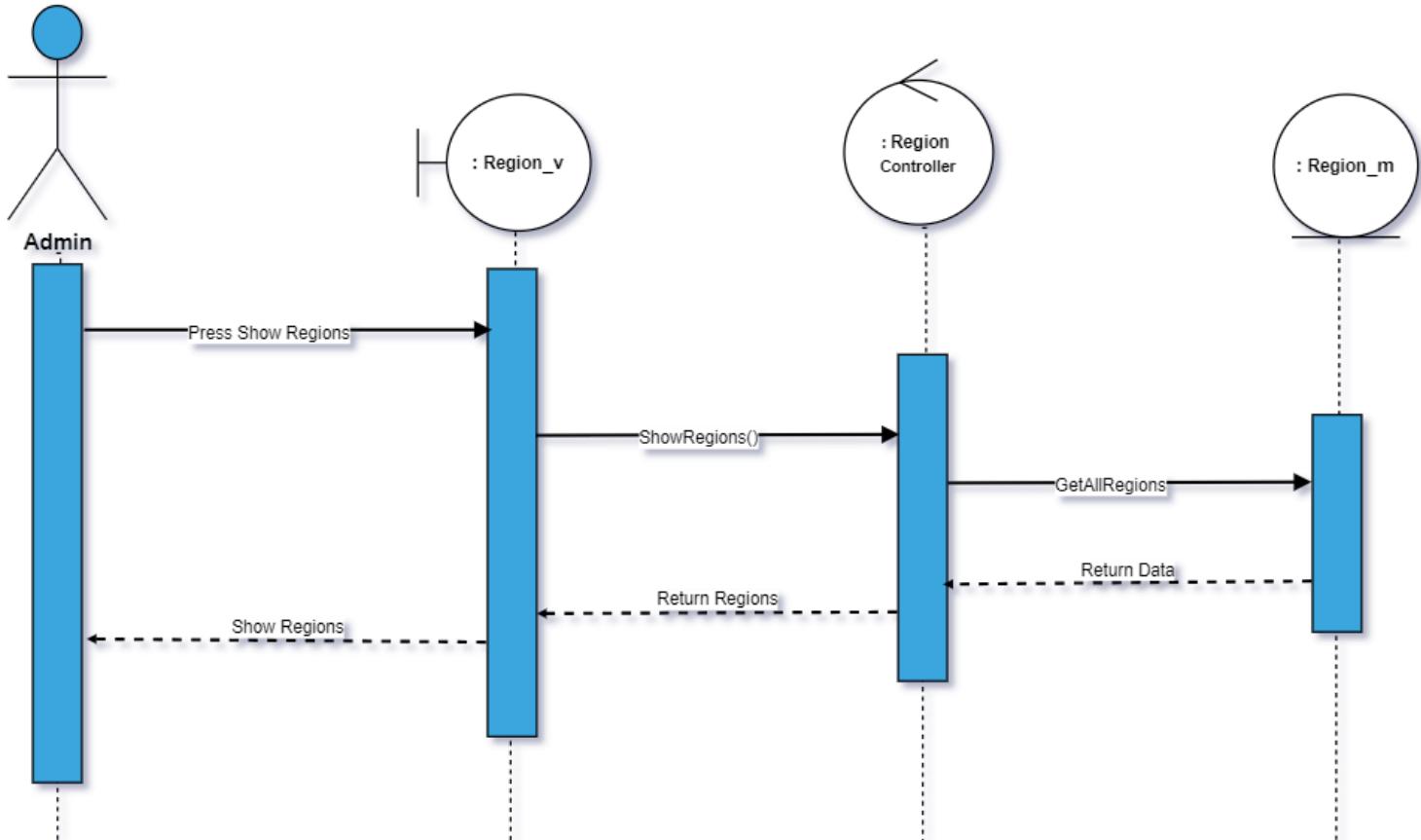
66)show street



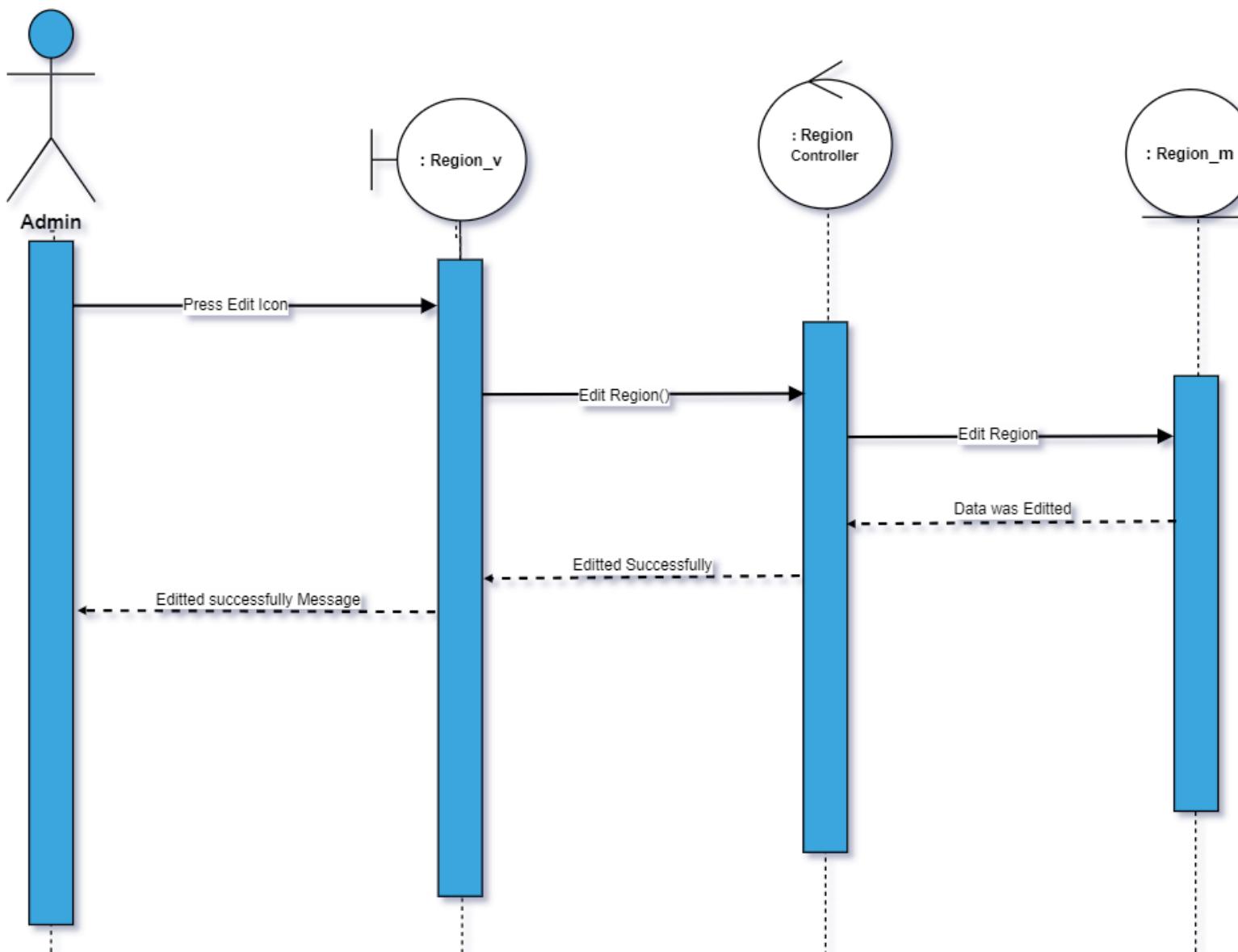
67) Delete Region



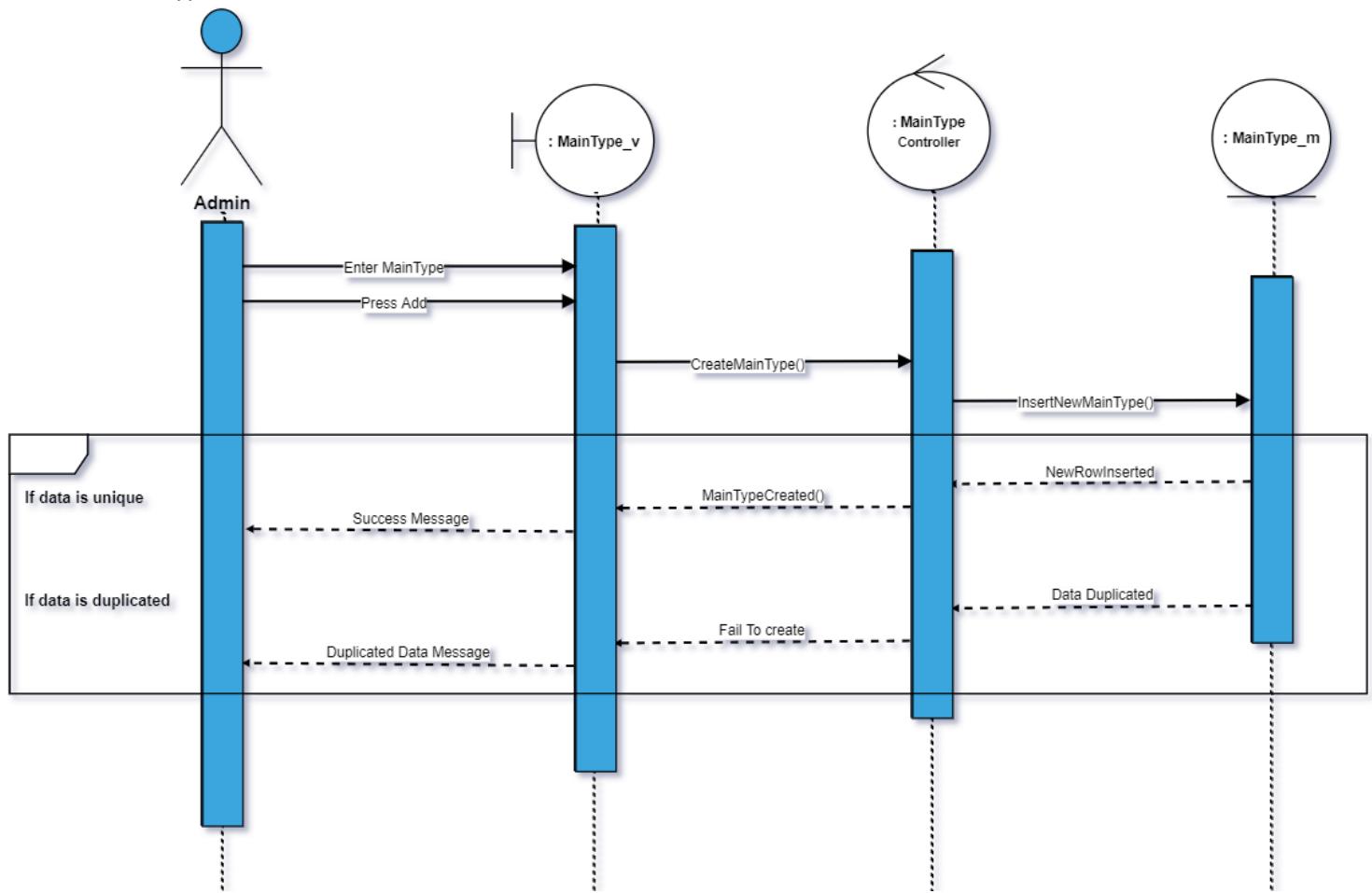
68) Show Region



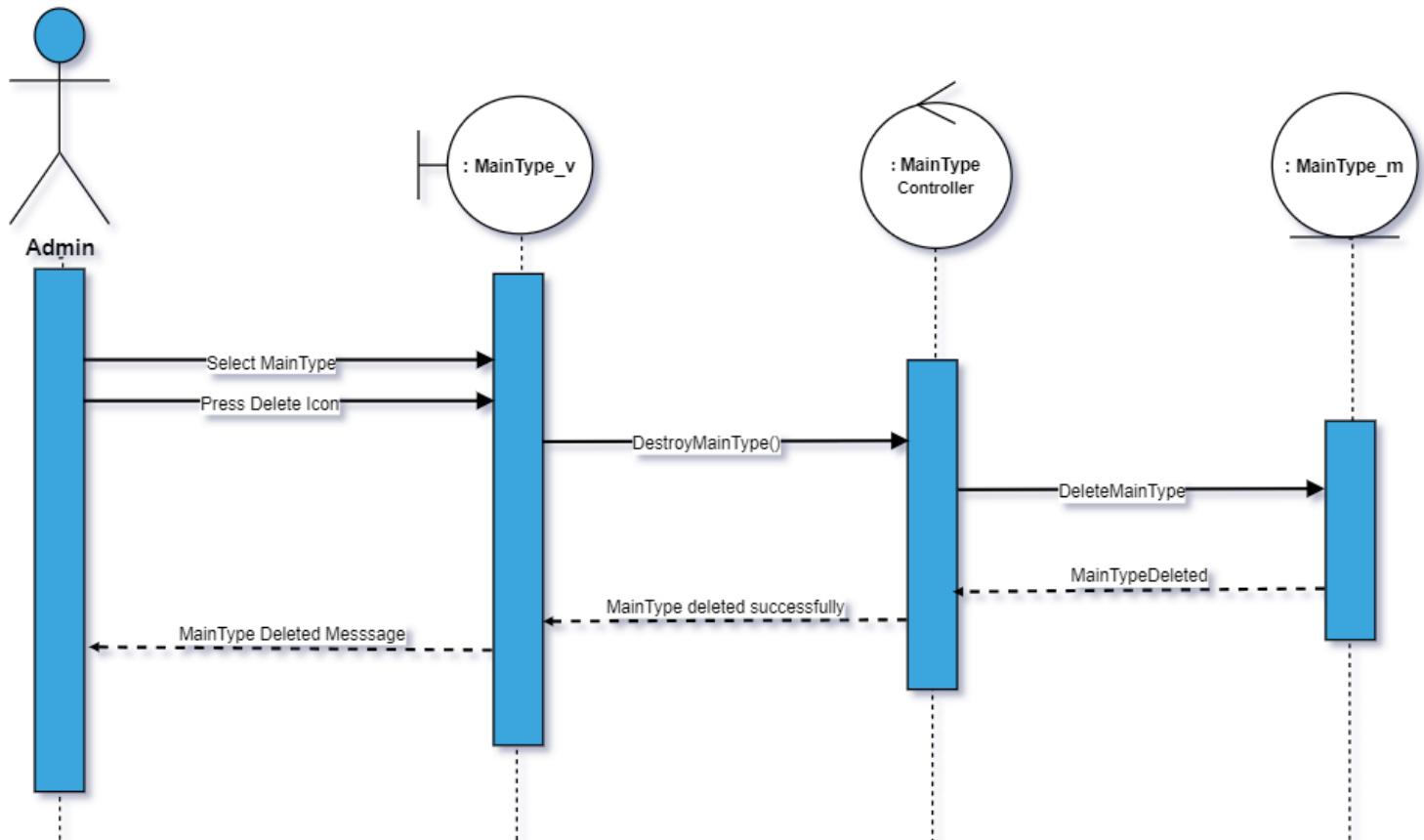
69)Edit Region



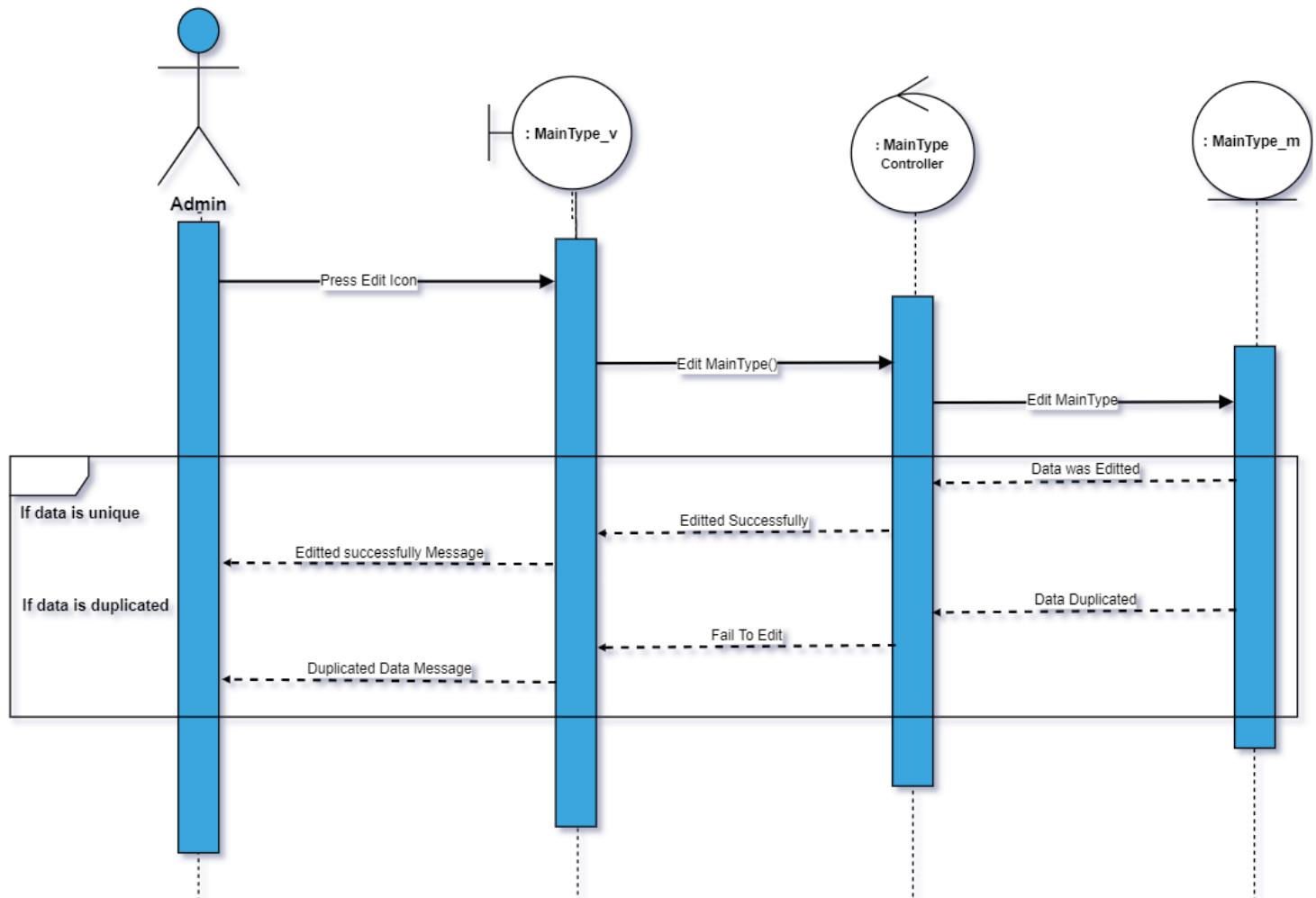
70) Add Main Type



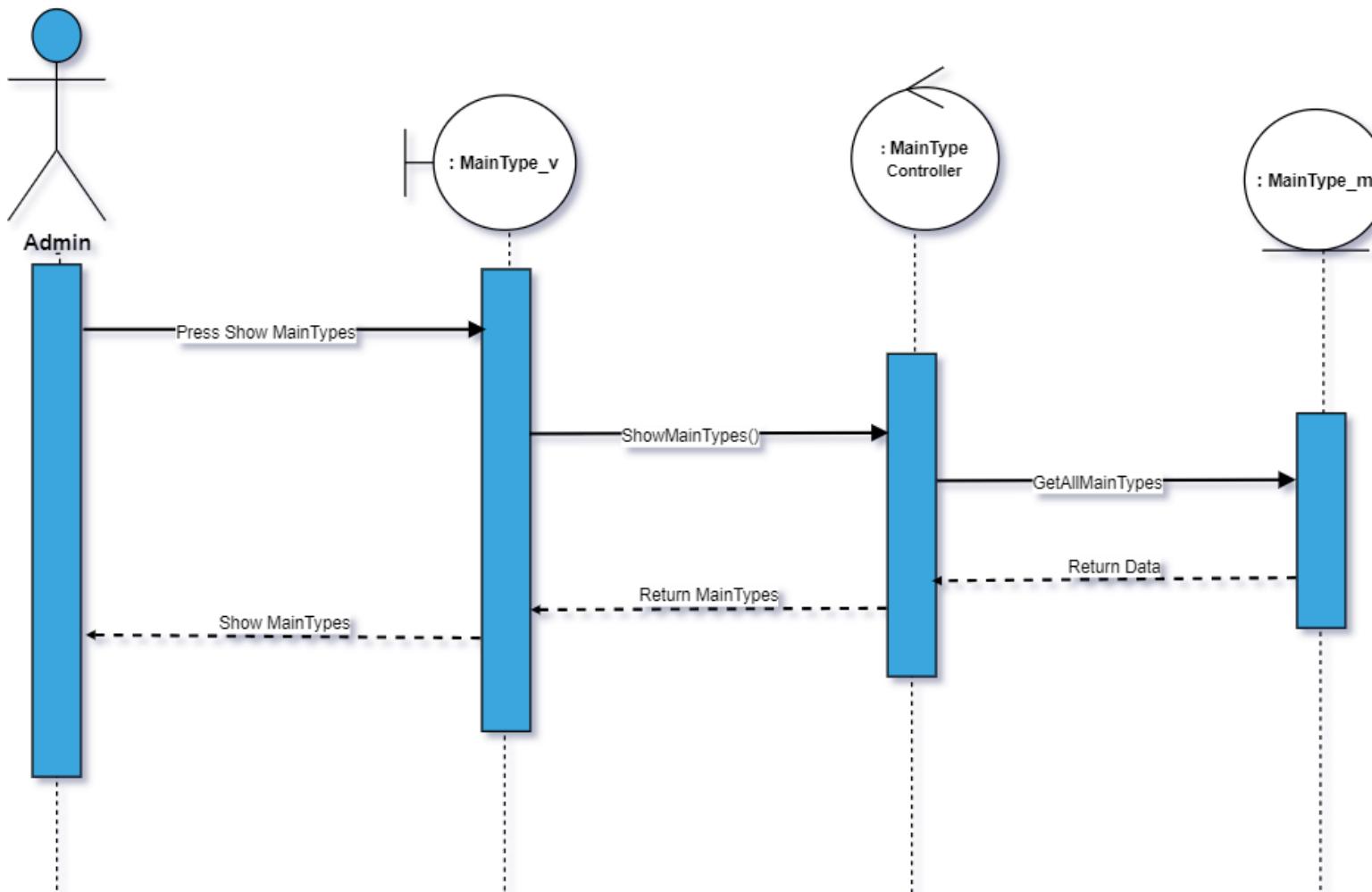
71)Delete main type



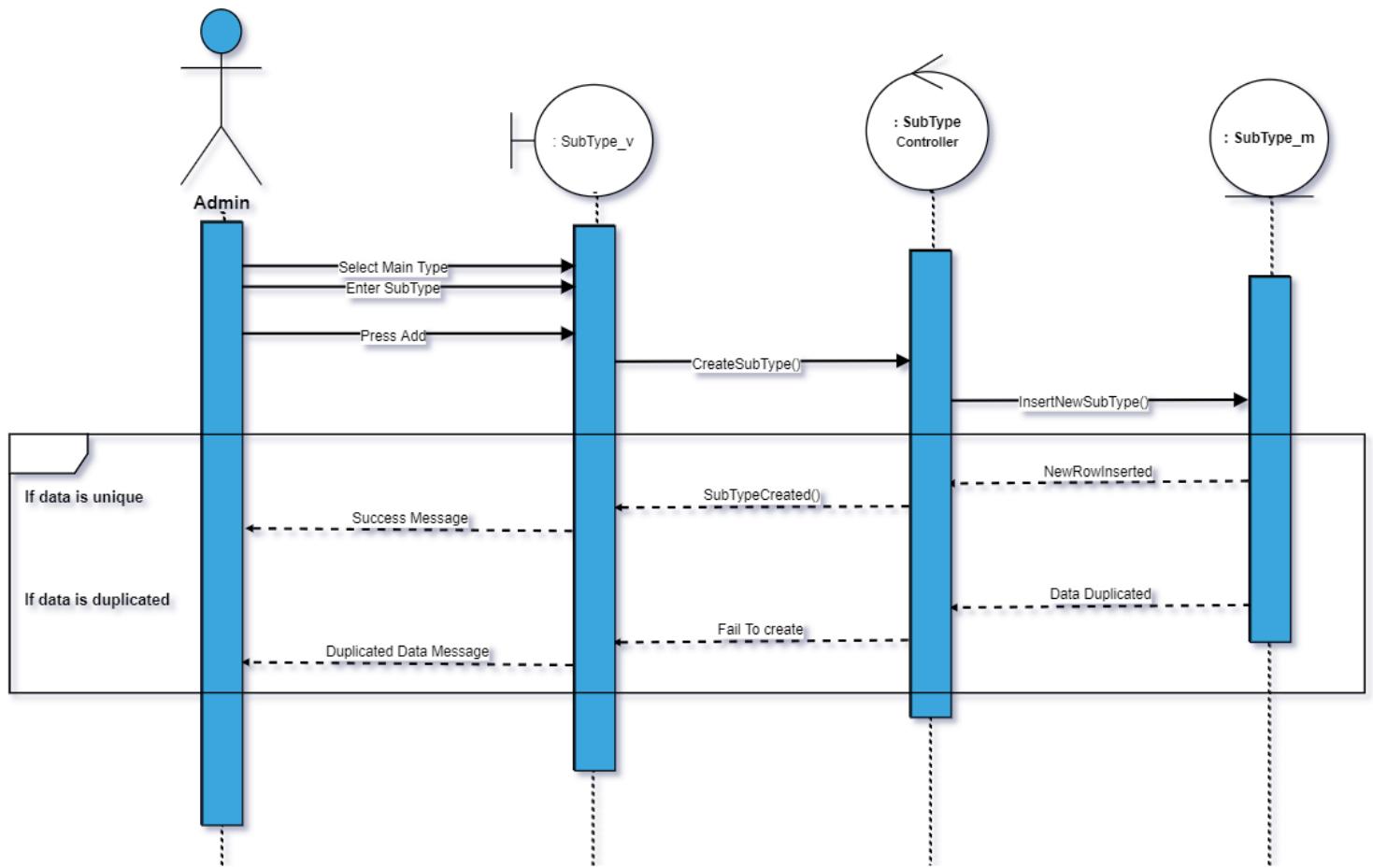
72)Edit main type



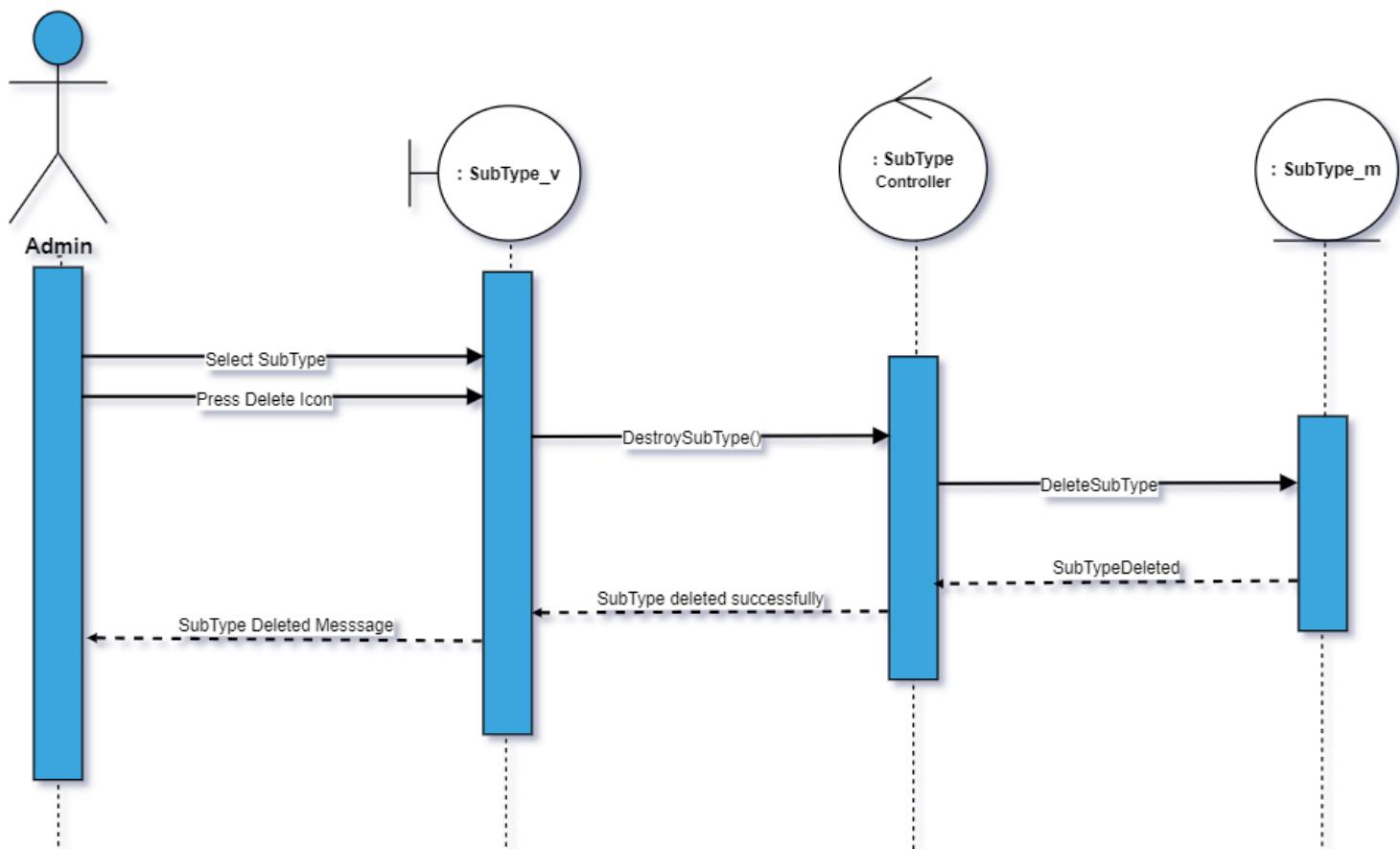
73) Show main type



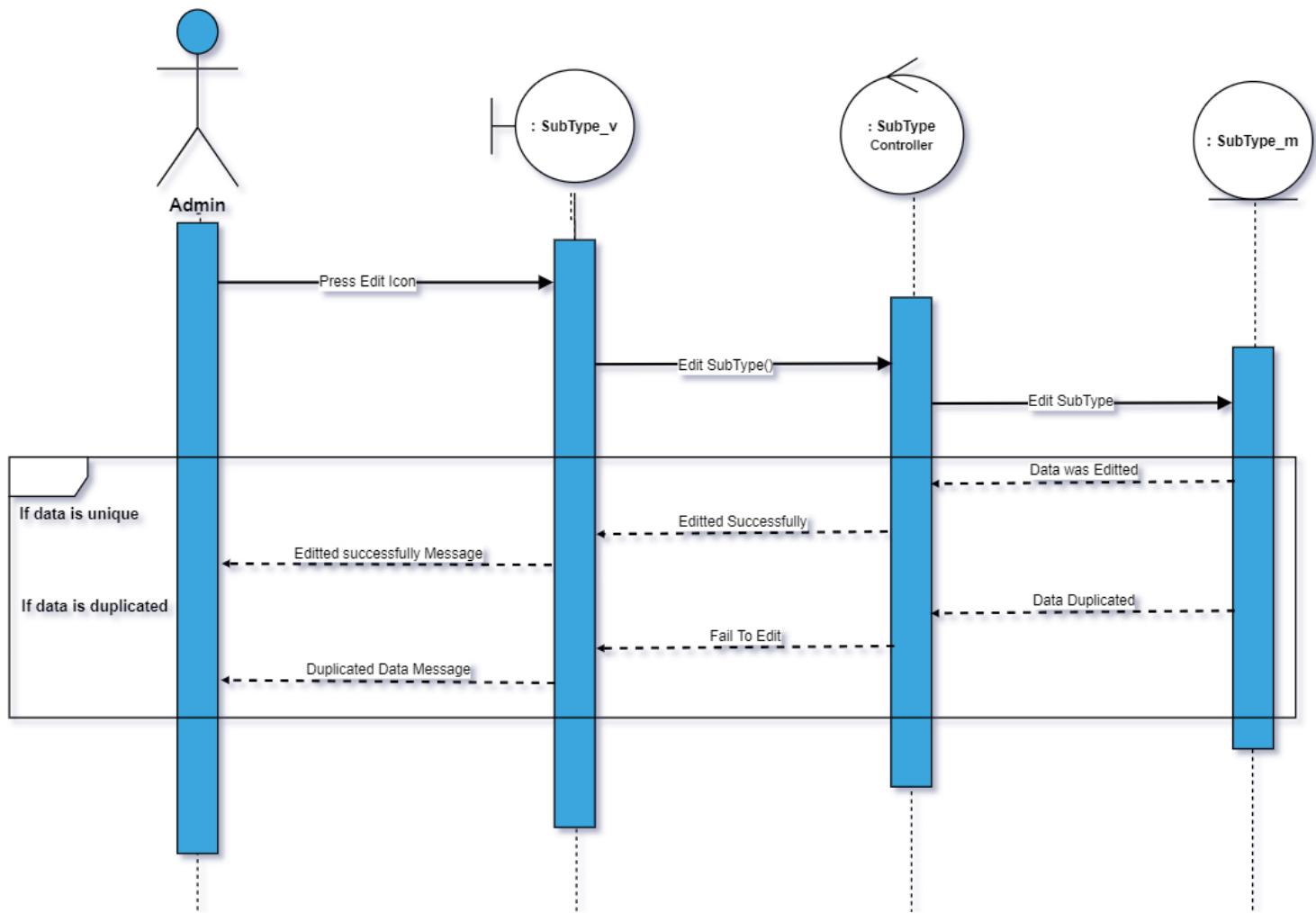
74)Add sub type



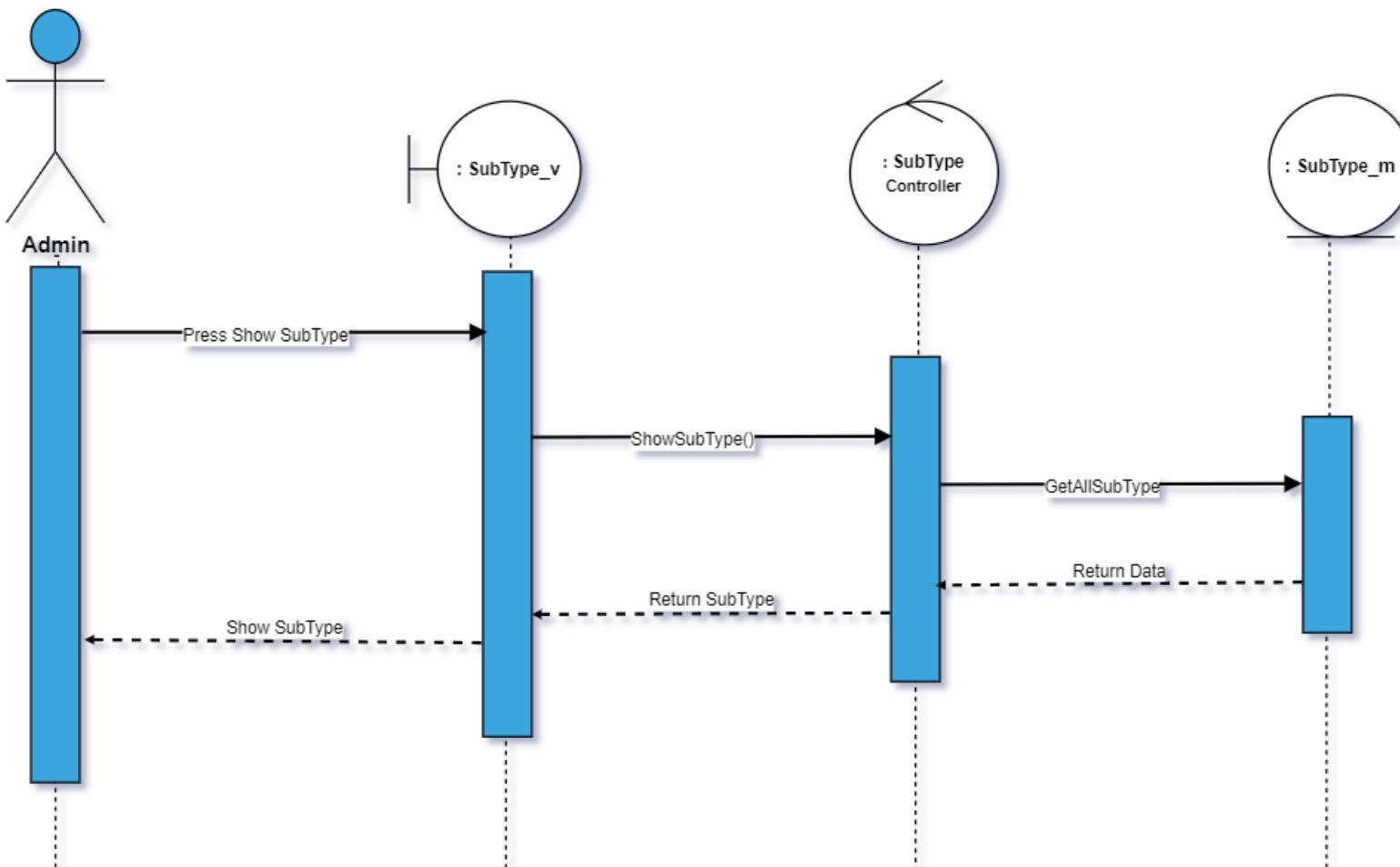
75) Delete Subtype



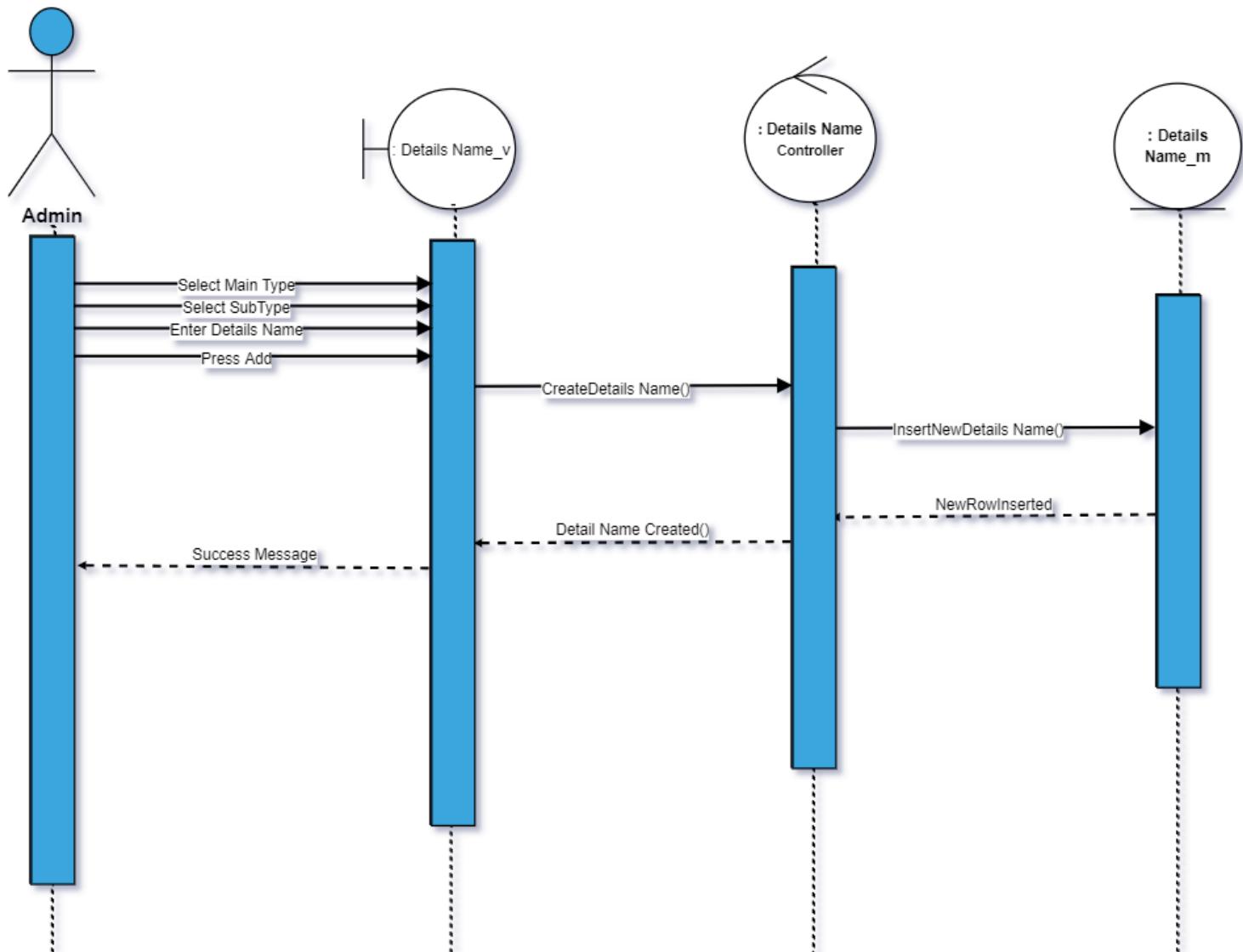
76)Edit subtype



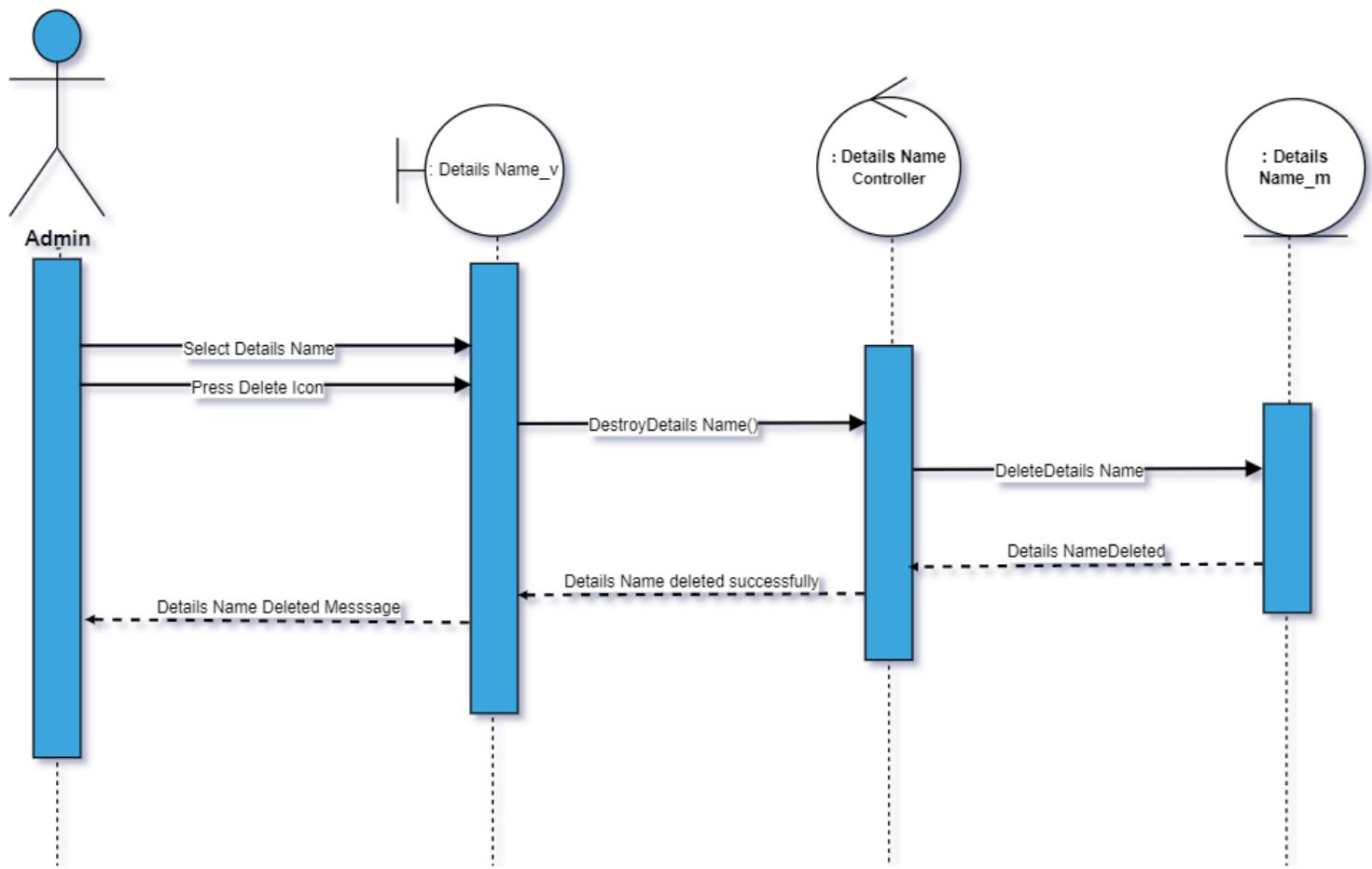
77) Show subtype



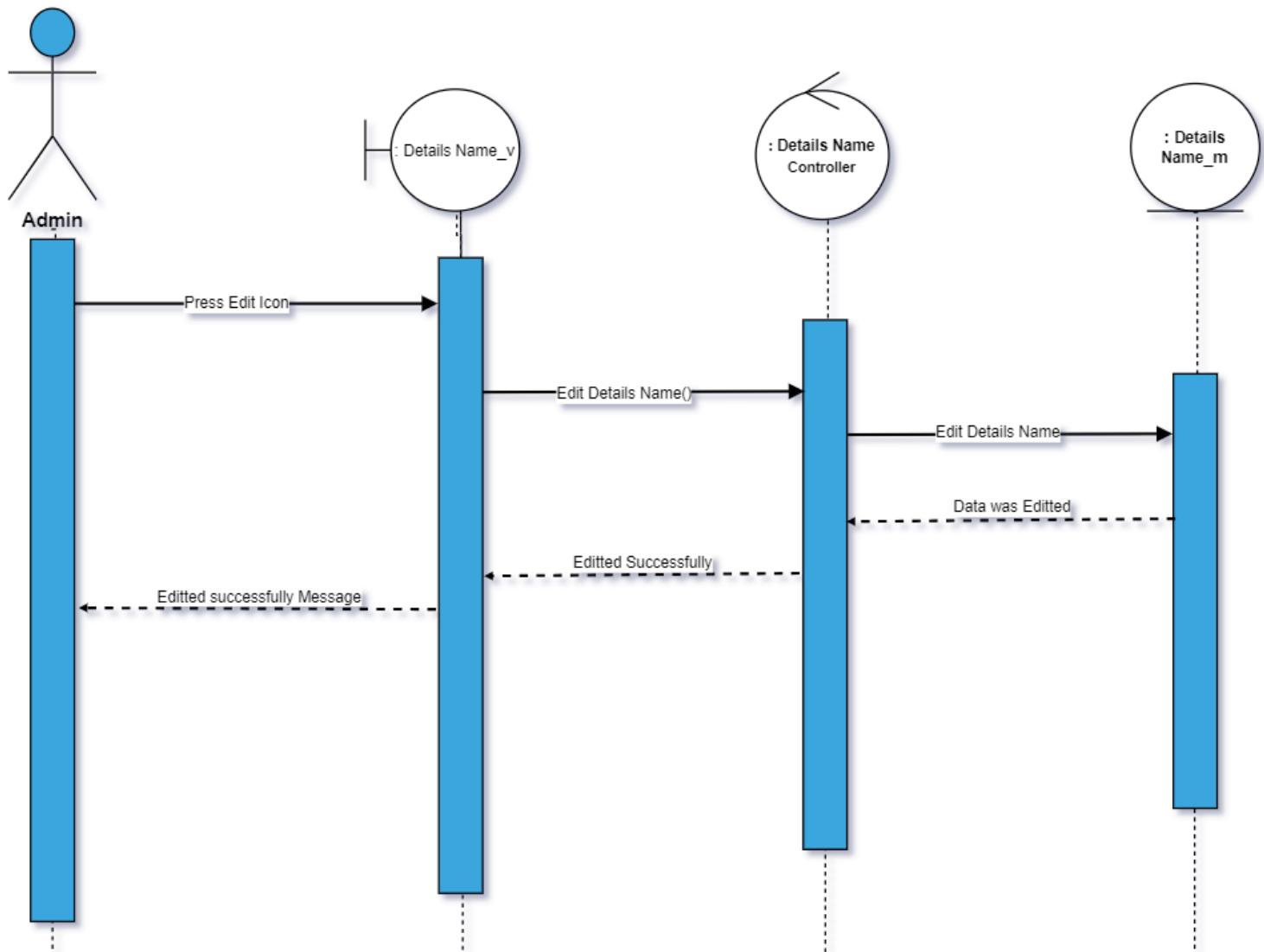
78)Add Detail Name



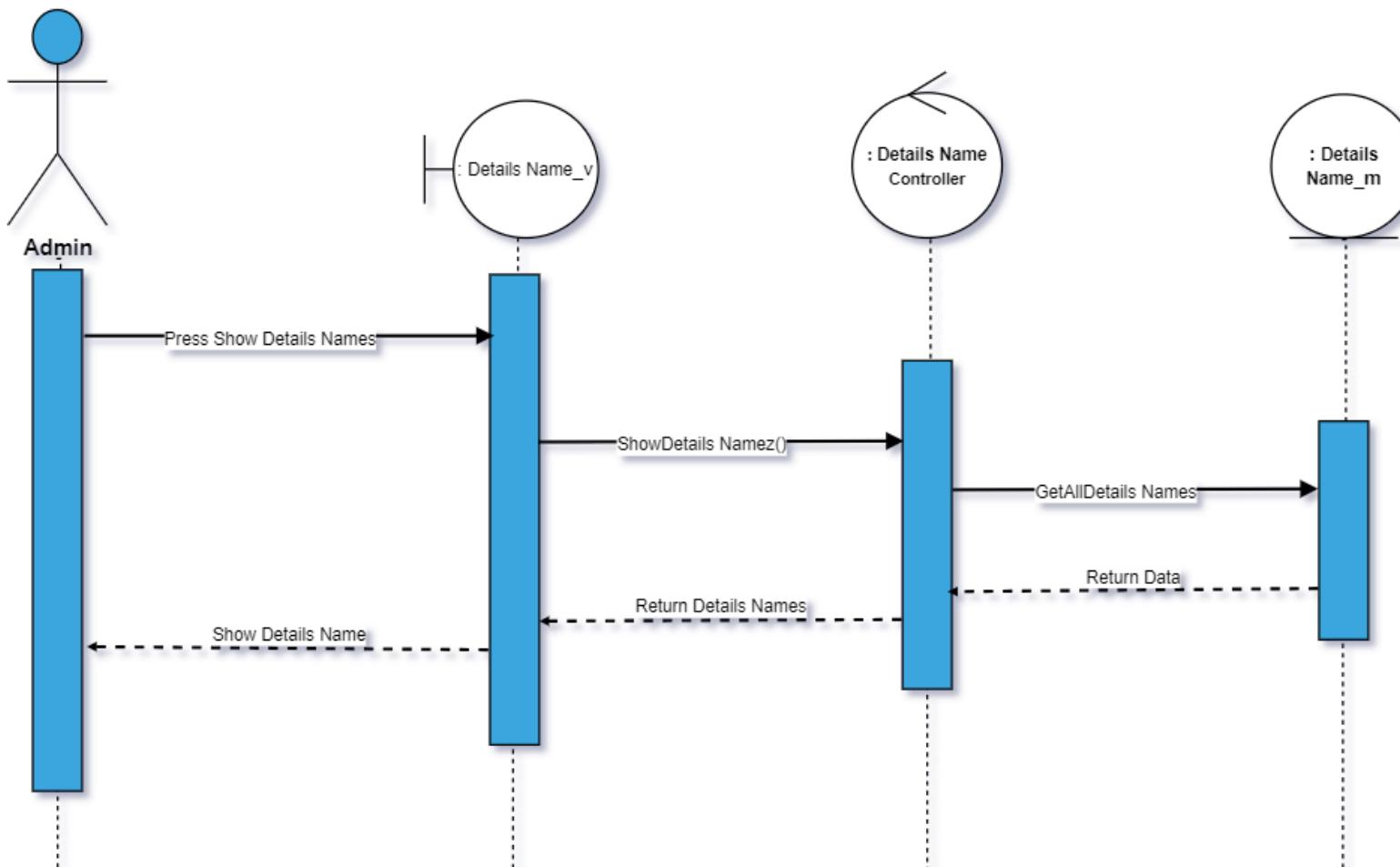
79)Delete Detail Name



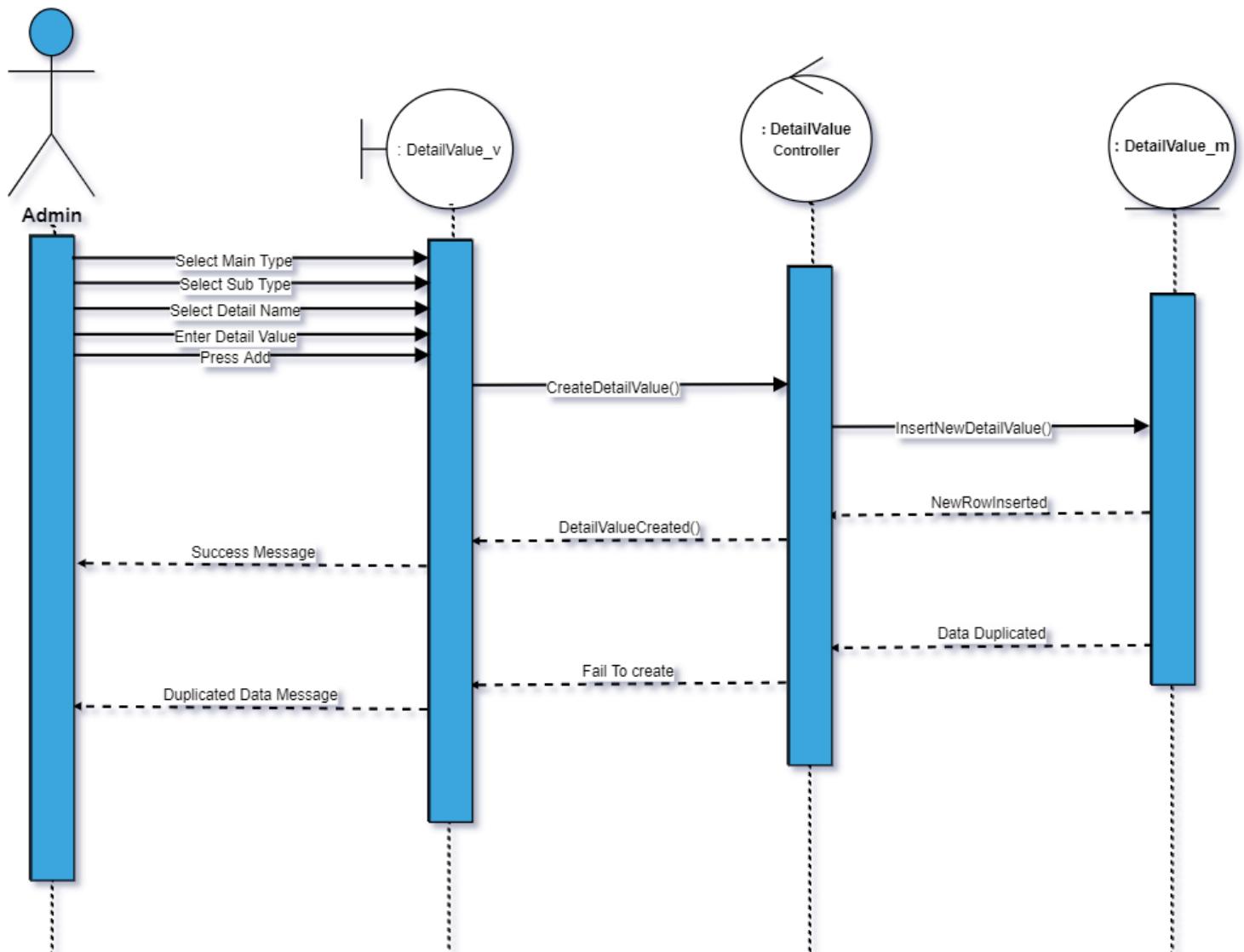
80)Edit Detail Name



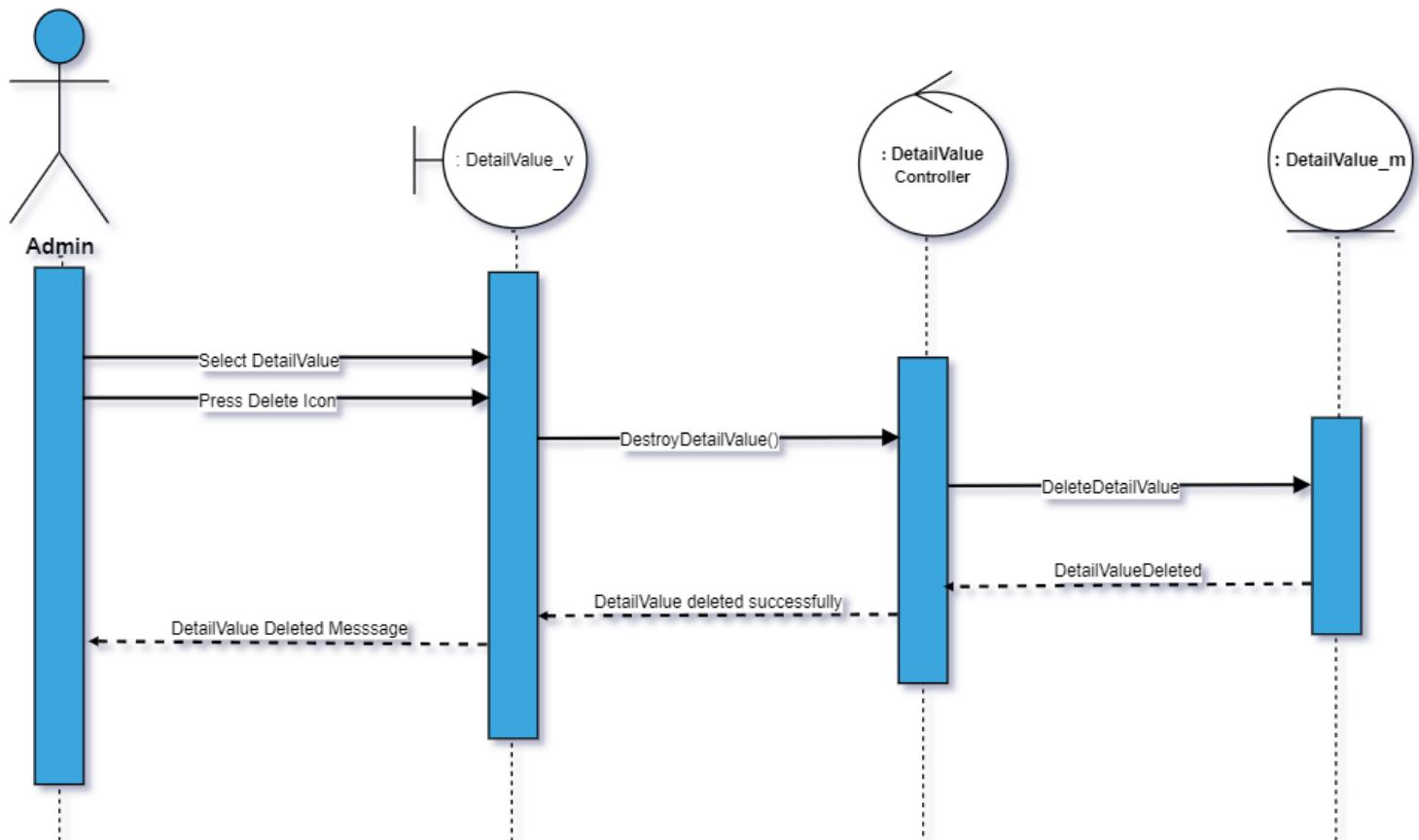
81) Show Detail Name



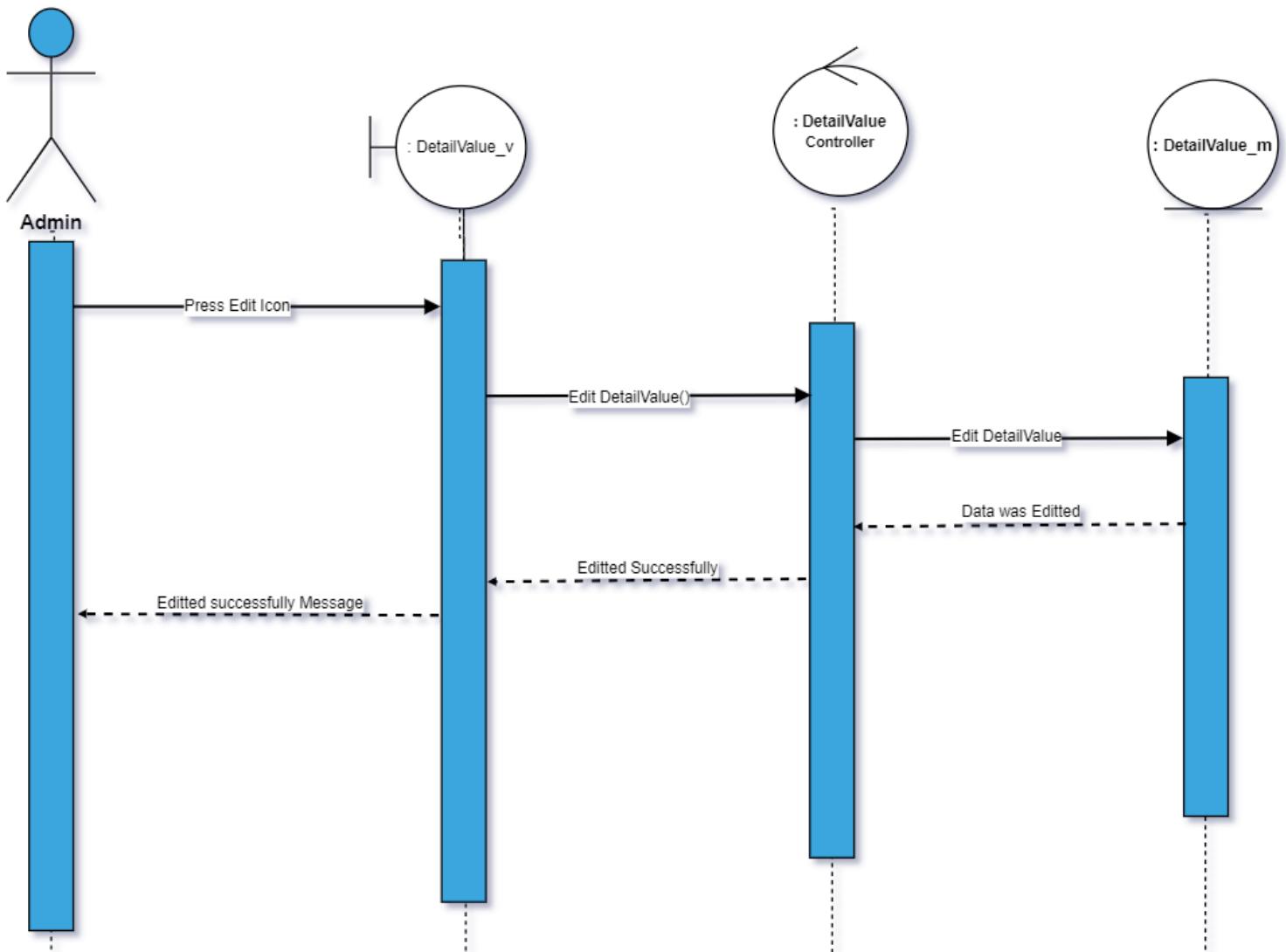
82)Add detail value



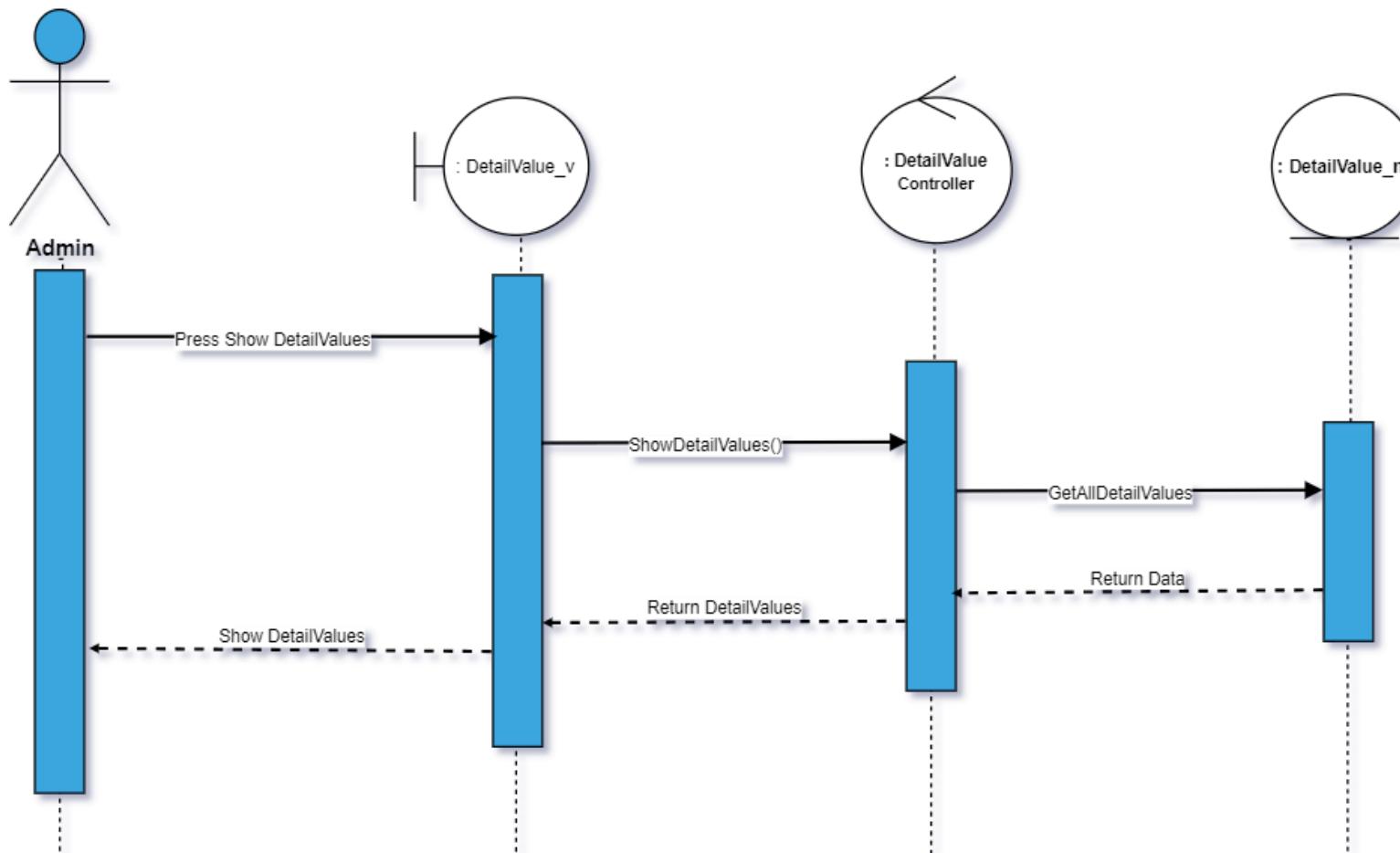
83)Delete Detail Value



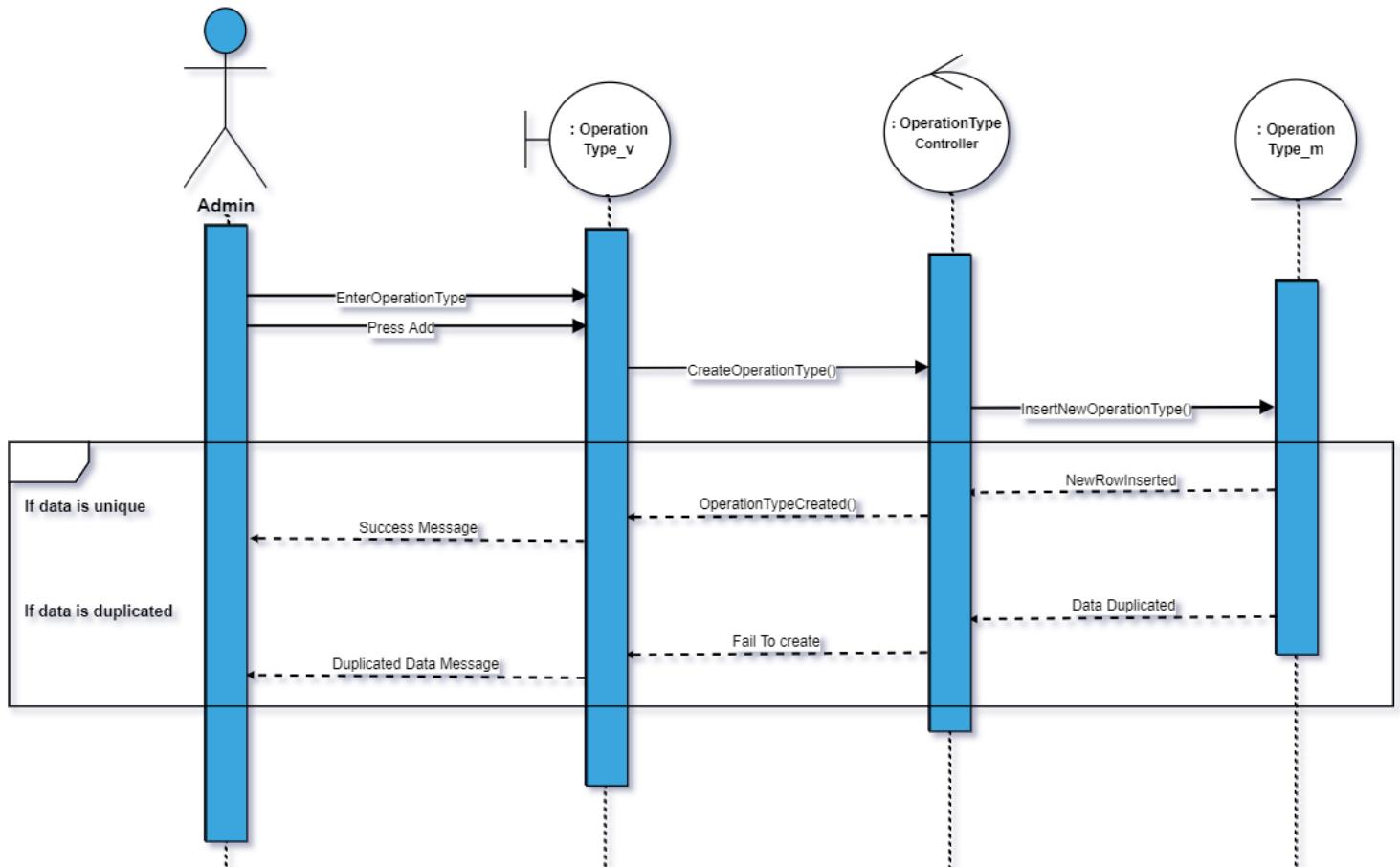
84)Edit Detail Value



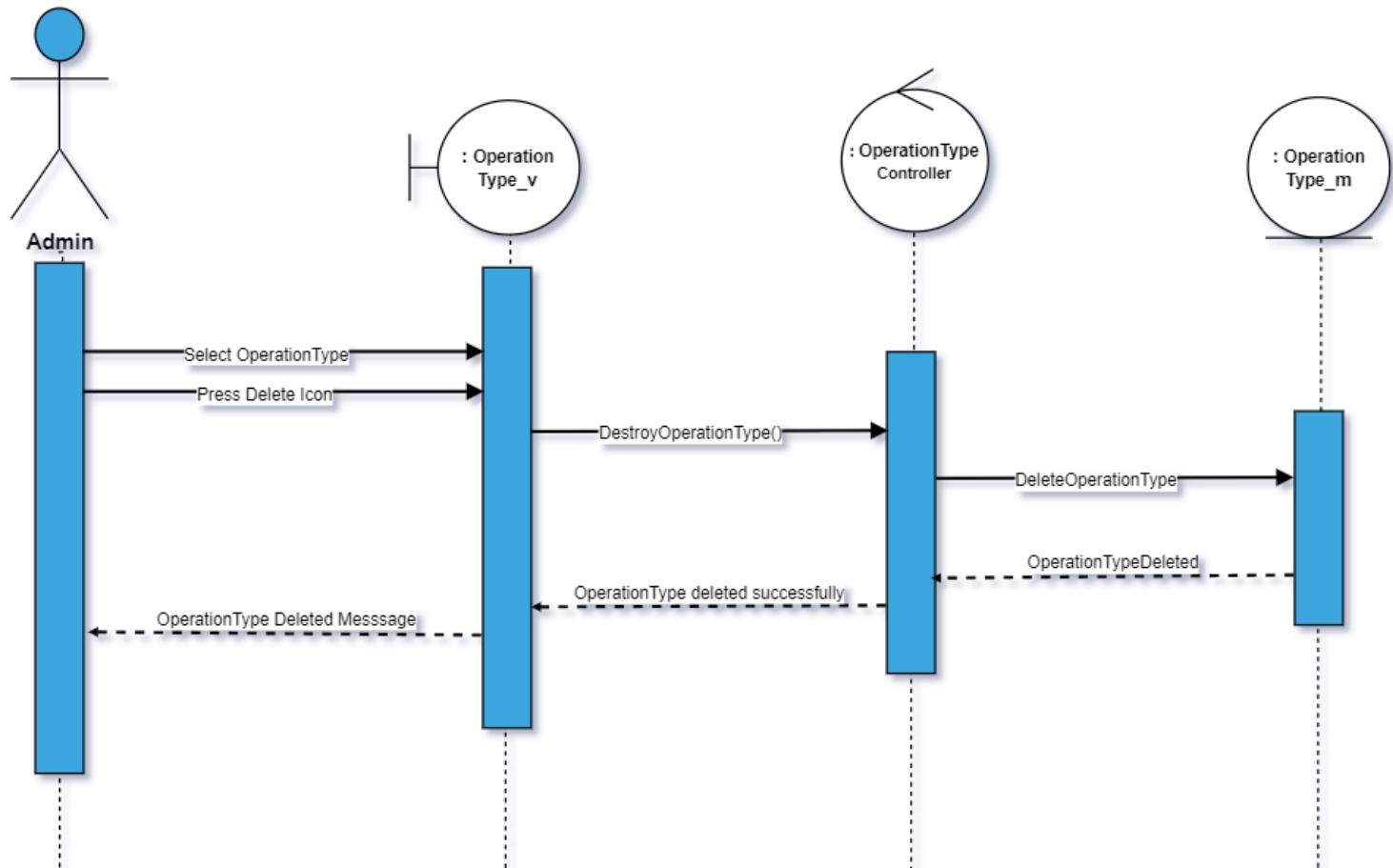
85) Show Detail Value



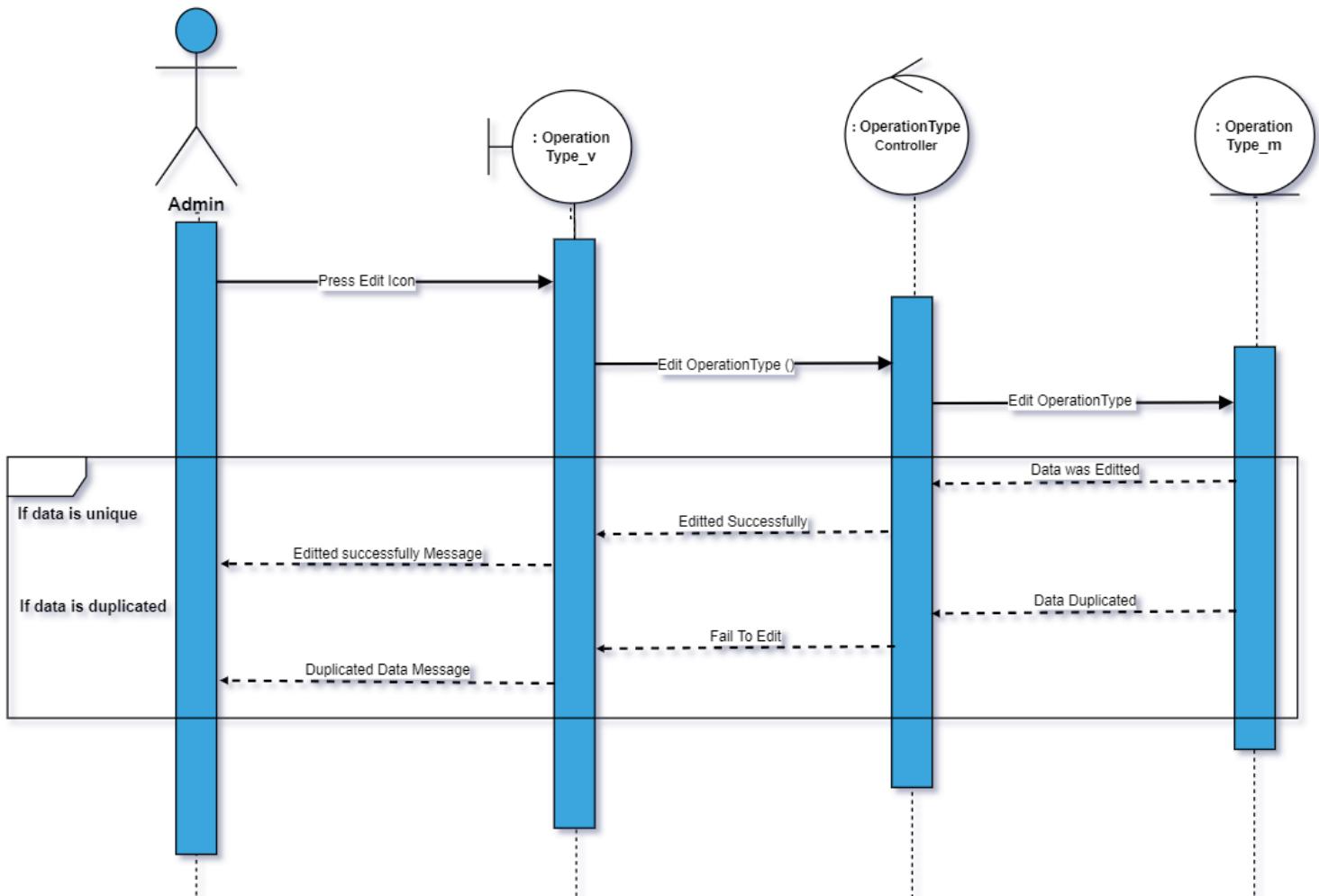
86) Add Operation Type



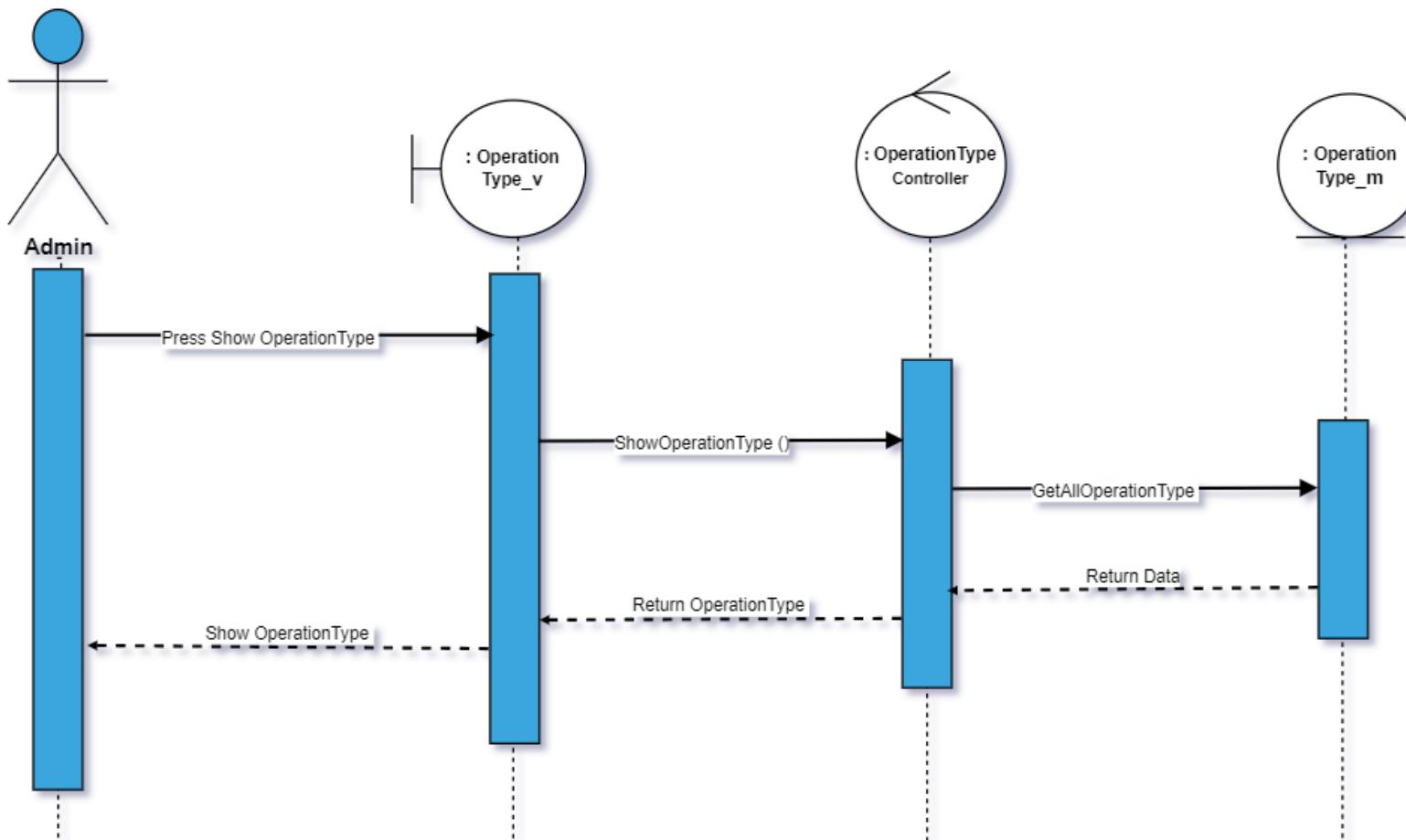
87)Delete Operation type



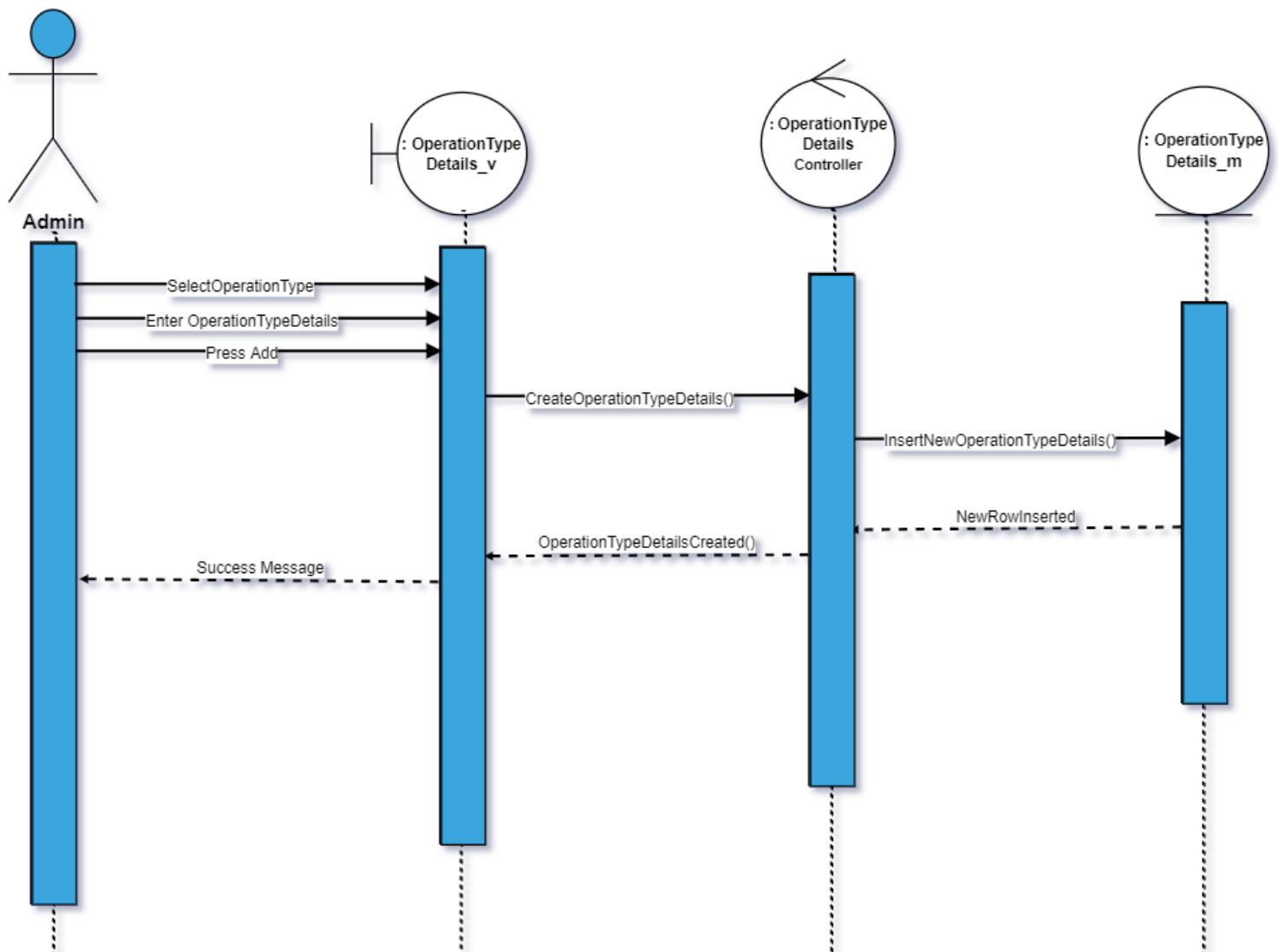
88)Edit operation type



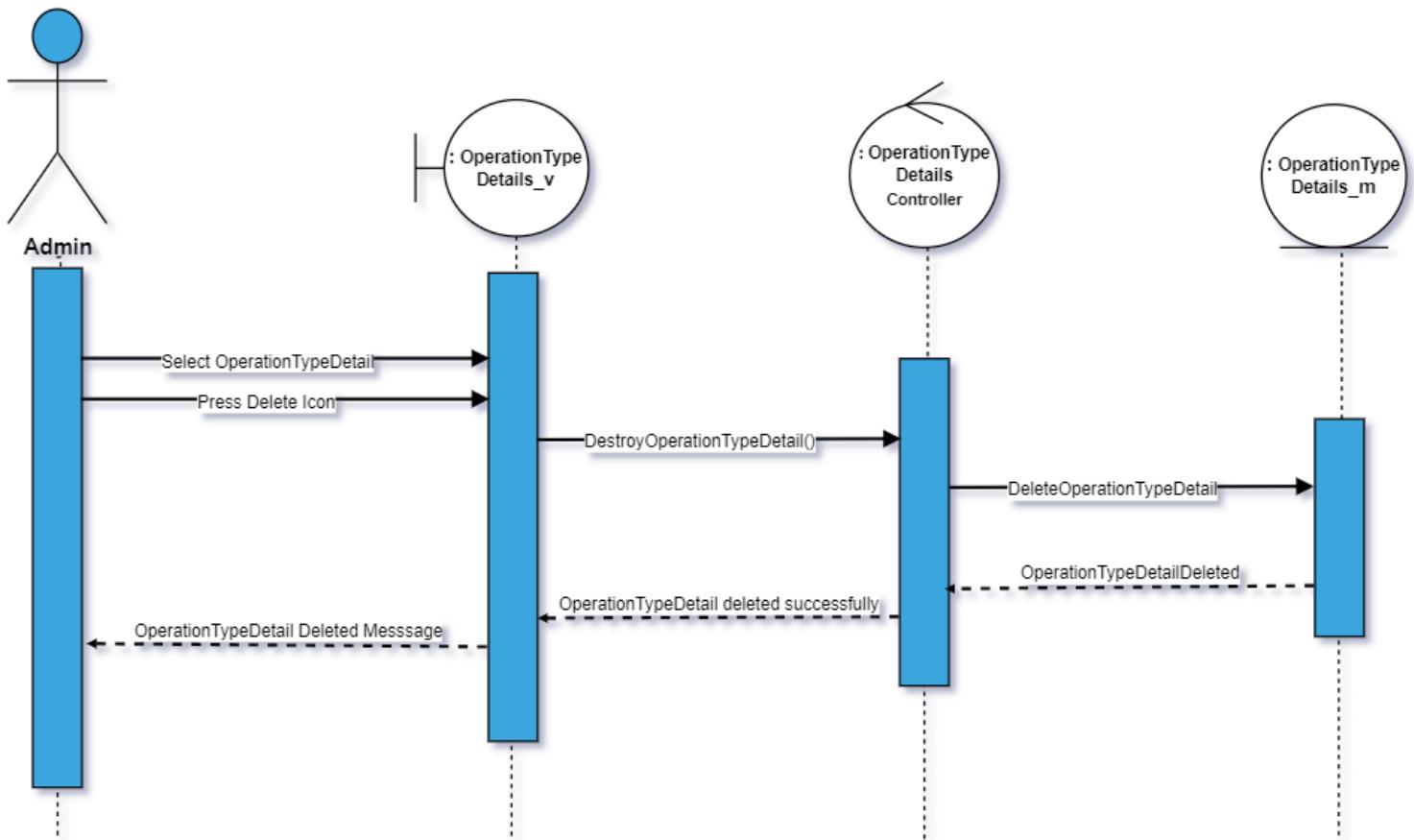
89) Show operation type



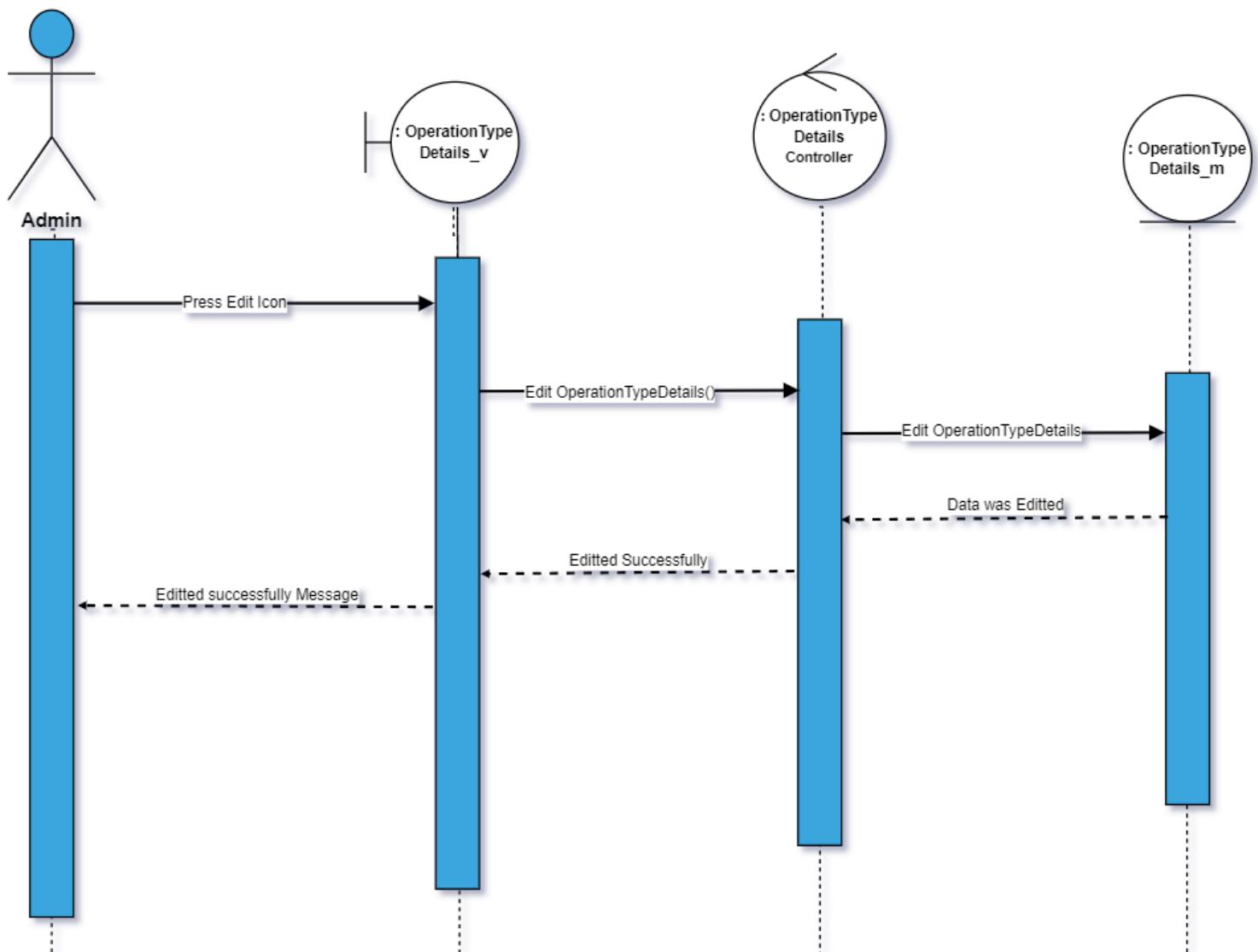
90)Create operation type Details



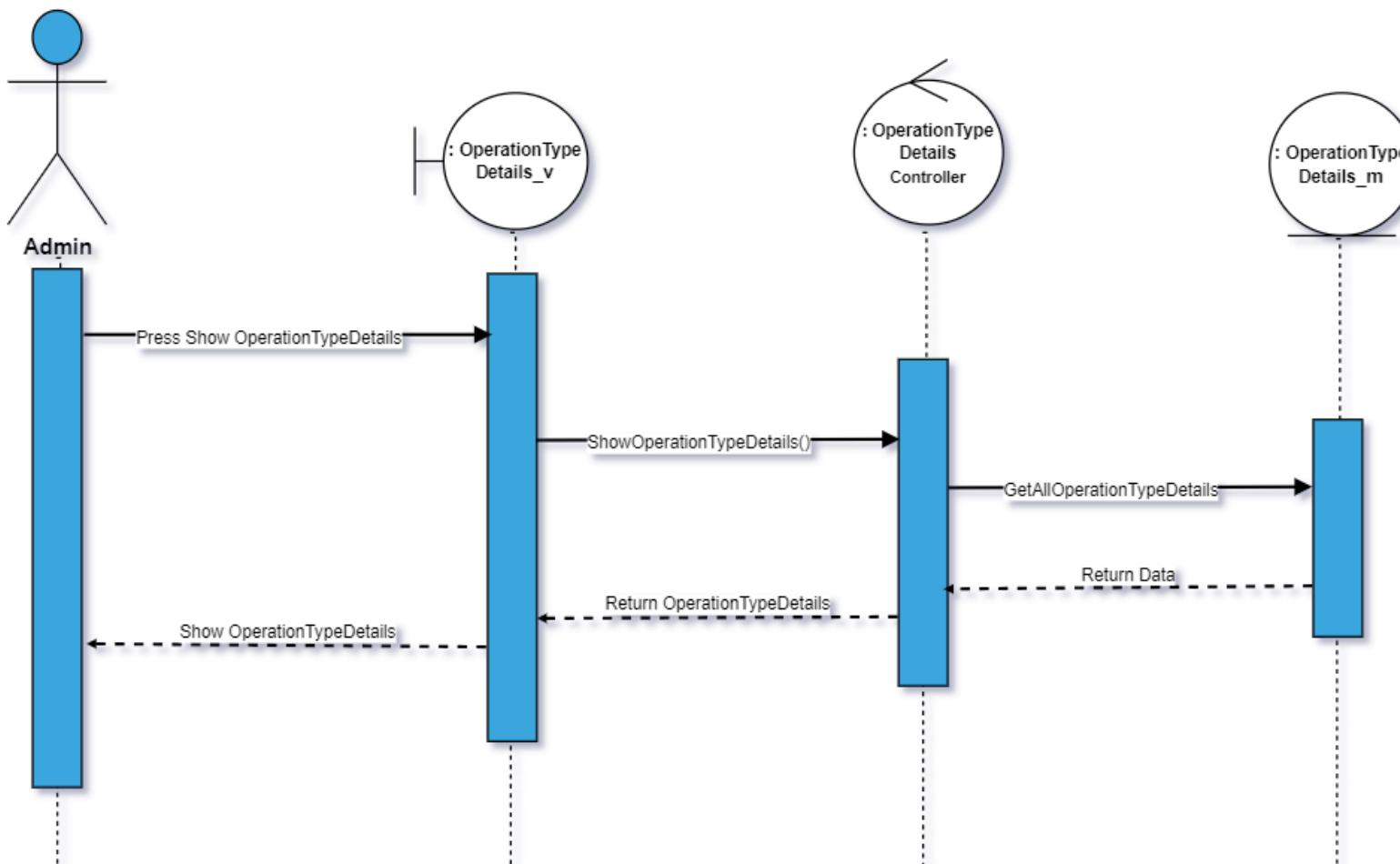
91)Delete operation type details



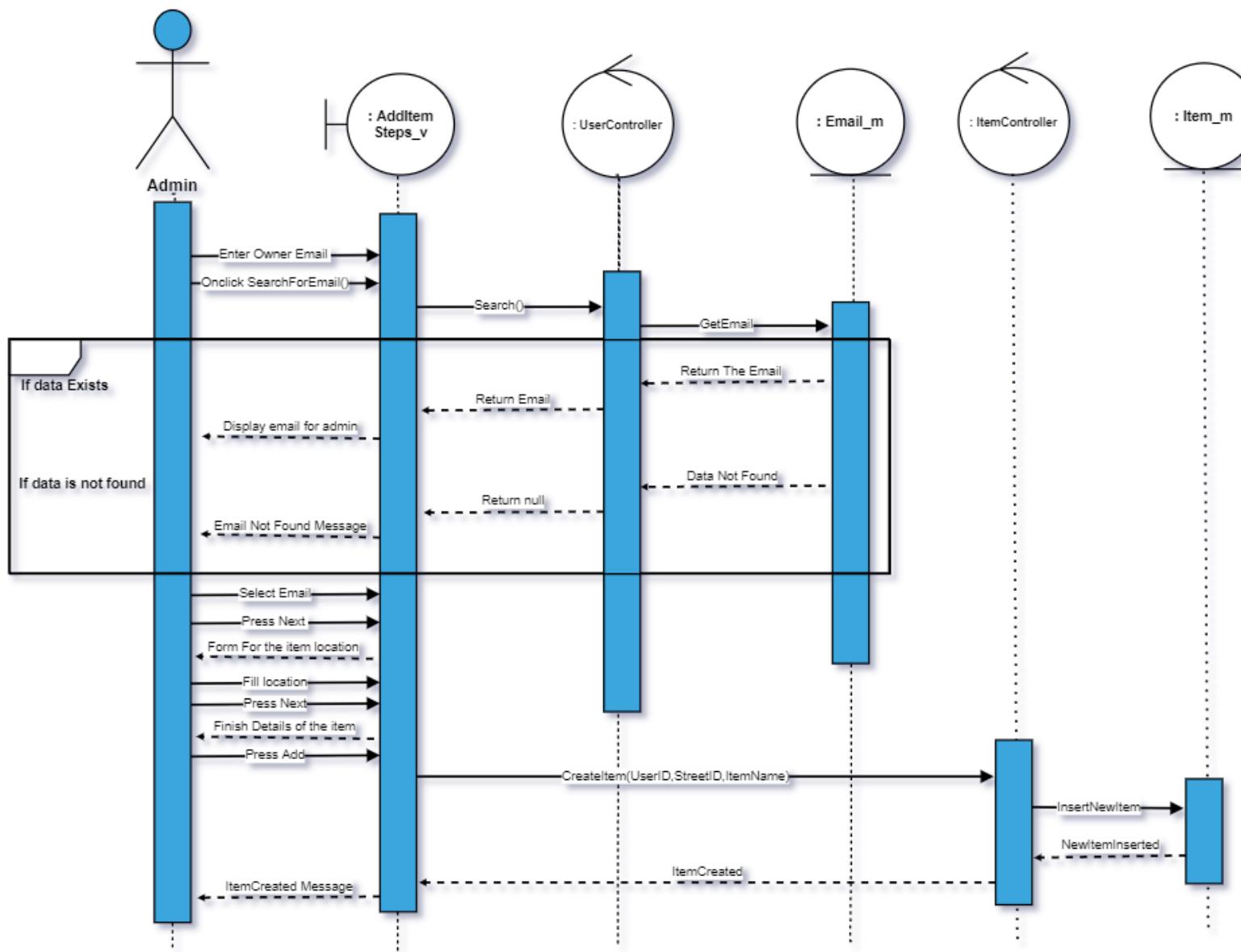
92)Edit Operation type details



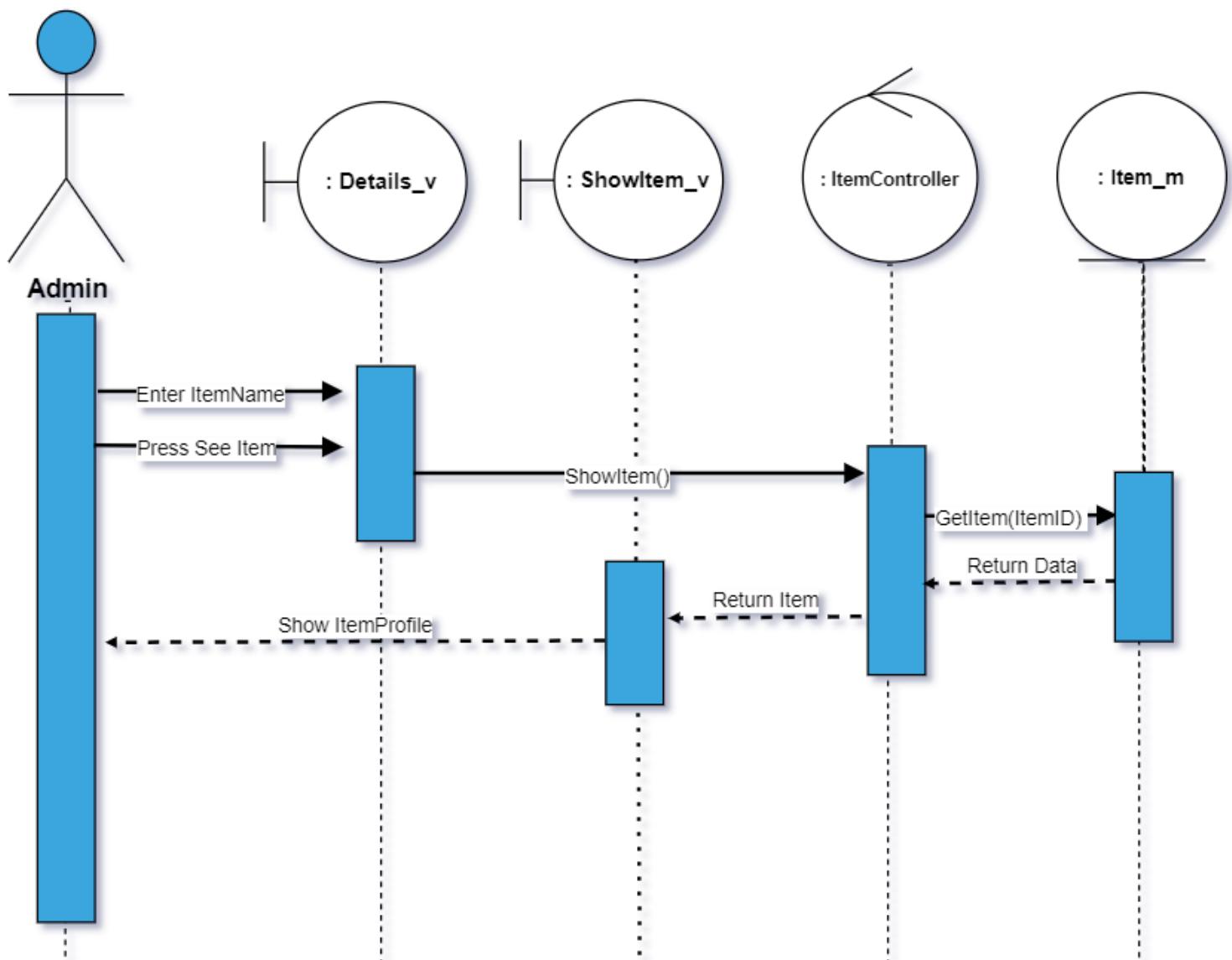
93) Show operation type details



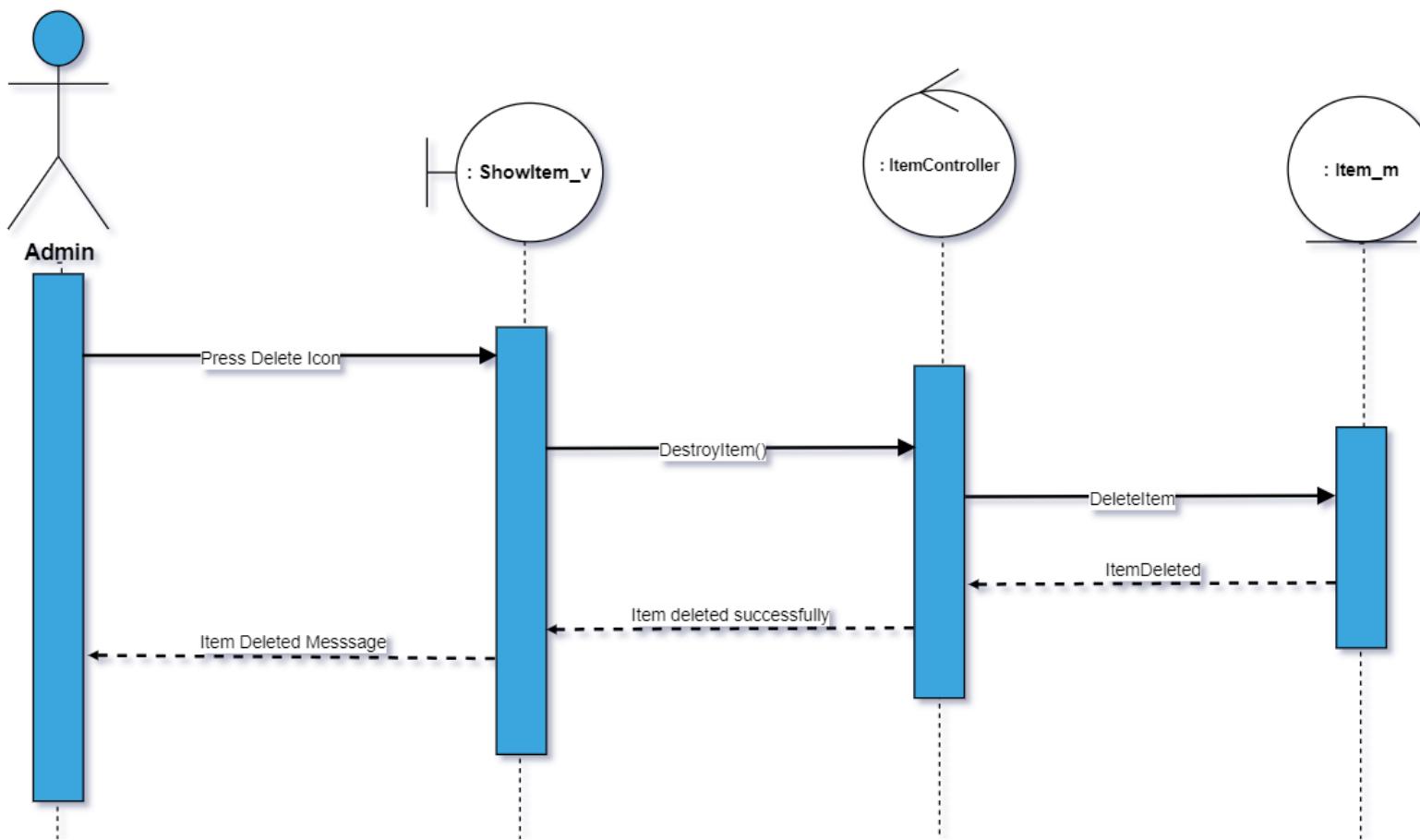
94) Add item



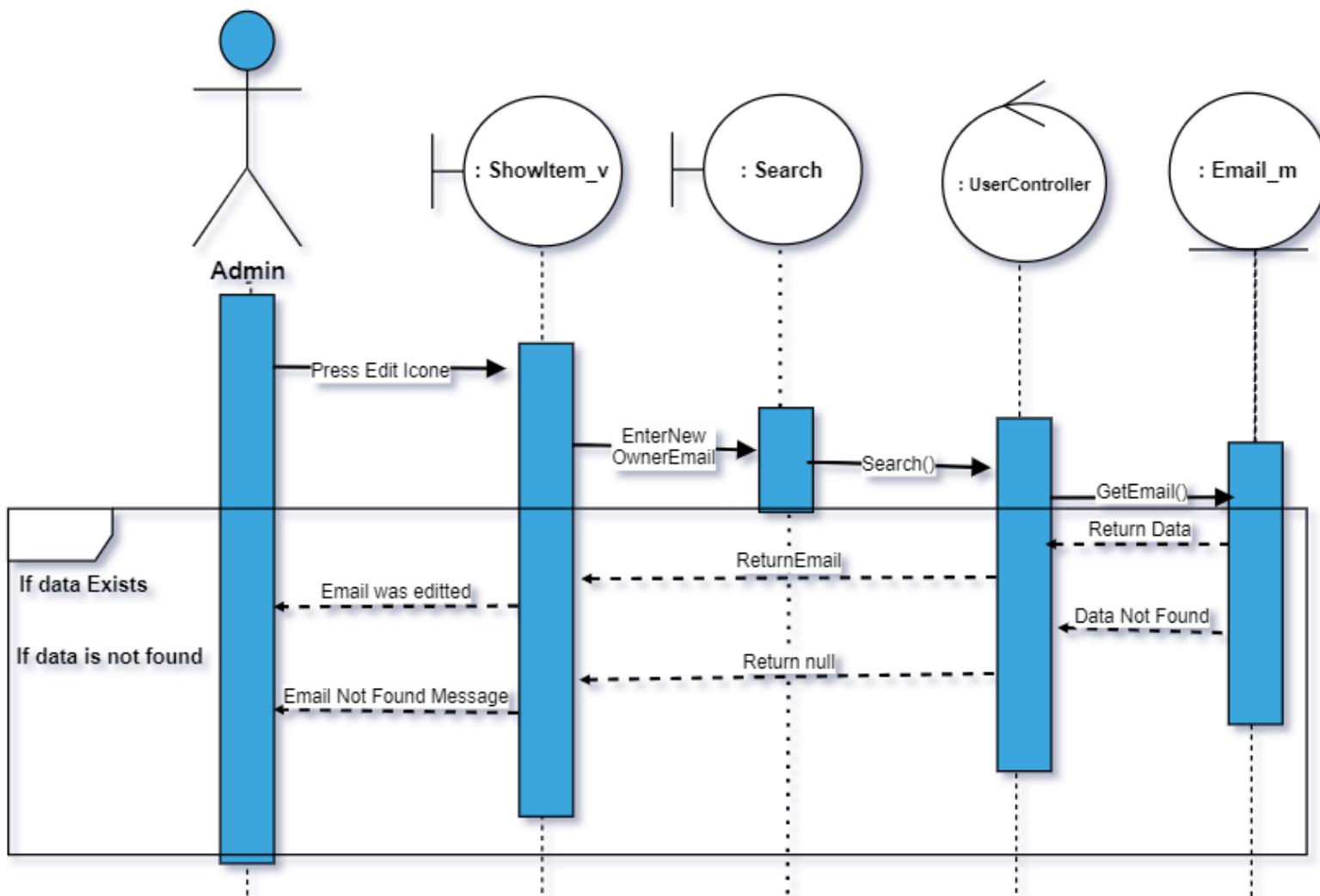
95) show item profile



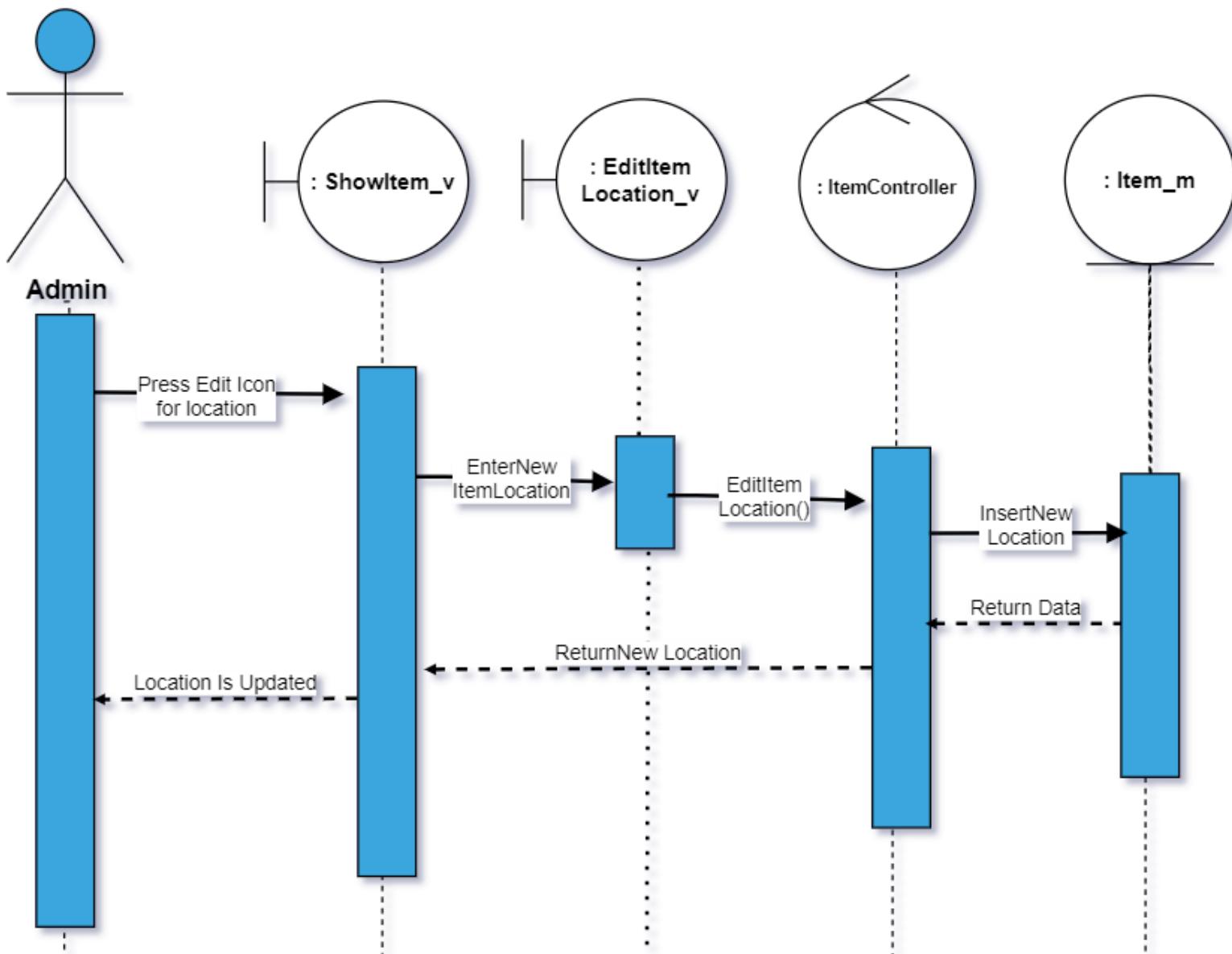
96)Delete Item



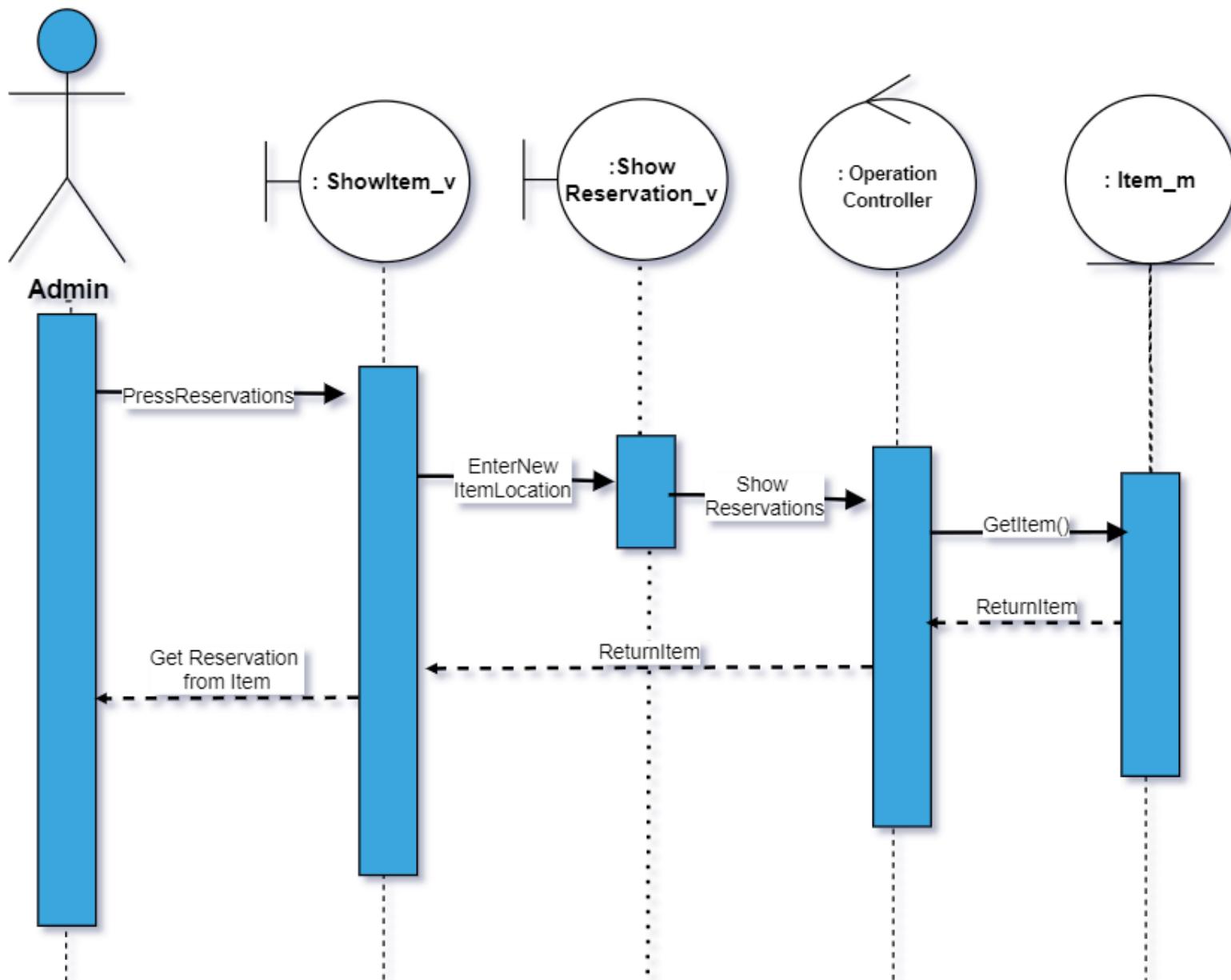
97)Edit owners email



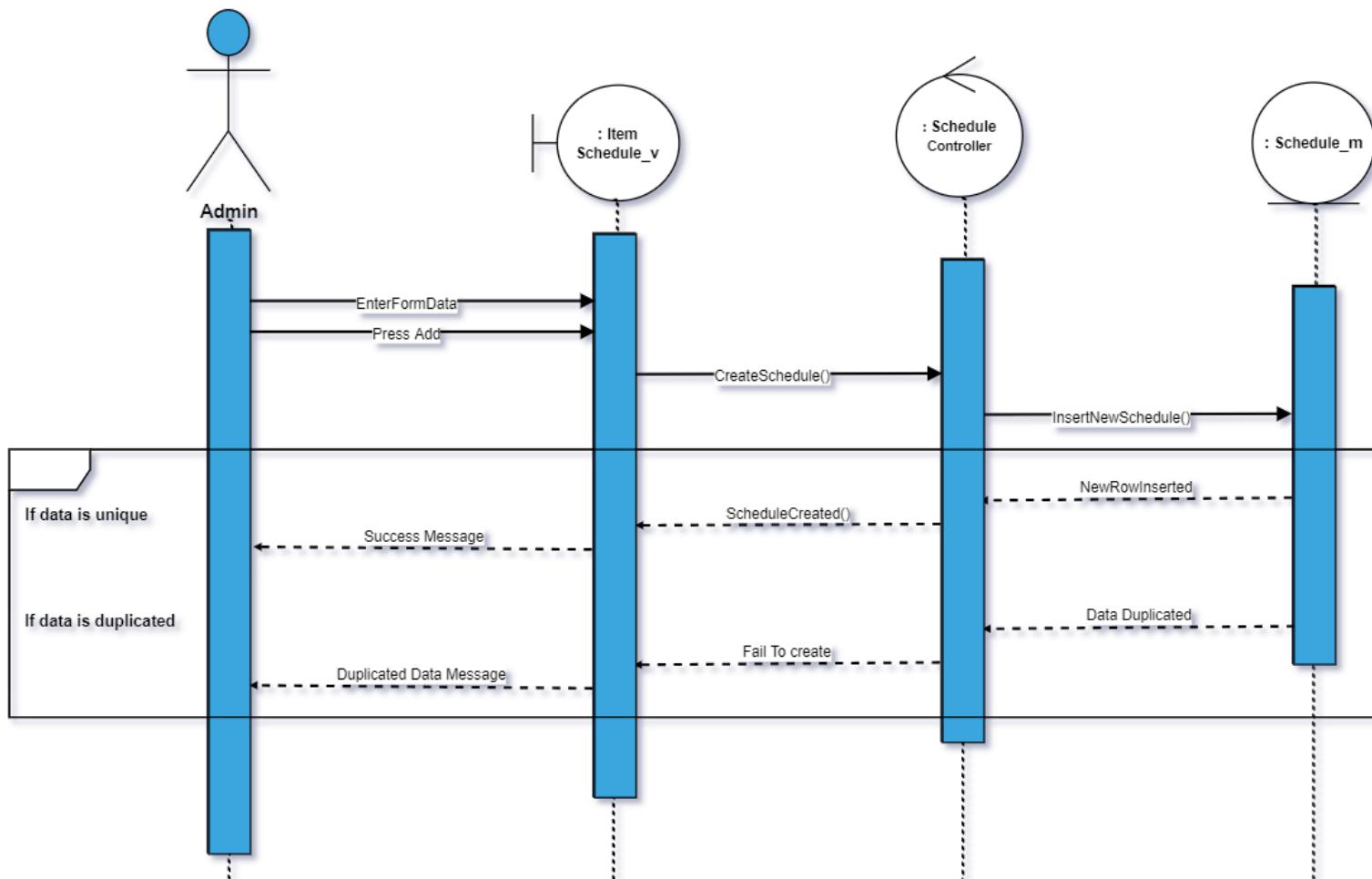
98)Edit Location



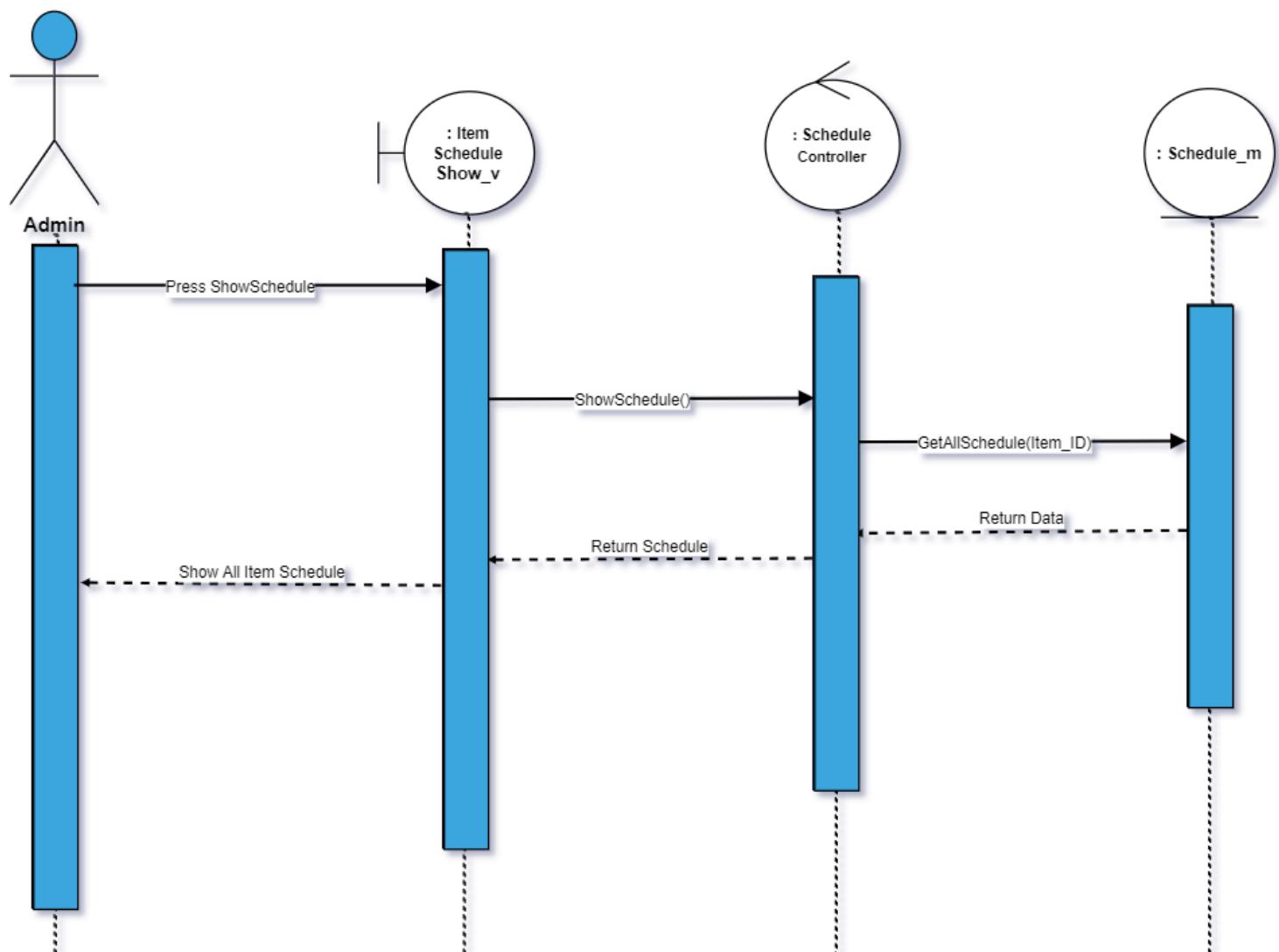
99)Resve



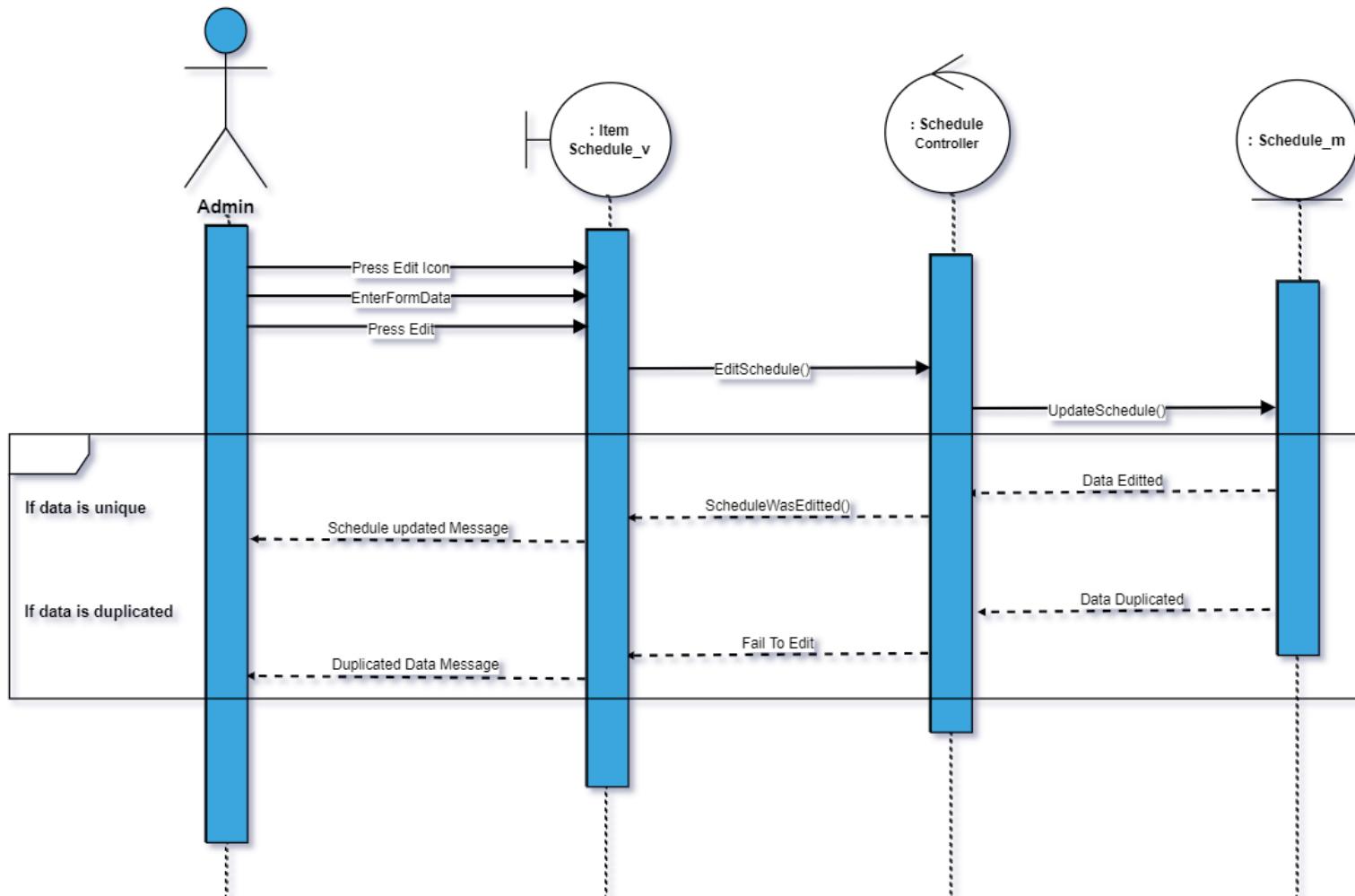
100) Schedule an item



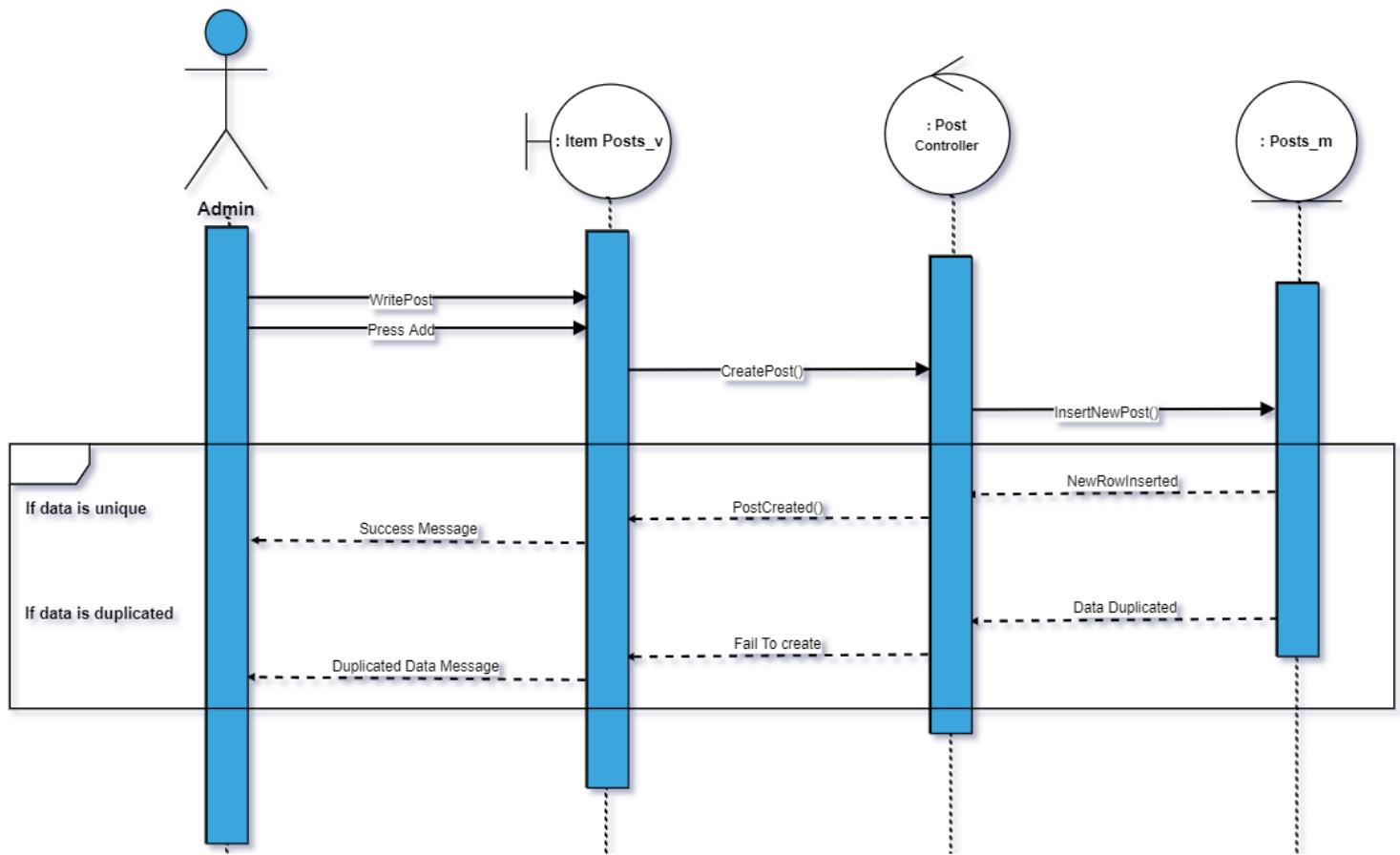
101) show item schedule



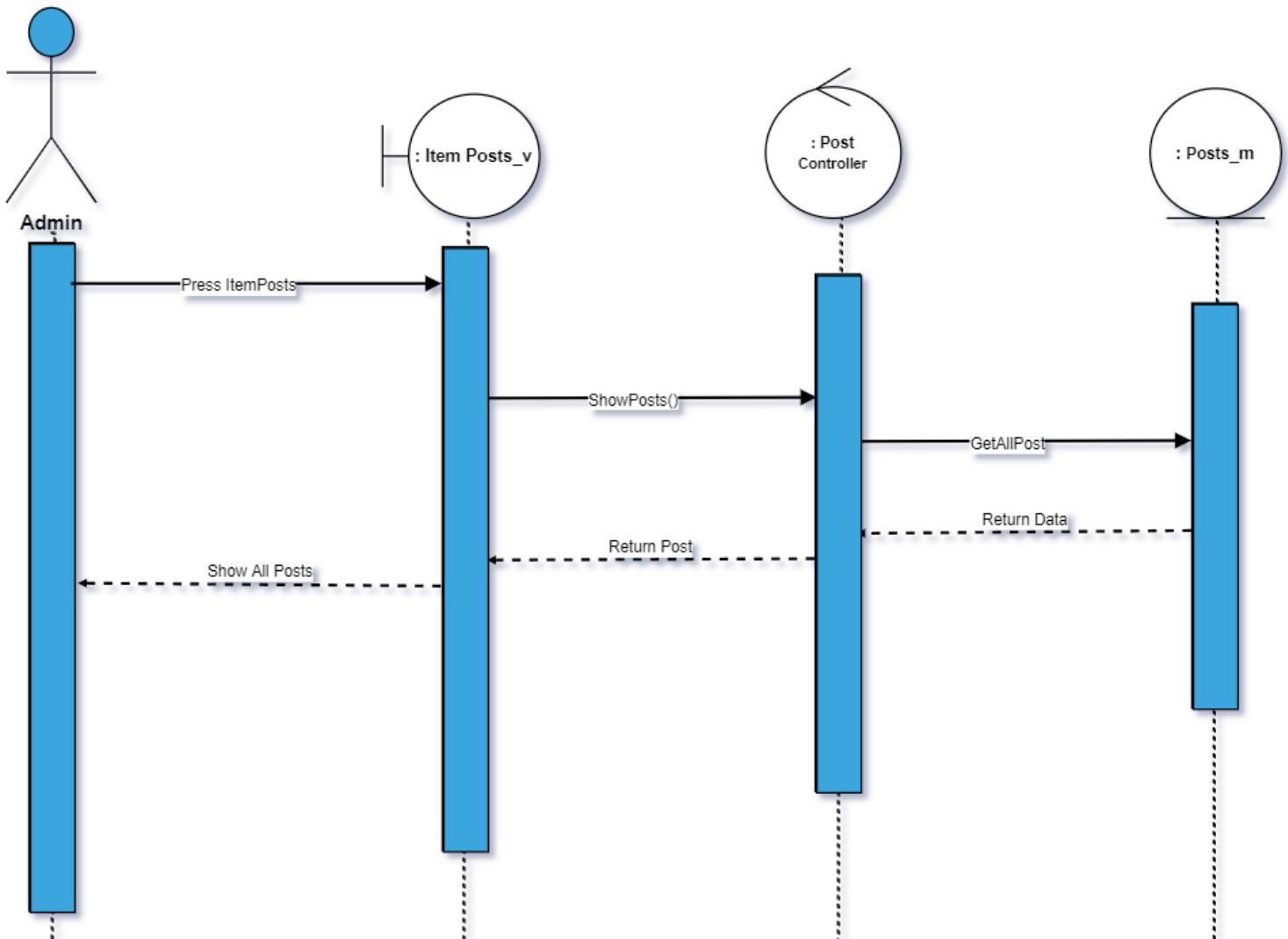
102)Edit schedule



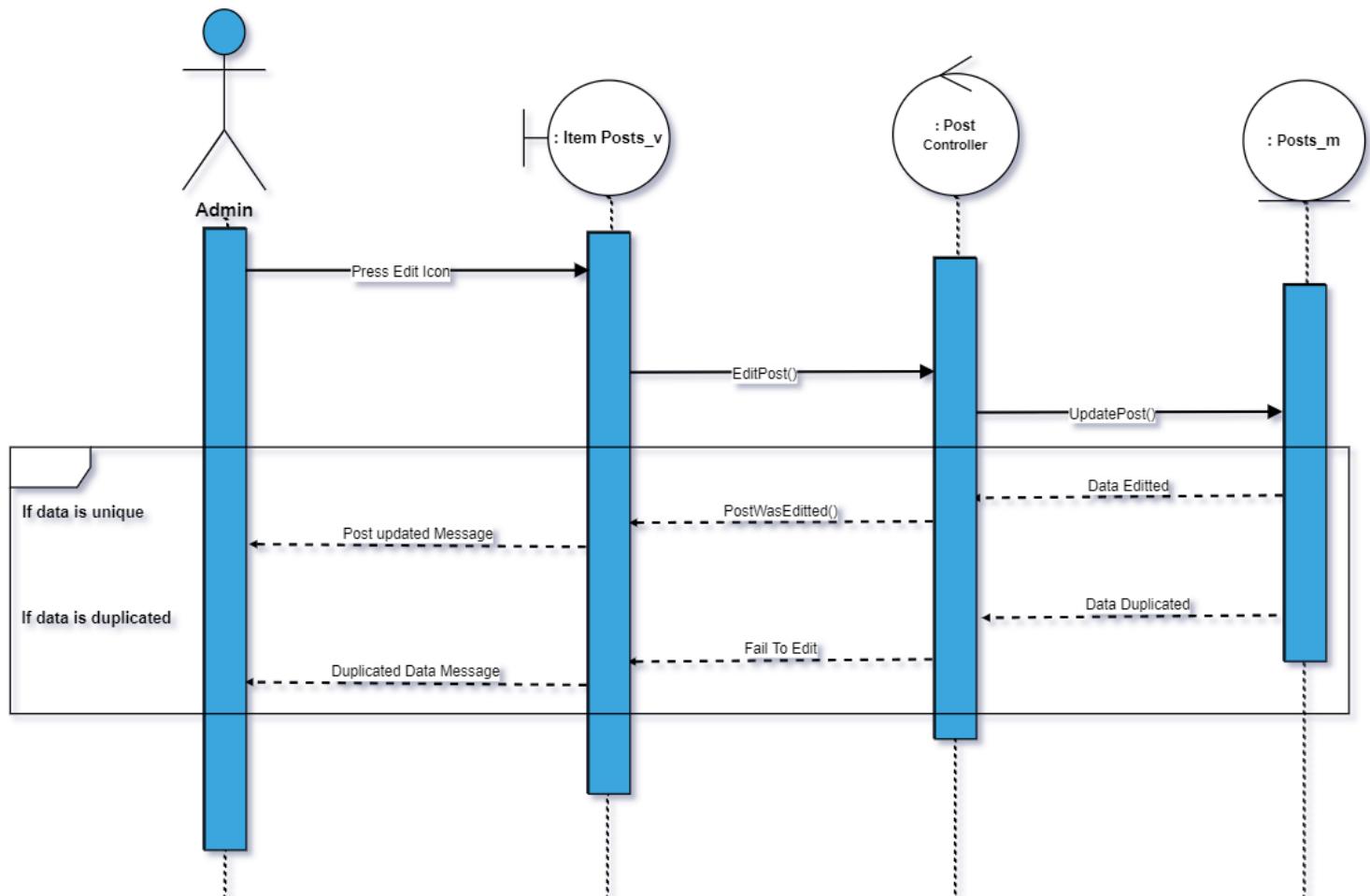
103) post a post



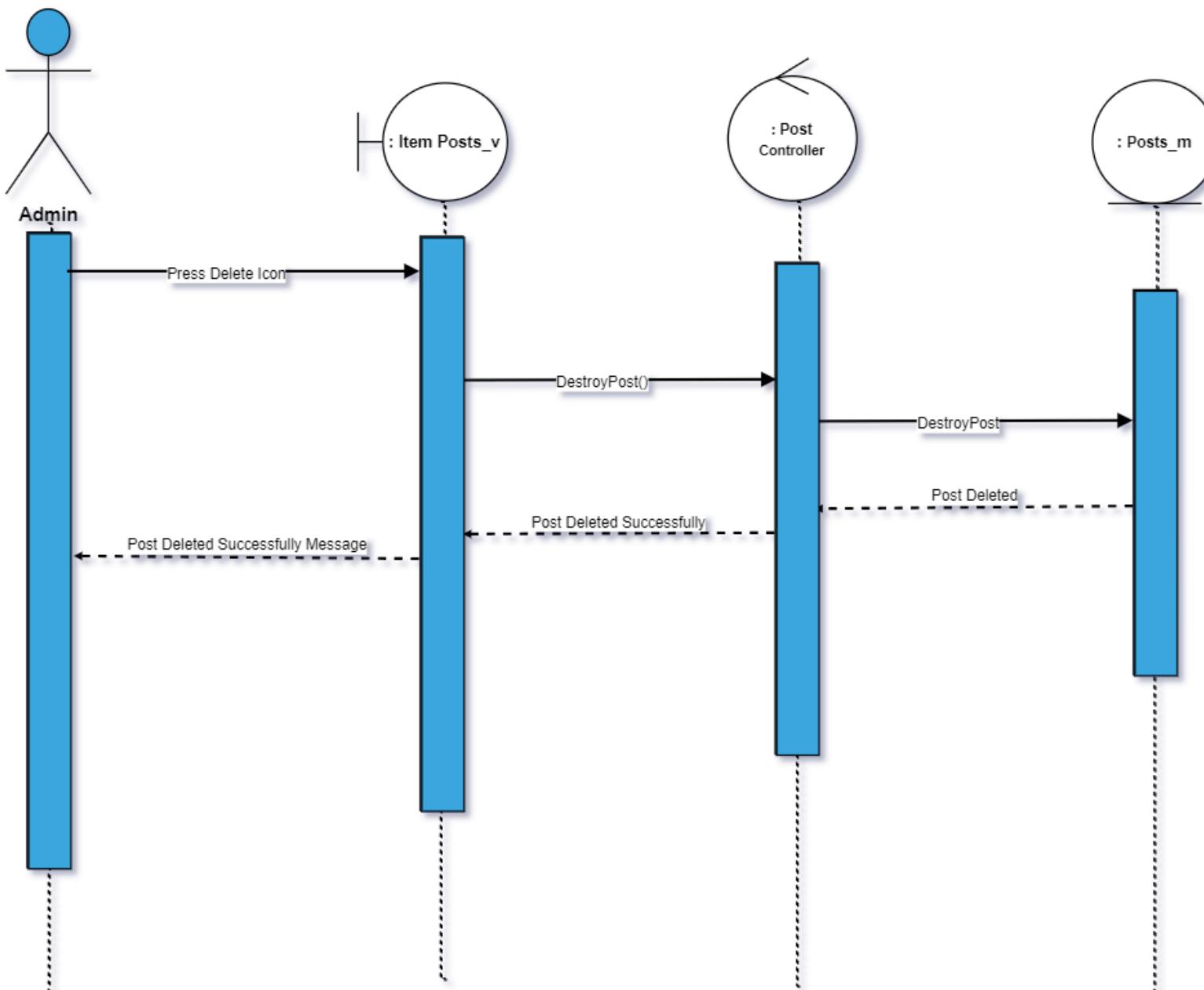
104)show item posts



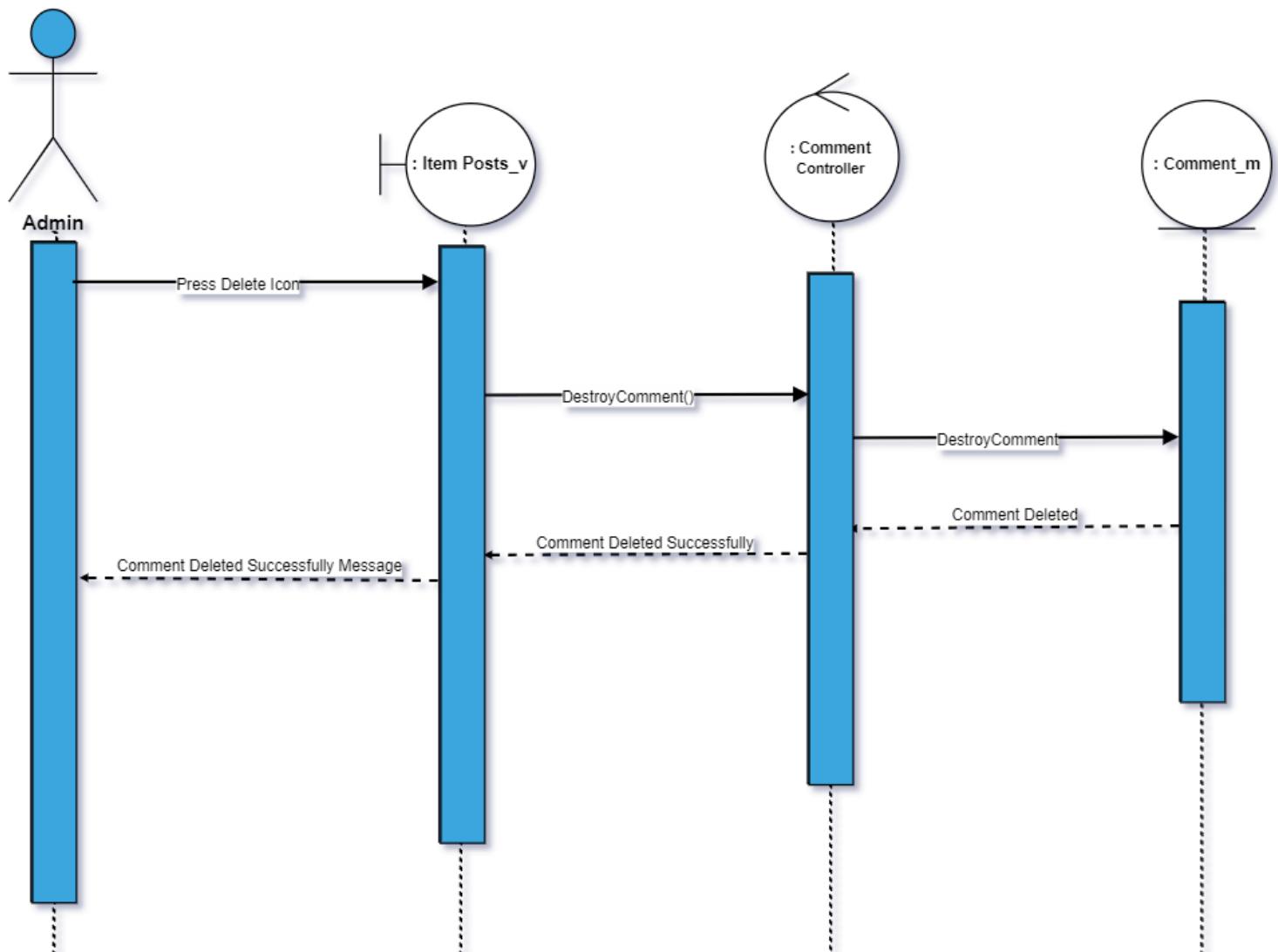
105)Edit post



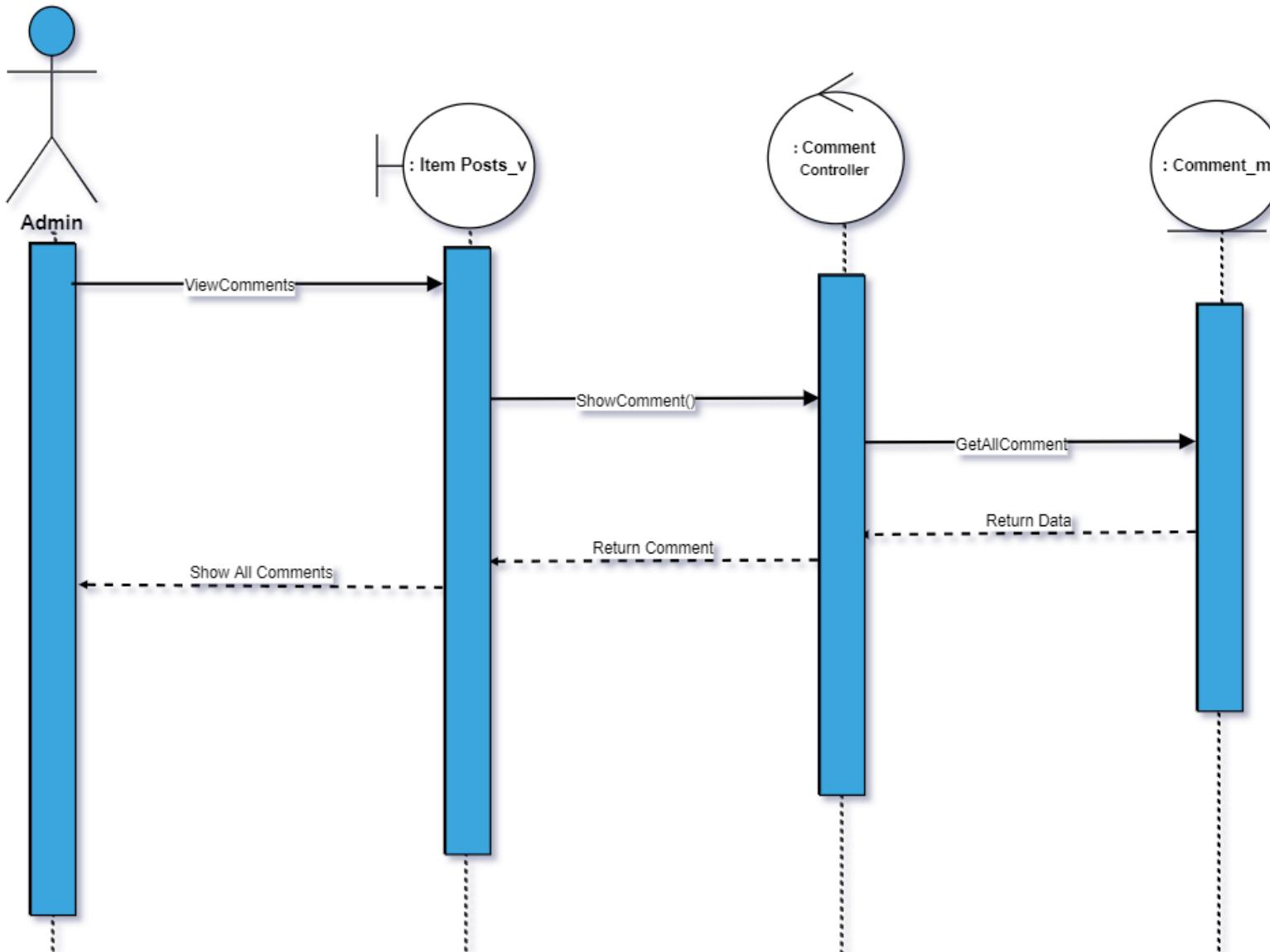
106)Delete post



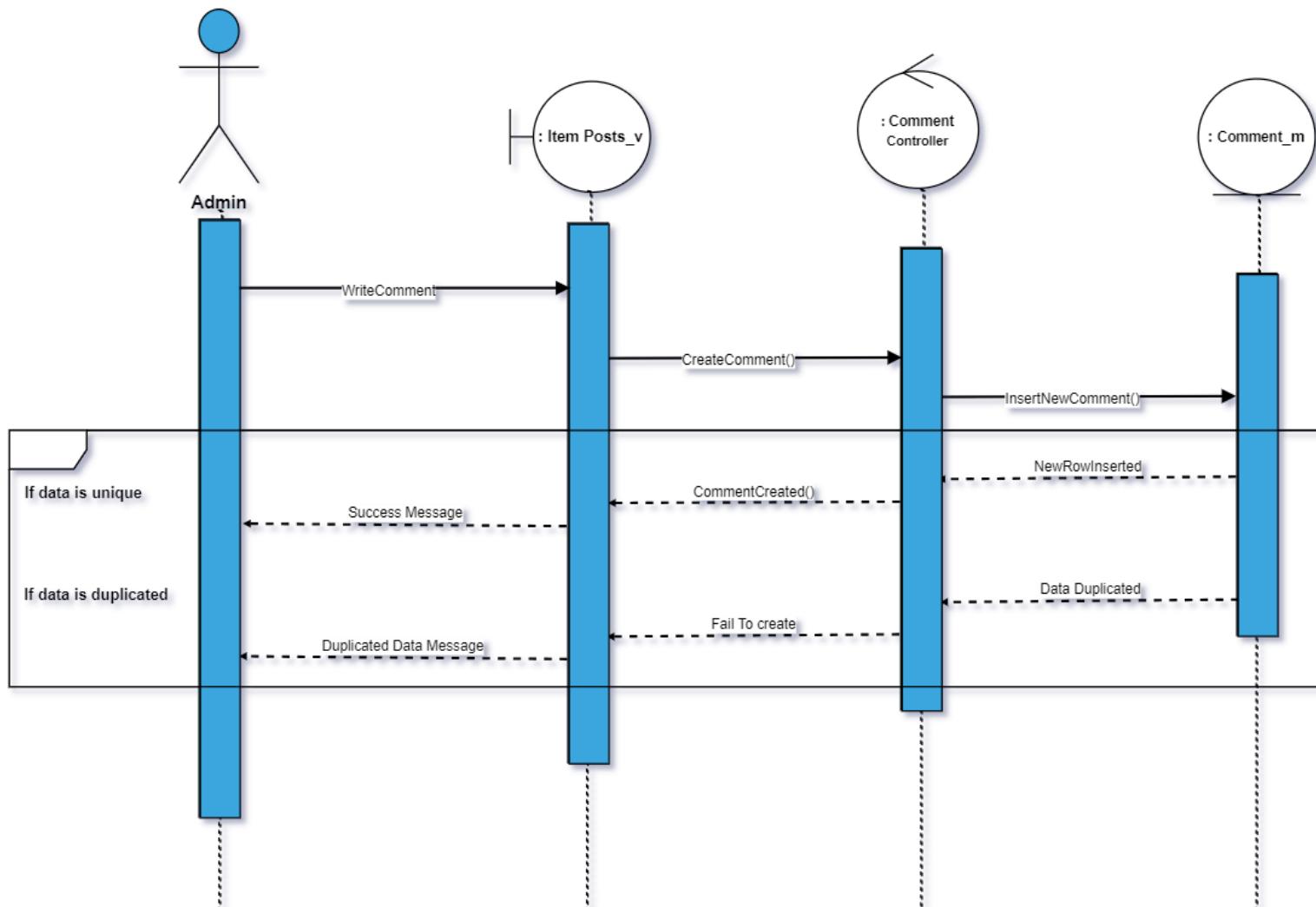
107)destroy comments



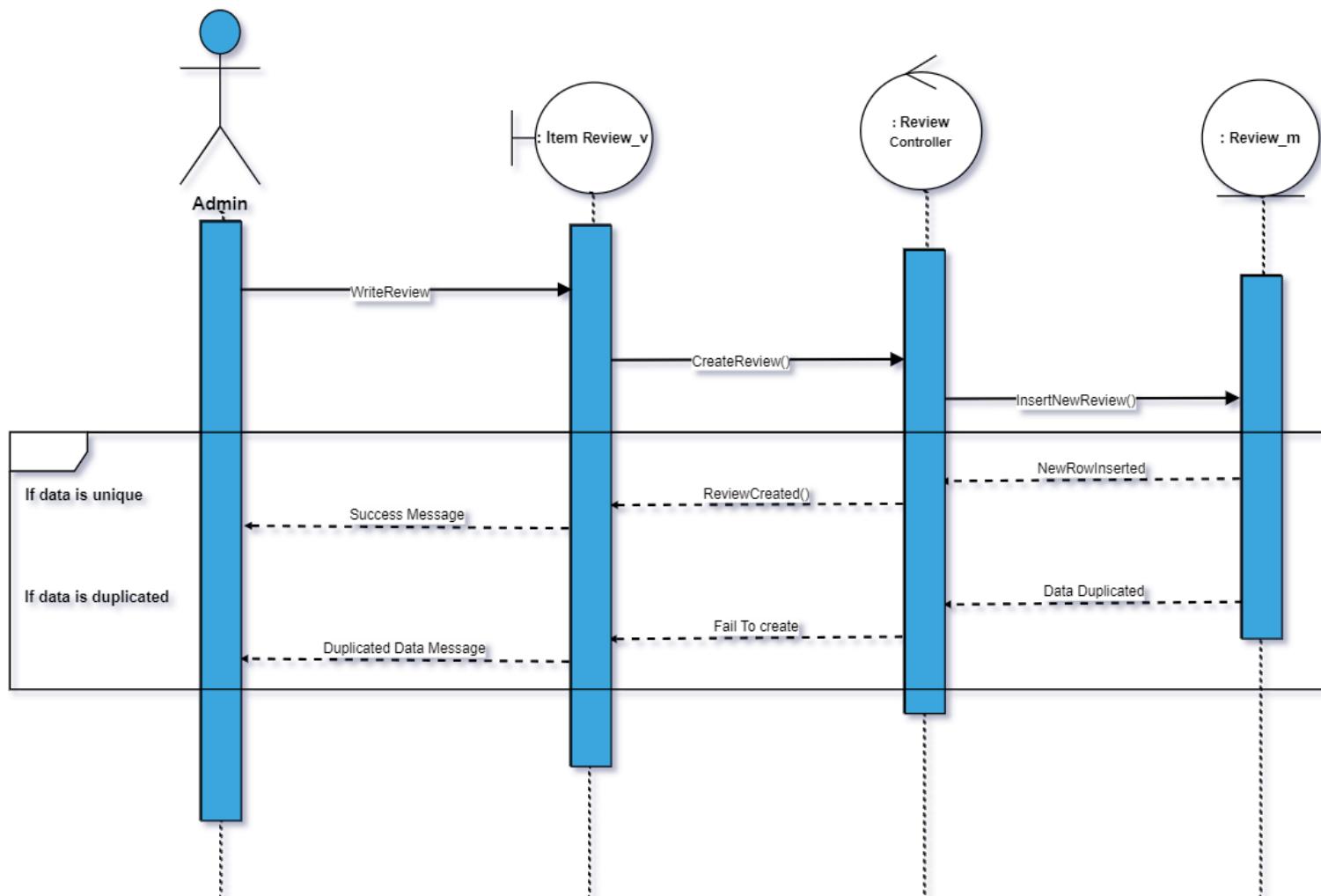
108) view comments



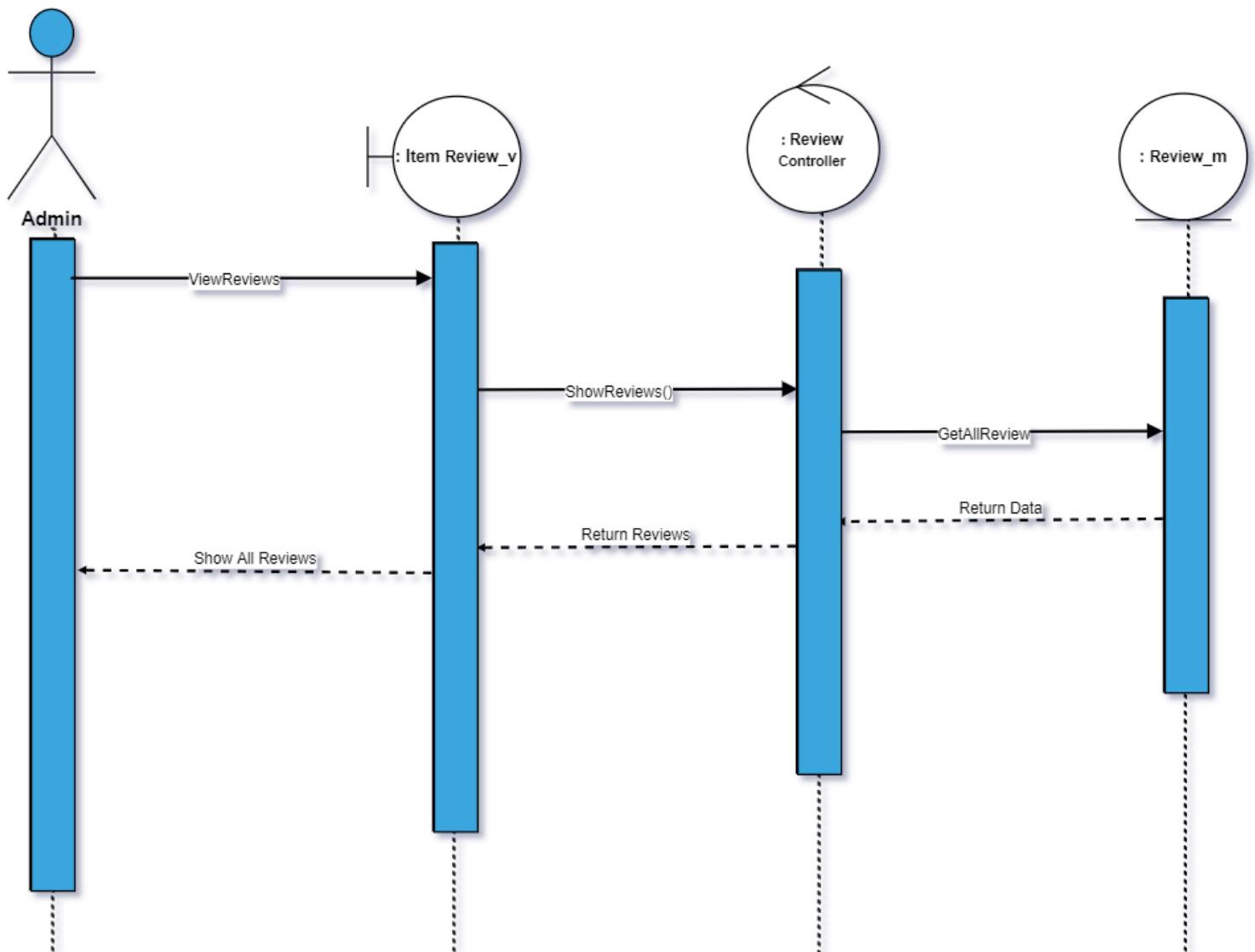
108) Write a comment



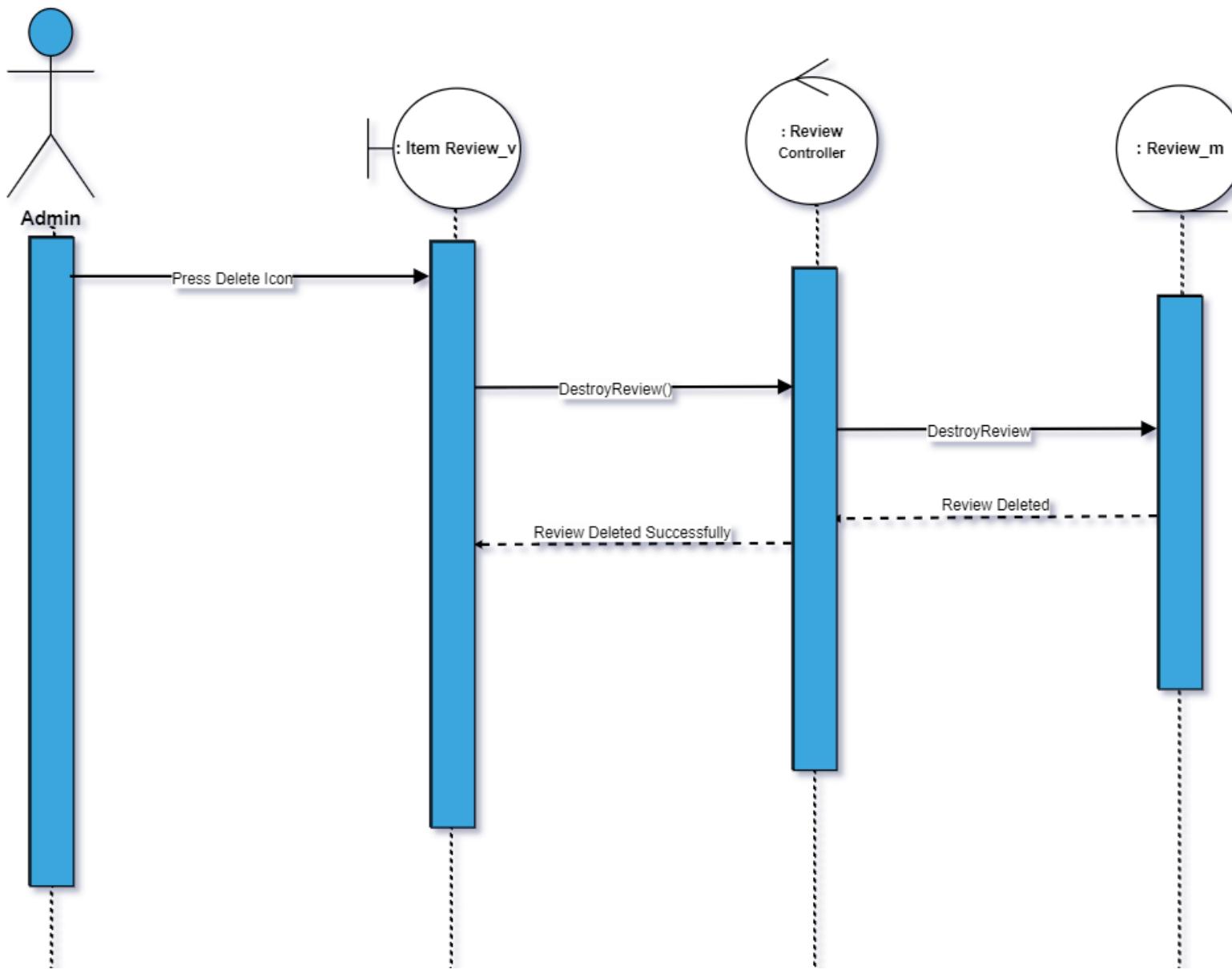
109)review



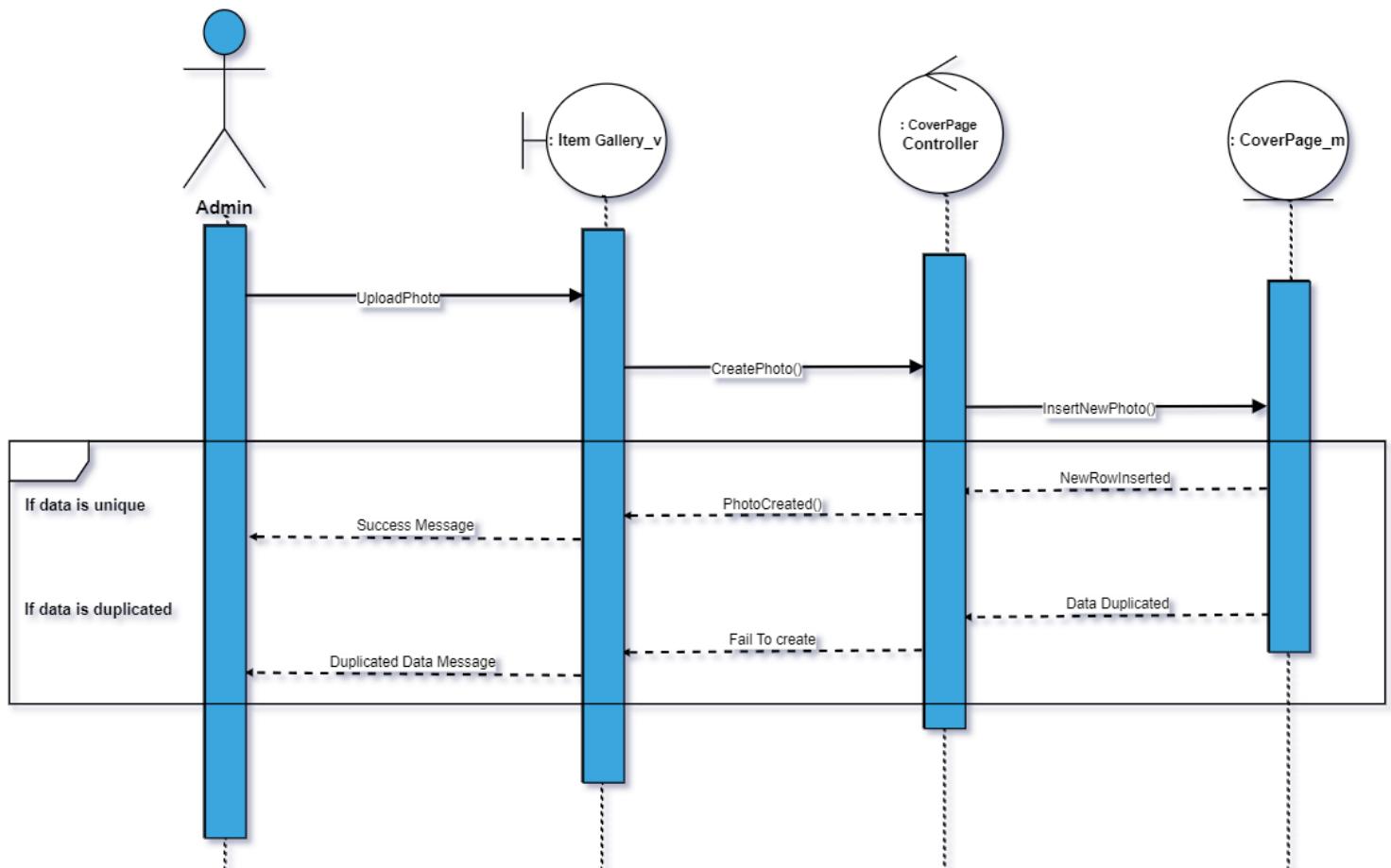
110) Show Review



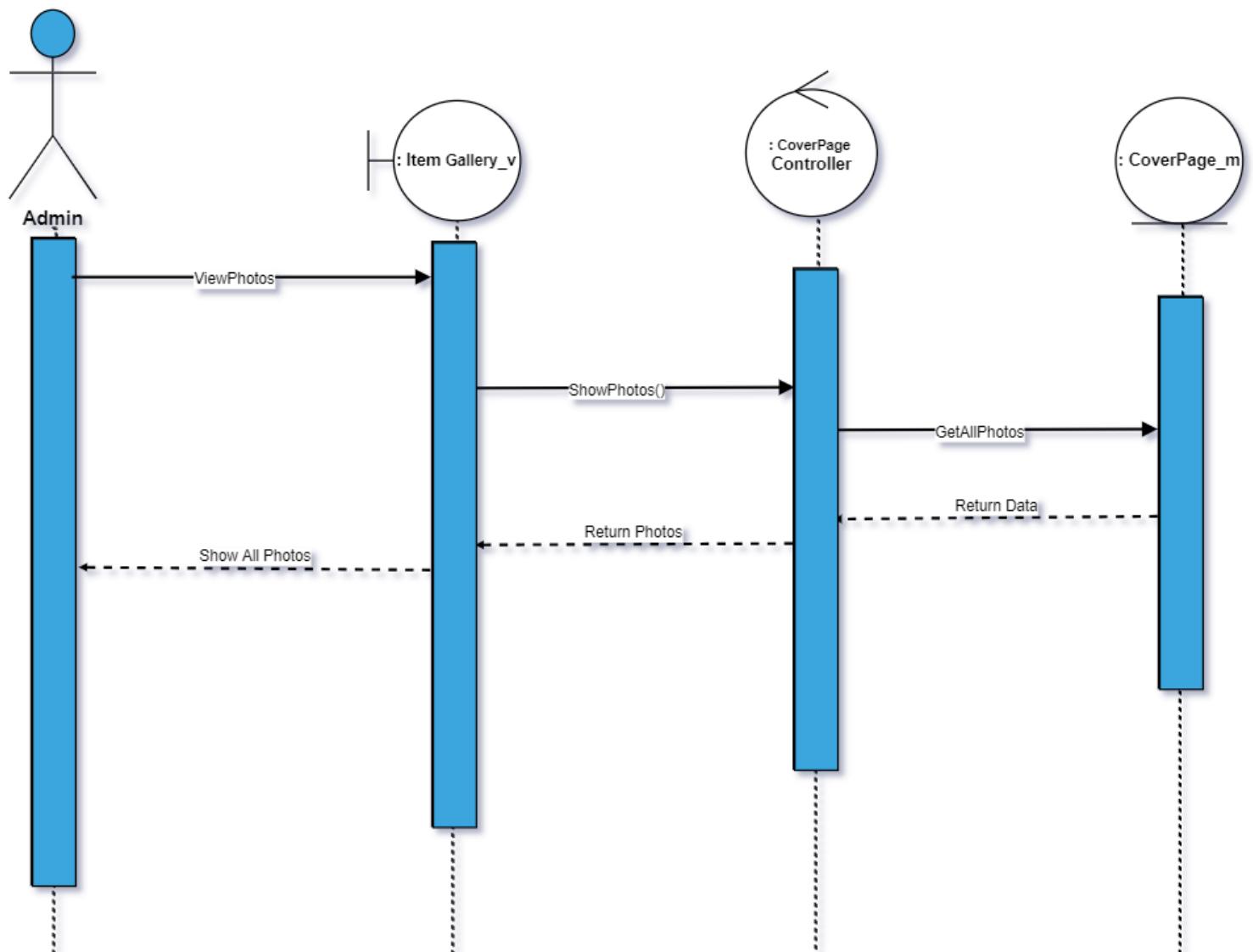
111)Delete review



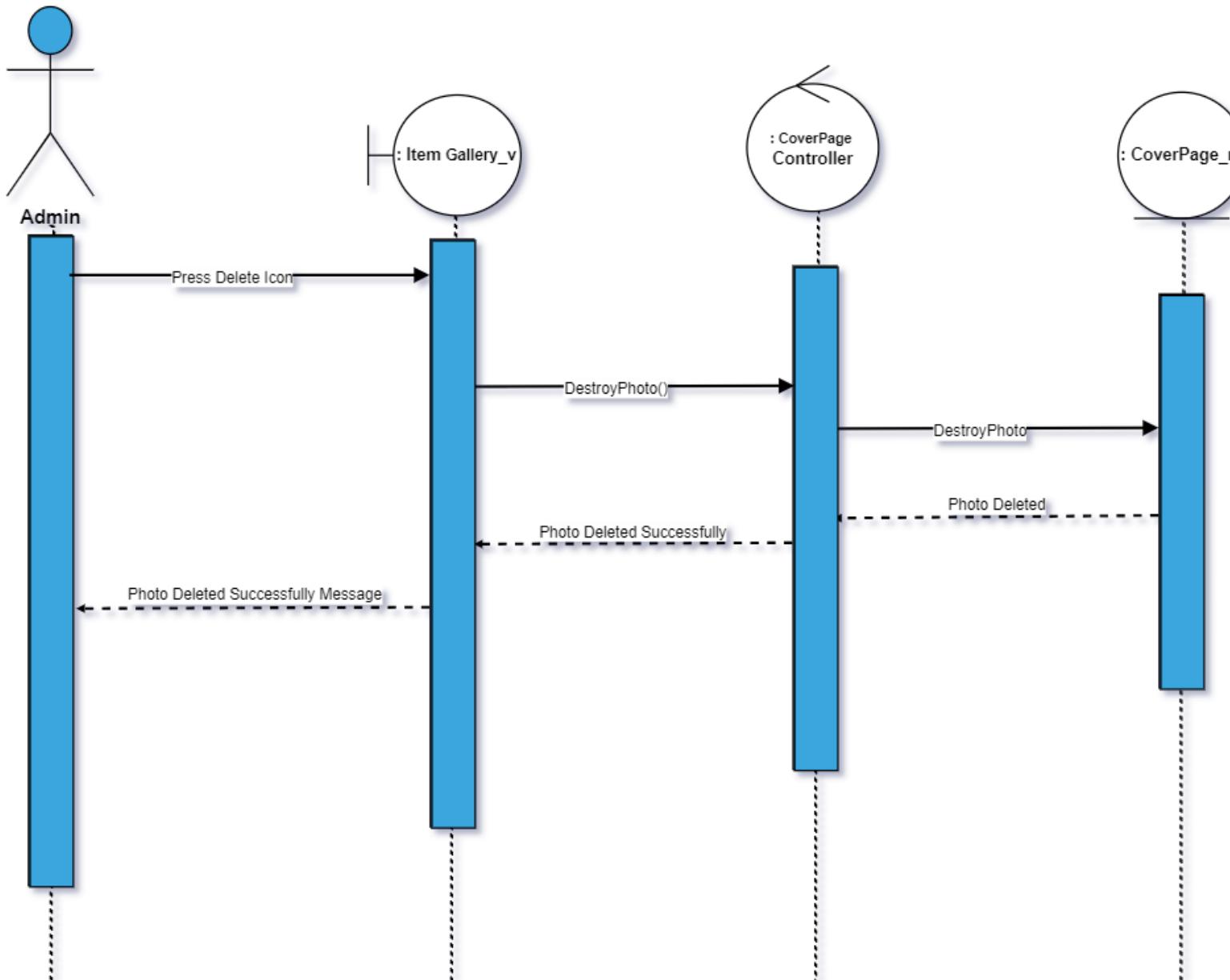
112) Upload cover page



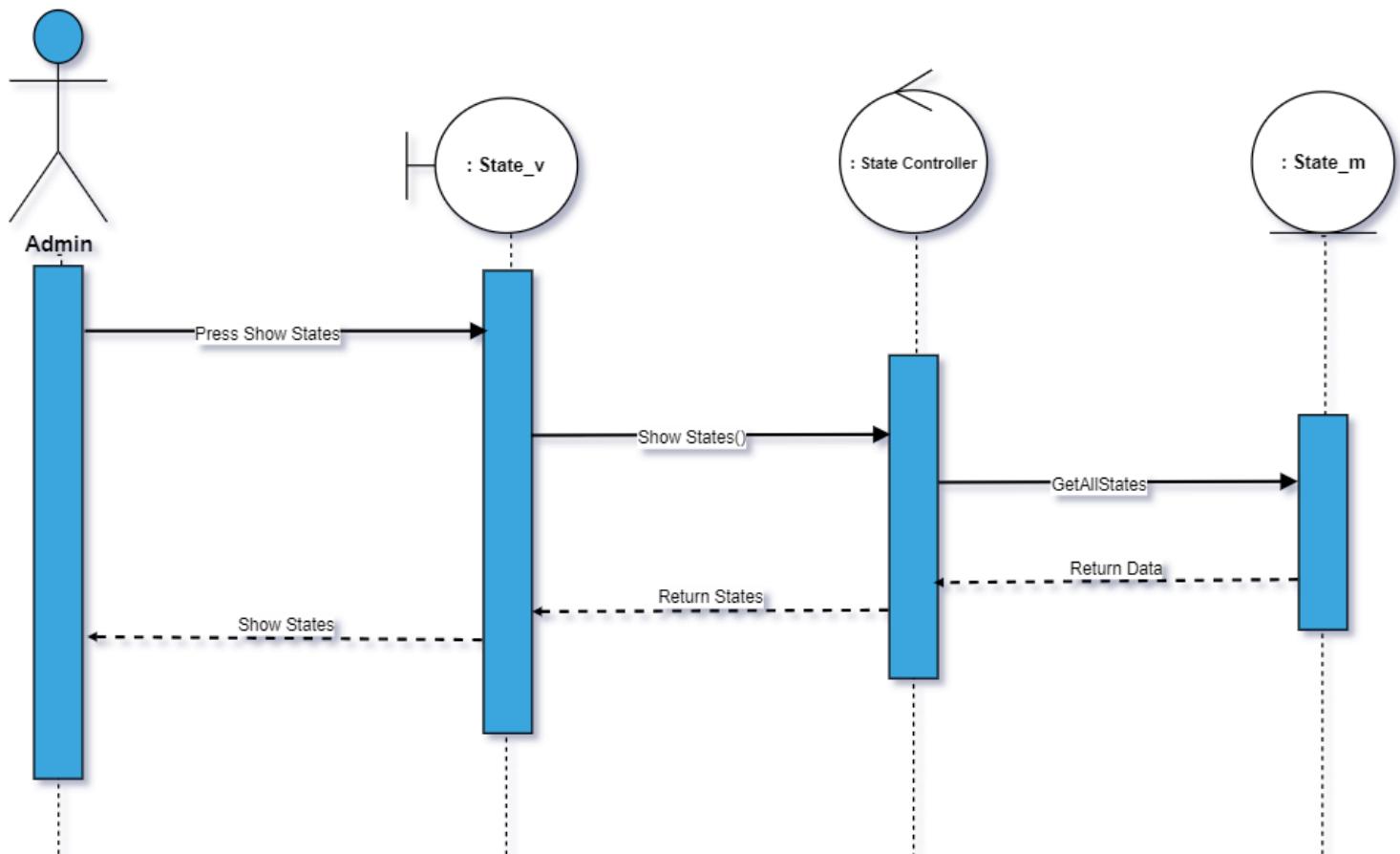
113)show photos



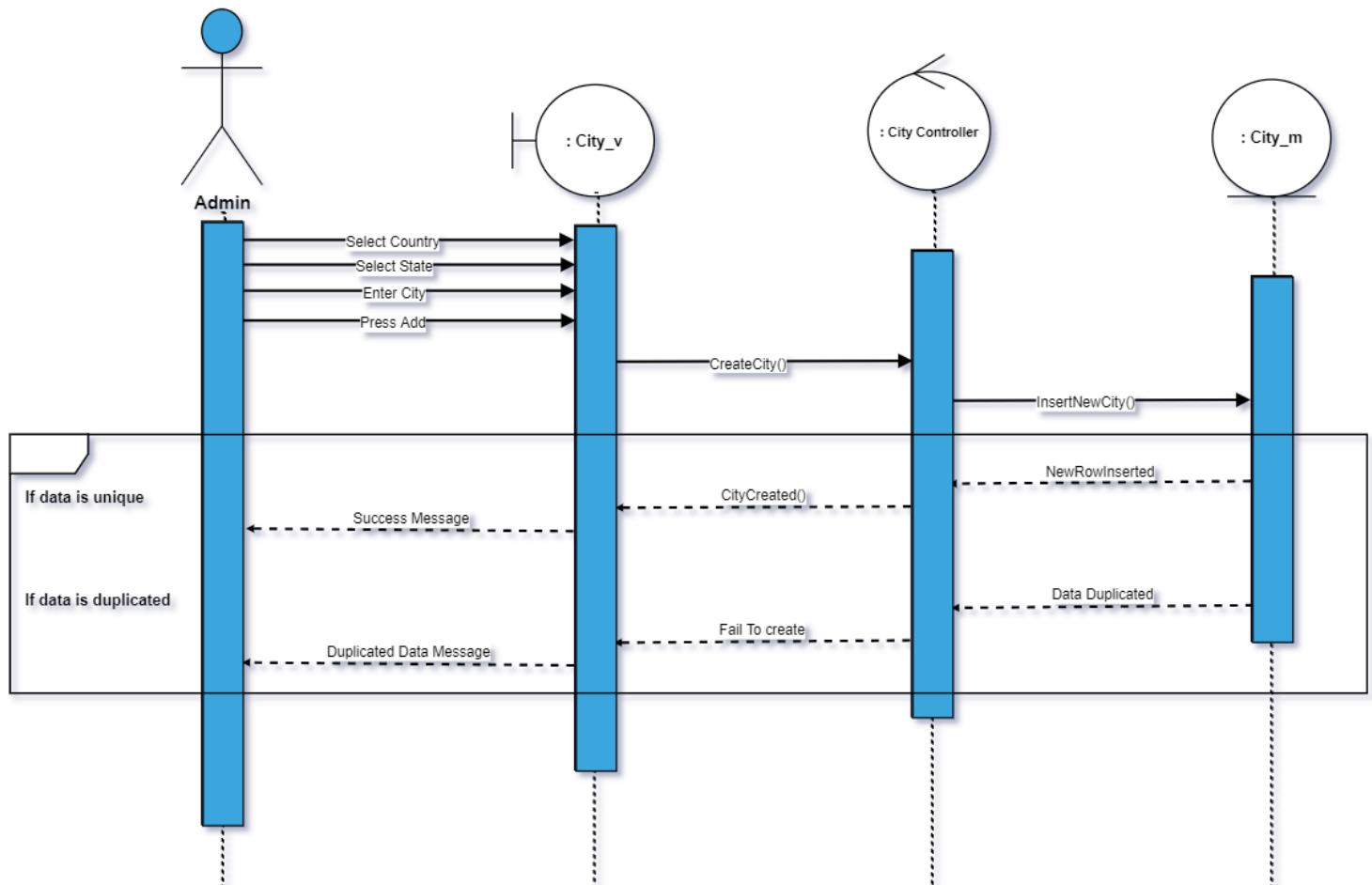
114)Delete photo



115) Show state



116) Add city



Chapter 5: System Implementation

5.1. Controllers

- AttachmentController

Controller Name	Description	Method & Description
AttachmentController	This controller was created for any process on attachments.	<ul style="list-style-type: none"> Index(\$id) This method takes item id as a parameter and returns all photos used in all posts of this item and return to Item_Gallery blade. Create(\$id) This method takes item id and create a post attachment for item with the id from the parameter and return back. getAttachment(\$id) This method takes attachment id as a parameter and return attachment getPostAttachments(\$item_id) This method takes item id as a parameter and return post attachments of all posts of this item. getAttachmentsOfPosts(\$id) This method takes post id as a parameter and return all attachments of a post. destroy(\$id) This method takes attachment id and destroy the attachment with the id from the parameter. getAttachmentOfuser(\$id) This method takes user id as a parameter and return all attachments of the post was uploaded by the user who created the post.

- **CityController**

Controller Name	Description	Method & Description
CityController	This controller was created for any process on Cities.	<ul style="list-style-type: none"> • Index(\$id) This method returns all cities , countries and states for the drop down list in the blade where to enter a new city. • Create() This method add a new city to the database. • show() This method return all cities to show cities blade. • destroy(\$id) This method finds all cities with state id from the request and return on json. • editCity() This method find city with the id from the request and changes city name by the name from the request and save it in the database then return back.

- CommentsController

Controller Name	Description	Method & Description
CommentsController	This controller was created for any process on Comments and replays.	<ul style="list-style-type: none"> create() This method for adding a new comments and saving it into database then send notification to the user who created the post and show the comment. reply() This method for adding a new replay and saving it into database then send notification to the user who entered the comment and show the replay. CommentCreatedBy(\$id) This method takes user id as a parameter and return comment created by the user destroyComment(\$id) This method takes comment id as a parameter and destroy this comment. destroyReplay(\$id) This method takes replay id as a parameter and destroy this replay. getPostComments(\$item_id) This method takes item id as a parameter and return all comments of all posts on this item and group them by post id then return the comments. getPostreplies(\$item_id) This method takes item id as a parameter and return all replies of all comments of all posts on this item and group them by parent comment then return the replies. getPostCommentHomePage(\$post_id) This method takes post id as a parameter and return all comments on this post with the name of the users and profile pictures. getPostRepliesHomePage(\$post_id) This method takes post id as an attachment and return all replies on this post with the name of the users and profile pictures. GetCommentReply() This method finds all replies of comment and return with users name , profile pictures and all replies. GetComments()



This method finds all comment of a post and return them.

- **editComment()**

This method takes comment id , edited comment from the request and update it then return back.

- **CountryController**

Controller Name	Description	Method & Description
CountryController	This controller was created for any process on Countries.	<ul style="list-style-type: none"> • index() This method for returning all countries. • create() This method for adding a new country to database and save it. • editCountry() This method takes country id and new country name form request and update the row in the database. • destroy() This method takes country id from the request and destroy this country.

- **CoverPageController**

Controller Name	Description	Method & Description
CoverPageController	This controller was created for any process on Coverpages of items.	<ul style="list-style-type: none"> • create() This method for adding a new coverpage it takes item id as a parameter and saving the new cover page into database then return back. • getCoverPhotoOfItem(\$id) This method takes item id as a parameter and return coverpage with this item id. • edit(\$id) This method takes coverpage id as a parameter and update a new cover page for this id then return back. • destroy(\$id,\$path) This method takes coverpage id and the path to delete the coverpage from the database and the folder.

- **CoverPhotoController**

Controller Name	Description	Method & Description
CoverPhotoController	This controller was created for any process on CoverPhotos.	<ul style="list-style-type: none"> • create() This method takes the file name by the request and creates a new cover photo for the user and save it in database and folder then return back. • edit () This method takes user id from authentication and edit his cover photo by the new one from the request and save it in the folder and database then return back. • getPhoto(\$id) This method takes user id as a parameter and return his cover photo. • update() This method get the photo with thephoto id from the request and update filename with the new one from the request and save it in the database and the folder then return back. • destroy(\$id,\$path) This method takes photo id as a parameter and photo name then delete the photo from the folder and also from the database by it's id. • sendCoverPhotoToProfile(\$id) This method takes user id as a parameter and return cover photo and it's id.

- **CustomerHomeController**

Controller Name	Description	Method & Description
CustomerHomeController	This controller was created for any process on Customer Home page.	<ul style="list-style-type: none"> • index() This method for sending all states to the customer home page to make the customer able to search for item in any state preferred and also to check if the user is owner or customer then return to the customer home page with results. • indexPhoto() This method returns all states' photos for the customer home page. • findItemInState() This method takes state name entered from the user by request and returns all items in this state and all states. • findItemInStateAndDate() This method takes state name entered from the user , arrival and departure by the request and returns all items in this state but also available for rent in TimeLine page . • FollowedItemPosts(\$item_id) This method takes item id as a parameter and returns all posts for this followed item for the TimeLine page. • HomePagePosts() This method returns all posts of all items the user follows and also all newitems and most popular items to HomePage posts blade. • HomePageUserPosts() This method returns all posts of all userss the user follows and also all newitems and most popular items to HomePageUser posts blade. • showMyProfile() This method return user name , cover photo , profile photo , posts and post images to customer own profile page

- **CustomerController**

Controller Name	Description	Method & Description
CustomerController	This controller was created for any process on Customer.	<ul style="list-style-type: none"> • itemProfile(\$id)



		<p>This method takes item id as a parameter and return all item profile information for the customer</p> <ul style="list-style-type: none">• itemDetails(\$id) <p>This method takes item id as a parameter and return all item details for item details page .</p> <ul style="list-style-type: none">• itemProfileGallery(\$id) <p>This method takes item id as a parameter and return all item galleries for item profile gallery page.</p> <ul style="list-style-type: none">• itemProfileReviews(\$id) <p>This method takes item id as a parameter and return all item reviews for item profile reviews page.</p> <ul style="list-style-type: none">• Show(\$id) <p>This method takes user id as a parameter and return all his posts , profile photo , cover photo and images to customer own profile if the user id is from the authentication else return to customer profile.</p>
--	--	---

- **OwnerController**

Controller Name	Description	Method & Description
OwnerController	This controller was created for any process on Owner.	<ul style="list-style-type: none"> • itemProfile(\$id) This method takes item id as a parameter and return all item profile information for the Owner. • itemDetails(\$id) This method takes item id as a parameter and return all item details for item details page to the owner . • itemProfileGallery(\$id) This method takes item id as a parameter and return all item galleries for item profile gallery page to the owner. • itemProfileReviews(\$id) This method takes item id as a parameter and return all item reviews for item profile reviews page to the owner. • itemReservations(\$item_id) This method takes item id as a parameter and returns all reservations and all it's details of this item for the item profile reservation page for the owner. • itemManageSchedule(\$item_id) This method takes item id as a parameter and returns all schedule of this item for the item profile manage schedule page for the owner. • index() This method returns all states to Customer Home blade for the owner.

- **DataTypeController**

Controller Name	Description	Method & Description
DataTypeController	This controller was created for any process on DataType.	<ul style="list-style-type: none"> • index() This method return to data type page in admin pages. • create() This method takes data type name from the request and save it in the data base then return back. • show() This method return all data types to show data type page in admin pages. • edit() This method takes data type id and the new name from the request and edit it in the data base then return back. • destroy() This method takes data type id from the request and delete this data type from the data base then return back.

- **DetailsController**

Controller Name	Description	Method & Description
DetailsController	This controller was created for any process on Details.	<ul style="list-style-type: none"> • index() This method return all main types , sub types , property and property details for details page in admin pages. • create() This method takes all details by the request and save each detail in the data base then return back. • editDetails() This method takes details from the request and edit each detail and save it in the data base then return back. • AddImage() This method takes file name from the request and save it in the database if the user want to enter the detail value as a image then return back. • show() This method return all details for show details page in admin pages. • edit() This method takes detail id and the new detail name from the request and edit it then save it in the database and return back. • destroy() This method takes detail id from the request and delete this detail from the database • destroydetails() This method takes all details of property and delete all its details then return back. • FindDetailsForShow() This method takes property diff id from the request and get all it's details for the show.

- **FollowedUsersController**

Controller Name	Description	Method & Description
FollowedUsersController	This controller was created for any process on Follow.	<ul style="list-style-type: none"> • FollowedUser(\$id) This method takes the id of the user followed as a parameter and save a relation ship “follow” between the user from the authentication and the user he want to follow then send a notification to the user he was followed then return back. • UnfollowUser(\$id) This method takes id of the user was followed as a parameter and delete this relation from the database then return back. • checkFollow(\$id) This method takes id of the user was followed as a parameter and check if there is a relation ship between him and the user from the authentication.

- **MainTypesController**

Controller Name	Description	Method & Description
MainTypesController	This controller was created for any process on MainTypes.	<ul style="list-style-type: none"> • index() This method return to main types page in admin pages. • create() This method takes Main Type name from the request and save it in the database then return back. • show() This method returns all main types to show them in main types show page. • Edit() This method takes the new main type and its id then edit it and save it in the data base then return back. • Destroy() This method takes main types id and delete it from the database then return back.

- **HomeController**

Controller Name	Description	Method & Description
HomeController	This controller was created for any process on MainTypes.	<ul style="list-style-type: none">• index() This method return to Home page.
		<ul style="list-style-type: none">• __construct() This method allow to access all pages if user is authenated.

- **NotificationController**

Controller Name	Description	Method & Description
NotificationController	This controller was created for any process on Notification.	<ul style="list-style-type: none"> • index() This method return all notifications • create(\$from_id,\$to_id,\$notification) This method takes notification , from and to as a parameter and save it in the data base then return back. • create(\$from_id,\$to_id,\$notification,\$redirect_to) This method takes notification , from , to and redirect to as a parameter and save it in the data base then return back. • viewNotification() This method takes notification id from the request and return the notification as a viewed then return back. • show() This method return all notifications.

- **PaymentController**

Controller Name	Description	Method & Description
PaymentController	This controller was created for any process on Payment.	<ul style="list-style-type: none"> • create() This method return all notifications • create(\$from_id,\$to_id,\$notification) This method takes notification , from and to as a parameter and save it in the data base then return back. • create(\$from_id,\$to_id,\$notification,\$redirect_to) This method takes notification , from , to and redirect to as a parameter and save it in the data base then return back. • viewNotification() This method takes notification id from the request and return the notification as a viewed then return back. • show() This method return all notifications.

- **PostAttachmentController**

Controller Name	Description	Method & Description
PostAttachmentController	This controller was created for any process on PostAttachment.	<ul style="list-style-type: none">• deleteImgFromGallery() This method takes attachment id from request and return all attachment

- PostsController

Controller Name	Description	Method & Description
PostsController	This controller was created for any process on Posts.	<ul style="list-style-type: none"> • index() This method takes item id as a parameter and return all posts of this item to item posts page .
		<ul style="list-style-type: none"> • create(\$id) This method takes item id as a parameter and create post for this item with it's attachments then return back.
		<ul style="list-style-type: none"> • postCreatedBy(\$id) This method takes post id as a parameter and return the id of the user created this post.
		<ul style="list-style-type: none"> • userPosts(\$id) This method takes user id as a parameter and return all posts of this user.
		<ul style="list-style-type: none"> • Destroy(\$id) This method takes post id as a parameter and delete this post from the data base then return back.
		<ul style="list-style-type: none"> • getItemPosts(\$item_Id) This method takes item id as a parameter and return all posts of this item.

- **ProfilePhotoController**

Controller Name	Description	Method & Description
ProfilePhotoController	This controller was created for any process on Profile Photo.	<ul style="list-style-type: none"> • getPhoto(\$id) This method takes profile photo id as a parameter and return this photo.
		<ul style="list-style-type: none"> • create() This method takes profile photo name from the request and save it in the database and the folder then return back.
		<ul style="list-style-type: none"> • edit() This method takes profile photo name from the request and edit this photo in the data base by user id with the new photo name and return back.
		<ul style="list-style-type: none"> • update() This method takes profile photo and photo id from the request and update the photo with this id with the new profile photo then return back.
		<ul style="list-style-type: none"> • Destroy(\$id,\$path) This method takes profile photo id and the path the delete this photo from the folder and the database then return back.

- **PropertyDetailsController**

Controller Name	Description	Method & Description
PropertyDetailsController	This controller was created for any process on Property Details.	<ul style="list-style-type: none"> • index() This method return all property details for property details page.
		<ul style="list-style-type: none"> • create() This method takes property details from the request and save it in the database then return back.
		<ul style="list-style-type: none"> • show() This method return all property details for show in property details show page.
		<ul style="list-style-type: none"> • find() This method takes property detail id from the request and return property detail.
		<ul style="list-style-type: none"> • FindDetailsForForm() This method takes property id and return all property details of this property.
		<ul style="list-style-type: none"> • findDetailsForFormInOwner() This method takes property diff id and property id from the request and return all property details and all properties.

- **RegionController**

Controller Name	Description	Method & Description
RegionController	This controller was created for any process on Regions.	<ul style="list-style-type: none"> • index() This method return all countries , states and cities for region page in admin pages.
		<ul style="list-style-type: none"> • create() This method takes country , state , city and region from the request and save region in the database then return back.
		<ul style="list-style-type: none"> • show() This method return all regions for show in region show page.
		<ul style="list-style-type: none"> • Destroy(id) This method takes region id as a parameter and delete this region from the database then return back.
		<ul style="list-style-type: none"> • Findregion() This method takes city id from the request and return all regions in this city.
		<ul style="list-style-type: none"> • EditRegion() This method takes region id and region name from the request and edit it in the database then return back. .

- **UserTypesController**

Controller Name	Description	Method & Description
UserTypesController	This controller was created for any process on UserTypes.	<ul style="list-style-type: none"> • index() This method return all UserTypes for User Types Page in admin pages.
		<ul style="list-style-type: none"> • create() This method takes User Type Name from the request and save User Type Name in the database then return back.
		<ul style="list-style-type: none"> • show() This method return all UserTypes for User Types Show Page in admin pages.
		<ul style="list-style-type: none"> • edit () This method takes user type id from the request and edit it's name by the new one from the request and save it in the database then return back.
		<ul style="list-style-type: none"> • destroy() This method takes user type id from the request and delete it from the database then return back to user types show page.
		<ul style="list-style-type: none"> • get_user_types() This method return all user types and all users for users show page. • getUser() This method takes user type id from the request and check in table type of users for all users with this user type id then return all these users and their phone number , email by json .

- **ReservationController**

Controller Name	Description	Method & Description
ReservationController	This controller was created for any process on Reservations.	<ul style="list-style-type: none">• show() <p>This method return all reservation operation details to reservation show page .</p>

- **ReviewController**

Controller Name	Description	Method & Description
ReviewController	This controller was created for any process on Reviews.	<ul style="list-style-type: none"> • index() This method takes item id as a parameter and return all item reviews to item reviews page.
		<ul style="list-style-type: none"> • create() This method takes item id and user id from the request and check if there is a review with these two ids and edit it with the new review content and stars from the request and save it if there is no review with these two ids will create a new review and fill it's information by the data from the request , save it in the data base and return done.
		<ul style="list-style-type: none"> • getItemReviews(\$item_id) This method takes item id as a parameter and return all item reviews .
		<ul style="list-style-type: none"> • getItemRate(\$item_id) This method takes item id as a parameter and calculate the rate of item by summing all stars for this item and how many reviews on this item then calculating the rate .
		<ul style="list-style-type: none"> • destroy(\$id) This method takes review id as a parameter and delete this review from the data base and return back.
		<ul style="list-style-type: none"> • EditRegion() This method takes region id and region name from the request and edit it in the database then return back. .

- **StateController**

Controller Name	Description	Method & Description
StateController	This controller was created for any process on States.	<ul style="list-style-type: none"> • index() This method return all countries for add state page in admin pages.
		<ul style="list-style-type: none"> • create() This method takes country id and state name from the request and save state in the database then return back.
		<ul style="list-style-type: none"> • show() This method return all states for show in states show page.
		<ul style="list-style-type: none"> • Destroy(id) This method takes state id as a parameter and delete this state from the database then return back.
		<ul style="list-style-type: none"> • findstate() This method takes country id from the request and return all states in this country.
		<ul style="list-style-type: none"> • edutState() This method takes state id and new state name from the request and edit it in the database then return back.
		<ul style="list-style-type: none"> • Findstatebyname(\$statename) This method takes state name as a parameter and return state id with this state name.
		<ul style="list-style-type: none"> • getStates() This method return all state names.

- **StatePhotoController**

Controller Name	Description	Method & Description
StatePhotoController	This controller was created for any process on StatePhotos.	<ul style="list-style-type: none"> • index() This method return all states for state photo page.
		<ul style="list-style-type: none"> • create() This method takes file name from the request and save it in the folder and create a new attachment then take the attachment id to save it in state photo table by state id and create a state photo then return back

- **StreetController**

Controller Name	Description	Method & Description
StreetController	This controller was created for any process on Streets.	<ul style="list-style-type: none"> • index() This method return all countries , states , cities and regions for add street page in admin pages.
		<ul style="list-style-type: none"> • create() This method takes country , state , city , region ids and street name from the request and save street in the database then return back.
		<ul style="list-style-type: none"> • show() This method return all streets for show in street show page.
		<ul style="list-style-type: none"> • Destroy(id) This method takes street id from the request and delete this street from the database then return back.
		<ul style="list-style-type: none"> • findstreet() This method takes region id from the request and return all streets in this region.
		<ul style="list-style-type: none"> • editStreet() This method takes street id and new street name from the request and edit it in the database then return back.

- **SubTypePropertyController**

Controller Name	Description	Method & Description
SubTypePropertyController	This controller was created for any process on SubTypeProperties.	<ul style="list-style-type: none"> • index() This method return all maintypes and sub types for SubTypeProperty page.
		<ul style="list-style-type: none"> • create() This method takes maintype id and subtype id and Property name from the request and save Property name in the database then return back.
		<ul style="list-style-type: none"> • show() This method return all Property names for show in SubTypeProperty show page.
		<ul style="list-style-type: none"> • Destroy(id) This method takes property id from the request and delete this property from the database then return back.
		<ul style="list-style-type: none"> • find() This method takes subtype id from the request and return all properties in this region.
		<ul style="list-style-type: none"> • editStreet() This method takes property id and new property name from the request and edit it in the database then return back.
		<ul style="list-style-type: none"> • Property_select(\$item_id,\$sub_type_id) This method takes subtype id as a parameter and return all properties for this sub type with item id to properties select page .

- SubTypeController

Controller Name	Description	Method & Description
SubTypeController	This controller was created for any process on SubType s.	<ul style="list-style-type: none"> • index() This method return all maintypes for SubType page.
		<ul style="list-style-type: none"> • create() This method takes maintype id and subtype name from the request and save subtype name in the database then return back.
		<ul style="list-style-type: none"> • show() This method return all maintypes names and subtypes names for show in SubType show page.
		<ul style="list-style-type: none"> • Destroy(id) This method takes subtype id from the request and delete this subtype from the database then return back.
		<ul style="list-style-type: none"> • find() This method takes maintype id from the request and return all subtypes in this maintype.
		<ul style="list-style-type: none"> • update(\$id) This method takes subtype id as a parameter and maintype id and subtype name from the request and edit the main type of this sub type then return back.
		<ul style="list-style-type: none"> • Property_select(\$item_id,\$sub_type_id) This method takes subtype id as a parameter and return all properties for this sub type with item id to properties select page .
		<ul style="list-style-type: none"> • getSubTypeById(\$id) This method takes subtype id as a parameter and return sub type name.

- **UserController**

Controller Name	Description	Method & Description
UserController	This controller was created for any process on User.	<ul style="list-style-type: none"> • create() This method for adding a new user information (name , gender , password , email , phone number and user type) saving it into database and return back. • index() This method return all user types for add user page. • AdminProfile() This method takes user id from the authentication and return user name , email , phone number to admin profile page. • EditUserProfileVeiw() This method takes user id from the authentication and return user name , email , phone number and profile photo to edit user profile page. • EditUserProfile(\$id) This method takes user id from the authentication and user name , email and phone number from request to edit user profile page and return back. • checkIfOwner() This method takes user id from the authentication and check in table type of users if this user id has a usertype id of the owner or not. • BeOwner(\$user_id) This method takes user id as a parameter and create a row data for this id in table type of user making his user type equal owner id and filling all his data then return back. • getItemWithOwnerName(\$item_id) This method takes item id as a parameter and return item with the owner name of this item. • search() This method takes email from the request and return it from the data base if found . • destroy() This method takes user id from the request and delete this user , his phone number and his email from data base then return back. • editUserName()



This method takes user id , new first , middle and last name then edit it in the database then return back.

- **editUserEmail()**

This method takes user id and the new email then edit it in the database then return back.

- **editUserPhoneNumber()**

This method takes user id and the new phone number then edit it in the database then return back.

- **getUserName(\$user_id)**

This method takes user id as a parameter and return user name.

- **FollowedItem(\$item_id)**

This method takes item id as a parameter and make relation between the user from the authentication and this item “follow” then send notification to the owner of this item that this user started to follow his item.

- **UnFollowedItem(\$item_id)**

This method takes item id as a parameter and the user id from the authentication and delete row of data of these two ids from table followed item by user then return back.

- **changePassword()**

This method takes user id from authentication and the new password from the request and changes it in the database then return back.

- **ItemController**

Controller Name	Description	Method & Description
ItemController	This controller was created for any process on Items.	<ul style="list-style-type: none"> • create() This method for adding a new user information (name , gender , password , email , phone number and user type) saving it into database and return back. • index() This method return all main types and items for item page in admin pages. • Index1() This method return all user types and countries for add item steps. • Show() This method takes item id from the request and return owner of item , item location and item details to show item page in admin pages. • itemShow() This method return to item page. • SubTypeShow() This method return all main types and subtypes to item sub type show. • OwnerSelectProperty(\$item_id,\$subtype_id) This method takes item id and subtype id as a parameter and return subtype and properties to owner select details page. • SelectSubType(\$item_id) This method takes item id as a parameter and return subtypes and main types to owner select sub type. • OwnerAddItem() This method takes user id from authentication , street id , item name , longitude and latitude from request and create item with owner id of this user id then return to function SelectSubType(\$item_id) to continue adding item. • create() This method takes user id ,street id and item name from the request and add item in the data base then return to function SubTypeShow(\$item_id). • ShowEditLocation(id) This method return all regions , countries , states , cities and streets to edit item location page. • EditLocation(\$id)

This method takes item id as a parameter and find the item then edit street id by the new one from the request and return to function show(\$item_id).

- **showEditUser(\$id)**

This method takes item id as a parameter and return back to edit item user page.

- **EditUser(\$item_id)**

This method takes item id as a parameter and find the item then edit user id by the new one from the request and return to function show(\$item_id).

- **searchEmail()**

This method takes email from the request and find it in the database then return back.

- **destroy(\$item_id)**

This method takes item id as a parameter and delete this item from the database then return back.

- **EditItemMap(\$Item_Id)**

This method takes item id as a parameter and the new latitude and longitude from the request and changes it in the database then return back.

- **Getnewestitems()**

This method order items by created at date then return all items from the newest to the oldest

- **Getpopularitems()**

- **Getowner(\$id)**

This method takes item id as a parameter and return user id of this item .

- **OperationController**

Controller Name	Description	Method & Description
OperationController	This controller was created for any process on Operations.	<ul style="list-style-type: none"> • create(\$item_id) This method takes item id as a parameter and user id from the authentication then create a new operation the data base then return back. • index() This method return all operation types for operation detail. • calculatedays() This method takes schedule id start and end date from the request then calculate total price and price per date and return them. • Show() This method return all operation types to operation types show page. • finddetail() This method takes operation detail id from the request and return operation detail • edit() This method takes operation types id , operation type name and edit operation type then save it in the data base and return back. • editDetail() This method takes operation detail id , operation detail name and edit operation detail then save it in the data base and return back. • destroy() This method takes operation type id and delete this operation type then return back. • destroydetail() This method takes operation detail id and delete this operation detail then return back. • showDetail() This method return operation details and operation types to operation details show SubTypeShow(\$item_id). • Showreservations(\$item_id) This method takes item id as a parameter and return this item to reservation show page. • Showuserreservations() This method takes user id from authentication and get all operations of this user then return to show reservation page. • destroyoperation(\$id)

	<p>This method takes operation id as a parameter and delete this operation from the database then return back.</p> <ul style="list-style-type: none">• createType() <p>This method takes operation type name from the request and add it in the data base then return back.</p> <ul style="list-style-type: none">• createDetail() <p>This method takes operation detail and operation type id from the request then save it in the database then return back.</p> <ul style="list-style-type: none">• createValue(\$operation_id,\$type_id,\$detail_id,\$value) <p>This method takes operation_id, type_id, detail_id, value as a parameters and create a new value and save it in the database then return back.</p>
--	---

- ScheduleController

Controller Name	Description	Method & Description
ScheduleController	This controller was created for any process on Schedules.	<ul style="list-style-type: none"> • checkIfScheduleExists(\$item_id,\$start) This method takes item id and start date as a parameter and check if this schedule is exists. • index(\$item_id) This method return to item schedule page with item id. • create(\$item_id) This method takes item id as a parameter and check if this schedule not exists then create if is exists will create error message. • ownercreate() This method takes item id from the request and check if this schedule not exists then create if is exists will create error message. • createWithVariables(\$id,\$start\$end\$price) This method takes item id , start date , end date and price as a parameter and check if this schedule not exists then create if is exists will create error message. • getWholeSchedule(\$item_id) This method takes item id and return schedule of this item. • getAvailableTime() This method return all available schedules of this • getdays(\$star,\$end , \$schedule_id) This method takes schedule id , start and end date as a parameter and return all available days between start date and end date. • cutSchedule(\$schedule_id,\$start,\$end) This method takes schedule_id,start,end as a parameter and cut the reserved dates from the schedule • show(\$id) This method takes item id and return schedule of this item to item schedule show page. • edit() This method takes item id , start date , end date and price from the request and update it in the data base then return back. • destroy() This method takes schedule id from the request and delete it from the data base then return back

e

Chapter 6: System Testing

6.1. System Testing

1) Login

Input	Expected Output	Actual Output	Success
Valid Email valid Password	Sign in Go to Home page	Sign in Go to Home page	yes
Invalid Email valid Password	Display message: "Wrong email or password"	"Wrong email or password"	Yes
Valid Email Invalid Password	Display message: "Wrong email or password"	"Wrong email or password"	Yes

2) Add Country

Input	Expected Output	Actual Output	Success
Valid Input (Egypt)	Display message: "Country added Successfully".	"Country created Successfully"	yes
Invalid Input (egypt)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	yes

3) Add State

Input	Expected Output	Actual Output	Success
Valid Input (Cairo)	Display message: "state added Successfully".	"State created Successfully"	yes
Invalid Input (cairo)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	yes

4)Add City

Input	Expected Output	Actual Output	Success
Valid Input (Nasr City)	Display message: "City added Successfully".	"City created Successfully"	yes
Invalid Input (nasr city)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	yes

5)Add Region

Input	Expected Output	Actual Output	Success
Valid Input (Seven th District)	Display message: "Region added Successfully".	"Region created Successfully"	yes
Invalid Input (Seven th district)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	yes

6)Add street

Input	Expected Output	Actual Output	Success
Valid Input (Ibn Kotayba)	Display message: "Street added Successfully".	"Street created Successfully"	yes
Invalid Input (Ibn kotayba)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	yes

7)Edit Country

Input	Expected Output	Actual Output	Success
Valid Input (Egypt)	Display message: "Country edited Successfully".	"Country edited Successfully"	yes
Invalid Input (egypt)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	yes

8)Edit State

Input	Expected Output	Actual Output	Success
Valid Input (Cairo)	Display message: "state edited Successfully".	"State edited Successfully"	yes
Invalid Input (cairo)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	yes

9)Edit City

Input	Expected Output	Actual Output	Success
Valid Input (Nasr City)	Display message: "City edited Successfully".	"City edited Successfully"	yes
Invalid Input (nasr city)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	yes

10)Edit Region

Input	Expected Output	Actual Output	Success
Valid Input (Seven th District)	Display message: "Region edited Successfully".	"Region edited Successfully"	yes
Invalid Input (Seven th district)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	yes

11)Edit Street

Input	Expected Output	Actual Output	Success
Valid Input (Ibn Kotayba)	Display message: "Street added Successfully".	"Street edited Successfully"	yes
Invalid Input (ibn kotayba)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	yes

17)Add main type

Input	Expected Output	Actual Output	Success
Valid Input (Property)	Display message: "Item added Successfully".	"Item Created Successfully"	yes
Invalid Input (property)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	yes

18)Add sub type

Input	Expected Output	Actual Output	Success
Valid Input (Flat)	Display message: "Type added Successfully".	"Type created Successfully"	yes
Invalid Input (flat)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	yes

19)Add sub type property

Input	Expected Output	Actual Output	Success
Valid Input (Room)	Display message: "Property added Successfully".	"Property created Successfully"	yes
Invalid Input (room)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	Yes

20)Add Detail of property

Input	Expected Output	Actual Output	Success
Valid Input (Area)	Display message: "Property Detail added Successfully".	"Property Detail created Successfully"	yes
Invalid Input (area)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	yes

21)Edit Main Type

Input	Expected Output	Actual Output	Success
Valid Input (Property)	Display message: "Item edited Successfully".	"Item edited Successfully"	yes
Invalid Input (property)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	yes

22)Edit Sub type

Input	Expected Output	Actual Output	Success
Valid Input (Flat)	Display message: "Type edited Successfully".	"Type edited Successfully"	yes
Invalid Input (flat)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	yes

23)Edit Sub Type Property

Input	Expected Output	Actual Output	Success
Valid Input (Room)	Display message: "Property edited Successfully".	"Property edited Successfully"	yes
Invalid Input (room)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	Yes

24)Edit detail of property

Input	Expected Output	Actual Output	Success
Valid Input (Area)	Display message: "Property Detail edited Successfully".	"Property Detail edited Successfully"	yes
Invalid Input (area)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	Yes

25)Add user

Input	Expected Output	Actual Output	Success
Valid all fields in the wizard form	Display message: "User created successfully"	user created successfully	yes
Invalid one or more fields in the wizard form	Display message: "Please matches the requested format"	Please matches the requested format	Yes
No entered data	Please fill the out this field	Please fill the out this field	Yes

26)Add Item

Input	Expected Output	Actual Output	Success
Valid all fields in the wizard form	Display message: "Item created successfully"	Item created successfully	yes
Invalid one or more fields in the wizard form	Display message: "Please matches the requested format"	Please matches the requested format	Yes
No entered data	Please fill the out this field	Please fill the out this field	Yes

27)Add Operation

Input	Expected Output	Actual Output	Success
Valid Input (Rent)	Display message: "Operation added Successfully".	"Operation Created Successfully"	yes
Invalid Input (rent)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	Yes

28)Add Operation type

Input	Expected Output	Actual Output	Success
Valid Input (Rent)	Display message: "Operation type created Successfully".	"Operation type Created Successfully"	yes
Invalid Input (rent)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	Yes

29)Add Comment

Input	Expected Output	Actual Output	Success
Valid Input ()	Display message: "Comment added Successfully".	"comment added Successfully"	yes
Invalid Input (empty)	Please fill the out this field	Please fill the out this field	Yes

30)Edit comment

Input	Expected Output	Actual Output	Success
Valid Input (2 rooms)	Display message: "Comment edited".	"Comment edited"	yes
Invalid Input (empty)	Please fill the out this field	Please fill the out this field	Yes

31)Edit Operation

Input	Expected Output	Actual Output	Success
Valid Input (Rent)	Display message: "Operation added Successfully".	"Operation Created Successfully"	yes
Invalid Input (rent)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	Yes

32)Edit Operation type

Input	Expected Output	Actual Output	Success
Valid Input (Rent)	Display message: "Operation type edited Successfully".	"Operation type edited Successfully"	yes
Invalid Input (rent)	Display message: "First Letter must be Capital"	"First Letter must be Capital"	Yes
No entered data	Please fill the out this field	Please fill the out this field	Yes

33)Registration

Input	Expected Output	Actual Output	Success
Valid Email input valid Password input	Go to sign in	Go to sign in	yes
Invalid Email input valid Password	Display message: " Please matches the requested format"	"Please matches the requested format "	Yes
Valid Email input Invalid Password input	Display message: " Please matches the requested format"	"Please matches the requested format "	Yes
Duplicate data	"This Email is already taken"	"This Email is already taken"	Yes
No entered data	Please fill the out this field	Please fill the out this field	Yes

34)Be Owner

Input	Expected Output	Actual Output	Success
Valid Input All model View	Popup to agree continue owner	Popup to agree continue owner*	yes
Invalid input any model view	Display message: " Please matches the requested format"	"Please matches the requested format "	Yes
Duplicate data	"This Phone is already taken"	"This phone is already taken"	Yes
No entered data	Please fill the out this field	Please fill the out this field	Yes

35)Edit Detail

Input	Expected Output	Actual Output	Success
Valid input (Room)	Change in the details	Change in the details*	yes
Invalid input (room)	Display message: " Please matches the requested format"	"Please matches the requested format "	Yes
No entered data	Please fill the out this field	Please fill the out this field	Yes

36)Add Reply

Input	Expected Output	Actual Output	Success
Valid Input ()	Display message: "Reply added Successfully".	"Reply added Successfully"	yes
Invalid Input (empty)	Please fill the out this field	Please fill the out this field	Yes

37)Edit Reply

Input	Expected Output	Actual Output	Success
Valid Input (2 rooms)	Display message: "Reply edited".	"Reply edited"	yes
Invalid Input (empty)	Please fill the out this field	Please fill the out this field	Yes

38)Forget Password

Input	Expected Output	Actual Output	Success
Valid Input (Test@Gmail.com)	Forgot Password Page	Forgot Password Page	yes
Invalid input(hgujik)	Display message: " Please matches the requested format"	" Please matches the requested format"	yes
Invalid Input (empty)	Please fill the out this field	Please fill the out this field	Yes

Chapter 7: System Deployment

7.1 Tool Setup

Steps to Install Laravel 7 and all needed Software to Launch The Project

7.1.1 Install XAMPP

First, download the XAMPP from the official website given below.

<https://www.apachefriends.org/download.html>

Laravel 7 requires PHP >= 7.2.5. So download the latest version (7.3.3/PHP 7.3.3).



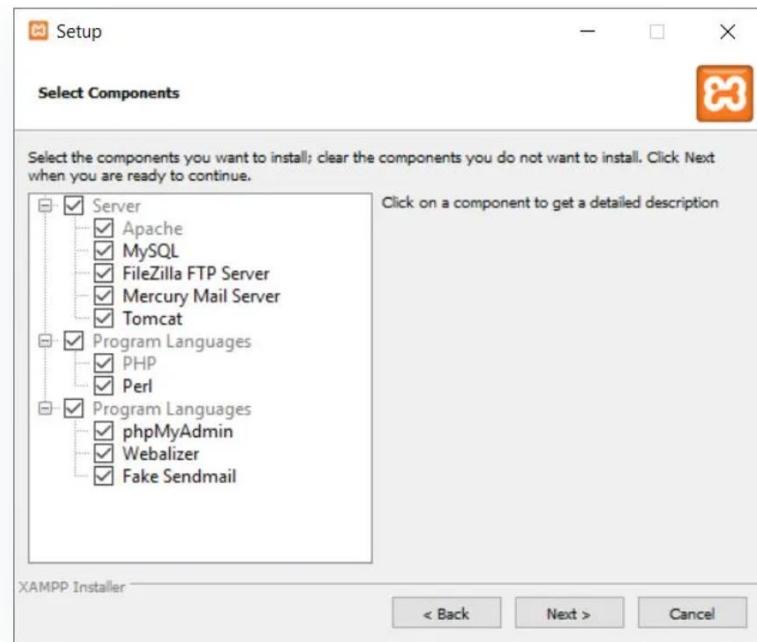
Version	Checksum	Size
7.1.27 / PHP 7.1.27	What's Included? md5 sha1	Download (64 bit) 141 Mb
7.2.16 / PHP 7.2.16	What's Included? md5 sha1	Download (64 bit) 146 Mb
7.3.3 / PHP 7.3.3	What's Included? md5 sha1	Download (64 bit) 146 Mb

Requirements Add-ons More Downloads »

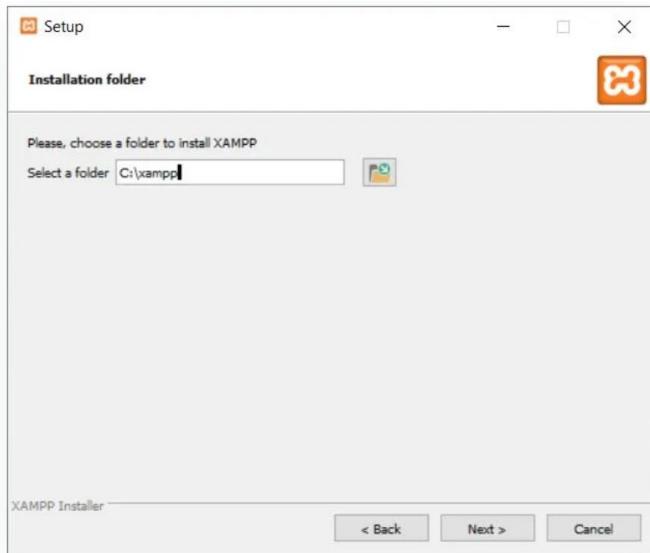
Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms [here](#).

Next steps Explain the XAMPP installation with the screenshots.

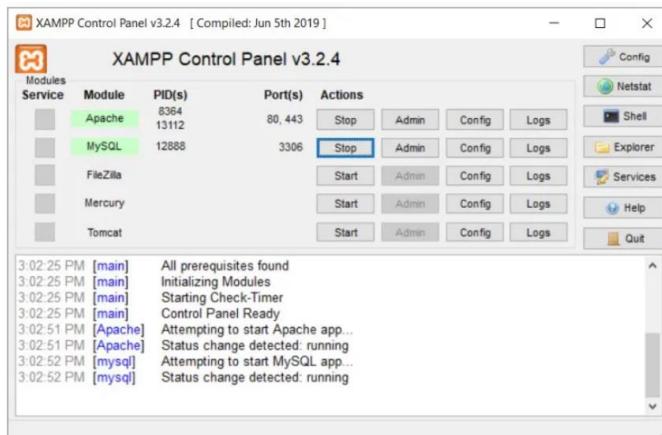
1. Select the components to install with XAMPP. Actually we only need Apache, MySQL, and phpMyAdmin. But here, I have selected all.



2- Select the installation folder. In default, it will be C:\xampp



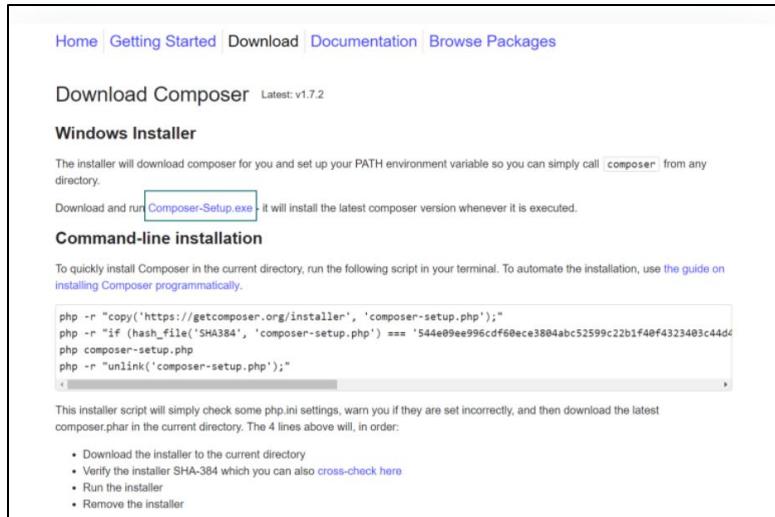
3- Finished and Now start the Apache and MySQL.



2. Install Composer

follow the below URL and download the **Composer-Setup.exe** file.

<https://getcomposer.org/download/> .



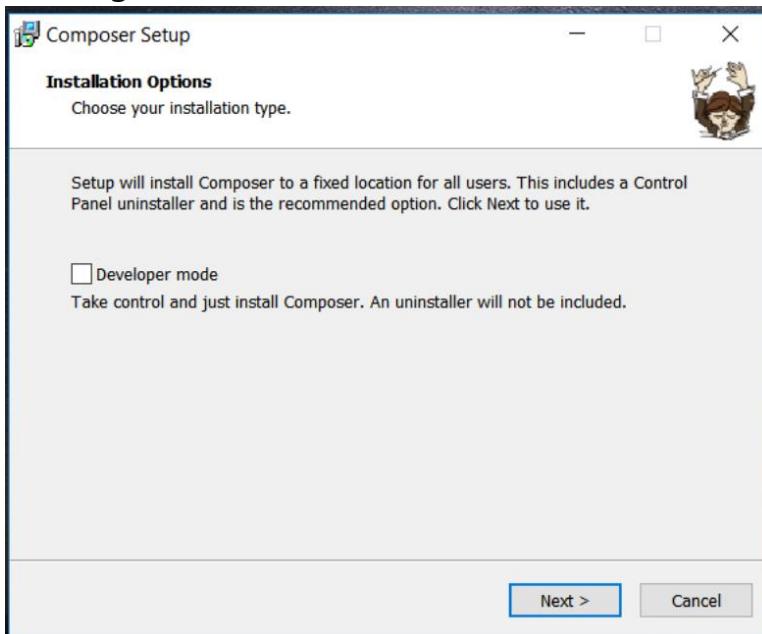
The screenshot shows the official Composer download page. The 'Windows Installer' section is highlighted. It contains instructions for running the installer and a command-line installation script. A note at the bottom explains the purpose of the script and provides steps for manual installation.

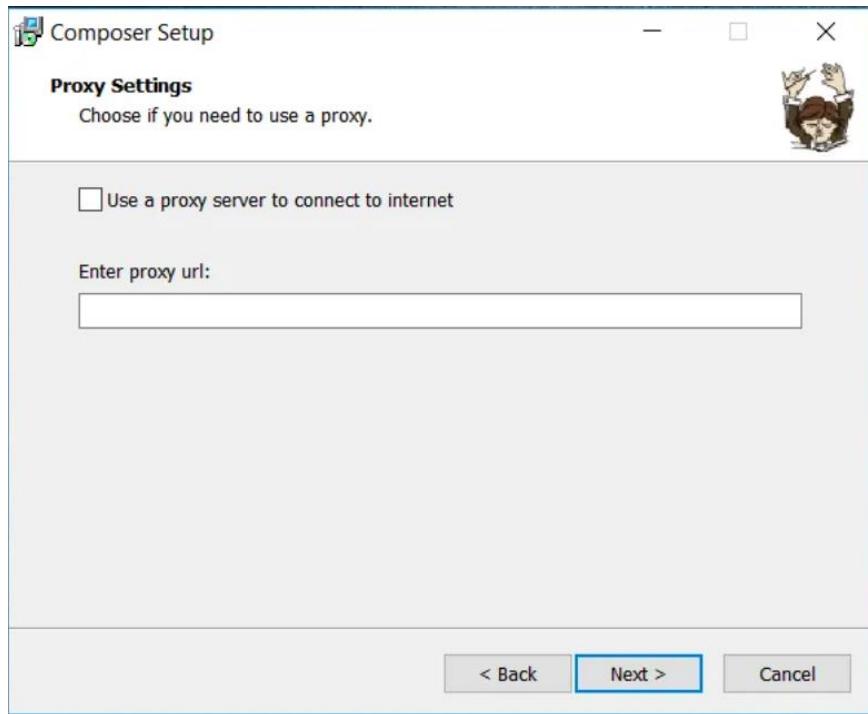
```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('SHA384', 'composer-setup.php') === '544e09ae996cdf60ece3804abc52599c22b1f40f4323403c44d4'
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

This installer script will simply check some php.ini settings, warn you if they are set incorrectly, and then download the latest composer.phar in the current directory. The 4 lines above will, in order:

- Download the installer to the current directory
- Verify the installer SHA-384 which you can also [cross-check here](#)
- Run the installer
- Remove the installer

Next figures are screenshots from the installation steps.





3. Install Laravel

This can be done by running the below command in the **Command prompt**.

```
composer global require "laravel/installer"
```

4- Create a Database.

using phpMyAdmin.

- Open the link below. <http://localhost/phpmyadmin>
- Now Enter username and password(As a default, the username will be **root** password will be empty).



Now you can see phpMyAdmin panel where you can manage all the MySQL databases.

To create a database by clicking **New** from the left menu. **Enter the database name** and press **Create**.

- Click on the **New** tab
- Enter a database name
- Press **Create**

Database	Collation	Action
information_schema	utf8_general_ci	Check privileges
mysql	latin1_swedish_ci	Check privileges
performance_schema	utf8_general_ci	Check privileges
phpmyadmin	utf8_bin	Check privileges
test	latin1_swedish_ci	Check privileges
Total: 5		

Note: Enabling the database statistics here might cause heavy traffic between the web server and the MySQL server.

[Enable statistics](#)

5-Modify .env File

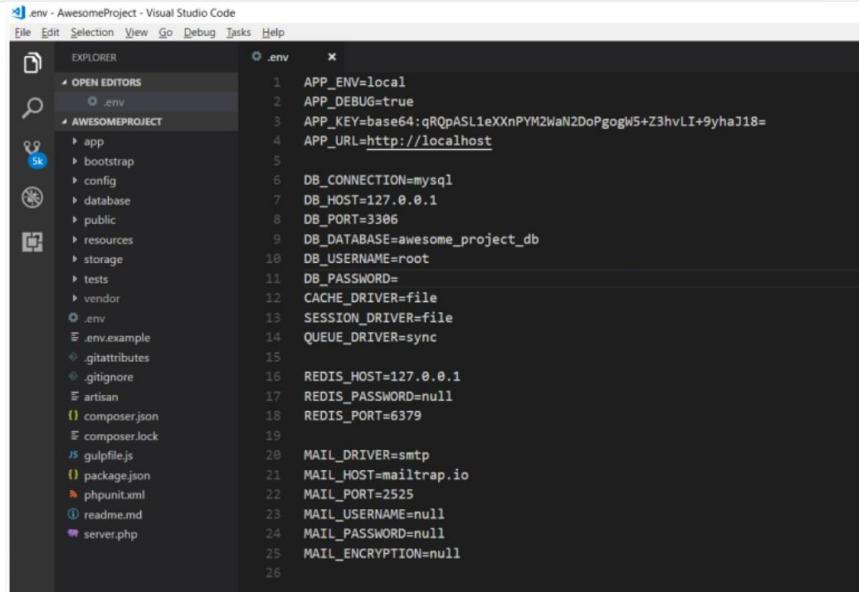
Now open the Project in any editor . recommend using [Visual Studio code](#) or [php storm](#) as the source-code editor for working with Laravel projects.

Check **.env** file inside the root directory of our project. Add our database details in this file which shown below.

```
DB_DATABASE=(The database name for the projec - realstatedb)
```

```
DB_USERNAME=(Your Mysql username. Default: root)
```

```
DB_PASSWORD=(Your Mysql password. As a default, blank)
```



```

.env - AwesomeProject - Visual Studio Code
File Edit Selection View Go Debug Tasks Help
EXPLORER
OPEN EDITORS .env
AWESOMEPROJECT
app
bootstrap
config
database
public
resources
storage
tests
vendor
.env
.env.example
.gitattributes
.gitignore
artisan
composer.json
composer.lock
gulpfile.js
package.json
phpunit.xml
readme.md
server.php
1 APP_ENV=local
2 APP_DEBUG=true
3 APP_KEY=base64:qRQpASL1eXXnPJM2WaN2DoPgogW5+Z3hvLI+9yhaJ18=
4 APP_URL=http://localhost
5
6 DB_CONNECTION=mysql
7 DB_HOST=127.0.0.1
8 DB_PORT=3306
9 DB_DATABASE=awesome_project_db
10 DB_USERNAME=root
11 DB_PASSWORD=
12 CACHE_DRIVER=file
13 SESSION_DRIVER=file
14 QUEUE_DRIVER=sync
15
16 REDIS_HOST=127.0.0.1
17 REDIS_PASSWORD=null
18 REDIS_PORT=6379
19
20 MAIL_DRIVER=smtp
21 MAIL_HOST=mailtrap.io
22 MAIL_PORT=2525
23 MAIL_USERNAME=null
24 MAIL_PASSWORD=null
25 MAIL_ENCRYPTION=null
26

```

6-Migrate Database

This can be done using the command below.

```
php artisan migrate
```

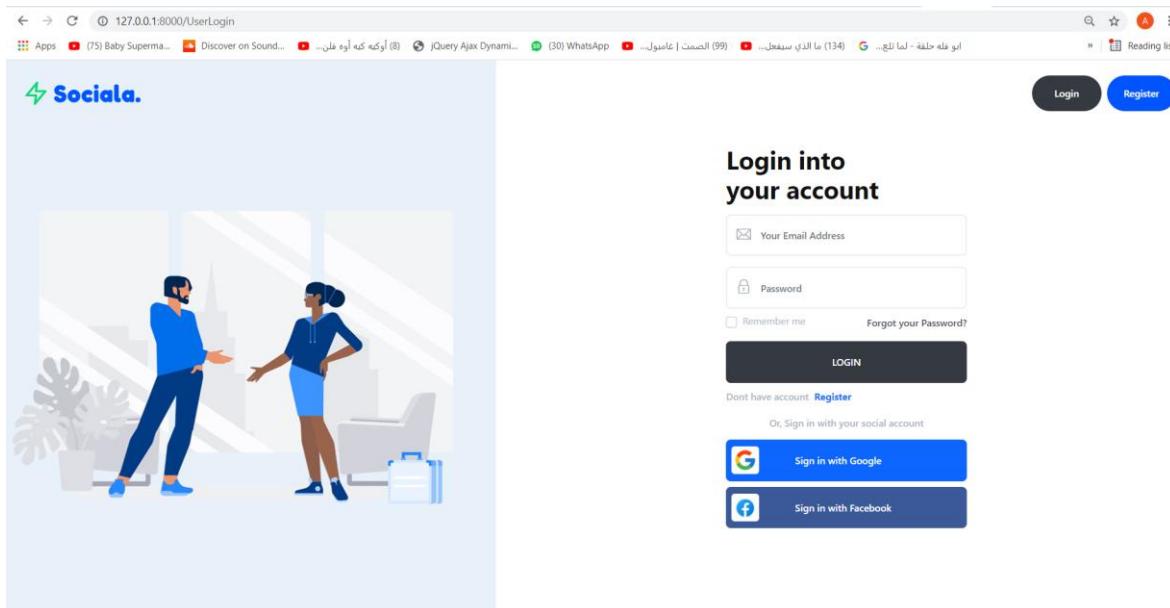
7-Running the Application

Our application can be run using the command below.

```
php artisan serve
```

This will open up a new tab on our browser with the URL below. This is the home page of our Laravel application.

http://localhost:8000



Chapter 8: User Manual

Chapter 9: Conclusion and Future Works

9.1. Conclusion

Our aim was to give real estate web application the advantage of social web application to serve the customer to find any detail in an organized , simple and categorized way while interacting and socializing with other customer. We built our web application in a dynamic way to be able easily add any category without need to do many changes in the system. We have done structured and unstructured interviews with users who use real estate websites then we collected the data extracted from these interviews and we added all their needs to our web application.

9.2. Future Works

We intend to do more improvement and updates in the system to help more our users and make the web application easier, friendly, and efficient:

1. There will be a mobile application for iOS and android.
2. There will be analysis for each property and for all properties for the owner to track his properties progress and history.
3. The customer can select a transportation from the web and there will be a new user type called drivers to upload their personal information and their cars details.
4. Chat between customers will be produced to help them get in touch for more information and sociality.
5. There will be a feature called Event in the web application to be displayed for the customers and all properties near to each event and available in the date of the event will be displayed for the customers to rent and there will be a recommendations to the customers for interesting and famous places near to the customer .
6. More languages to the application will be added to allow many nationalities to access this application.
7. There will be some updates and enhancement in admin panel.
8. Their will be enhancement for showing of each property by virtual tool to be more real.
9. The owner will be able to report misuse in his property to be fixed or bad behavior.
10. The system will send alert to the customers who misuse in the property they rented and will take an action if they repeat it.

11. There will be Analysis for customer reviews to return feedback to suggestion owner how to enhance his properties.
12. Customer will be able to buy a property.