**Requirements Specifications**

**09:Petswala**

|  |  |
| --- | --- |
| **Student ID** | **Name** |
| **22100298** | **Roshan A. Aziz** |
| **22100004** | **Adil Aslam Chaudry** |
| **22100201** | **Mohid Yousaf** |
| **22100111** | **Ayan Tabassum** |
| **22100248** | **Sabahat** |

|  |  |  |
| --- | --- | --- |
| **Content** | **Totals** | **Obtained** |
| Introduction & system actors | 5 | 5 |
| Use case diagram | 10 | 10 |
| Use case descriptions | 20 | 12 |
| Class diagram | 20 | 16 |
| Sequence diagram | 20 | 20 |
| State diagram | 5 | 5 |
| Non-functional requirements | 5 | 5 |
| Who did what | 5 | 5 |
| Review checklist | 5 | 5 |
| Overall formatting/template | 5 | 5 |
| Late submission penalty | -20 |  |
| **Total** | **100** | **88** |
| Review | 20 | **16** |
| **Grand Total** |  |  |

**Table of Contents**

1. Introduction. 3

2. System Actors. 4

3. Use Cases. 5

3.1 Use Case Diagrams. 6

3.2 Description of Use Cases. 7

4. Class Diagram.. 53

4.1 Diagram.. 54

4.2 Description. 55

5. Sequence Diagrams. 55

6. State Diagrams. 80

6.1 Diagram details. 80

6.2 Diagram.. 82

7. Non-functional Requirements / Quality Attributes. 83

8. Who Did What?. 85

9. Review checklist 86

# **1.** **Introduction**

The project comprises an application catered towards pet owners, pet shops, sellers of pet accessories, veterinary doctors, pet rescue volunteers, and pet shelters/rescue teams. The application aims to link all these different entities through a variety of useful and helpful functionalities; pet owners would be able to search for different accessories, as well as finding good veterinary doctors for their pets, making their lives much more comfortable and better. They would also be able to interact with other pet owners and share photos and stories of their own pets as well. They could share helpful information with other pet owners. Any user of the app can also help with the animal rescue process by reporting pets to be rescued to the relevant rescue teams in the area.

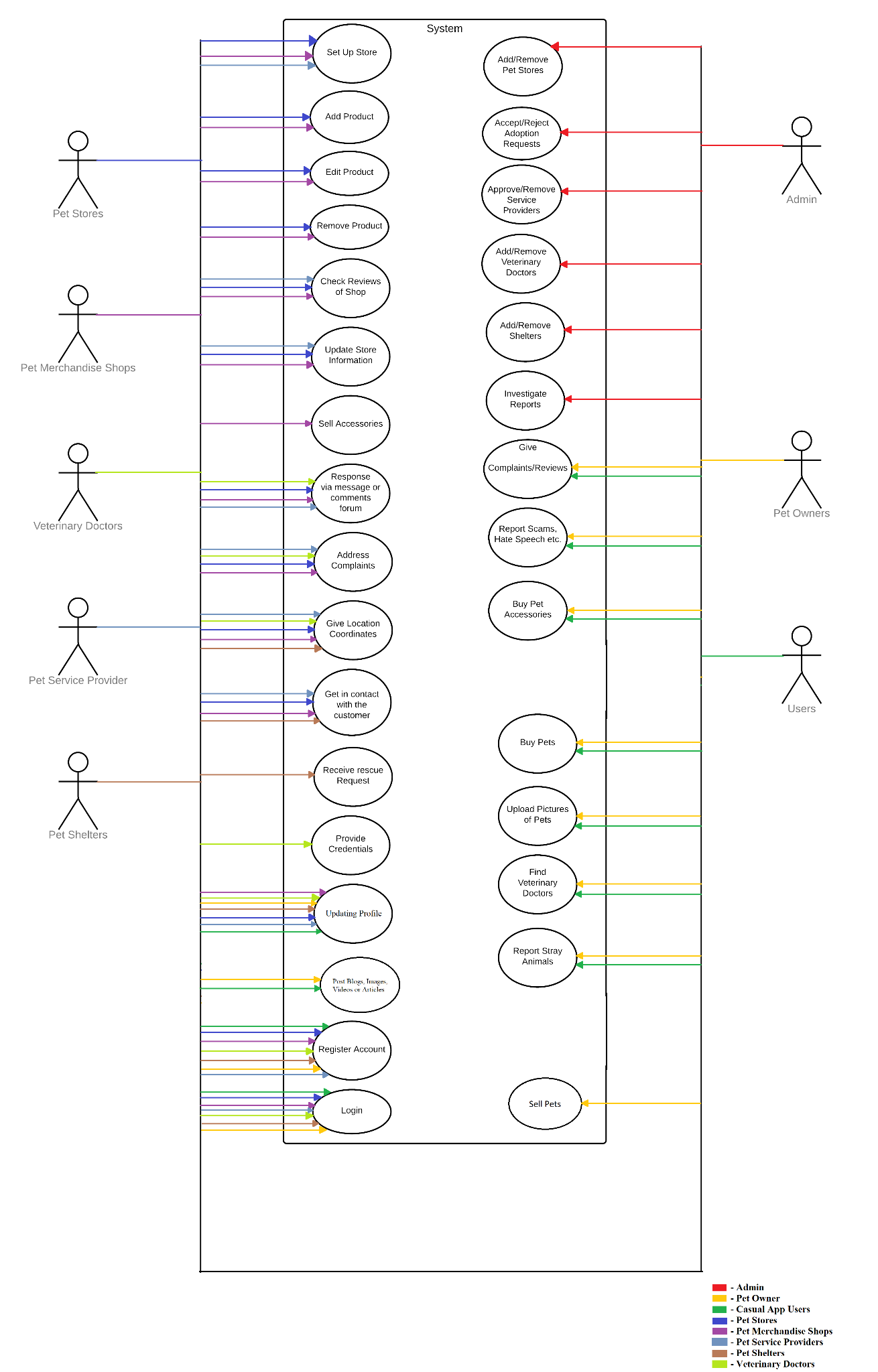
# **2.** **System Actors**

<List down the actor names and give a 2-3 lines description of the role of each actor>

|  |  |
| --- | --- |
| **Actor Name** | **Description** |
| Admin | The admin is in charge of managing the entire application. The admin can add/delete/edit/filter any of the profiles and data in the application. |
| Casual App Owners | They can register themselves and can access various facilities including veterinary opportunities and buying pets and accessories. They can also report pets to the rescue teams. They can also publish pet-related events in their surroundings. |
| Pet Owners | Those who are looking for guidance to treat their pets in the best possible way, can also rate the veterinary-related facilities and give their feedback and reviews. |
| Veterinary Doctors | They can register themselves on the platform, so pet owners can look for veterinary opportunities nearby. |
| Pet Stores | They are selling pets mainly, so people can search for the desired pets. |
| Pet Merchandise Shops | They are selling accessories required for pet care and grooming |
| Pet Service Provider | These are service providers who help the pet owners with various tasks such as pet daycares, dog walkers, pet grooming, etc. |
| Pet Shelters | Pet shelters are places where the rescued pets are provided with a temporary home until they are adopted. A lot of the pet shelters have their own rescue teams. |

# **3.** **Use Cases**

## **3.1** **Use Case Diagrams**



## **3.2** **Description of Use Cases**

[ In general, there are issues with overall content including pre conditions, format of description and alternative paths. See my comments in some selected use cases below for specific examples.]

**3.2.1 Register Account**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-001 | |
| **Purpose** | | The user makes an account. | |
| **Pre-conditions** | | The user has the app installed. | |
| **Post-conditions** | | An account is created for the respective type of user. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The user selects the register/sign up option. | |  |
| 2. | The user fills in their details. [You should mention some details of the information being filled.] | |  |
| 3. | The user selects the type of user they want to be. [Mention one type here and rest can be handled in alternate courses of action.] | |  |
| 4. | The user confirms their information. | |  |
| 5. | An account is created. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | In step 3, if the user selects the store, veterinary, shelter, merchandise option, they will be transferred to another page to set up their store. [What will happen in each of these situations. You should mention one case in the main flow and rest in the alternate courses of actions or if there are separate use cases for each type of selection, they should be highlighted as well.] | |  |
| **Step #** | **Exception Paths** | |  |
|  | The user has filled in all the essential information and selected the type of account they want to create. [What makes this an exception path and when does it occur?] | |  |
|  |  |  |  |

### **3.2.2** **Report Stray Animals**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-002 | |
| **Purpose** | | The user can report the location of the stray animals to nearby animal rescue teams. | |
| **Pre-conditions** | | 1. The *Register Account* use case has been completed. 2. The user is a casual app user or a pet owner. 3. The user has their GPS turned on. | |
| **Post-conditions** | | A post is created with the location of the pet, this can be seen by nearby pet rescue teams. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The user will select the rescue option. | |  |
| 2. | The system displays a map, with the option to pinpoint users own location. | |  |
| 3. | The user selects the gps option to pinpoint their own location. | |  |
| 4. | The user selects the type of animal. | |  |
| 5. | The user selects the post option. | |  |
| 6. | An automatic post, with the location embedded, is created for the rescue teams. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | There are none. | |  |
| **Step #** | **Exception Paths** | |  |
|  | If the GPS is not active, the user is given a pop-up to turn the location services on. | |  |
|  |  |  |  |

### **3.2.3** **Buy Pets**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-003 | |
| **Purpose** | | The casual app users and pet owners can buy pets from the pet store. | |
| **Pre-conditions** | | 1. The user has completed the *Register Account* use case. | |
| **Post-conditions** | | The user will reserve the pet they choose to be purchased and picked up in the store. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The user would select the marketplace option. | |  |
| 2. | The user would browse for the store they want to choose. | |  |
| 3. | The user would select the store. | |  |
| 4. | The system will present the user with the store page. | |  |
| 5. | The user would browse the store to see the pets they have. | |  |
| 6. | The user would select the pet to learn of its breed and name. | |  |
| 7. | The user would reserve the pet by clicking on the reserve button. | |  |
| 8. | The status of the pet would change to reserve on the store page by the system. | |  |
| 9. | A notification will be sent to the store page by the system. | |  |
| 10. | The user would get a notification that they reserved a pet for a few days. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | 1. The user can at any point go back to the store list. 2. After step 4, the user can filter between types of animals to make his/her search easier. | |  |
| **Step #** | **Exception Paths** | |  |
|  | If the animal is already reserved, the user can not reserve the animal. | |  |
|  |  |  |  |

### **3.2.4** **Find Veterinary Doctors**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-004 | |
| **Purpose** | | The user can find the location of a veterinary doctor. | |
| **Pre-conditions** | | 1. The user must have completed the *Register Account* use case. | |
| **Post-conditions** | | The user learns of the location of the veterinary doctor. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The user goes to the services section. | |  |
| 2. | The user browses the services section and looks for the right veterinary doctor. | |  |
| 3. | The user selects a veterinary doctor. | |  |
| 4. | The system shows the user the credentials as well as the information regarding the veterinary doctor. | |  |
| 5. | The user reads the location of the doctor. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | After step 4, the user can click on a map to be led to google maps, with their coordinates. They can then visit the store themselves. | |  |
| **Step #** | **Exception Paths** | |  |
|  | In the alternative path, if the veterinary doctor has not uploaded their address, that path cannot happen. | |  |
|  |  |  |  |

**3.2.5 Check Reviews of Shop**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-005 | |
| **Purpose** | | The user can check reviews of Pet stores,shops, veterinary doctors, and shelters on their store page. | |
| **Pre-conditions** | | 1. The user must have completed the *Register account* use case. 2. The user must be of the correct type (Pet Owners and Casual App Owner). | |
| **Post-conditions** | | The user can see all reviews of a store posted by other users. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The user will go to the marketplace. | |  |
| 2. | The user will select a particular store. | |  |
| 3. | The user selects the information section by pressing on “i”. | |  |
| 4. | The system takes the user to the information section. | |  |
| 5. | The user clicks on the review section | |  |
| 9. | The system makes visible all the reviews of that store. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | None. | |  |
| **Step #** | **Exception Paths** | |  |
|  | None. | |  |
|  |  |  |  |

**3.2.6 Provide Credentials**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-006 | |
| **Purpose** | | The veterinary doctors can edit information about themselves on their store page. | |
| **Pre-conditions** | | 1. The user must have completed the *Register account* use case. 2. The user must be of the correct type. | |
| **Post-conditions** | | The credentials of the doctors will be uploaded on the store page of the veterinary doctor. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The user selects the profile. | |  |
| 2. | The user selects the store page option. | |  |
| 3. | The system takes the user to the store page. | |  |
| 4. | The user selects the information section by pressing on “i”. | |  |
| 5. | The system takes the user to the information section. | |  |
| 6. | The user selects the pencil icon to add their information. | |  |
| 7. | The user browses the form to find the correct text field. | |  |
| 8. | The user enters their credentials. | |  |
| 9. | The user presses the “save button” | |  |
| 10. | The system has updated their credentials. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | The user can add their credentials at the time of setting up their store. | |  |
| **Step #** | **Exception Paths** | |  |
|  | None. | |  |
|  |  |  |  |

**3.2.7 Update Store Information**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-007 | |
| **Purpose** | | The stores,shops, veterinary doctors, shelters can edit information about themselves on their store page. | |
| **Pre-conditions** | | 1. The user must have completed the *Register account* use case. 2. The user must be of the correct type. | |
| **Post-conditions** | | The information will be updated on the store page. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The user selects the profile. | |  |
| 2. | The user selects the store page option. | |  |
| 3. | The system takes the user to the store page. | |  |
| 4. | The user selects the information section by pressing on “i”. | |  |
| 5. | The system takes the user to the information section. | |  |
| 6. | The user selects the pencil icon to add their information. | |  |
| 7. | The user browses the form to find the correct text field. | |  |
| 8. | The user enters new information. | |  |
| 9. | The user presses the “save button” | |  |
| 10. | The system has updated the store’s information. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | None. | |  |
| **Step #** | **Exception Paths** | |  |
|  | None. | |  |
|  |  |  |  |

**3.2.8 Receive Rescue Request**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-008 | |
| **Purpose** | | The user will get the request of an animal in need and can update the status of the animal rescue. | |
| **Pre-conditions** | | 1. The user must have completed the *Register account* use case. 2. The user must be of the correct type. 3. There should be a request made by one of the casual app users or pet owners. | |
| **Post-conditions** | | The user will get the notification and use the location. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The user gets the notification. | |  |
| 2. | The user taps on the notification. | |  |
| 3. | The system takes the user to the post. | |  |
| 4. | The user selects the location coordinates. | |  |
| 5. | The system gives the location of the stray animal. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | The user can reply to the customer afterwards about the success of the rescue or not. | |  |
| **Step #** | **Exception Paths** | |  |
|  | None. | |  |
|  |  |  |  |

### **3.2.9** **Buy Pet Accessories**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-009 | |
| **Purpose** | | The user can buy accessories for their pets. | |
| **Pre-conditions** | | 1. The user must have completed the *Register Account* use case. 2. The user must be of type *Pet Owner.* | |
| **Post-conditions** | | The accessories are dispatched, and the user can buy cash on delivery or visit the store itself. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The user selects the marketplace option. | |  |
| 2. | The user browses the marketplace to find the merchandise shop they want to buy from. | |  |
| 3. | The user selects the store. | |  |
| 4. | The system presents the user with the store page. | |  |
| 5. | The user browses the store page to find the accessories they are looking for. | |  |
| 6. | The user selects the accessories they want to purchase. | |  |
| 7. | The user confirms the purchase and selects the cash on purpose option. | |  |
| 8. | The user enters their information. | |  |
| 9. | The user confirms the order. | |  |
| 10. | The system sends a notification to the seller. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | 1. The user can at any point go back to the store list. 2. After step 4, the user can filter between types of accessories to make his/her search easier. 3. The user can continue to add items after step 6. | |  |
| **Step #** | **Exception Paths** | |  |
|  | If the item is out of stock, the user cannot order it. | |  |
|  |  |  |  |

### **3.2.10** **Add/Remove Pet Stores**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-010 | |
| **Purpose** | | Accepting or removing pet store requests. | |
| **Pre-conditions** | | The user has admin access. | |
| **Post-conditions** | | The requester is notified of the decision. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
| 1. | The user views a request of the list of submitted requests. | |  |
| 2. | The user goes through the request details and contacts requesters. | |  |
| 3. | The admin accepts the request. | |  |
| 4. | System notifies the user of the decision. | |  |
|  |  | |  |
|  |  | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | In step 3, the request gets rejected, the user gets notified and their info is discarded. | |  |
| **Step #** | **Exception Paths** | |  |
|  | In step 2, if mandatory details are not provided, (contact, mandatory questions etc..) the request doesn’t proceed. | |  |
|  |  |  |  |

### **3.2.11** **Add/Remove Veterinary Doctors**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-011 | |
| **Purpose** | | Accepting or removing veterinary doctors. | |
| **Pre-conditions** | | The user has admin access. | |
| **Post-conditions** | | The requester is notified of the decision. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
| 1. | The user views a request of the list of submitted requests. | |  |
| 2. | The user goes through the request details and contacts requesters. | |  |
| 3. | The admin accepts the request. | |  |
| 4. | System notifies the user of the decision. | |  |
|  |  | |  |
|  |  | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | In step 3, the request gets rejected, the user gets notified and their info is discarded. | |  |
| **Step #** | **Exception Paths** | |  |
|  | In step 2, if mandatory details are not provided, (contact, mandatory questions etc..) the request doesn’t proceed. | |  |
|  |  |  |  |

### **3.2.12** **Add/Remove Shelters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-012 | |
| **Purpose** | | Accepting or removing animal shelters. | |
| **Pre-conditions** | | The user has admin access. | |
| **Post-conditions** | | The requester is notified of the decision. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
| 1. | The user views a request of the list of submitted requests. | |  |
| 2. | The user clicks on request details of a particular request. | |  |
| 3. | The user goes through the request details and contacts requesters. | |  |
| 4. | The user accepts the request. | |  |
| 6. | Requesters get notified of the decision by the system. | |  |
|  |  | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | In step 4, the request gets rejected, the user gets notified and their info is discarded. | |  |
| **Step #** | **Exception Paths** | |  |
|  | In step 3, if mandatory details are not provided, (contact, mandatory questions etc..) the request doesn’t proceed. | |  |
|  |  |  |  |

### **3.2.13** **Get in contact with the Customer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-013 | |
| **Purpose** | | The pet stores,shelters, merchandise shops can chat with their customers. | |
| **Pre-conditions** | | 1. The types of communicator and the one they are communicating should be appropriate. [How would this pre-condition be satisfied?] 2. The shops have entered correct communication information. [How would the correctness of information be ensured?] | |
| **Post-conditions** | | The customer and the seller can get in contact with each other. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The shop will open the customer's profile by looking through their orders.  [Is shop an actor? Not listed in your actor list?] | |  |
| 2. | The shops will press the chat button. | |  |
| 3. | The shops will be led to the chat window, by the system. | |  |
| 4. | The shops will type the text they want. | |  |
| 5. | The shops will send the text.  [Does chat mean to send a single message once? Isn’t it two-way communication?] | |  |
|  |  | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | None | |  |
| **Step #** | **Exception Paths** | |  |
|  | There are none. | |  |
|  |  |  |  |

### **3.2.14** **Post Blog, Videos or Articles**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-014 | |
| **Purpose** | | The user can post anything pet related, whether text, video or images.. | |
| **Pre-conditions** | | 1. The user must have completed the *Register account* use case. 2. The user must be of Pet Owners or Casual App Users type. | |
| **Post-conditions** | | The user's post is generated on a News Feed for everyone else to see. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The user goes to the News Feed, by selecting the News Feed. | |  |
| 2. | The user selects the “What's On Your Mind” section on the top of the News Feed. | |  |
| 3. | The user adds the text, image and/or video on the post. | |  |
| 4. | The user selects the post button. | |  |
| 5. | The post is created and posted on the News Feed, by the system. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | In steps 2, 3, and 4, the user can press the cancel button and go directly to step 6. | |  |
| **Step #** | **Exception Paths** | |  |
|  | If the post is empty, the system would give an error on pressing post. | |  |
|  |  |  |  |

### **3.2.15** **Add Product**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-015 | |
| **Purpose** | | The shops and stores can add a product(s) from their inventory on their store page. | |
| **Pre-conditions** | | 1. The *Register account* use case has been completed. 2. The users are of the correct type. | |
| **Post-conditions** | | The user will add a new item(s) or animal(s) in their inventory. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The user will visit their store page. | |  |
| 2. | The user would select on the Manage Inventory section | |  |
| 3. | The user selects the Add Product button. | |  |
| 4. | The system takes user to the Add Product page. | |  |
| 5. | The user enters the details and image of the new product. | |  |
| 6. | The user presses the add button. | |  |
| 7. | The user is shown a prompt to confirm their action, by the system. | |  |
| 8. | The user presses the confirm button. | |  |
| 9. | The product is added to their inventory by the system and is visible on the store page. | |  |
| 10. | The user is taken back to the Manage Inventory page, by the system. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | In steps 4, 5, 6, and 7, the user can press the cancel button and go directly to step 10. | |  |
| **Step #** | **Exception Paths** | |  |
|  | None | |  |
|  |  |  |  |

### **3.2.16** **Edit Product**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-016 | |
| **Purpose** | | The shops and stores can edit a product(s) from their inventory on their store page. | |
| **Pre-conditions** | | 1. The *Register account* use case has been completed. 2. The users are of the correct type. | |
| **Post-conditions** | | The user will edit an existing item(s) or animal(s) in their inventory. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The user will visit their store page. | |  |
| 2. | The user would select on the Manage Inventory section | |  |
| 3. | The user selects the Edit Product button. | |  |
| 4. | The system takes the user to the Edit Product page. | |  |
| 5. | The user presses the edit item button on an existing item in the inventory. | |  |
| 6. | The user enters the details and/or image of the product. | |  |
| 7. | The user presses the edit button. | |  |
| 8. | The user is shown a prompt to confirm their action, by the system. | |  |
| 9. | The user presses the confirm button. | |  |
| 10. | The product is updated in their inventory by the system and is visible on the store page. | |  |
| 11. | The user is taken back to the Manage Inventory page, by the system. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | In steps 5, 6, 7, and 8, the user can press the cancel button and go directly to step 11. | |  |
| **Step #** | **Exception Paths** | |  |
|  | None | |  |
|  |  |  |  |

### 

|  |
| --- |
|  |
|  |

### **3.2.17** **Give Location Coordinates**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-017 | |
| **Purpose** | | The store, veterinary doctors, shops, services can give their locations on their pages. | |
| **Pre-conditions** | | 1. The users must have completed the *Register account* use case. 2. The users must be of the correct type. | |
| **Post-conditions** | | The user's location will be given on their store page. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The user visits their profile. | |  |
| 2. | The user selects their store page. | |  |
| 3. | The user goes to the information section by pressing the button “i”. | |  |
| 4. | The user presses the pencil icon. | |  |
| 5. | A form is opened by the system. | |  |
| 6. | The user goes to the section of location. | |  |
| 7. | The user clicks on the add location button, | |  |
| 8. | The user uploads their location. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | The location can be uploaded on the time of creating the account/store page. | |  |
| **Step #** | **Exception Paths** | |  |
|  | If GPS is off on step 7, the GPS will need to be turned on. | |  |
|  |  |  |  |

### **3.2.18** **Set up Store**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-018 | |
| **Purpose** | | Set-up Store | |
| **Pre-conditions** | | The owner has a registered account. | |
| **Post-conditions** | | The store is visible to the general public upon search to buy pets/ accessories from. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
| 1. | The user submits a request for making a pet store, using relevant workflow.  (make a store etc.) | |  |
| 2. | The user enters relevant details, items available, address contact, pictures | |  |
| 3. | The account request goes to the admin for review. | |  |
| **4.** | The request gets approved. [You should mention here that some use case such as “Approve” completes this step.] | |  |
| 5. | The user gets a notification and access to their store online, by the system. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | In step 4, the request gets rejected, the user gets notified and their info is discarded. | |  |
| **Step #** | **Exception Paths** | |  |
|  | In step 2, if mandatory details are not provided, (contact, enough pictures etc.) the request doesn’t proceed. | |  |
|  |  |  |  |

### **3.2.19** **Accept/Reject adoption Requests**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-019 | |
| **Purpose** | | Accepting or rejecting the requests people place for adoption of pets | |
| **Pre-conditions** | | The user has admin access. | |
| **Post-conditions** | | The requester is notified of the decision. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
| 1. | The admin views a request of the list of submitted requests. | |  |
| 2. | The admin goes through the request details and contacts requesters. | |  |
| 3. | The admin accepts the request. | |  |
| **4.** | Users get notified of the decision by the system. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | In step 3, the request gets rejected, the user gets notified and their info is discarded. | |  |
| **Step #** | **Exception Paths** | |  |
|  | In step 2, if mandatory details are not provided, (contact, mandatory questions etc..) the request doesn’t proceed. | |  |
|  |  |  |  |

### **3.2.20** **Approve/Remove Service Providers**

### 

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-020 | |
| **Purpose** | | Accepting or removing service providers requests | |
| **Pre-conditions** | | The user has admin access. | |
| **Post-conditions** | | The requester is notified of the decision. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
| 1. | The admin views a request of the list of submitted requests. | |  |
| 2. | The admin goes through the request details and contacts requesters. | |  |
| 3. | The admin accepts the request. | |  |
| 4. | Users get notified of the decision by the system. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | In step 3, the request gets rejected, the user gets notified and their info is discarded. | |  |
| **Step #** | **Exception Paths** | |  |
|  | In step 2, if mandatory details are not provided, (contact, mandatory questions etc..) the request doesn’t proceed. | |  |
|  |  |  |  |

### **3.2.21 Upload Pictures of Pets**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-021 | |
| **Purpose** | | Sharing pets pictures on the platform | |
| **Pre-conditions** | | The user has a registered account. | |
| **Post-conditions** | | The shared pictures show up in other users' feeds. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
| 1. | The user logs in to their account. | |  |
| 2. | The user chooses the ‘share’ option and selects pictures from their gallery. | |  |
| 3. | The user enters a text description for the post. | |  |
| 4. | The user presses the post button. | |  |
| **4.** | The post gets uploaded on the News Feed, by the system. | |  |
| **5.** | The user is taken back to the News Feed, by the system. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | In steps 2, 3, and 4, the user can press the cancel button and go directly to step 5. | |  |
| **Step #** | **Exception Paths** | |  |
|  |  | |  |
|  |  |  |  |

### **3.2.22** **Give Complaints/ Reviews**

### 

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-022 | |
| **Purpose** | | The users submit their complaints or give reviews of the services they have availed. | |
| **Pre-conditions** | | The user has availed the respective service already. | |
| **Post-conditions** | | The complaint and reviews are displayed on the vet/service providers profile, and is visible to all users. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
| 1. | The user visits the profile of the veterinary doctor or service provider and chooses to ‘give feedback’. | |  |
| 2. | The user writes the review and submits it. | |  |
| 3. | The service provider gets notified of the review/complaint, by the system. | |  |
| **4.** | The review gets displayed on the clinic page/store page, by the system. | |  |
| **5.** | The user is taken back to the profile of the veterinary doctor or service provider, by the system. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | In step 2, the user can cancel the process and skip to step 5. | |  |
| **Step #** | **Exception Paths** | |  |
|  | None | |  |
|  |  |  |  |

### **3.2.23 Report Scams/ Hate Speech**

### 

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-023 | |
| **Purpose** | | The users can report a service provider or any hate speech they observe on the platform. | |
| **Pre-conditions** | | The user has an account and has availed the service. | |
| **Post-conditions** | | The admin is notified of the report. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
| 1. | The user visits the profile of the concerned account/ service provider. | |  |
| 2. | The user scrolls to the specific post or comment, if any. | |  |
| 3. | The user chooses the ‘report’ option and gives details of the reasons to report. | |  |
| **4.** | The user submits the report. | |  |
| **5.** | The admin is notified of the report to review. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | In step 3, the user can cancel the process. | |  |
| **Step #** | **Exception Paths** | |  |
| 5. | In case of a fake report/not enough information, the report gets discarded. | |  |
|  |  |  |  |

### **3.2.24** **Address Complaints**

### 

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-024 | |
| **Purpose** | | The store owners, service providers or vets can respond to the complaints received. | |
| **Pre-conditions** | | The user has respective account access. | |
| **Post-conditions** | | The complainant is notified of the answer. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
| 1. | The user views the list/ notifications of complaints on their profile/ store. | |  |
| 2. | The user writes the response to the complaint. | |  |
| 3. | The user confirms and submits the response. | |  |
| 4 | Complainants get notified of the store/ service provider’s response, by the system. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | In step2 and 3, The user can cancel the process of writing a response. | |  |
| **Step #** | **Exception Paths** | |  |
|  | None | |  |
|  |  |  |  |

### **3.2.25 Sell Accessories**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-025 | |
| **Purpose** | | The store owners can sell accessories on their online stores. | |
| **Pre-conditions** | | The user has an online store set up already. | |
| **Post-conditions** | | The accessories are dispatched and the cash is handed on delivery/in-person. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The user gets notified of the customer wishing to buy a certain accessory. | |  |
| 2. | The request is sent by the system in the ‘orders to be fulfilled’ section on the seller dashboard. | |  |
| 3. | The user changes the status of the order to ‘out for delivery’ after dispatching the package for delivery. | |  |
| 4. | The user changes the status to ‘order completed’ after the package is delivered. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | The order can be cancelled before step 3, and the request will be deleted from the seller's list. | |  |
| **Step #** | **Exception Paths** | |  |
|  | In case of a wrong address or non-payment of the package, it gets returned and the order is cancelled by the seller. | |  |
|  |  |  |  |

### **3.2.26 Login**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-0026 | |
| **Purpose** | | The user can access the account and avail the respective facilities authorized to  their account. | |
| **Pre-conditions** | | The user has registered an account on the app. | |
| **Post-conditions** | | The user can do different tasks, according to their type of account. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The user goes to the homepage of the app. | |  |
| 2. | The user enters their credentials. | |  |
| 3. | They click on ‘Login’ and get logged in to their account, by the system. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
| 1 | If the user does not have an account already, they can click on ‘sign up’ and proceed to create an account. | |  |
| **Step #** | **Exception Paths** | |  |
| 3 | In case of wrong credentials, an error message is displayed and page is redirected to the login screen. | |  |
|  |  |  |  |

### **3.2.27 Response via message or Comment forum**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-027 | |
| **Purpose** | | The store owners, service providers or vets can communicate with their customers. | |
| **Pre-conditions** | | The user has an online store or a service provider account. | |
| **Post-conditions** | | The messages and responses are displayed to the customers. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The user opens the message box or notifications for a new comment. | |  |
| 2. | The user chooses a certain message/ comment and writes their response in the message/reply box. | |  |
| 3. | The user posts the comment/ sends the message. | |  |
| 4. | The sent message/ comment is displayed on the screen by the system and the customer is notified. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | The user can cancel the process until step 2. | |  |
| **Step #** | **Exception Paths** | |  |
|  | None | |  |
|  |  |  |  |

### **3.2.28 Updating Profile**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-0028 | |
| **Purpose** | | The user can make desired changes to their profile. | |
| **Pre-conditions** | | The user has an account and is logged in. | |
| **Post-conditions** | | The changes are effectively displayed on the profile. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The user clicks on ‘profile settings’ on their account. | |  |
| 2. | The user makes the changes they want to (which may include,changing profile picture, updating interests and other details). | |  |
| 3. | The users saves the made changes. | |  |
| 4. | The new information is updated by the system and is visible to all the users visiting the profile. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | The user can cancel the process before step 3. | |  |
| **Step #** | **Exception Paths** | |  |
|  | None | |  |
|  |  |  |  |

### **3.2.29 Investigate Reports**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-029 | |
| **Purpose** | | The admin can investigate the report and take appropriate action. | |
| **Pre-conditions** | | The user has admin rights. | |
| **Post-conditions** | | The reported account/post is deleted or the report gets discarded. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The user views list/ notifications of the submitted reports. | |  |
| 2. | The user clicks on a report to review. | |  |
| 3. | The user deletes the account/ post after reviewing, or discards the report. | |  |
| 4. | The changes of deleted accounts/posts are visible to app users, after the system has processed the deletion or non deletion. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | At step 2, the user can send any message asking for more clarification,or to give a warning. | |  |
| **Step #** | **Exception Paths** | |  |
|  | None | |  |
|  |  |  |  |

### **3.2.30 Sell Pets**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-0030 | |
| **Purpose** | | The pet store owners can sell their pets. | |
| **Pre-conditions** | | The user has a store set up. | |
| **Post-conditions** | | The pet is sent to the customer and the amount is paid. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The user views list/ notifications of the reserved pets in the ‘orders in progress’ section. | |  |
| 2. | The user changes the status of the order after sending the pet for delivery/ handing in-person. | |  |
| 3. | The order is marked as completed after receiving cash on delivery or in-person. | |  |
| 4. | The sold pet is removed from available for sale pets and the changes are visible to the app users. | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | The customer can cancel the order before step 3, the order gets deleted from the list and the pet is available for sale for other users. | |  |
| **Step #** | **Exception Paths** | |  |
|  | In case of non-payment of the amount, the order is cancelled by the user and the pet is available for sale to other customers. | |  |
|  |  |  |  |

### **3.2.31** **Remove Product**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | UC-031 | |
| **Purpose** | | The shops and stores can remove a product(s) from their inventory on their store page. | |
| **Pre-conditions** | | 1. The *Register account* use case has been completed. 2. The users are of the correct type. | |
| **Post-conditions** | | The user will remove a new item(s) or animal(s) in their inventory. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The user will visit their store page. | |  |
| 2. | The user would select on the Manage Inventory section | |  |
| 3. | The user selects the Remove Product button. | |  |
| 4. | The system takes the user to the Remove Product page. | |  |
| 5. | The user clicks on the delete button next to a product available in their inventory. | |  |
| 6. | The system shows the user a prompt to confirm their action. | |  |
| 7. | The user presses the confirm button. | |  |
| 8. | The product is removed from their inventory and the store page, by the system. | |  |
| 9. | The user is taken back to the Manage Inventory page, by the system. | |  |
|  |  | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | In steps 4, 5, and 6, the user can press the cancel button and go directly to step 9. | |  |
| **Step #** | **Exception Paths** | |  |
|  | None | |  |
|  |  |  |  |

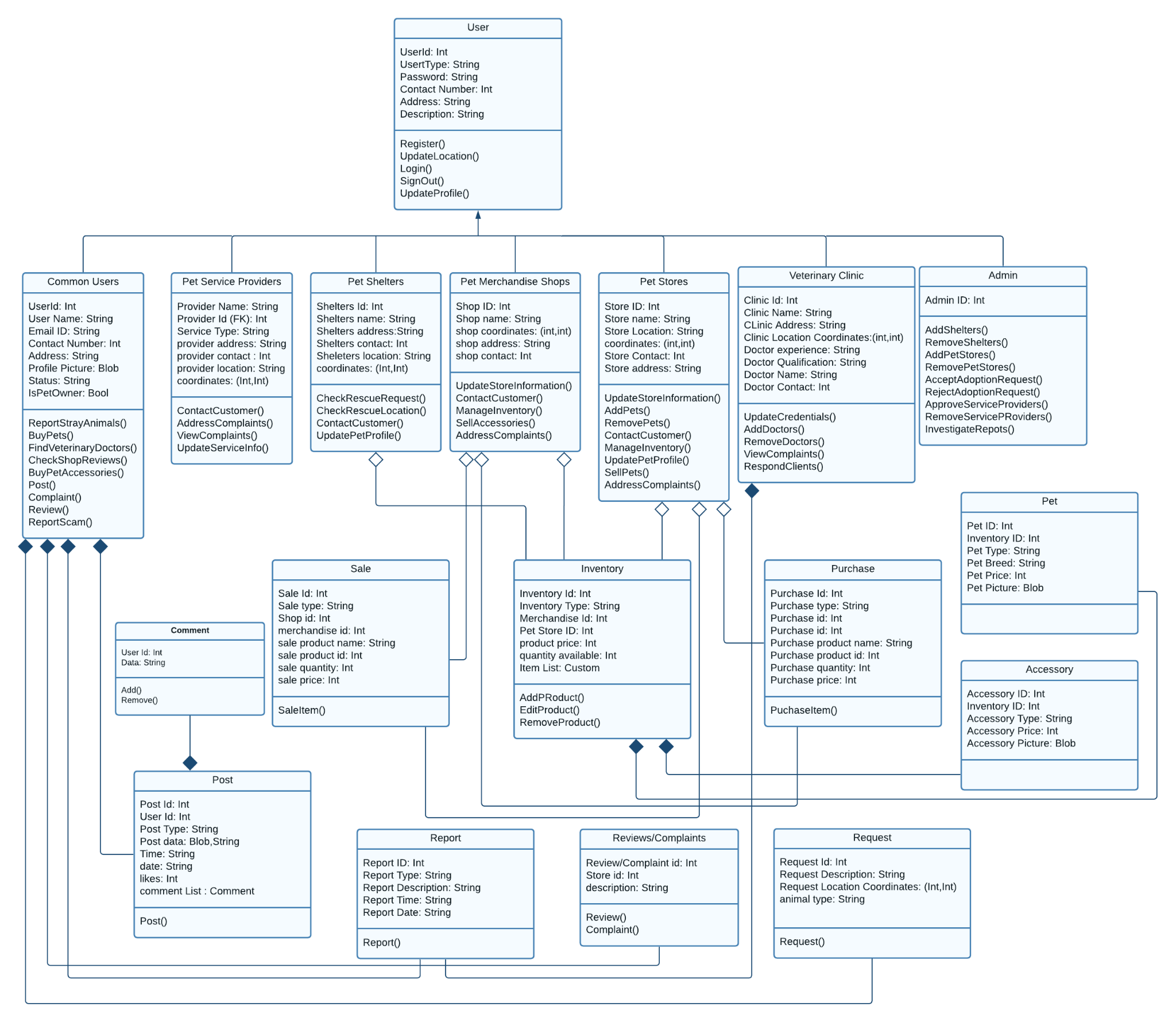
### 

# **4.** **Class Diagram**

## **4.1** **Diagram**

* The user roles must be modeled in a way that they can be granted and revoked on need basis.
* Missing classes for orders,

Comments: As discussed in meeting, the roles are rigid to the users, so we have specified the methods accordingly in the case where we can provide flexibility, we have provided as in the case of common users, where we have added a bool whether it is a common user or it can be a pet owner.



## **4.2** **Description**

<Give brief description/purpose of each class in the class diagram. Give readable names to classes, attributes and operations.>

**User** : It is a user of an application that can be subdivided into common users, Service Providers, Shelters , Veterinary Clinic, Pet Store and admin and it can register.

**Common Users:** Common users are those who are casual app owners and they can be pet owners, they have the functionalities of reporting stray animals, buy pets, finding veterinary options, post and review etc

**Pet Service Providers:** They are providing services related to pets, they can also update their services and address complaints if any.

**Pet Shelters:** Pet Shelters are entertaining rescue requests mainly.

**Pet Merchandise Shops:** Pet merchandise shops are the stores which sell products related to pets.

**Pet Stores:** Pet stores are the places where people can buy pets from. These are physical stores which are set up online on our app as well.

**Veterinary Clinic:** Veterinary clinic is a place where one or more than one veterinary doctors sit. These are also physical clinics with an online clinic set up on our app.

**Admin:** Admin is the owner of the app and they have special rights. They can add, remove people on the app, and modify several other things.

**Comment:** A comment is a message a user leaves on a post, video, blog or any advertisement of a product.

**Sale:** A sale is the exchange of goods (accessories or pets) in return for money. Sales on our app will be made in person for pets and can be cash on delivery for other accessories.

**Inventory:** An inventory is a list of items available at a particular store.

**Purchase:** A purchase is users buying the items in exchange for money. A purchase can be made for pets or accessories.

**Pet:** A pet is an animal kept by people for companionship or entertainment. A pet, on our app, can be in a store for sale, shelter for adoption or might belong to app users.

**Accessory:** An accessory is a product which might be used for decorative or useful purposes. An accessory is usually related to pets in the context of our app.

**Post:** A post is an image, video or text shard by an app user, which is visible to other app users and they can react or comment on it.

**Report:** A report is made by app users to draw admin’s attention to something which should not be on the app. The reports are reviewed by the admin and the relevant action is taken.

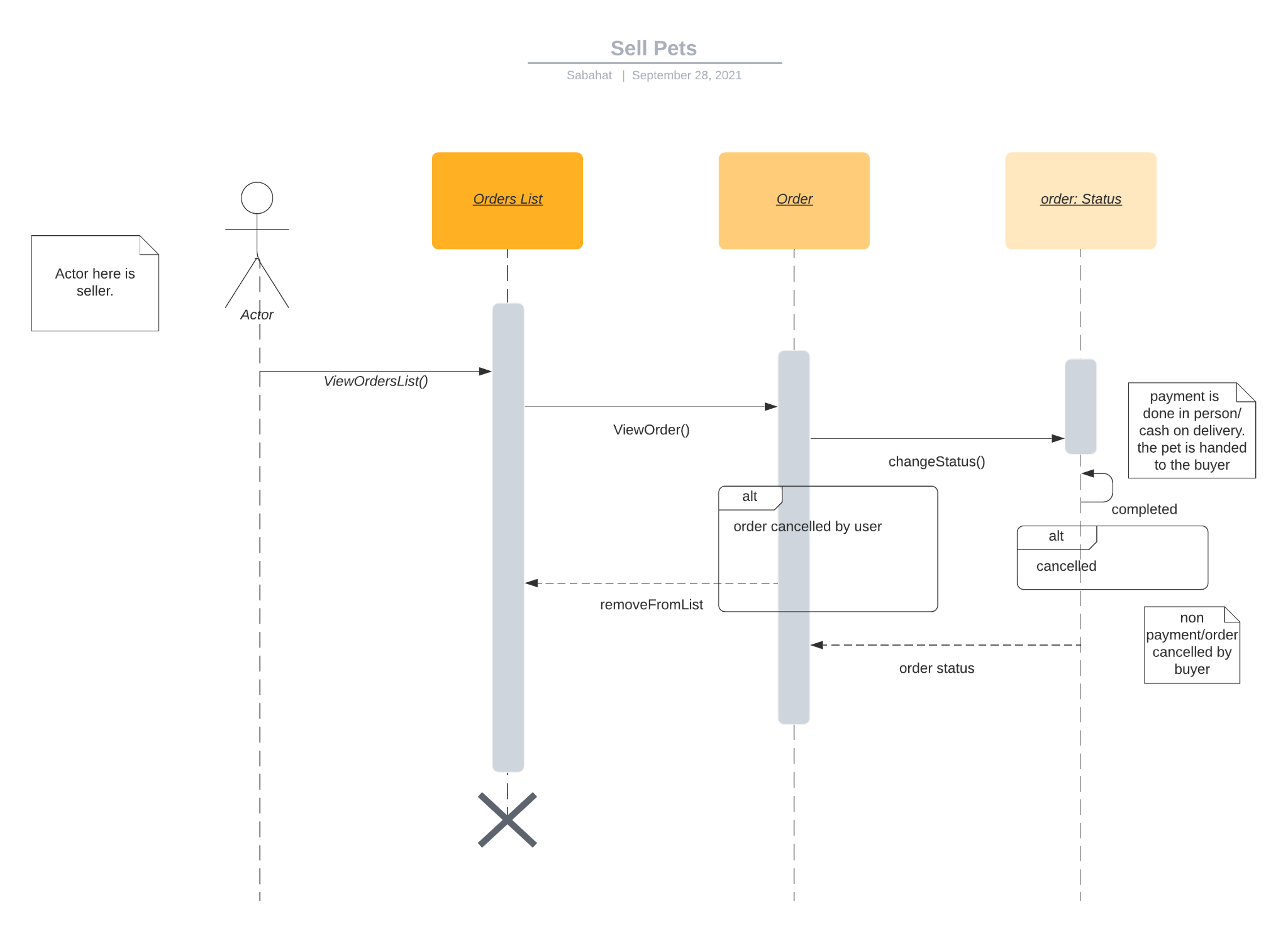
**Reviews / Complaints:** Reviews and complaints are the messages sent to store owners by buyers, to give feedback on their product or to register their complaints.

**Request:** A request is an application submitted by stores, shelters, vets and service providers to get permission to be allowed to sell their services on the app. The admin approves or deletes the request.

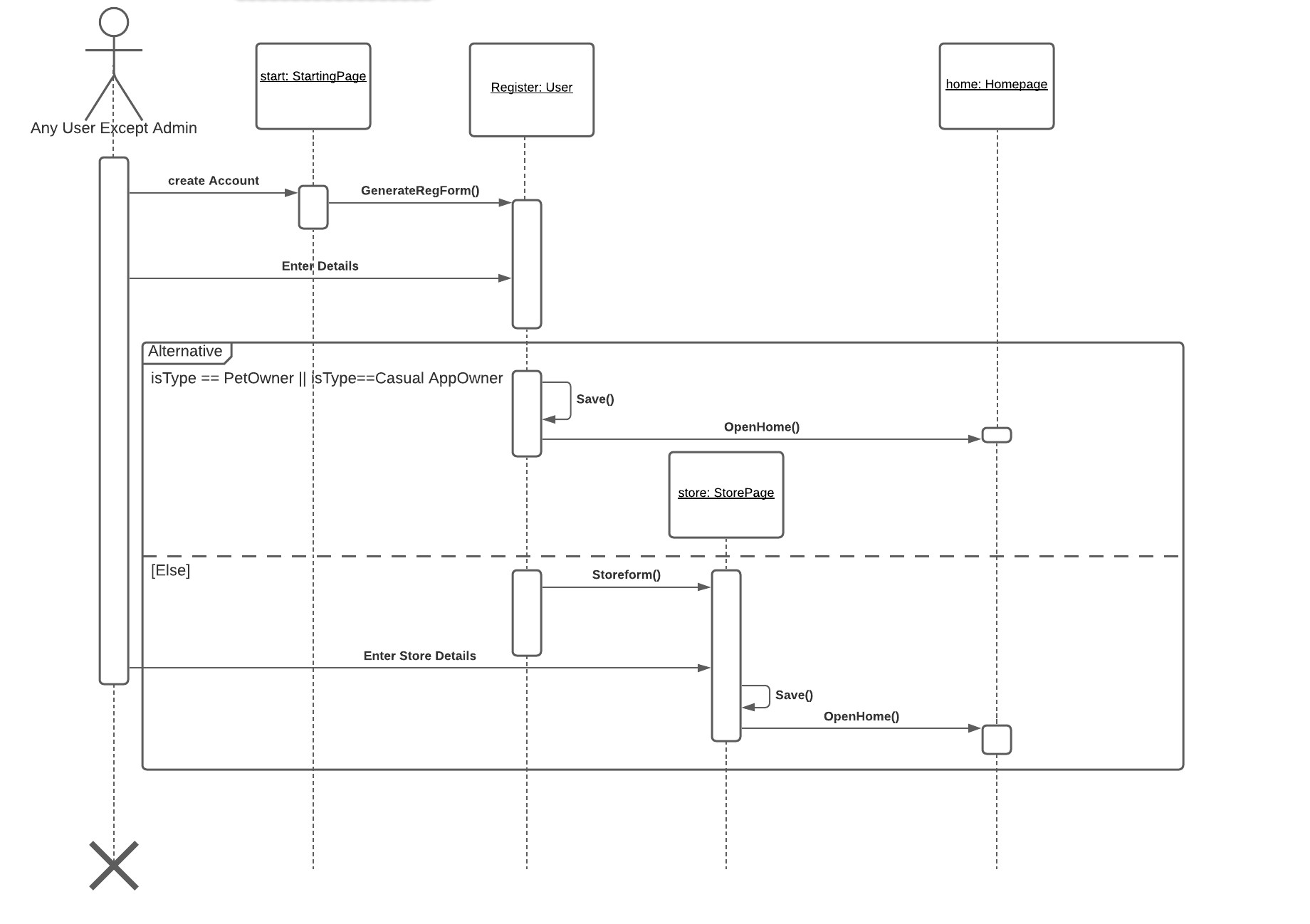
# **5.** **Sequence Diagrams**

<Create sequence diagram for each use case.>

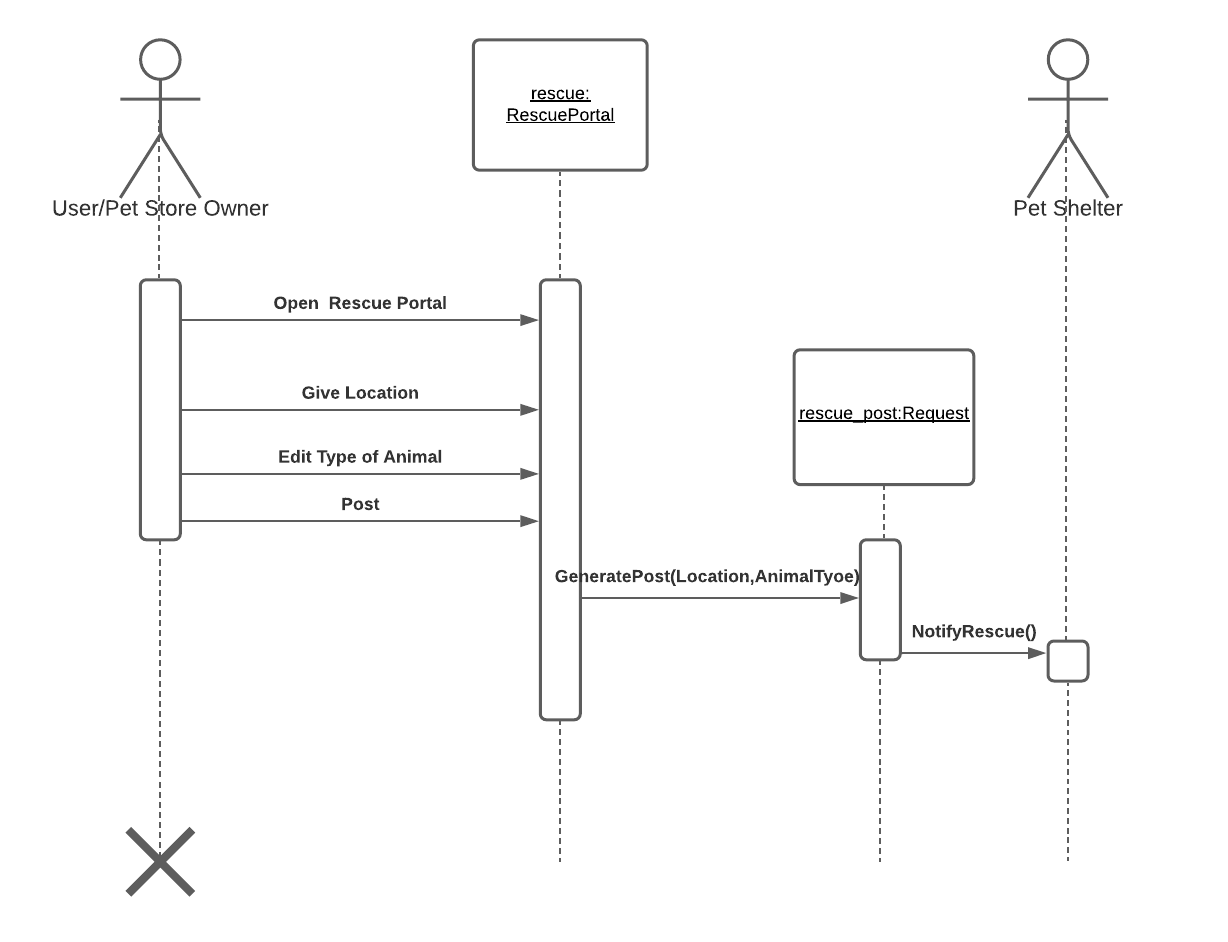
## **5.1** **Sell Pets**



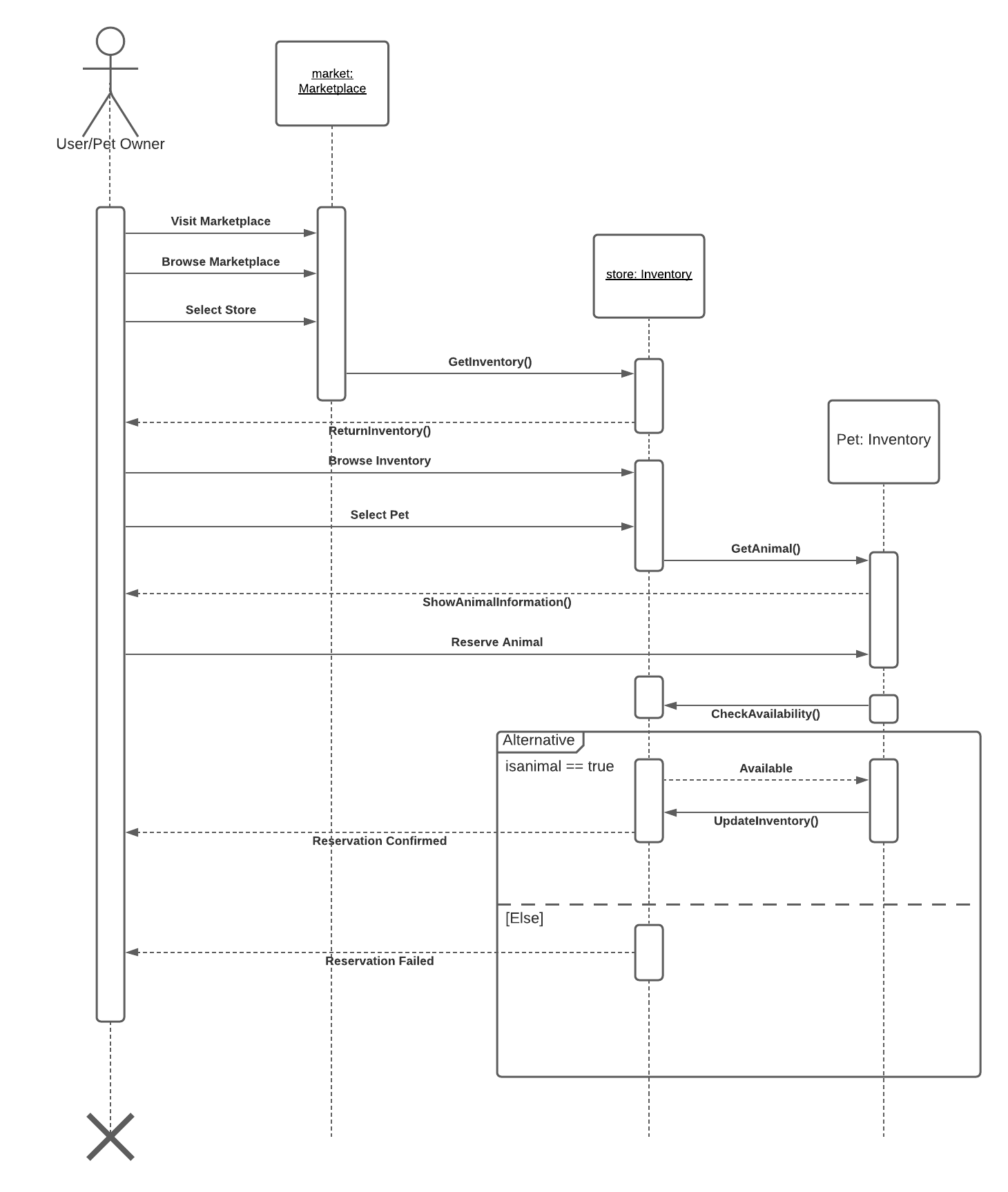
## **5.2** **Register Account**



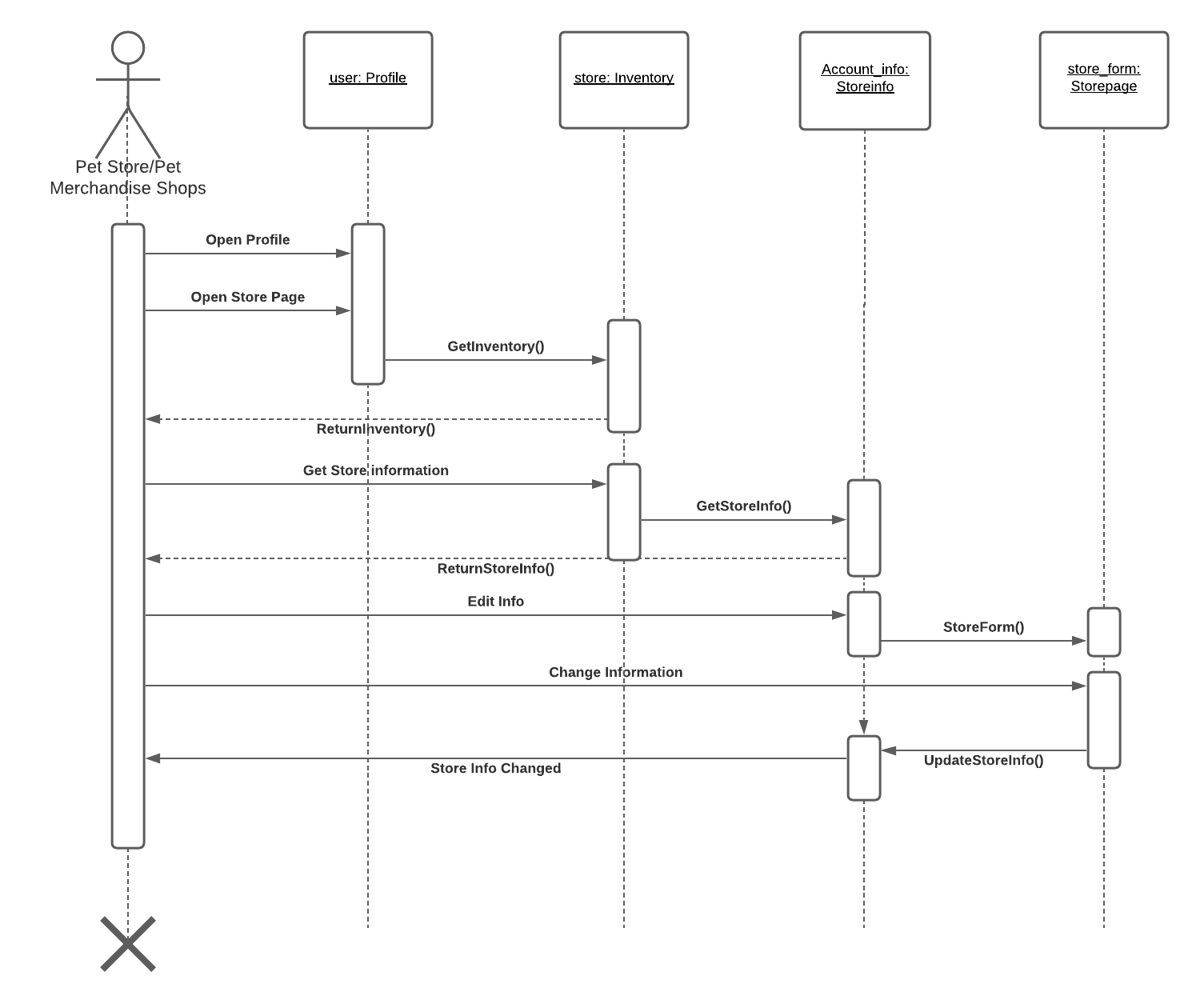
## **5.3** **Report Stray Animals**



## **5.4** **Buy Pets**



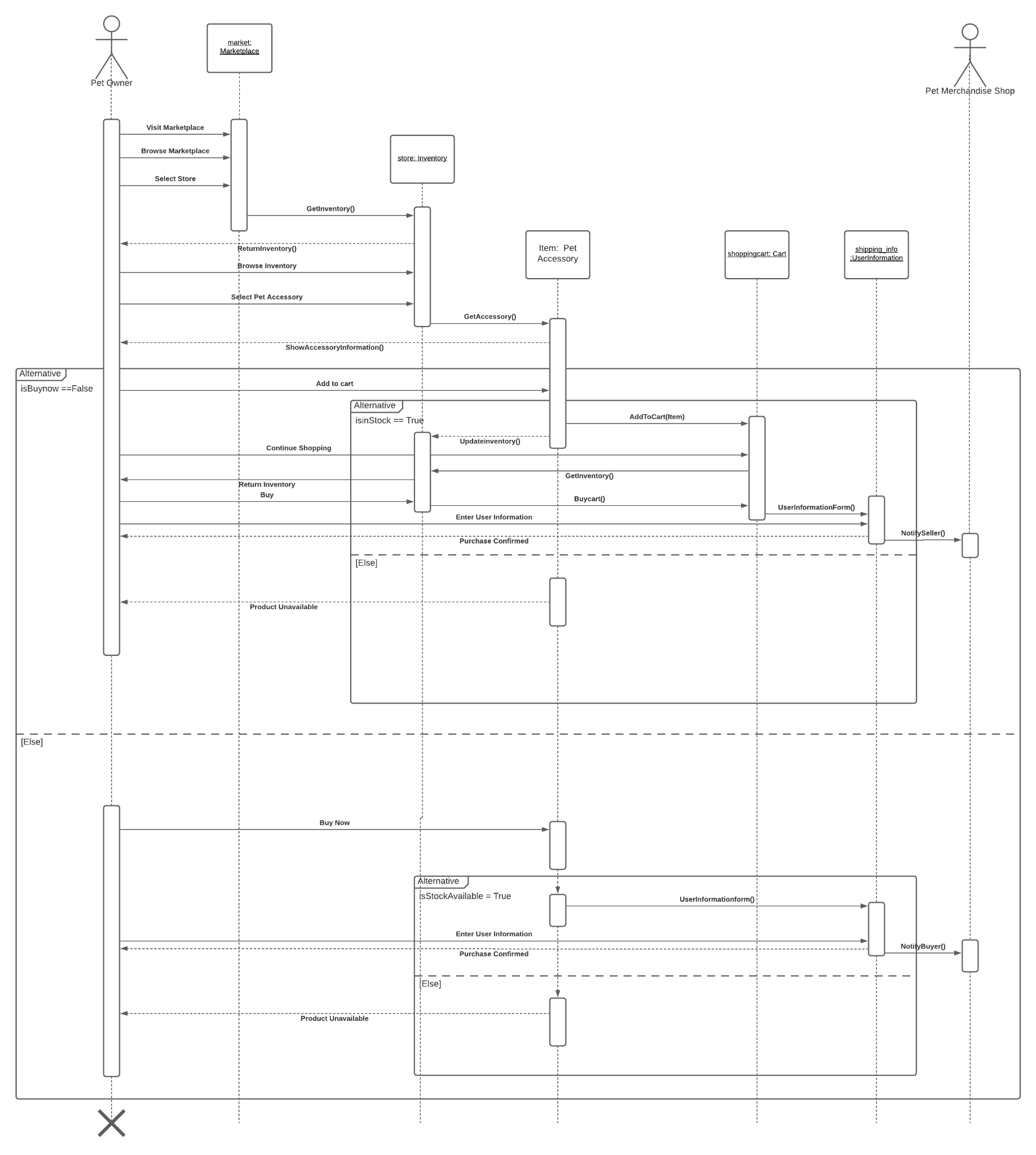
## **5.5** **Update Store Information**



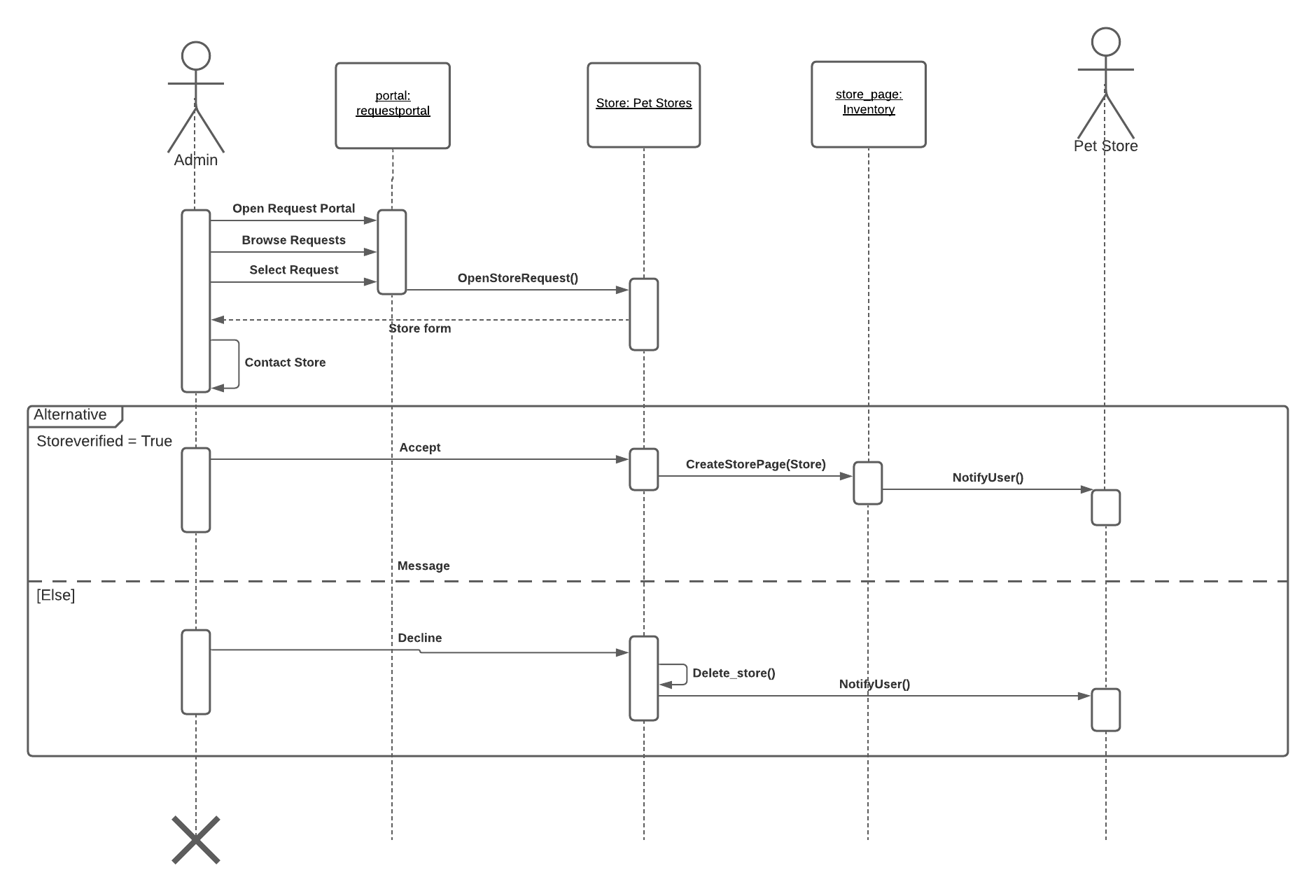
## **5.6** **Receive Rescue Request**



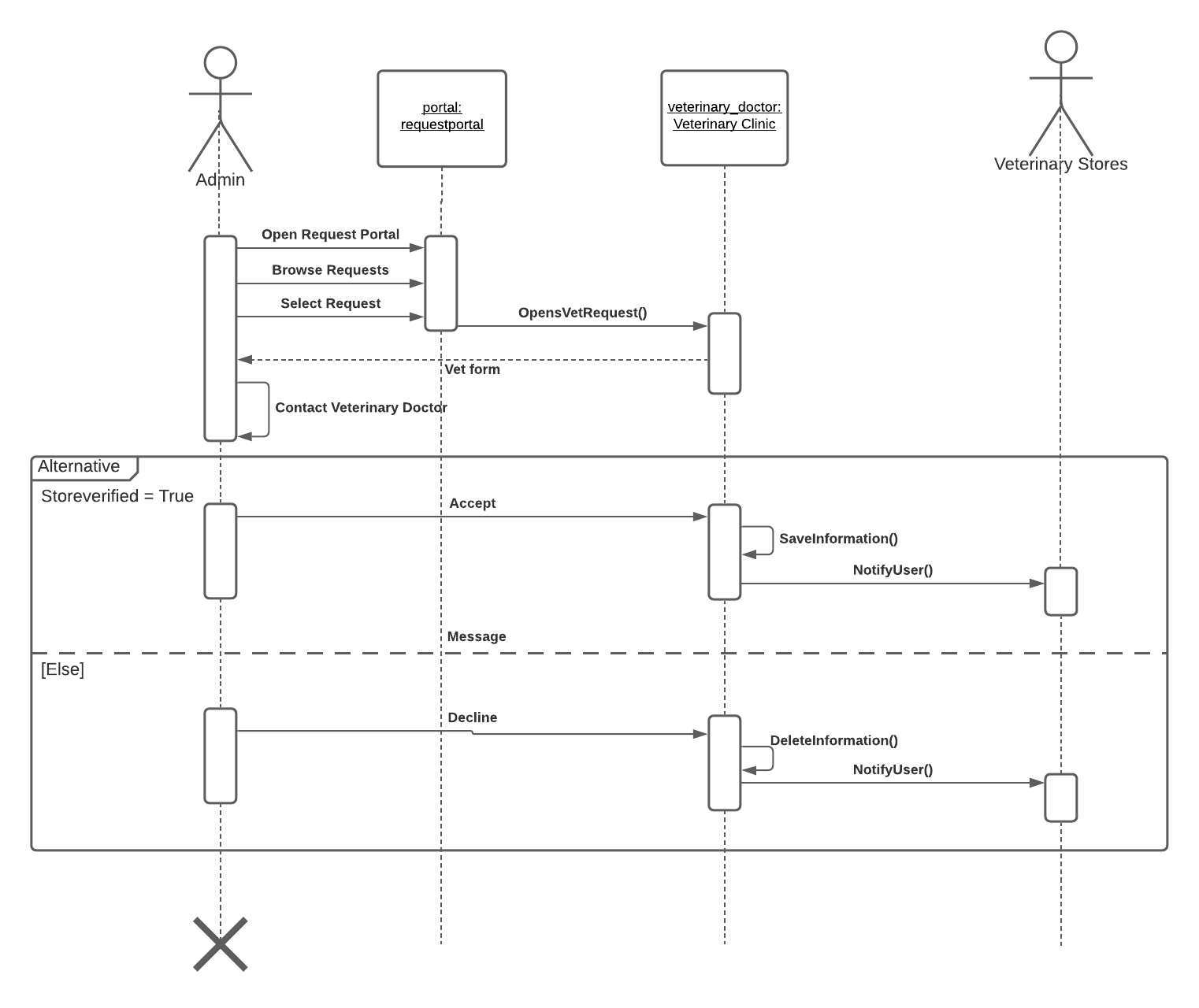
## **5.7** **Buy Pet Accessories**



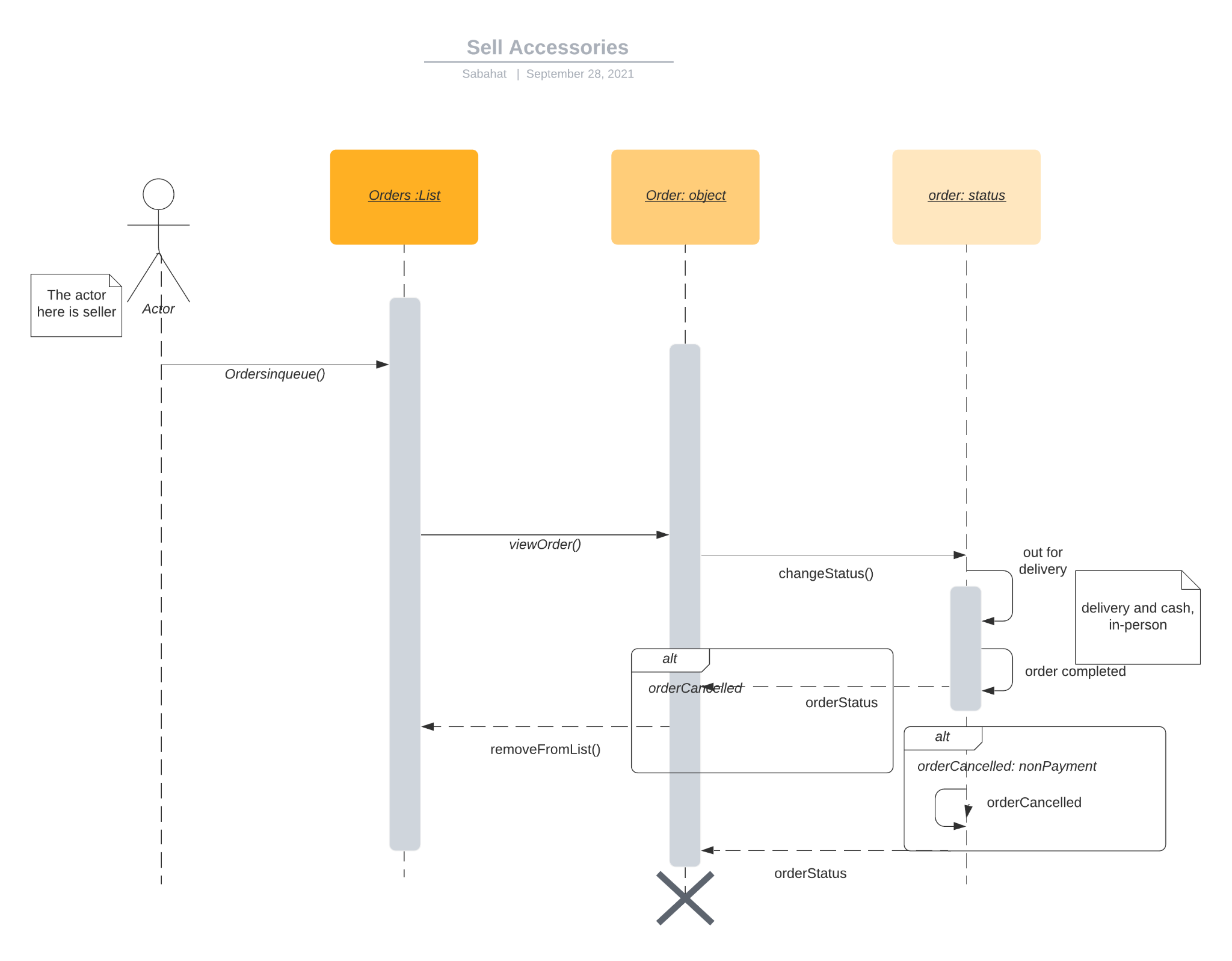
## **5.8** **Add/Remove Pet Stores**



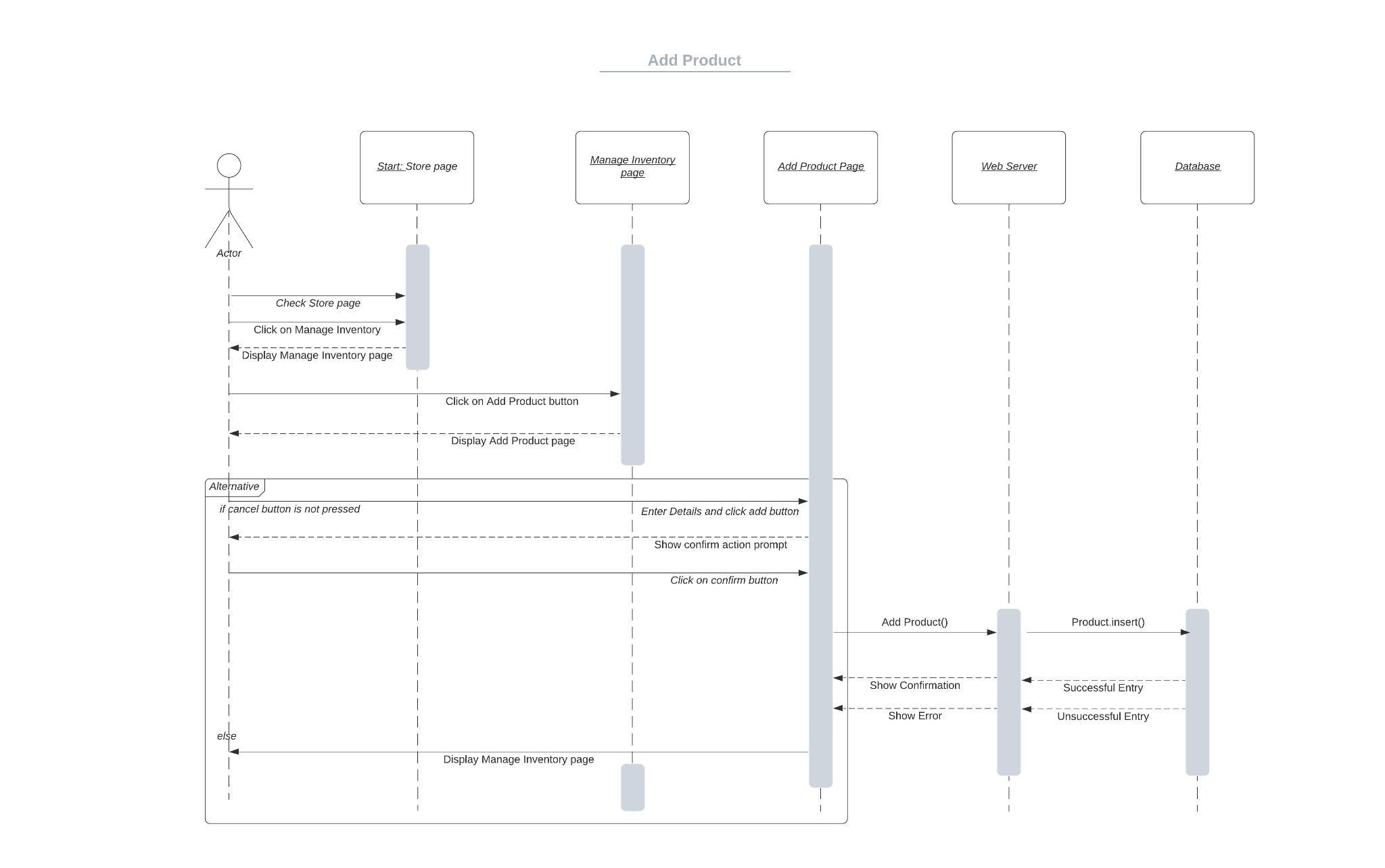
## **5.9** **Add/Remove Veterinary Doctors**



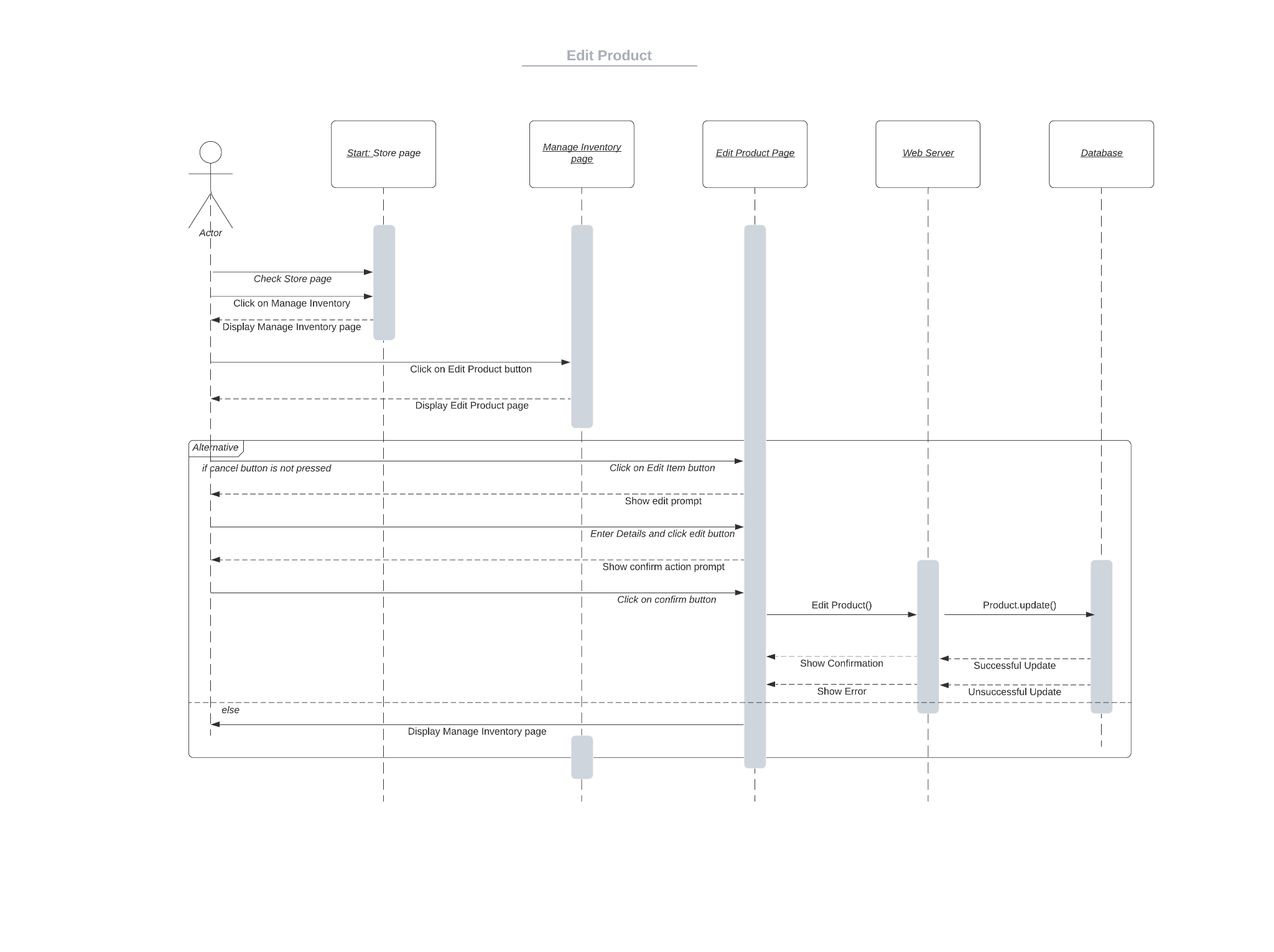
## **5.10** **Sell Accessories**

****

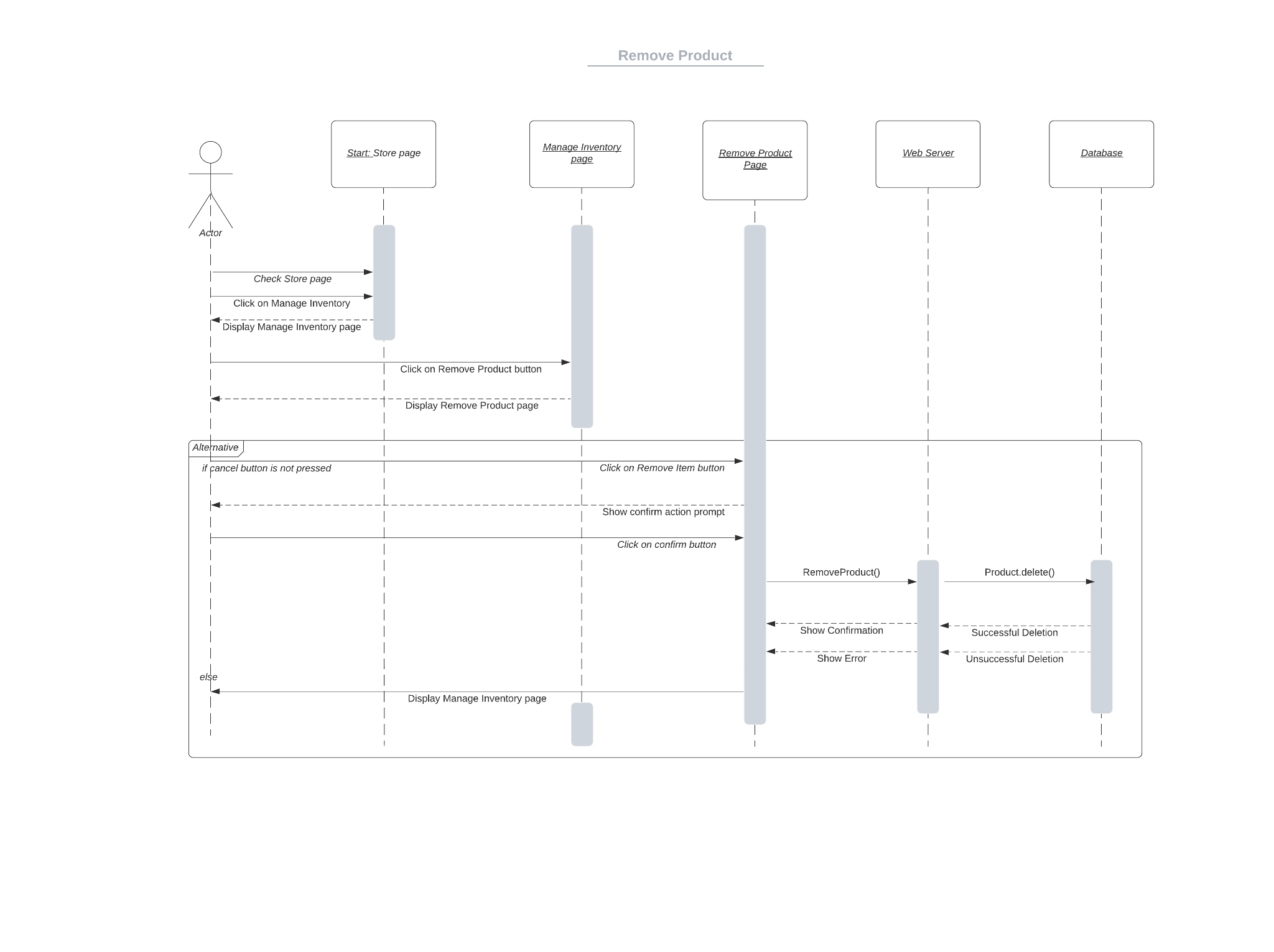
## **5.11** **Add Product**



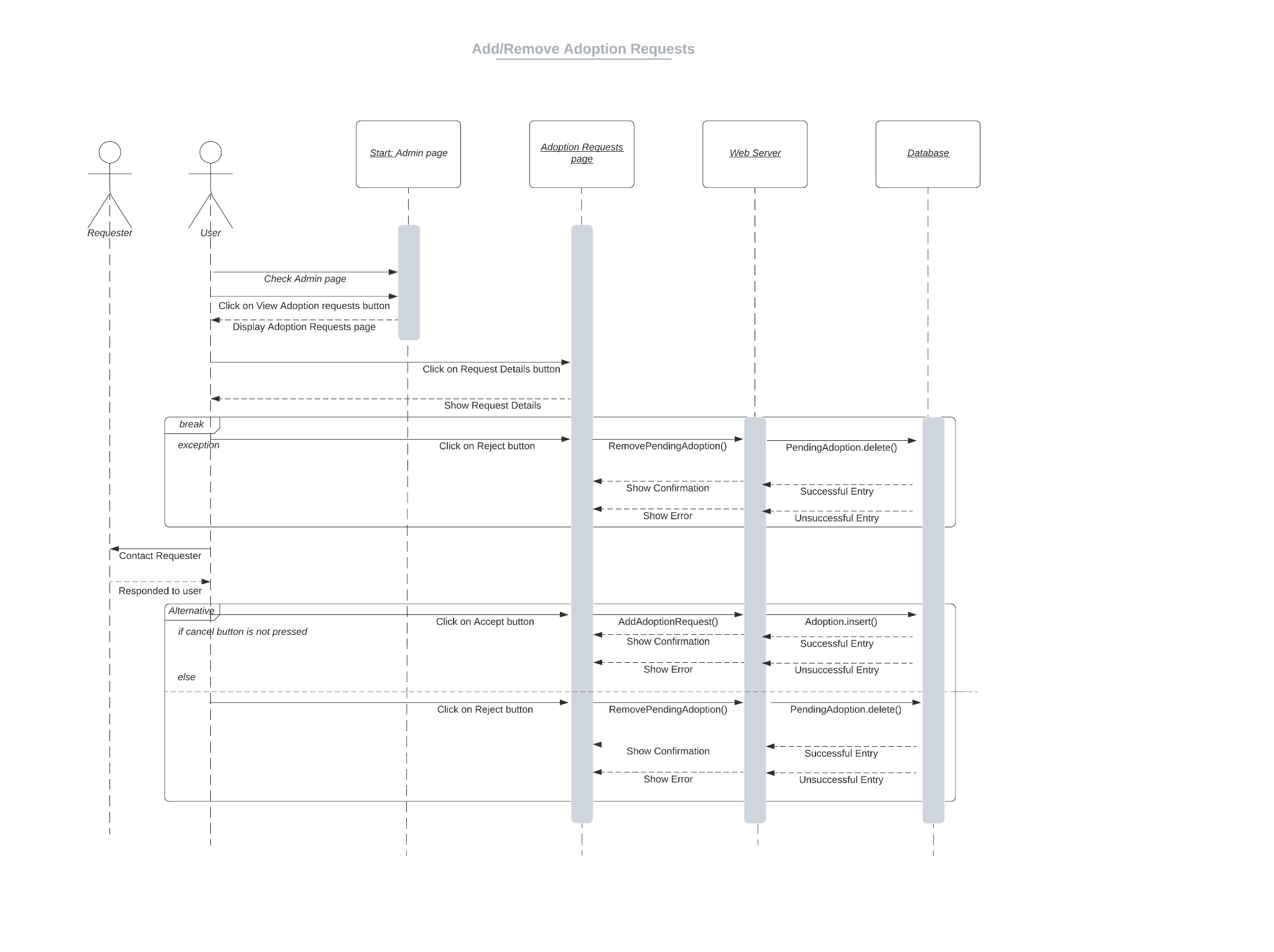
## **5.12** **Edit Product**



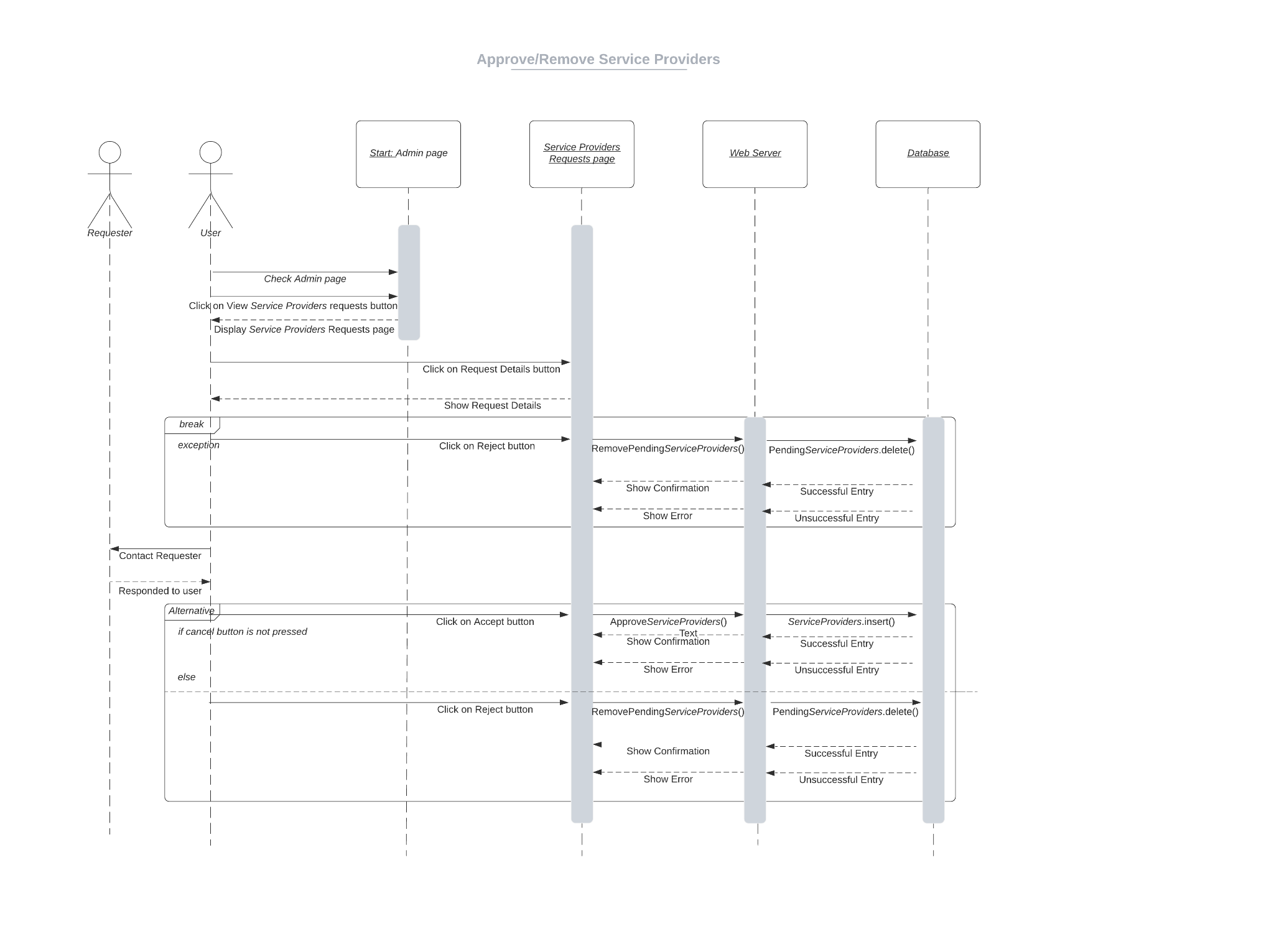
## **5.13** **Remove Product**



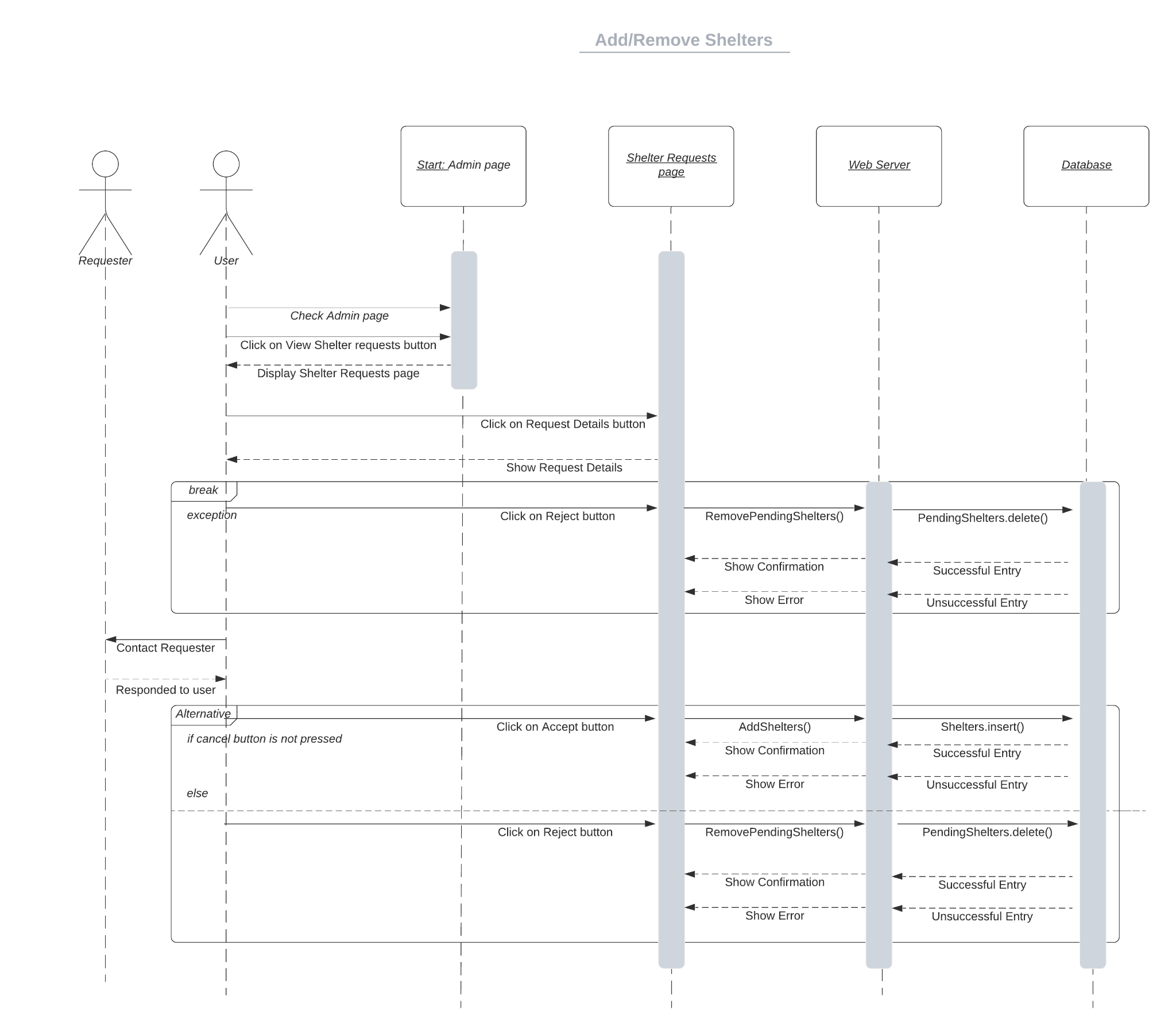
## **5.14** **Add/Remove Adoption Requests**



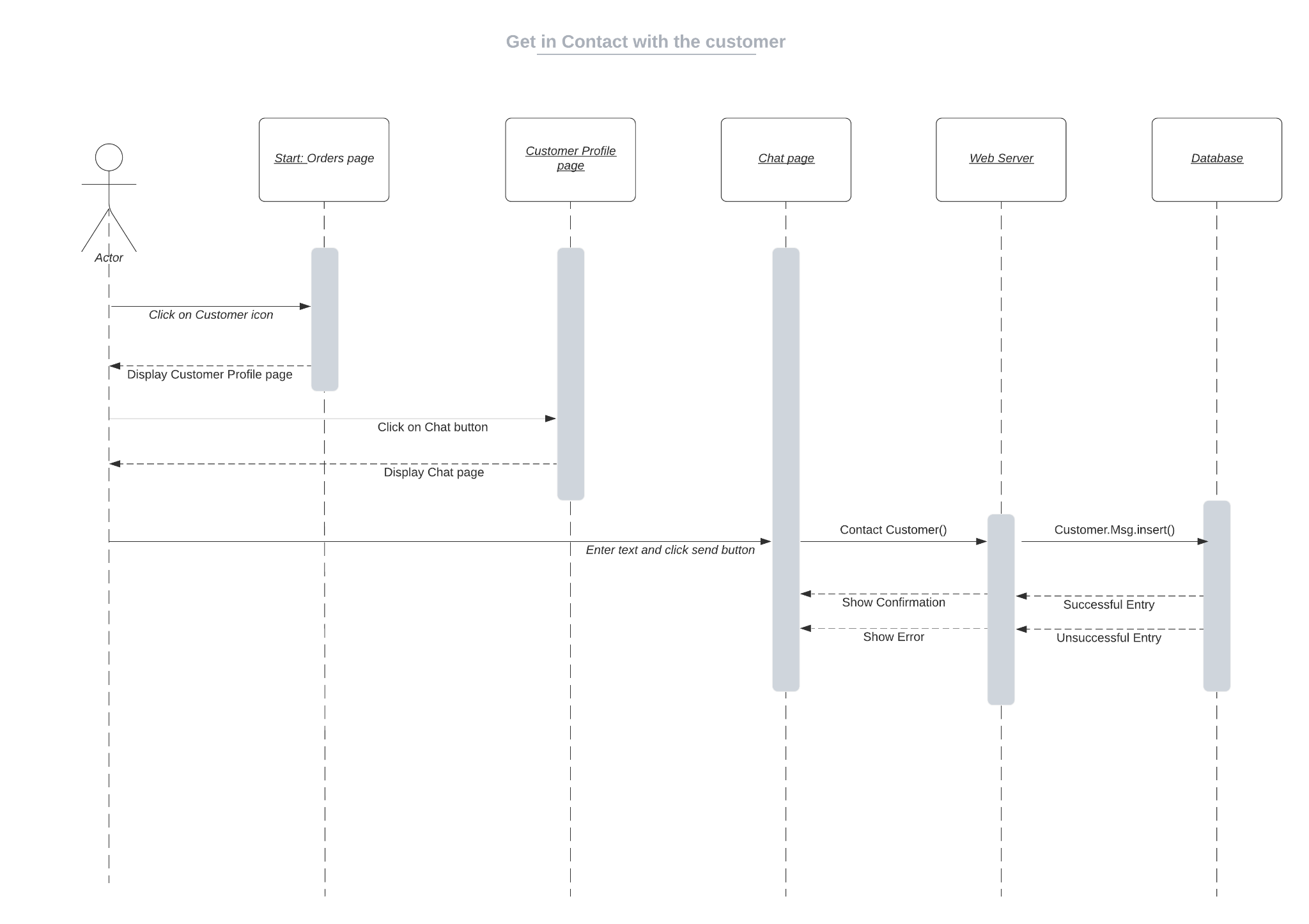
## **5.15** **Approve/Remove Service Providers**



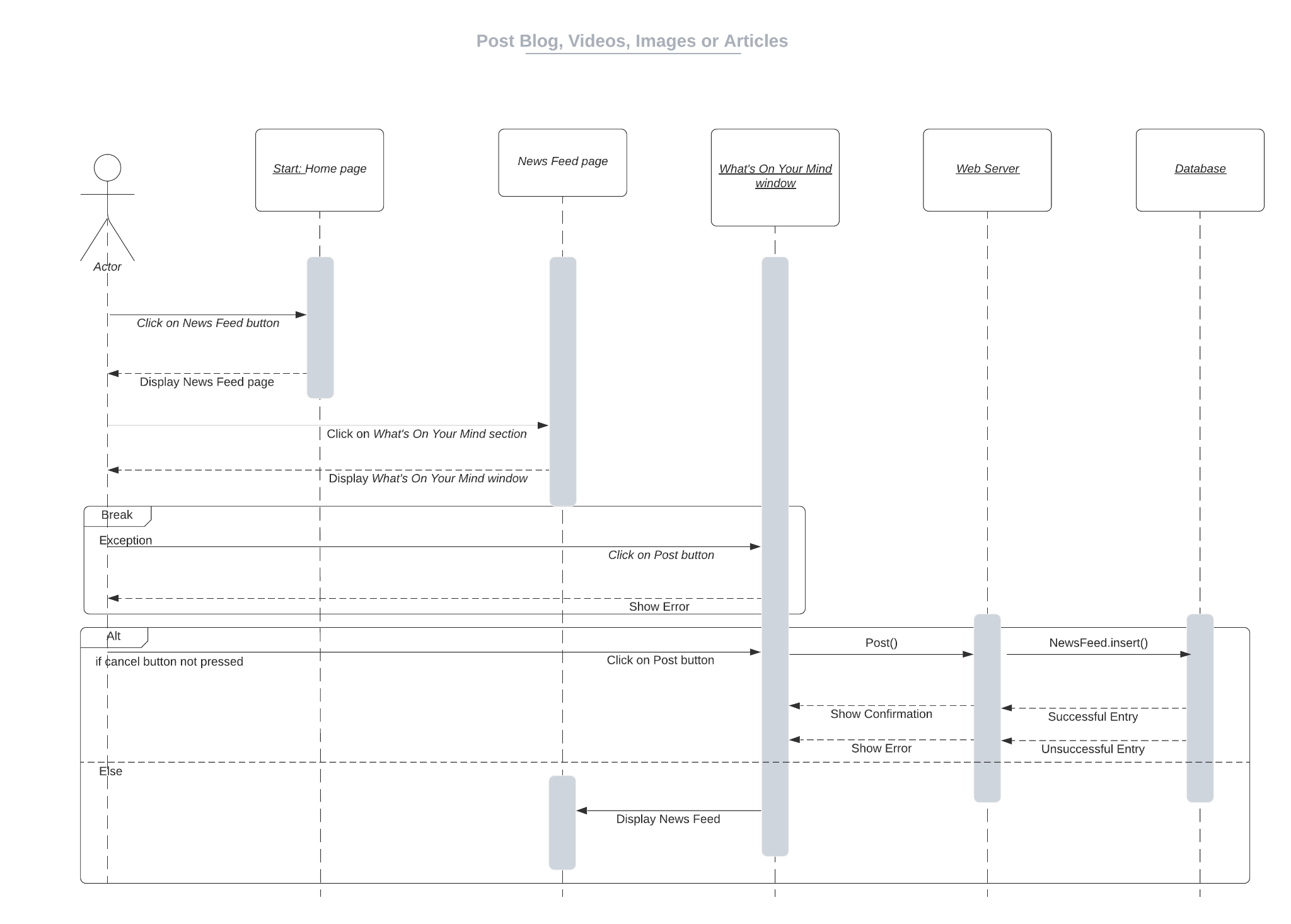
## **5.16 Add/Remove Shelters**



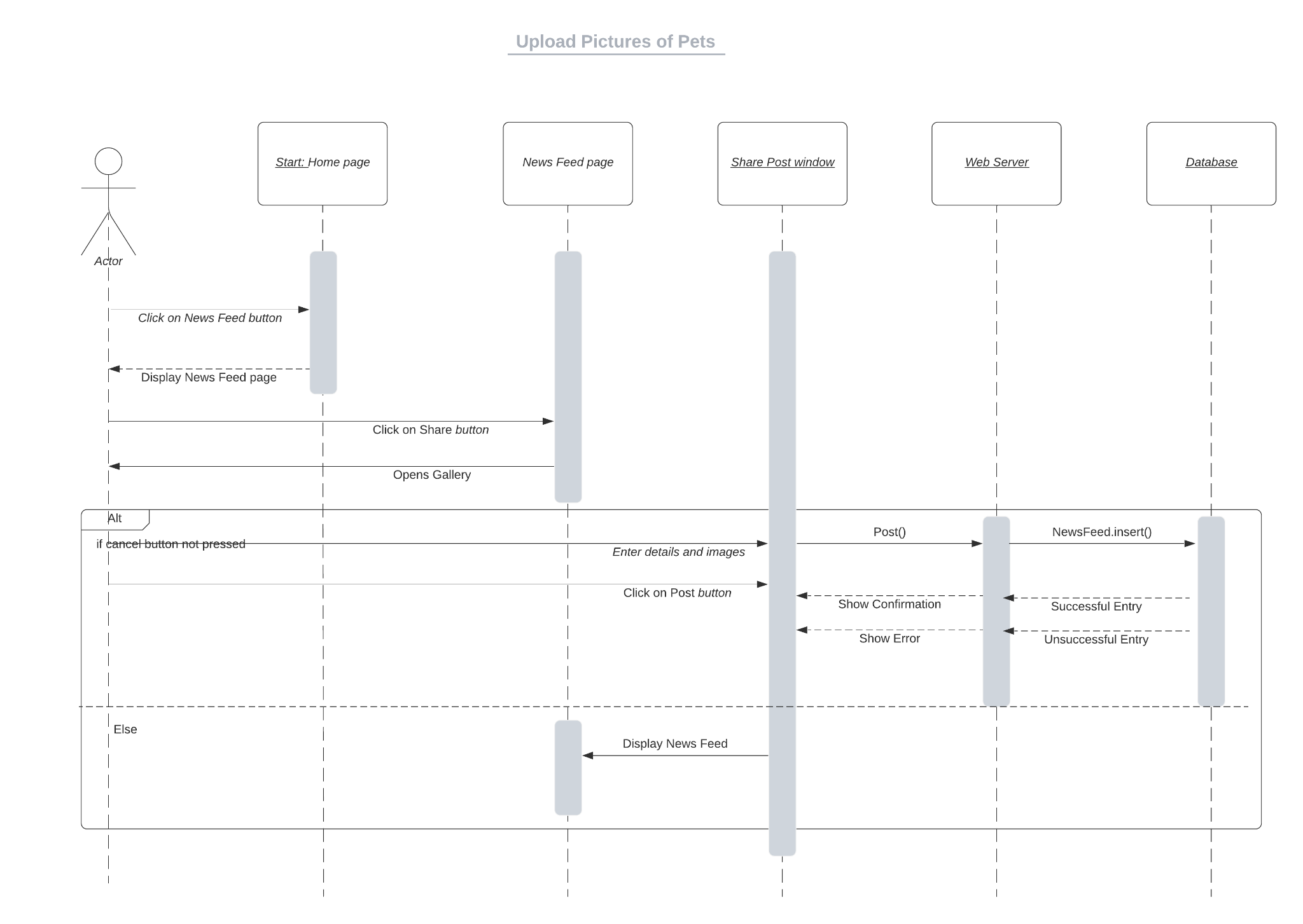
## **5.17 Get in Contact with the Customer**



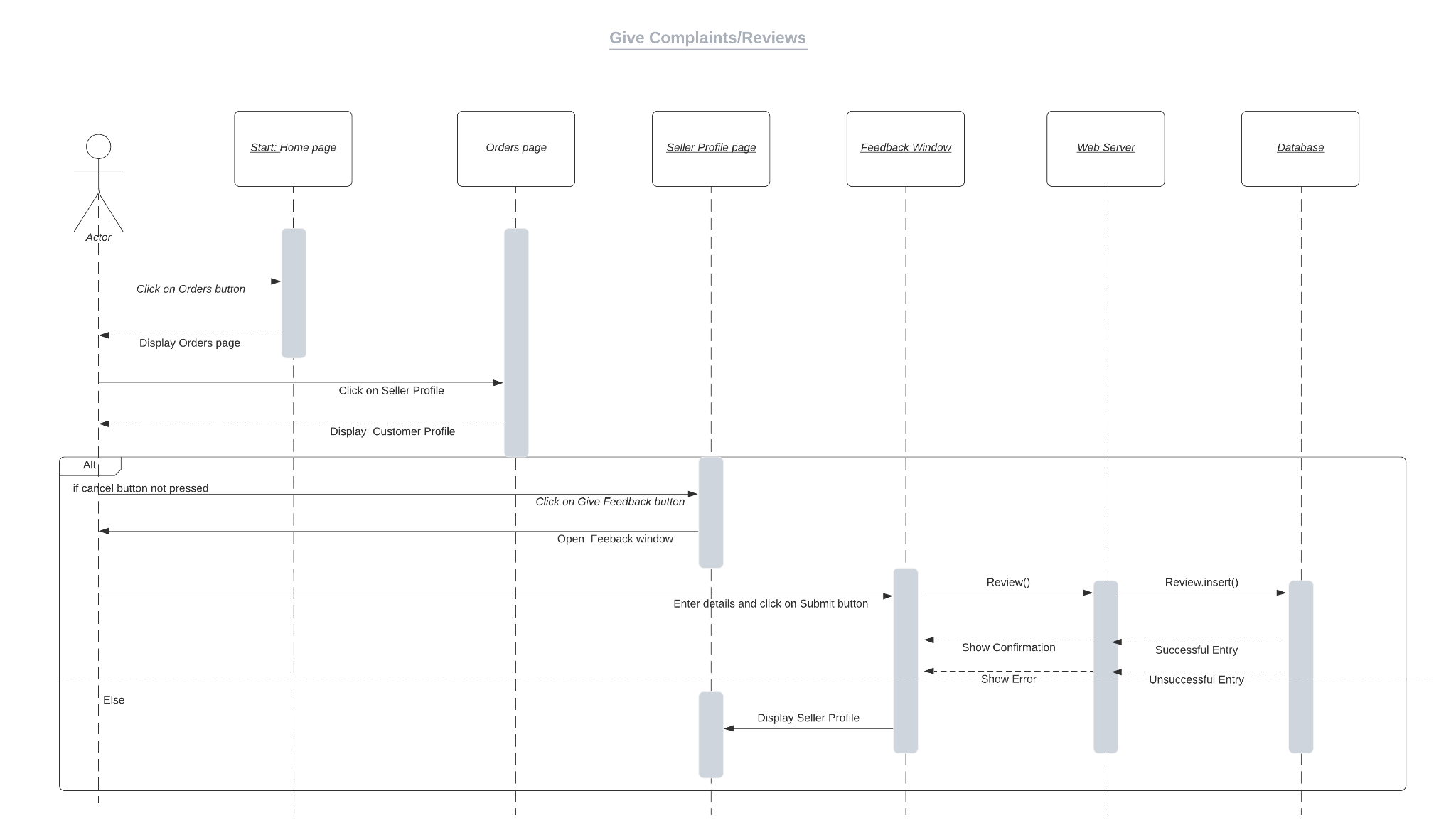
## **5.18 Post Blogs, Videos, Images or Articles**



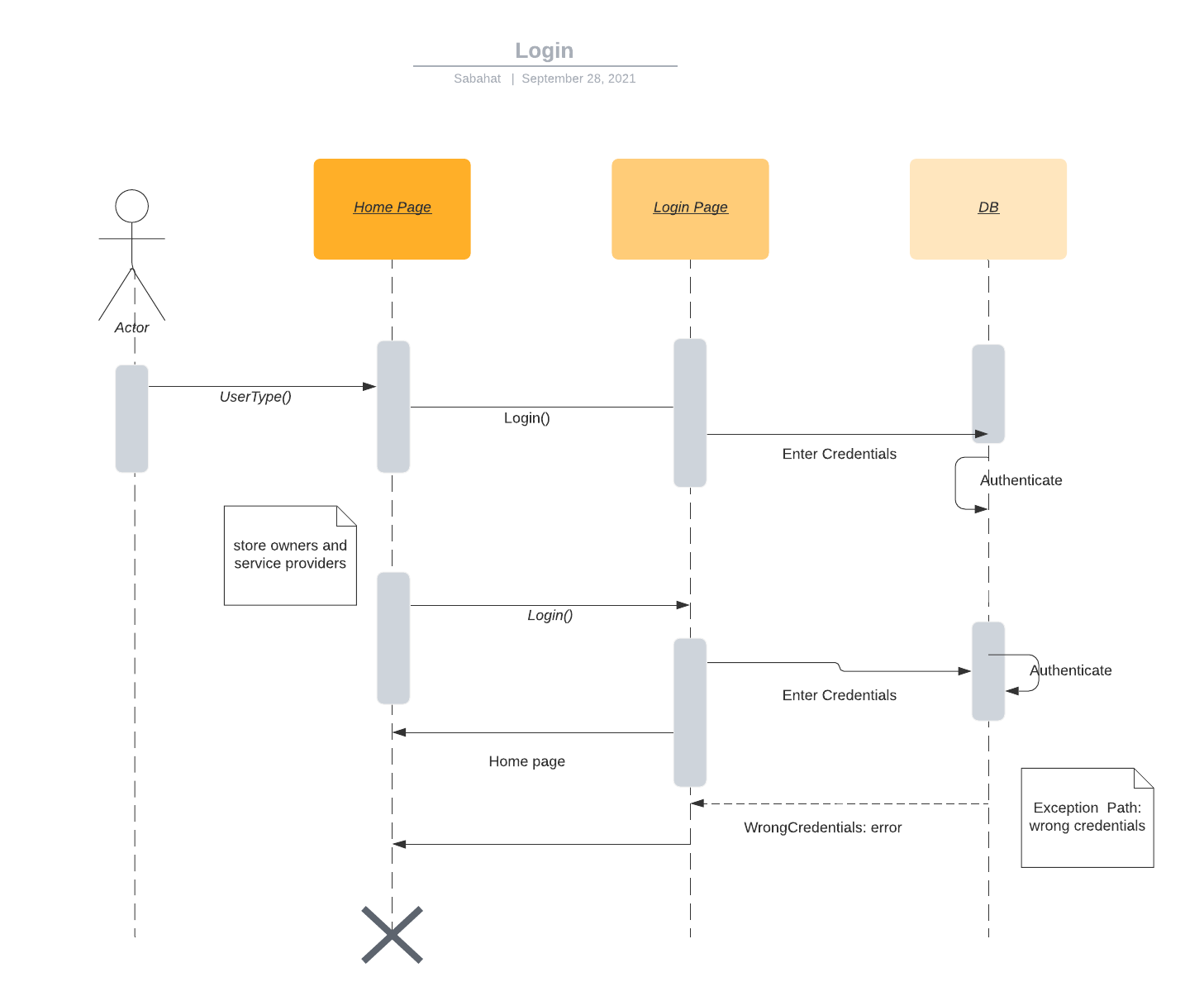
## **5.19 Upload Pictures of Pets**



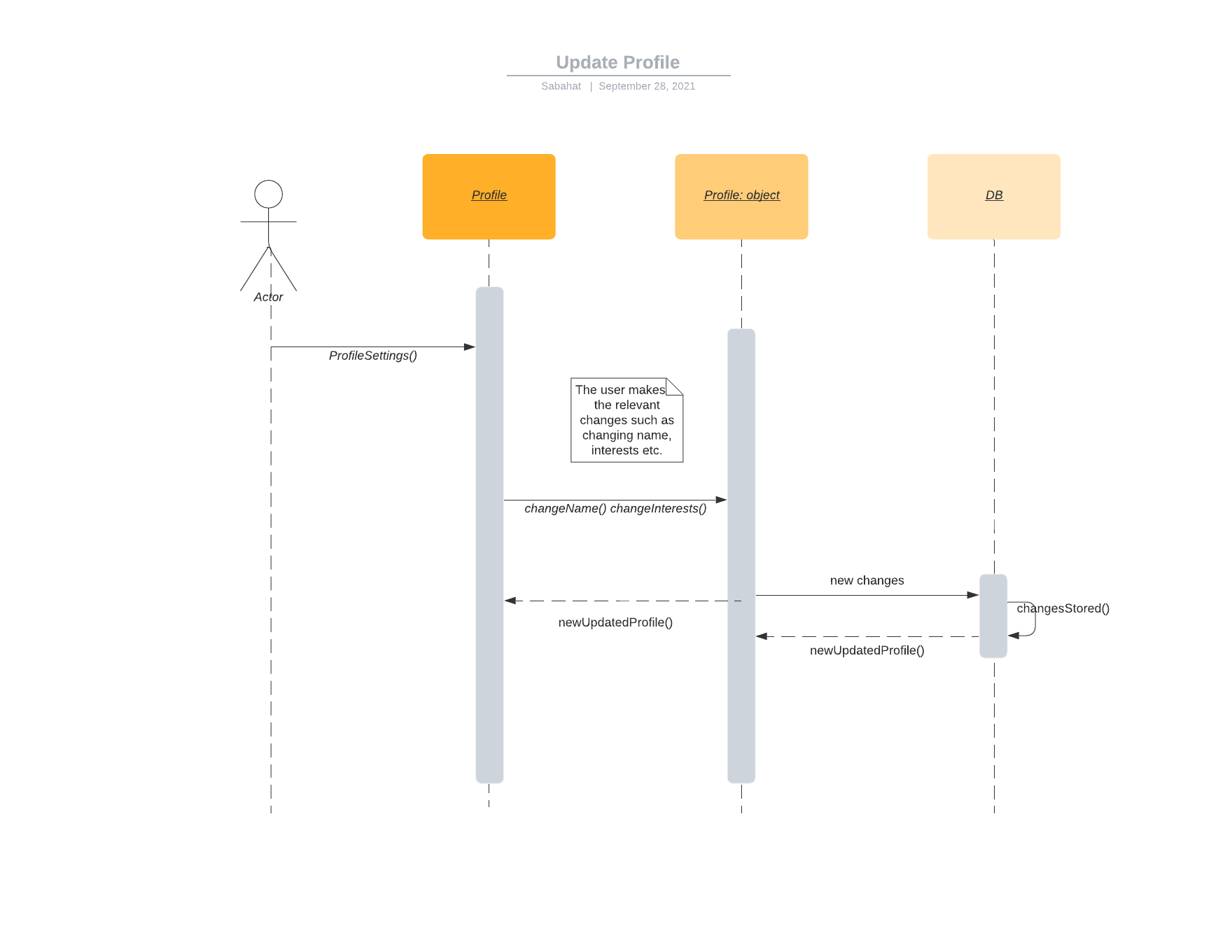
## **5.20 Give Complaints/Reviews**



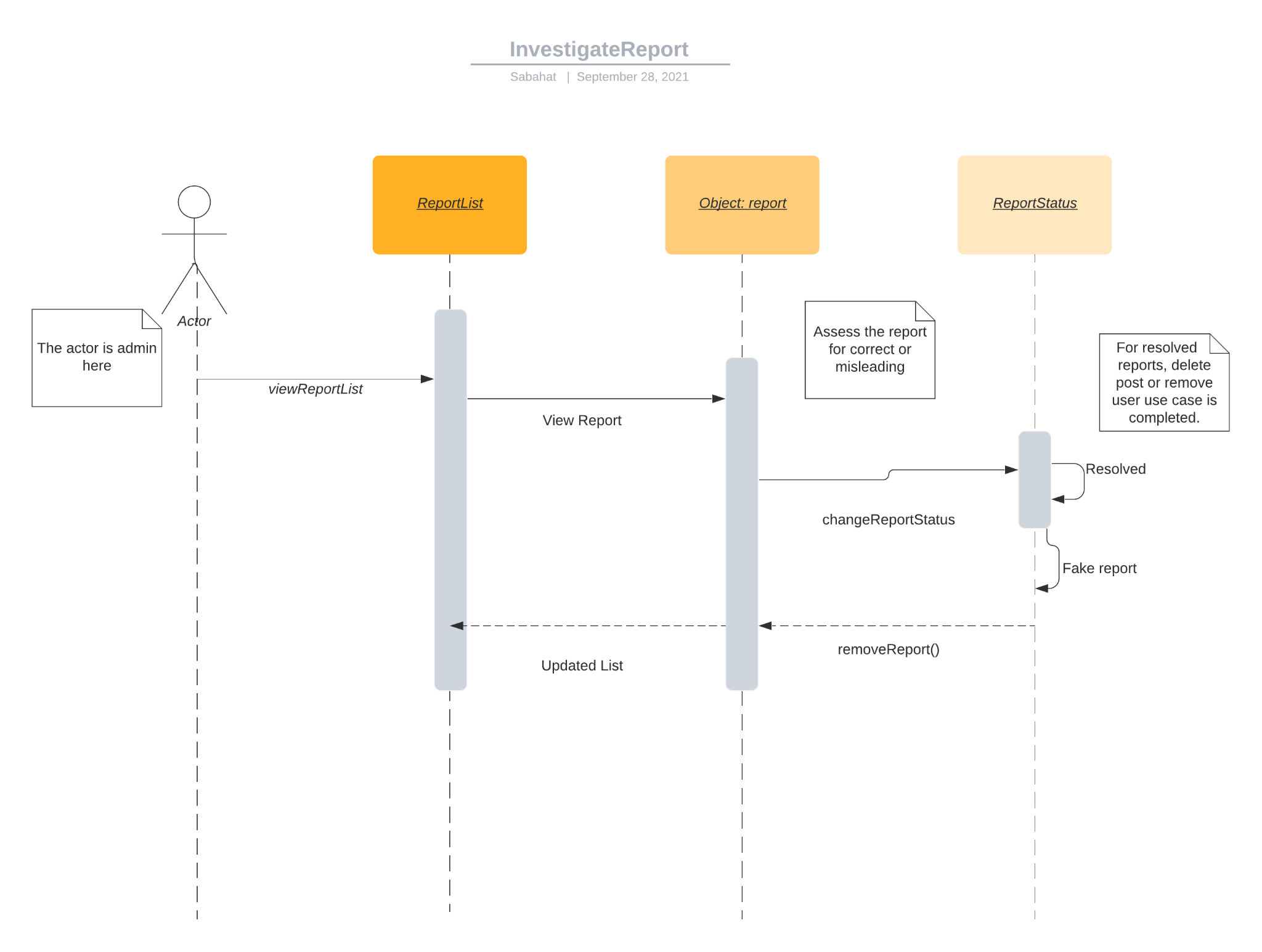
## **5.21 Login**



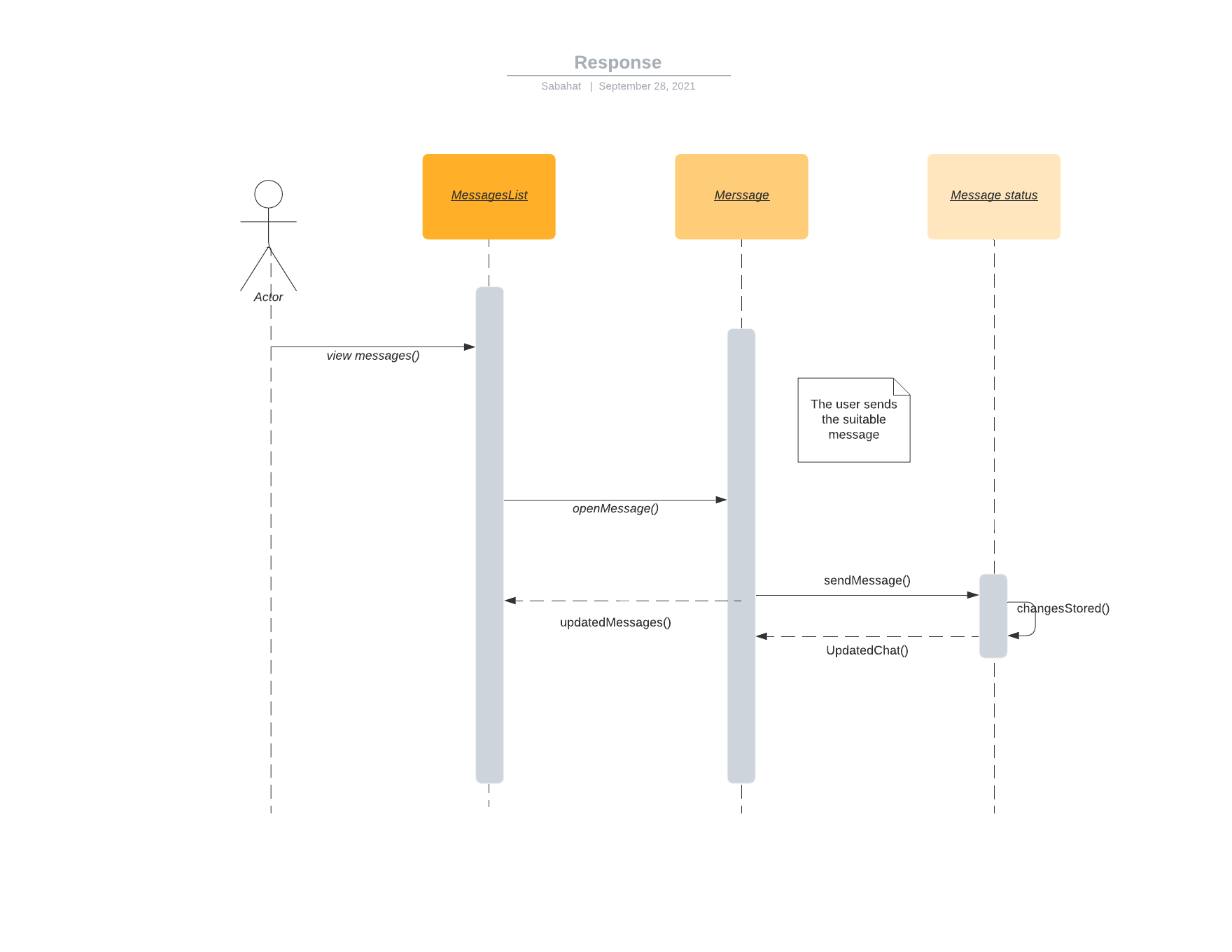
## **5.22 Update Profile**



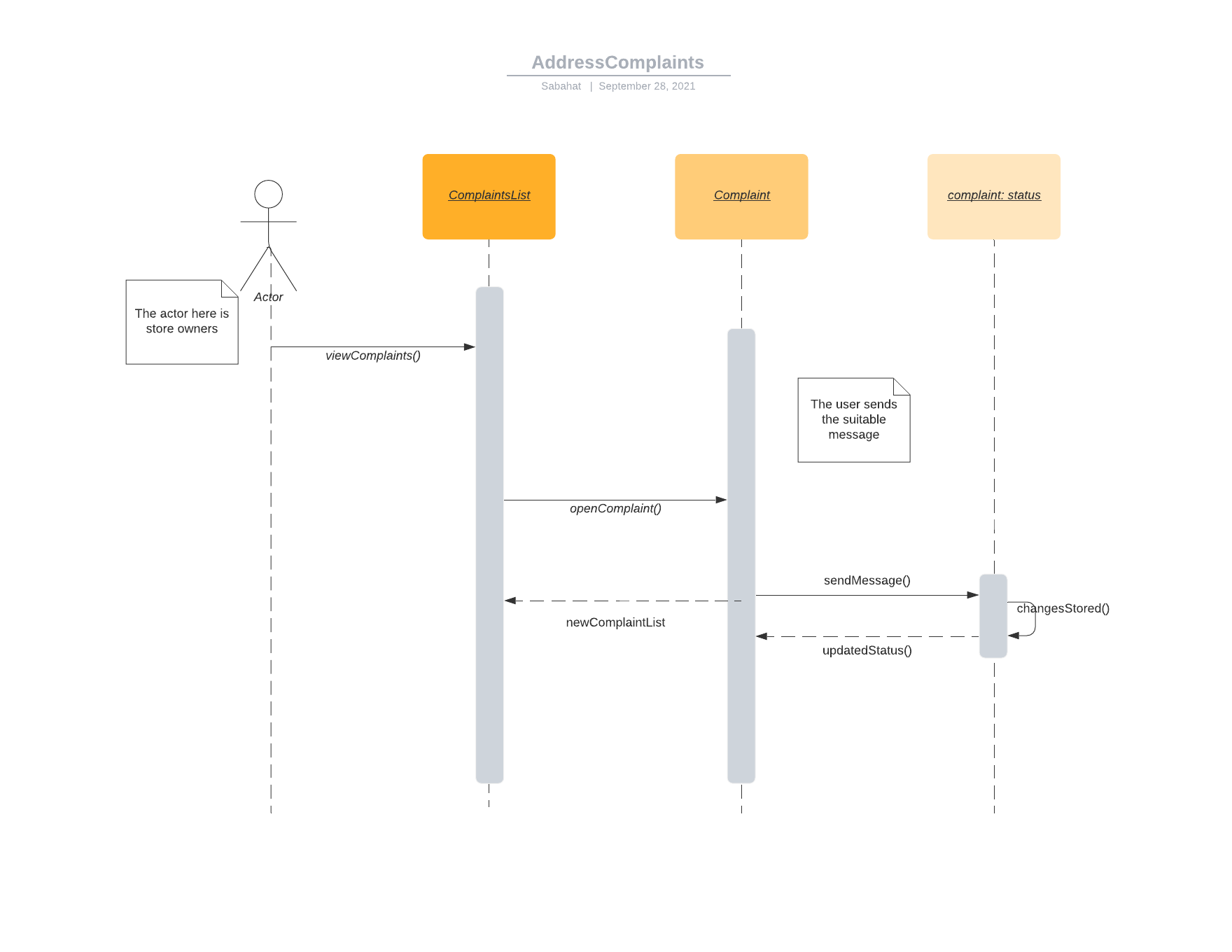
## **5.23 Investigate Report**



## **5.24 Respond via Messages**



## **5.25 Address Complaints**



# **6.** **State Diagrams**

<Repeat the following if you need to draw state diagrams of multiple objects>

## **6.1** **Diagram details**

**User** : It is a user of an application that can be subdivided into common users, Service Providers, Shelters , Veterinary Clinic, Pet Store and admin and it can register.

**Common Users:** Common users are those who are casual app owners and they can be pet owners, they have the functionalities of reporting stray animals, buy pets, finding veterinary options, post and review etc

**Pet Service Providers:** They are providing services related to pets, they can also update their services and address complaints if any.

**Pet Shelters:** Pet Shelters are entertaining rescue requests mainly.

**Pet Merchandise Shops:** Pet merchandise shops are the stores which sell products related to pets.

**Pet Stores:** Pet stores are the places where people can buy pets from. These are physical stores which are set up online on our app as well.

**Veterinary Clinic:** Veterinary clinic is a place where one or more than one veterinary doctors sit. These are also physical clinics with an online clinic set up on our app.

**Admin:** Admin is the owner of the app and they have special rights. They can add, remove people on the app, and modify several other things.

**Comment:** A comment is a message a user leaves on a post, video, blog or any advertisement of a product.

**Sale:** A sale is the exchange of goods (accessories or pets) in return for money. Sales on our app will be made in person for pets and can be cash on delivery for other accessories.

**Inventory:** An inventory is a list of items available at a particular store.

**Purchase:** A purchase is users buying the items in exchange for money. A purchase can be made for pets or accessories.

**Pet:** A pet is an animal kept by people for companionship or entertainment. A pet, on our app, can be in a store for sale, shelter for adoption or might belong to app users.

**Accessory:** An accessory is a product which might be used for decorative or useful purposes. An accessory is usually related to pets in the context of our app.

**Post:** A post is an image, video or text shard by an app user, which is visible to other app users and they can react or comment on it.

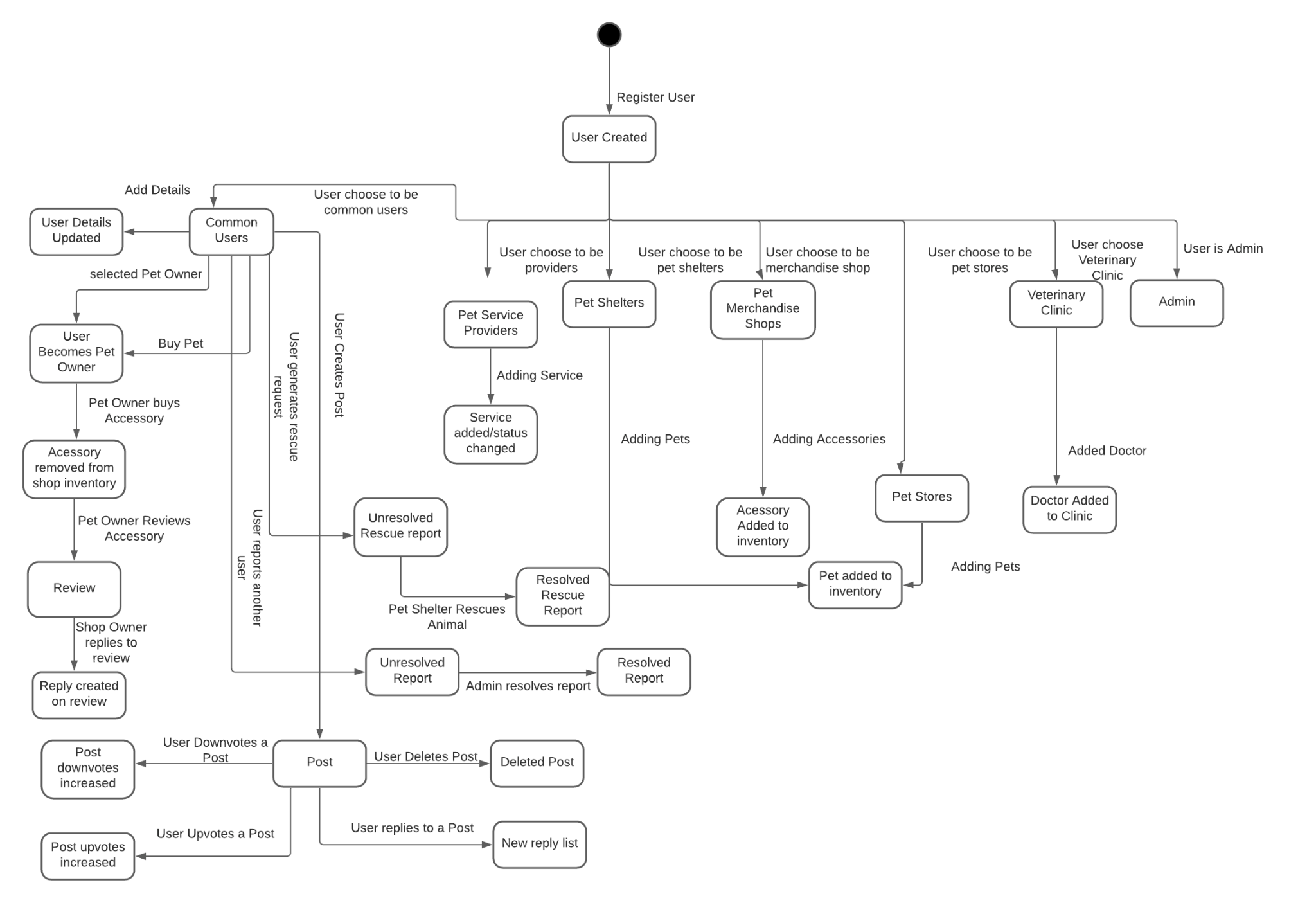
**Report:** A report is made by app users to draw admin’s attention to something which should not be on the app. The reports are reviewed by the admin and the relevant action is taken.

**Reviews / Complaints:** Reviews and complaints are the messages sent to store owners by buyers, to give feedback on their product or to register their complaints.

**Request:** A request is an application submitted by stores, shelters, vets and service providers to get permission to be allowed to sell their services on the app. The admin approves or deletes the request.

## 

## **6.2** **Diagram**



# 

# **7.** **Non-functional Requirements / Quality Attributes**

<Requirements must be testable>

|  |  |
| --- | --- |
| **Sr#** | **Requirements** |
| 1 | The system should not utilize more than 1 GB of memory at any time during its execution. |
| 2 | The system should not fail more than 3 times every 24 hours. In case of a failure, the system should restore to normal operations within 5 minutes of a failure. |
| 3 | The application should take no more than 5 seconds to load. |
| 4 | The system should protect all data inside it against malware attacks or unauthorized access. |
| 5 | User requests (ordering pets/accessories/offering services) should be processed within 5 seconds. |
| 6 | The user interface must be easily understandable by laymen having a basic understanding of English language, with users not having to spend more than 3 seconds to figure out how to proceed with a task. |
| 7 | Considering the application is targeted towards local people, measurements, time zones and currencies will be adjusted accordingly. (Kg, PKR etc.) |
| 8 | The time to notify rescue teams after the initiation of rescue requests should be no more than 20 seconds. |
| 9 | The system should initially be able to cater to a minimum of 1000 users simultaneously. |
| 10 | The system should have a clear distinction between the rights of the admin and the rights of the app users. |
| 11 | Users should be able to navigate their way around the application with ease after continuous use and exploration of it for at least 15 minutes. |
| 12 | The user should be able to make a post within 5 seconds |
| 13 | When the app gets interrupted by a call it should save state for when the call ends. |
| 14 | The locations of the stores or shops should be accurate within a radius of 10 metres. |
| 15 | The user should be able to reorder an item in under a minute. |
| 16 | The system should be compatible with Android OS versions 10 and above. |

# **8.** **Who Did What?**

# 

|  |  |
| --- | --- |
| **Name of the Team Member** | **Tasks done** |
| Adil Aslam Chaudry | State Diagram |
| Mohid Yousaf | Use Case Diagram, Class Diagram, State Diagram |
| Ayan Tabassum Saeed | Use case diagram, contributed to use case descriptions and sequence diagrams and proof read the document |
| Sabahat | contributed to use case descriptions and sequence diagrams |
| Roshan A. Aziz | Worked on use cases, both descriptions and diagrams. Made Sequence Diagrams. |

# **9.** **Review checklist**

Before submission of this deliverable, the team must perform an internal review. Each team member will review one or more sections of the deliverable.

# 

|  |  |
| --- | --- |
| **Section** **Title** | **Reviewer Name(s)** |
| Introduction | Roshan |
| System Actors | Adil |
| Use cases description and diagram | Sabahat, Mohid,Roshan, Ayan |
| Class Diagrams | Adil |
| Sequence diagrams | Sabahat,Mohid, Ayan |
| State Diagrams | Mohid, Adil |
| Non-Functional Requirements | Adil |