

## CHAPTER 0: PREFACE

In a new era of modernization and integration, logistic industry becomes more and more popular in society. Besides that, the overpopulation appears in many metropolitan makes the logistic industry get hard to deliver shipment from start point to end point because of many specific problems, including: new street, new city, government change the street name, traffic jam, ... These things make shipper find difficult when they are trying to deliver something for their customer. This is going to make the shipping process delay and user experience will reduce. In this situation, customer can choose for another company or application that can help them to deliver their items as fast as possible and also with a cheapest price. Because of many problems listed, we decide to build out an application that can help logistic company improve not only a speeding of shipment process and also the user experience.

⇒ Our application is aim to help customer easier to book a shipment order with a friendly graphic user interface. It can also help shipper can find out the shortest way to deliver items for customer. Shipper can also change the pathway if they have any problem, like refuel or mechanical fixing. System will re-calculate a pathway in these situations. Moreover, system will re-calculate if shipper stuck in a traffic jam or if shipper notify that that street is not good enough to put in a pathway: street under water, in construction. Finally, administrator can monitor shipper and can have a suitable to deal with shipper if they violate with company policies. Administrator can also modify the price of shipment based on a real condition of society and company's policies, for example: upgrade fee while price of petrol increase or reduce shipping fee to help customer can access an application, a company's services.

## CHAPTER 1: INTRODUCTION

### 1.1 Purpose

This document outlines all information of the implementation and usage of FAST DELIVERY APP.

### 1.2 Document Conventions

This Document was created based on the IEEE template for System Requirement Specification Documents.

### 1.3 Scope

This app is designed with three user objects (administrator, shipper and customer). So, this app is mainly for people who are having free time, intending to become a shipper, the delivery man in order to earn extra income. It can also be used for those wishing to transport goods in busy conditions that cannot be delivered on their own for objective reasons. In addition, the app can also be used for the administrator to manage the shippers and customers, operations, activities,

...

⇒ The reason why our group choose FAST DELIVERY APP is to simplify deliveries in the era there is a great need to transfer goods - technology 4.0, at the same time solving unemployment, increasing income for workers.

### 1.4 Overview

The first goal of our design was to avoid a dull structured user interface. As a result, we attempted to create a modern graphic user interface scheme. This program allows clients, shippers, and administrators to create new accounts in order to perform their respective functions. The user must enter a password and a username in order to use the features of our application. The app would ask the customer for some information at first, such as where they want their food shipped, the name of the shipping device, delivery forms, and so on. The app will show the results on the map interface based on the inputs. Further, in generally, the app also allows shippers to track their orders and receive directions. Customers may also track their orders, display their history, call for delivery, and make payments. You can handle the shipper and customer as an administrator, and you have the ability to block users, develop rules, and create events and promotions.

Because of its low cost and user-friendliness, our model framework can be applied to today's general business or strategic goals of the company commissioning the program. The app can be conveniently run on a variety of platforms (iOS, Android, ...), attracting a wide range of organizations.

## **1.5 Need for the System**

With the advancement in technology and technology, as well as the growth of consumer goods, the demand for transportation is growing day by day in the 4.0 period. For those who need to deliver something to friends or relatives, ... but must maintain the parcel's safety and integrity while meeting the shortest, optimized delivery deadline. In recent years, productivity and effectiveness have been in high demand. An application for quick delivery has been developed. Furthermore, the software can be used with the shipper and administrator interfaces, resulting in the development of work for the unemployed or those with spare time who want to earn more money and solve unemployment. It will ensure efficient service, build the best link between the shipper and the consumer, and deliver customer reward activities, as well as shipper event rewards, which will help the app attract users from both sides. With all of the software's features, from registration and login to order delivery, payment, promotion events, shipping management, and history management, we're dedicated to providing users with the most impressive app experience possible, while maintaining security, professionalism, and ease of use and management. The fast delivery app can quickly gain a large number of users and become a leader in the modern shipping industry.

## **CHAPTER 2: GLOSSARY**

## CHAPTER 3: USER REQUIREMENT DEFINITION

Num	Requirement	Explain in details
1	Login	User including: Customer, Shipper and Administration can log in their account into the system based on database by inputing their username/email and password in login form
2	Sign Up	Customer can create a new account into the system by inputing some basic information of them in sign up form
3	Making order to delivery	User can input the information of good that they want to deliver, information of start point (Including: customer name, address, phone number) and information of the end point (Information: customer name, address, phone number). After all information is filled, they can click to button to process and start searching for finding out the nearest shipper to help them deliver their items in 30 seconds, this is also a period that customer can cancel their order.
4	Various choice for customer	For each shipping transaction, customer can choose the different way to ship their good, for example: aircraft, ship, truck, motorcycle. With each transportation type, it will have a different time to finish and also different shipping cost.

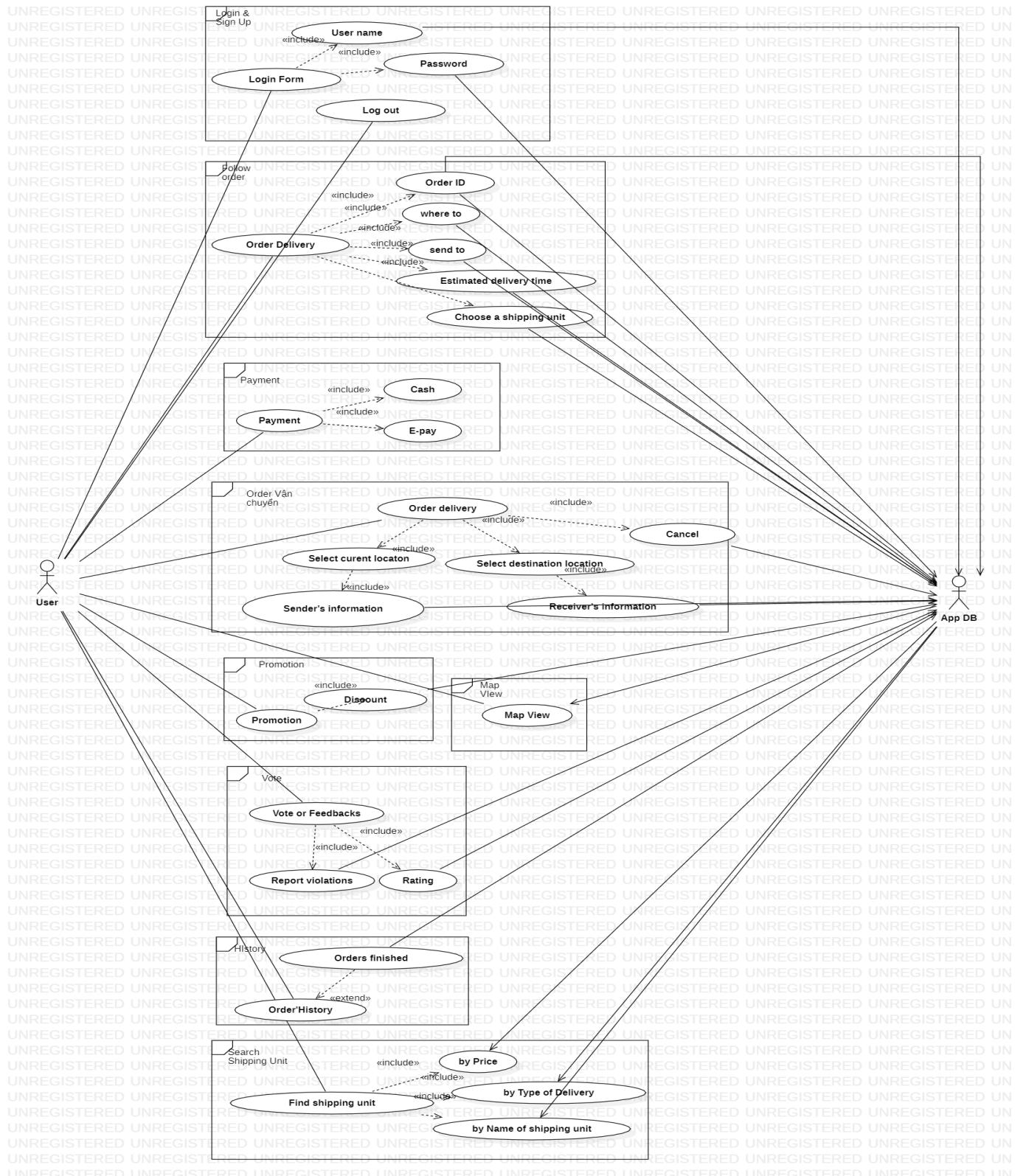
5	Cancel order at customer side	System provides customer at most 30 seconds to cancel their order if they change their idea or have any problem. After 30 seconds, system will start to find out the nearest shipper to delivery
6	Customer can access to the shipment history	Customer can access to the shipment history. Here they can find out all of their order in the past and can view all of them in details.
7	Vote or Feedback	After accessed in history of shipment and go to view the detail of order in the past, customer can choose to vote for shipper if they satisfied with shipper and system. Besides that, they can feedback if they are not satisfied.
8	Apply the promotion code	With this, customer can apply the promotion code which is sent or organized by administration.
8	Calculate the shortest way for shipper to deliver as fast as possible and calculate for estimate time	System will calculate the shortest and the fastest way to help shipper deliver items as fast as possible and base on that system can calculate the estimate time for shipment.
9	Deliver many items from many customers as the same time	Shipper can choose to deliver many items from many customers if they or on the same ways but at most 5 location in each shipping
10	Re-calculate the shipping way to help shipper exit traffic jam	If customer is on the way to deliver items but system find out that shipper is stuck at that street in 5-10 minutes so system need to recalculuate the delivery way again and help customer to get out that traffic jam.

11	Allow shipper add more location in journey	Shipper can add more location in their delivery way if it is necessary for them or if they think that it will help them to be faster. For example: shipper need to refuel their vehicles while they are delivering items to customer so that they need to reach the nearest gas station, at that moment, system need to calculate
		again for new way and new estimated time.
12	Cancel order at shipper side	Shipper can cancel order if they can not contact to customer after they reached the sender address or waiting time is more than 5 minutes or customer asked for cancellation.
13	Change shipper	After get items from customer, if shipper can not continue to delivery or they have any problem, violate with time policy of shipment, shipper can request system to find out other nearest shipper can help them to continue their job.
14	Manage shipper information	Admin can get all of information of shipper in the database to manage them. Show out good shipper and bad shipper to have suitable solution to enhance the shipping policy. This also help admin can manage the personal information of shipper and it will necessary in some cases. For example: Contact with family when shipper get problems send information to police if shipper steal goods from customer and hide away

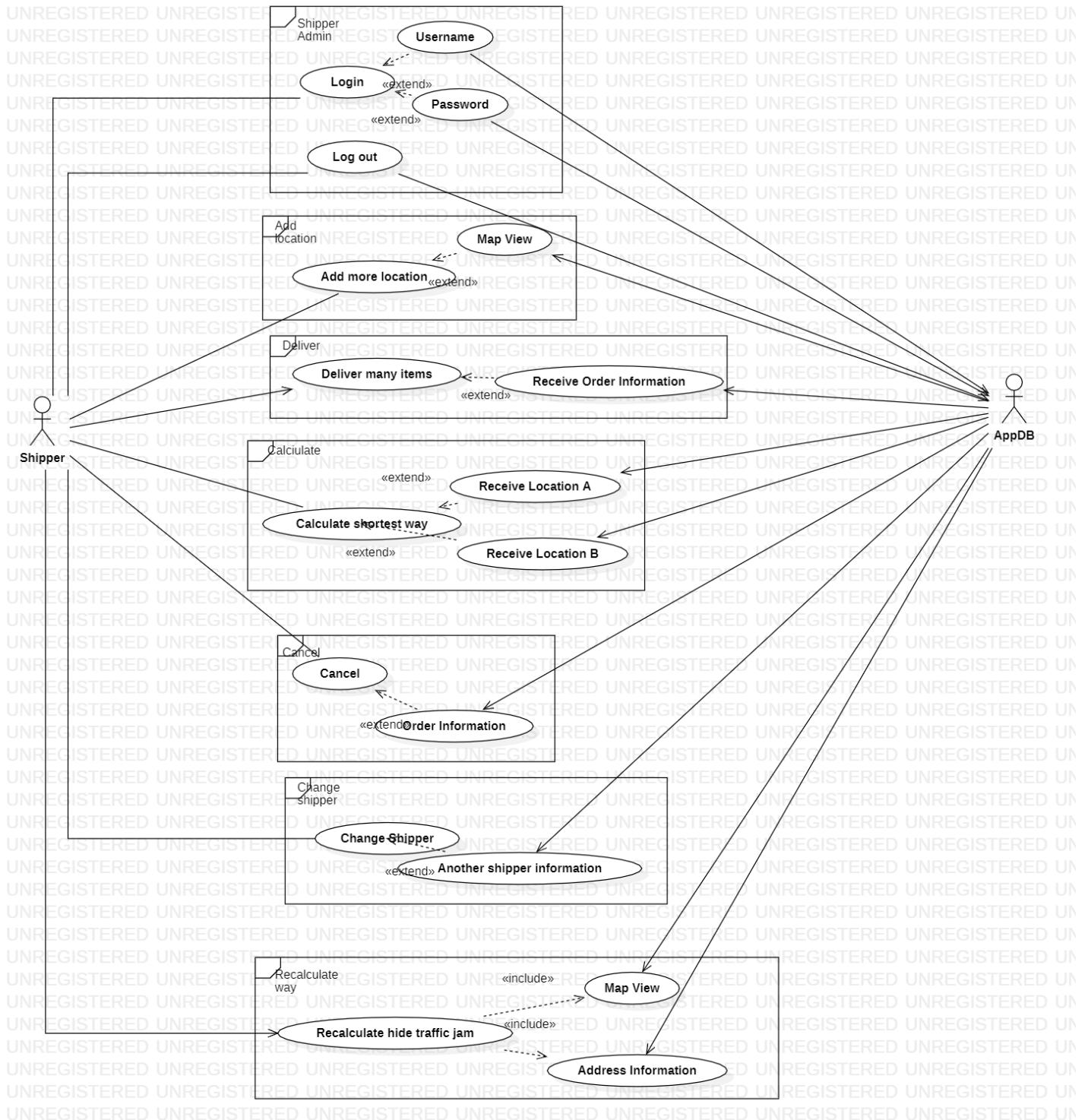
15	Manage customer information	Admin can get all of information of customer including: Name, address, phone number and number of using system from that customer. This is not only help admin can manage customer but also help them to find out the potential customer.
16	Shipping price can modify	The administrator can change the cost of shipping based on the policy of shipping and price of fuel. They can change the price per km or price per hour of shipping or the transport type.
17	Add new shipper and admin into the system	Sign up function of the system can not apply for admin and shipper because of the security problem so that it need to be added by administration.
18	Send invoice via customer email	Admin will send the invoice to customer when the order was finished
19	Block shipper if they are infringe	If shipper infringed the policy of shipment, admin can block their account.
20	Block customer if they are infringe	If customer infringed the policy of shipment, admin can block their account.
21	Send promotion to customer	Admin send promotions code to customer

# CHAPTER 4: SYSTEM ARCHITECTURE

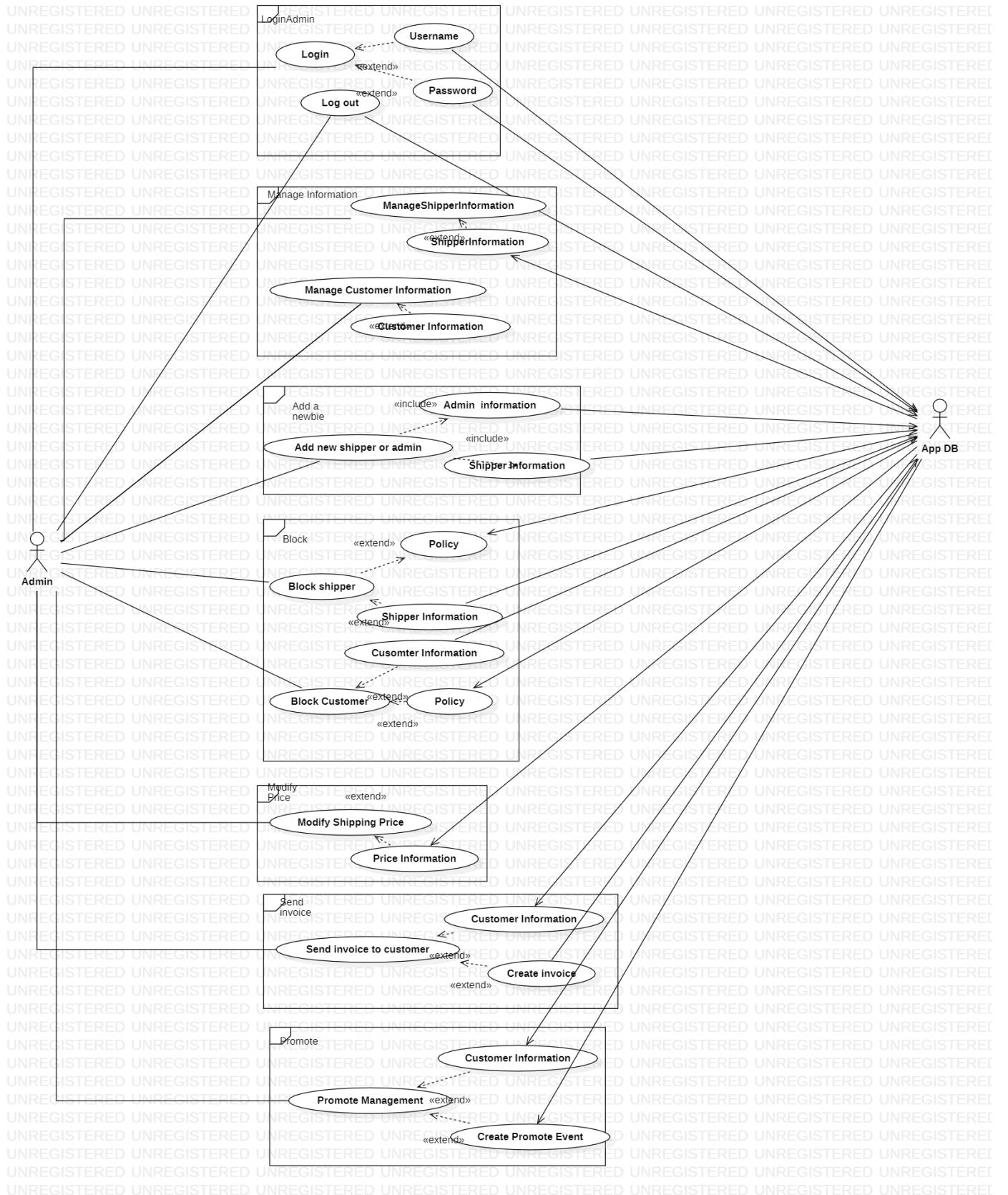
## Use Case Diagram for User



## Use Case Diagram for Shipper



## Use Case Diagram for Admin



## **CHAPTER 5: SYSTEM REQUIREMENT SPECIFICATION**

### **\* Functional requirements:**

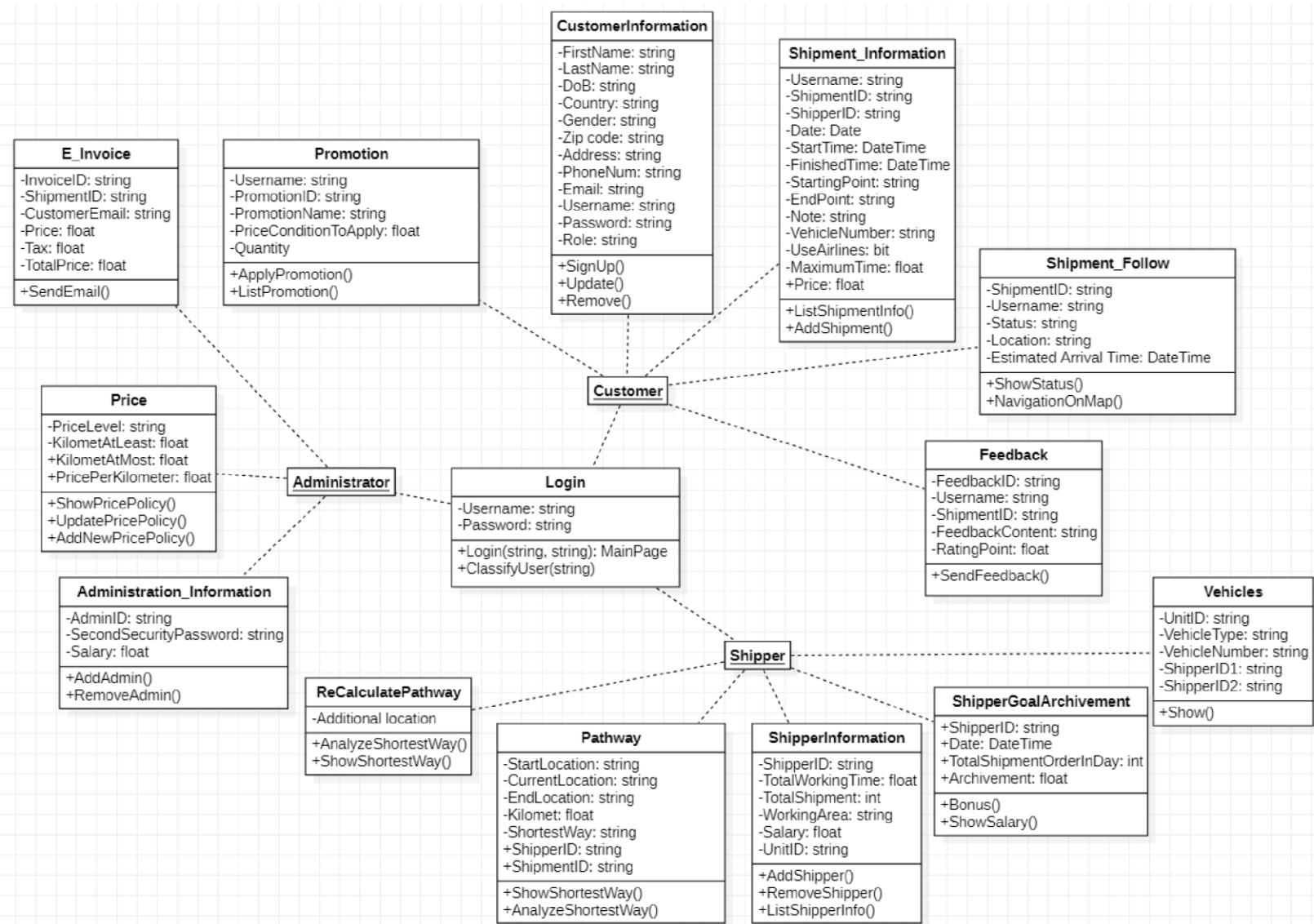
- Client requirements: has all the privileges on exercise seeker and the exercise seeker details. Client cannot make any change to the systems except register and add to my favorite.
- Backup/restore

### **\* Non-functional requirement (user related)**

- Resilience: Update routine, update listfriend (the user can change the number of friends/exercises in their private list)
- Usability: the app should be convenient, comfortable and intuitive.
- Efficiency: Speed of performing must be optimal. The timer calculation must be accurate. The routine/exercise should be effective.
- Compatibility: allows insert new song/exercises from an existing Excel file.

# CHAPTER 6: SYSTEM MODELS

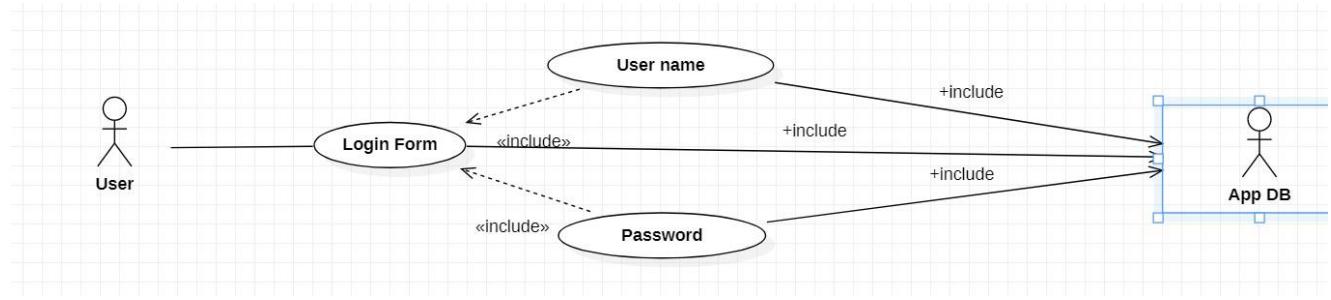
## 6.1 Data flow



## 6.2 Function design

### A. USE CASE DIAGRAM CUSTOMER SIDE

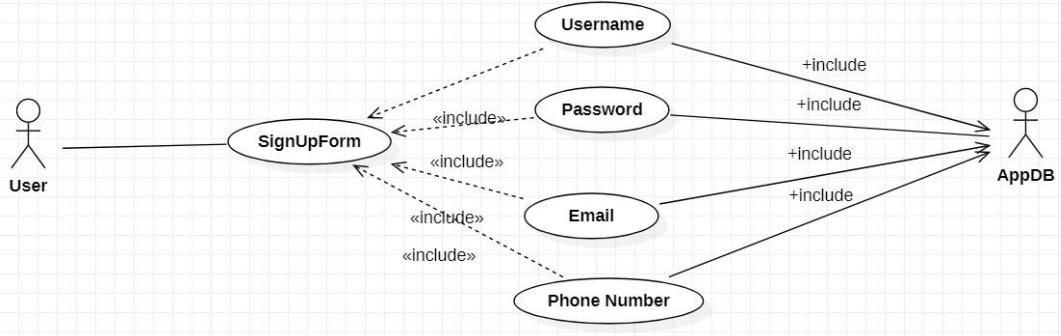
#### Usecase Login



#### Usecase Login

<b>Actors</b>	User, Database
<b>Description</b>	An user can log in their account into the system based on database by inputing their username/email and password in the login form
<b>Data</b>	Users (database)
<b>Stimulus</b>	Interface including form and button
<b>Response</b>	User's home page
<b>Comments</b>	The user must have a valid account and input correctly into the form. The user may have to confirm their account by using email/phone verification such as phone call, SMS, ... in some system

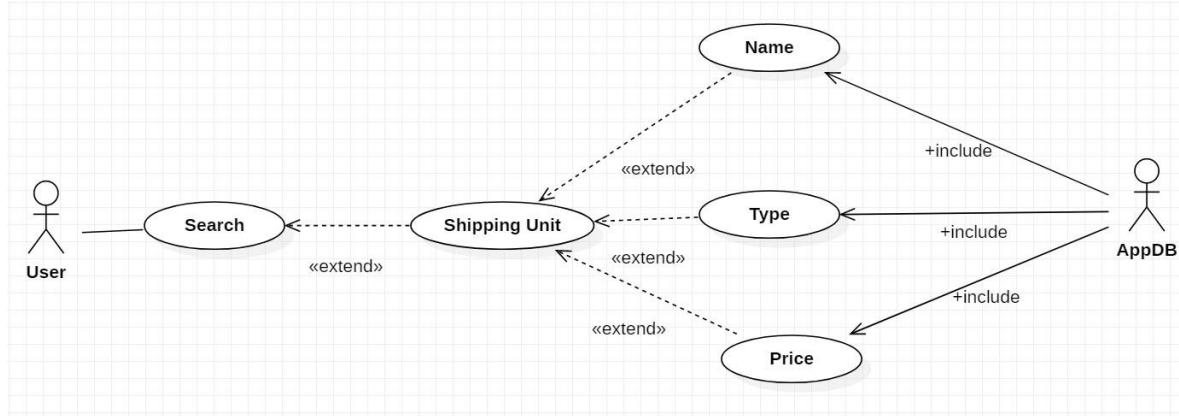
## Usecase Sign up



### Usecase SignUp

<b>Actors</b>	User, Database
<b>Description</b>	An user can create a new account in the system by filling some basic information in sign up form
<b>Data</b>	Users (database)
<b>Stimulus</b>	Interface including form and button
<b>Response</b>	Message: creating a new account successfully
<b>Comments</b>	The user must input some basic and valid information (username is unique) into the form. In some system, the user may have to verify by using email/phone or to wait for admin's acceptance

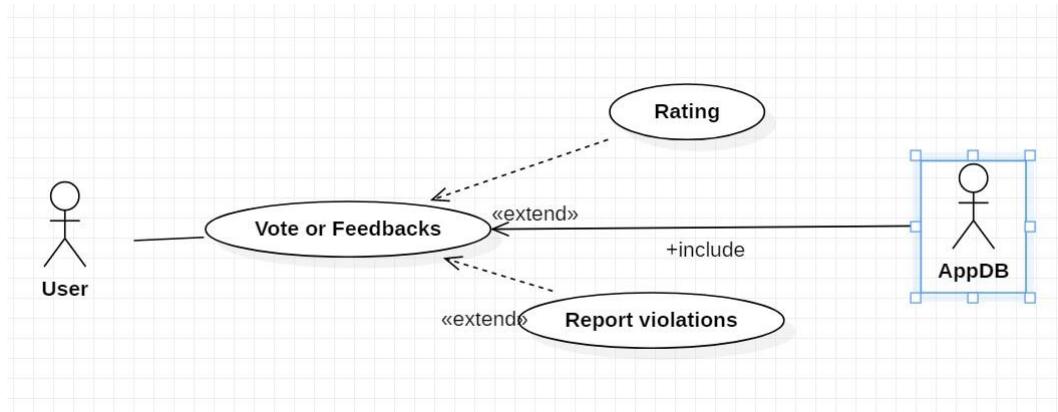
## Usecase: Search for shipping unit



## Usecase: Search for shipping unit

<b>Actors</b>	User, Database
<b>Description</b>	An user can search for shipping unit in 3 ways: price, name of Shipping Unit and Type of Delivery to booking
<b>Data</b>	Search (database)
<b>Stimulus</b>	Interface including the list of results and the button
<b>Response</b>	Message: There are ... results for your requirement.
<b>Comments</b>	The user may search shipping unit based on their needs in order to booking a shipper to delivery goods.

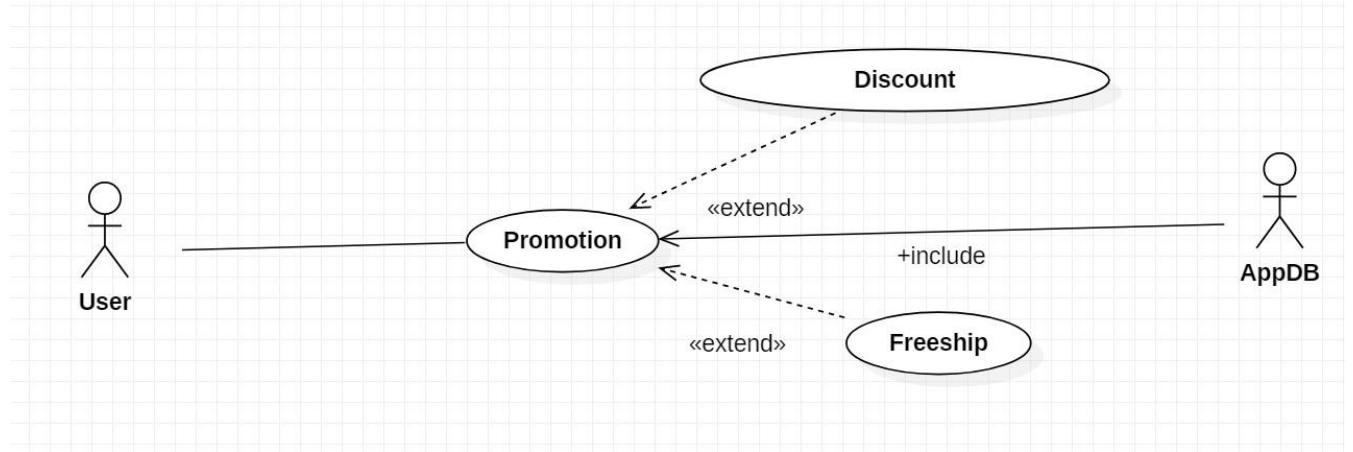
## Usecase: Vote or Feedbacks



### Usecase: Vote or Feedbacks

<b>Actors</b>	User, Database
<b>Description</b>	An user input some feedbacks and vote for shipper
<b>Data</b>	User's input
<b>Stimulus</b>	Interface include input form and submit button
<b>Response</b>	Message: You feedback is received
<b>Comments</b>	The user must input their felling about the shipper into the form for rating. Some system, users just need input a single line and using submit button in order to send to the system

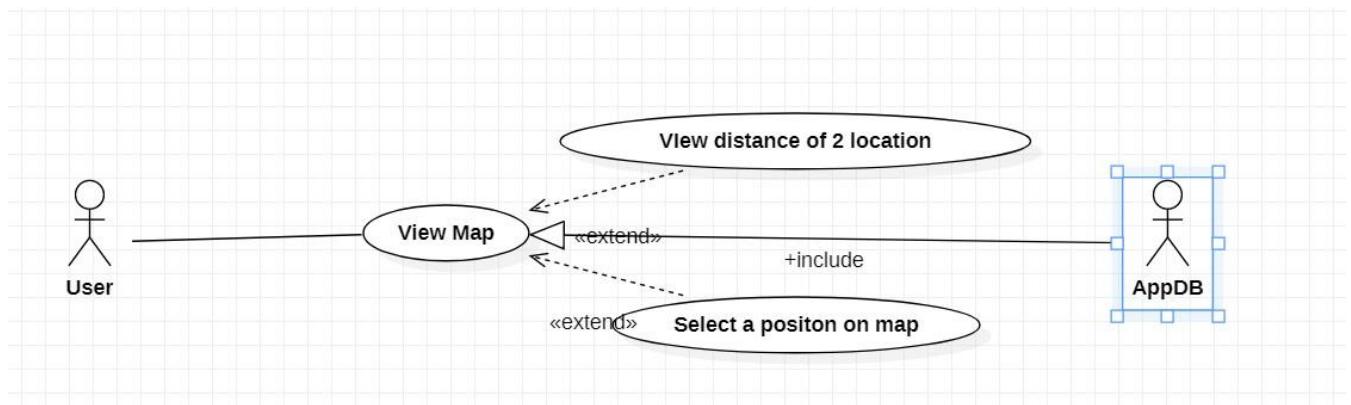
## Usecase: Promotion



## Usecase: Promotion

<b>Actors</b>	User, Database
<b>Description</b>	A user can choose promotion in a list of promotion displayed on the screen and then click on button Apply
<b>Data</b>	User's choice
<b>Stimulus</b>	Interface include a list view and button
<b>Response</b>	Apply promotion successfully
<b>Comments</b>	The user can choose a promotion or not. It can provide a discount or maybe freeship and it helps user save their money

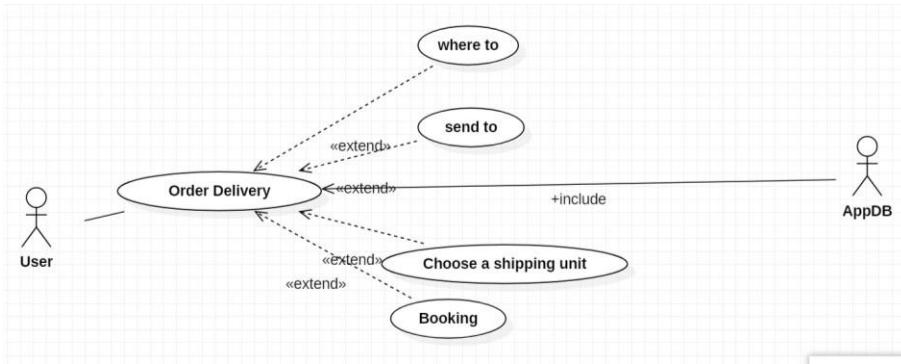
## Usecase: View map



### Usecase: View Map

<b>Actors</b>	User, Database
<b>Description</b>	User can view the map, manipulate on the map
<b>Data</b>	User's input, manipulate
<b>Stimulus</b>	Interface include input view and button
<b>Response</b>	The result on the screen
<b>Comments</b>	The user may use the map to view how far from position A to position B, predict the delivery time, find the shortest way,...

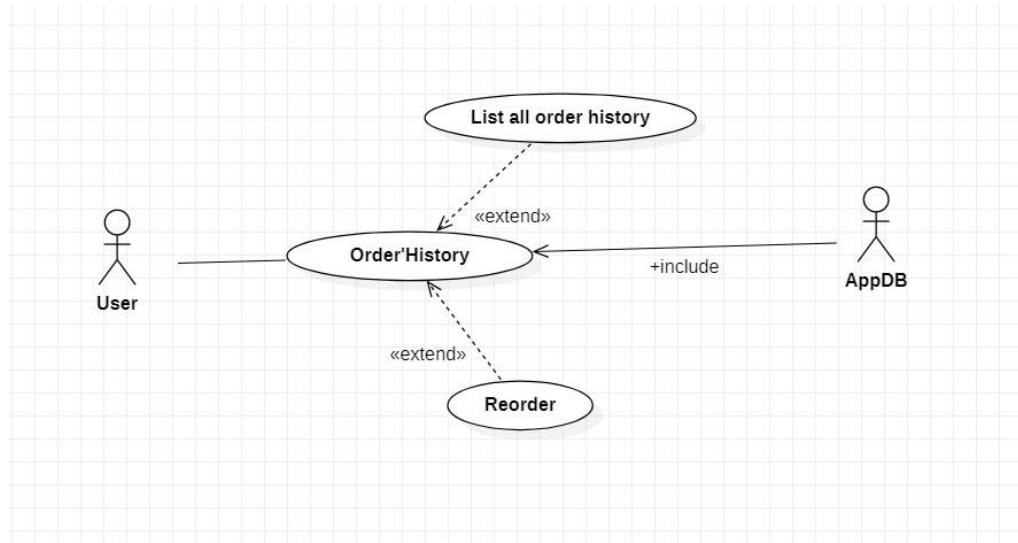
## Usecase: Order delivery



### Usecase: Order Delivery

<b>Actors</b>	User, Database
<b>Description</b>	User can order a shipper to delivery something they want
<b>Data</b>	User's input and submit
<b>Stimulus</b>	Interface include input form and button
<b>Response</b>	Successfully Return form
<b>Comments</b>	The user input 2 locations A, B and choose a shipping unit. After that, press button Booking to order

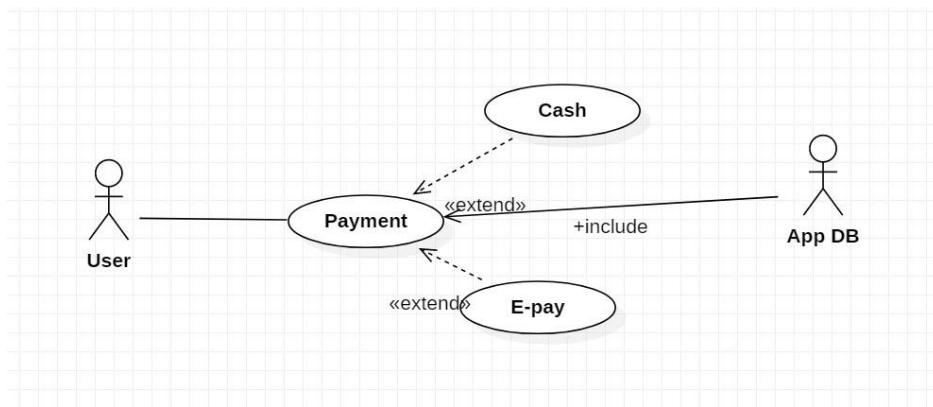
## Usecase: Orders's History



### Usecase: Order History

<b>Actors</b>	User, Database
<b>Description</b>	All history of ordering of user are stored here
<b>Data</b>	A list view and button
<b>Stimulus</b>	Interface include list view and button
<b>Response</b>	Message: All your finished order here
<b>Comments</b>	User can see the list of ordering accomplishmented and they can reorder by click Reorder button

## Usecase: Payment

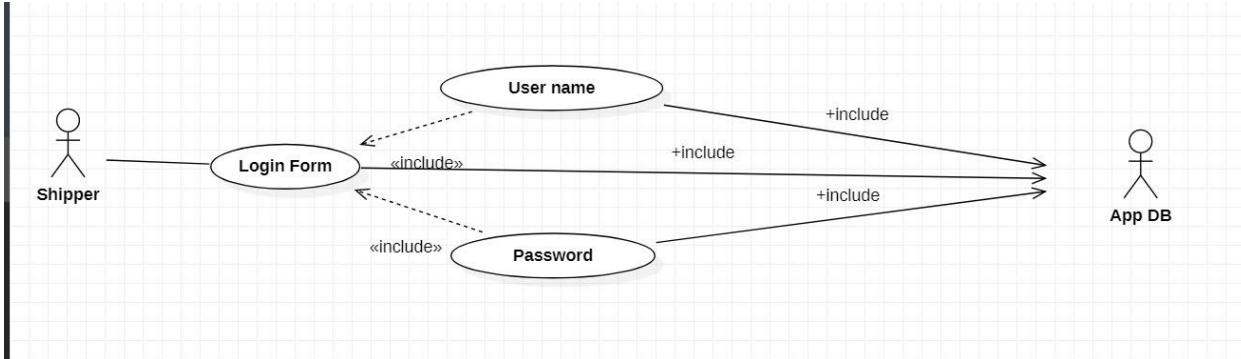


## Usecase: Payment

<b>Actors</b>	User, Database
<b>Description</b>	Access to Payment form to pay the money
<b>Data</b>	User choice
<b>Stimulus</b>	Interface include button and a view form
<b>Response</b>	Message: Successfully or not
<b>Comments</b>	The user can use this service to pay for delivery money before shipper starts work

## SHIPPER SIDE

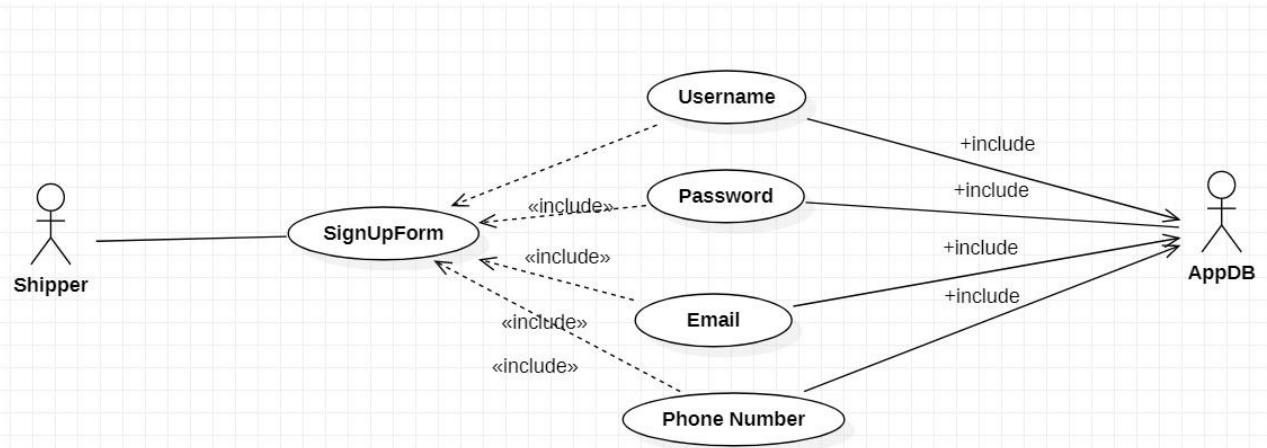
### Usecase: Shipper\_Login



#### Shipper usecase Login

<b>Actors</b>	Shipper, Database
<b>Description</b>	An shipper can log in their account into the system based on database by inputing their username/email and password in the login form
<b>Data</b>	Shippers (database)
<b>Stimulus</b>	Interface including form and button
<b>Response</b>	Shipper's home page
<b>Comments</b>	The shipper must have a valid account and input correctly into the form. The shipper may have to confirm their account by using email/phone verification such as phone call, SMS, ... in some system

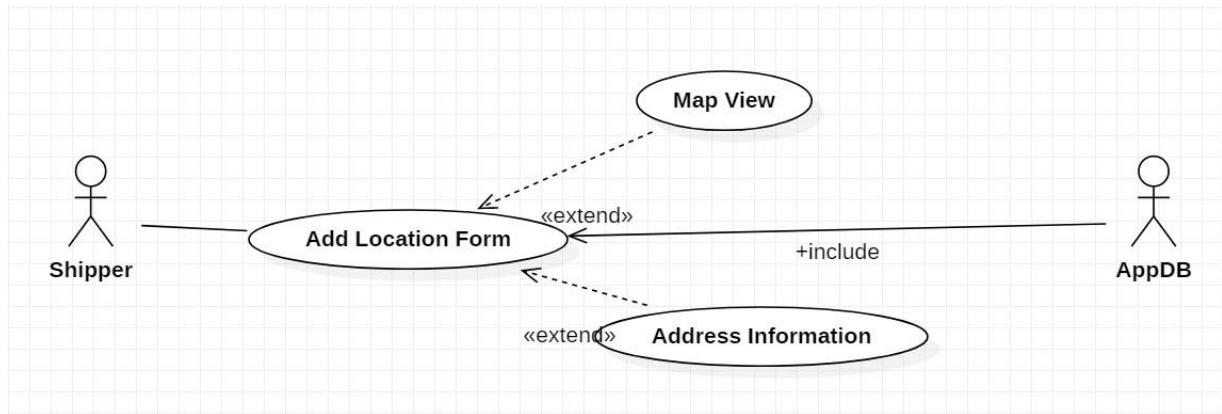
## Usecase: Login\_SignUp



### Shipper usecase SignUp

<b>Actors</b>	Shipper, Database
<b>Description</b>	An shipper can create a new account in the system by filling some basic information in sign up form
<b>Data</b>	Shippers (database)
<b>Stimulus</b>	Interface including form and button
<b>Response</b>	Message: creating a new account successfully
<b>Comments</b>	The shipper must input some basic and valid information (username is unique) into the form. In some system, the shipper may have to verify by using email/phone or to wait for admin's acceptance

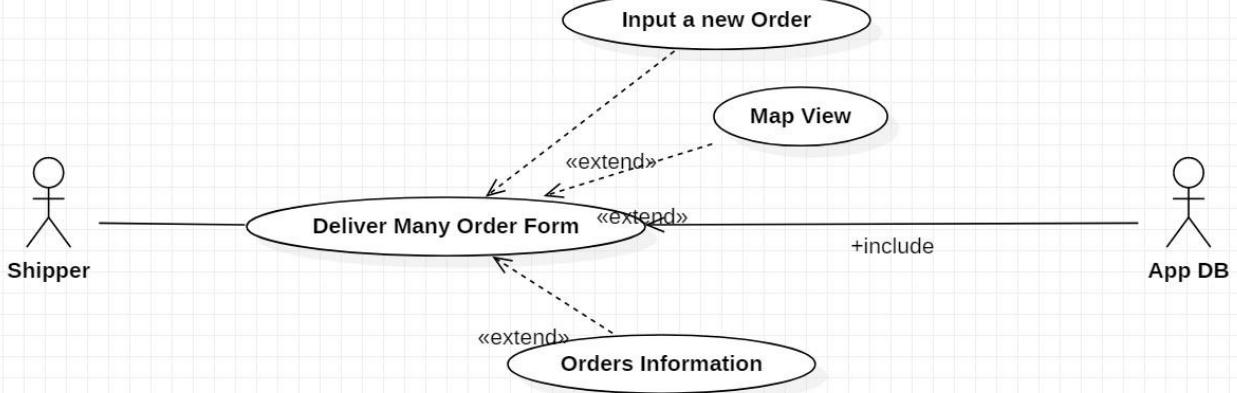
## Usercase: Add another location



### Shipper usecase: Add another location

<b>Actors</b>	User, Database
<b>Description</b>	Shipper can add more location in their delivery way if it is necessary for them or if they think that it will help them to be faster
<b>Data</b>	Shipper's input
<b>Stimulus</b>	Interface that is the map and button to add another location
<b>Response</b>	The list of shipper and button
<b>Comments</b>	The user can use this form to find and call for another's shipper help in case can not finish the order

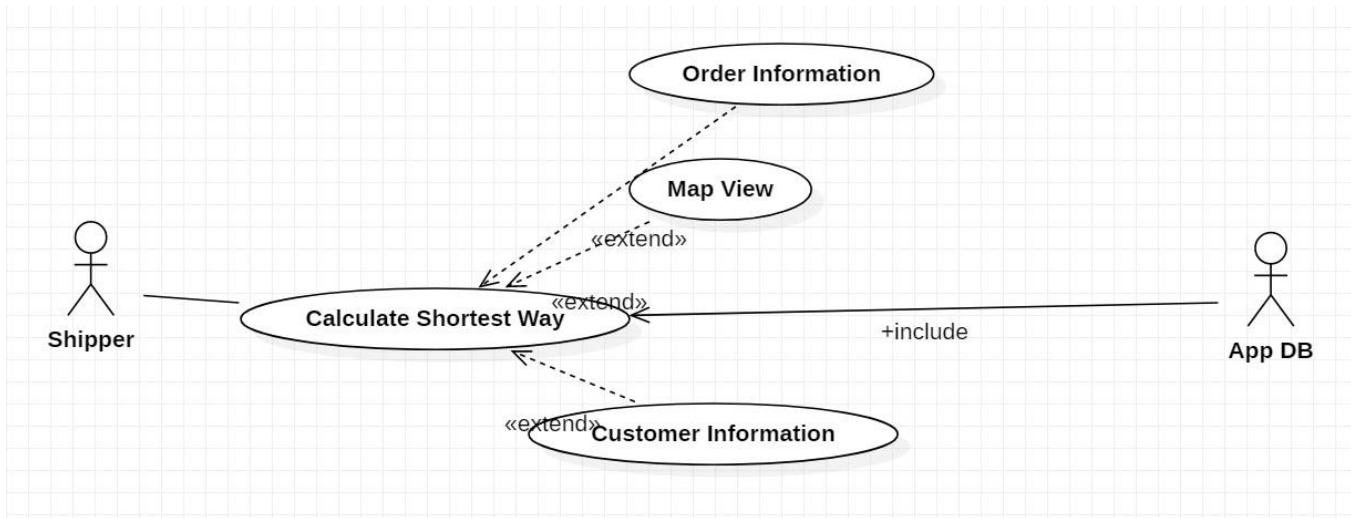
## Usercase: Deliver many orders



### Shipper usecase: Deliver many orders

<b>Actors</b>	User, Database
<b>Description</b>	Shipper can choose to deliver many items from many customer if they or on the same ways but at most 5 location in each shipping
<b>Data</b>	Shipper DB, Shipper's input
<b>Stimulus</b>	Interface include input form and submit button
<b>Response</b>	Add address successfully
<b>Comments</b>	The shipper can use this form to add another address so it can help them to deliver many location

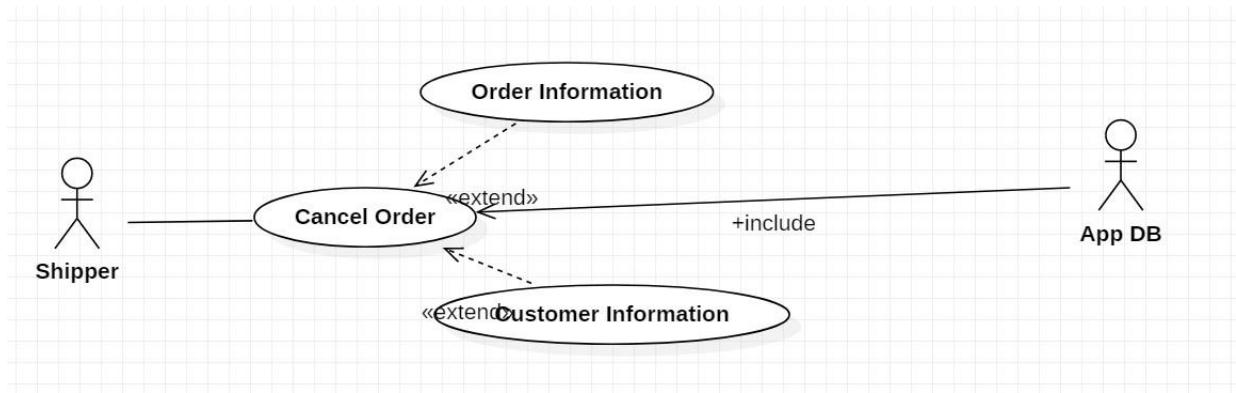
## Usercase: Shipper\_Calculate the shortest way to deliver in fastest way



### Shipper usecase: Calculate the shortest way to deliver in fastest way

<b>Actors</b>	Shipper, Database
<b>Description</b>	System will calculate the shortest and the fastest way to help shipper deliver items as fast as possible and base on that system can calculate the estimate time for shipment.
<b>Data</b>	Database,
<b>Stimulus</b>	Interface include input form and addfriend button
<b>Response</b>	Successfully added your new friends
<b>Comments</b>	The user can addfriend in order to share experience and create motivation when competing with your friend

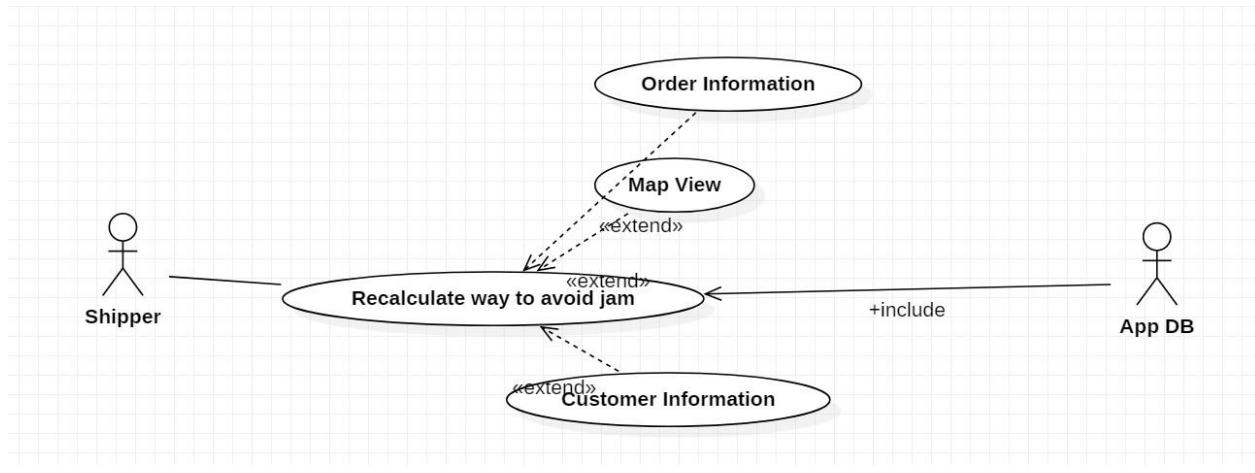
## Usercase: Shipper\_CancelOrder



### Shipper Usercase: CancelOrder

<b>Actors</b>	User, Database
<b>Description</b>	Shipper can cancel order if they can not contact to customer after they reached the sender address or waiting time is more than 5 minutes or customer asked for cancellation.
<b>Data</b>	Shipper's manipulate
<b>Stimulus</b>	Cancel button
<b>Response</b>	Canceled Order
<b>Comments</b>	The shipper can cancel order in case cannot connect to customer,... in order to stop and start another order

## Usecase: Shipper\_RecalculateWaytoAvoidJam



### Shipper Usecase: Recalculate way to avoid traffic jam

<b>Actors</b>	Shipper
<b>Description</b>	If customer is on the way to deliver items but system find out that shipper is stuck at that street in 5-10 minutes so system need to re-calcualate the delivery way again and help customer to get out that traffic jam.
<b>Data</b>	Shipper's Manipulate
<b>Stimulus</b>	Interface contains map, button
<b>Response</b>	Show result in map
<b>Comments</b>	Shipper can view on map and decide another way to deliver instead of continuing on the crowded way

## Usecase: Shipper\_ChangeShipper

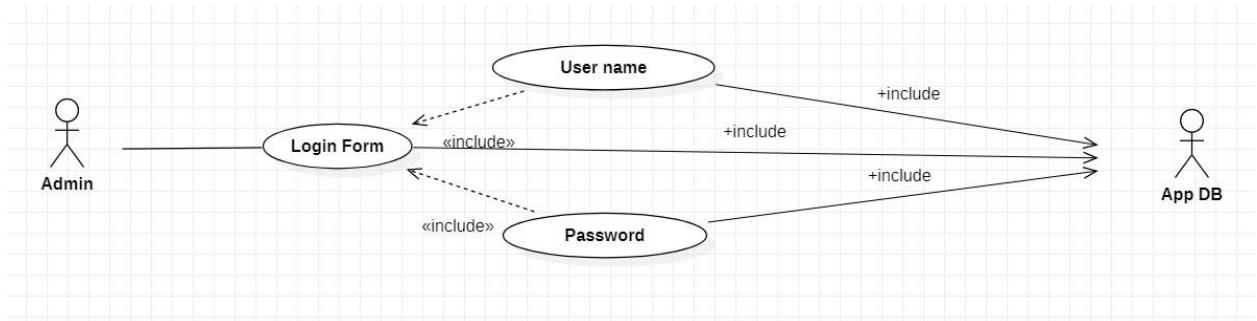
---

### Shipper Usecase: Change Shipper

<b>Actors</b>	User
<b>Description</b>	After get items from customer, if shipper can not continue to delivery or they have any problem, violate with time policy of shipment, shipper can request system to find out other nearest shipper can help them to continue their job.
<b>Data</b>	Shipper's button to find another one who is free to help finish the delivery
<b>Stimulus</b>	Interface has button to find another shipper, button to contact
<b>Response</b>	Show the list the shipper is in free time
<b>Comments</b>	Shipper use this to call for help from another one in case they cannot continue to delivery

## ADMIN SIDE

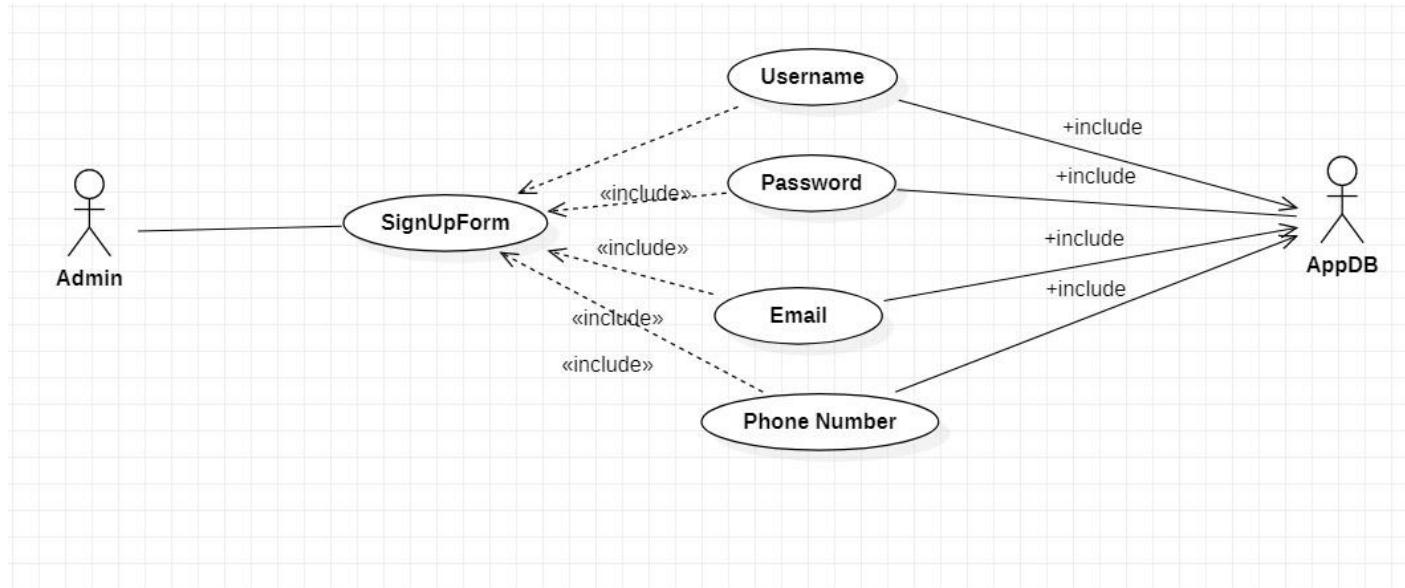
### Usercase: Admin\_Login



### Admin usercase: Login

<b>Actors</b>	Admin, Database
<b>Description</b>	An admin can log in their account into the system based on database by inputting their username/email and password in the login form
<b>Data</b>	Admin (database)
<b>Stimulus</b>	Interface including form and button
<b>Response</b>	Admin's home page
<b>Comments</b>	The admin must have a valid account and input correctly into the form. The admin may have to confirm their account by using email/phone verification such as phone call, SMS, ... in some system

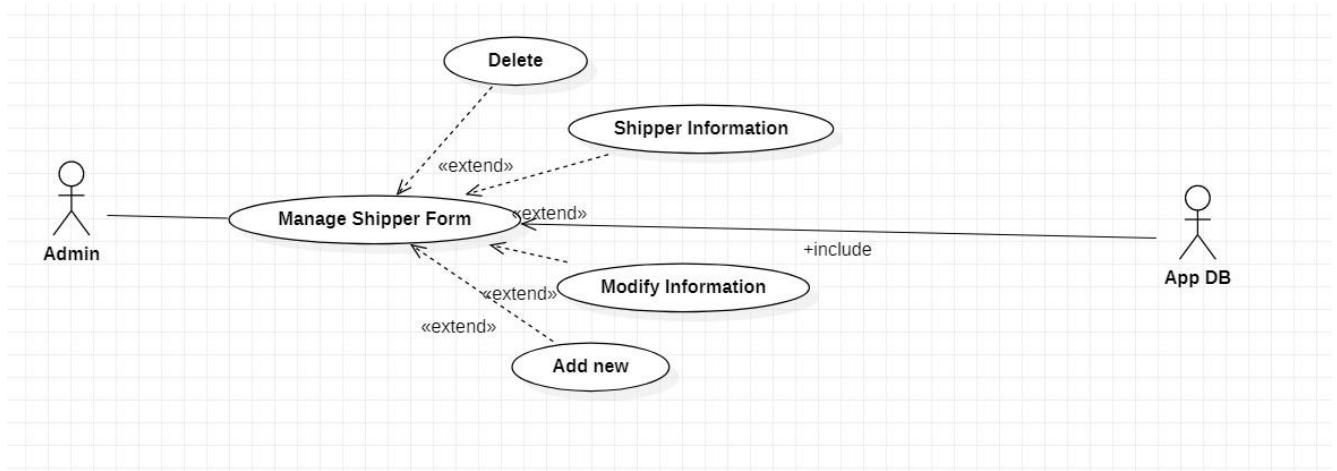
## Usercase: Admin\_SignUp



### Admin usercase: Sign Up

<b>Actors</b>	Admin, Database
<b>Description</b>	An admin can create a new account in the system by filling some basic information in sign up form
<b>Data</b>	Admin (database)
<b>Stimulus</b>	Interface including form and button
<b>Response</b>	Message: creating a new account successfully
<b>Comments</b>	The admin must input some basic and valid information (username is unique) into the form. In some system, the admin may have to verify by using email/phone or to wait for admin's acceptance

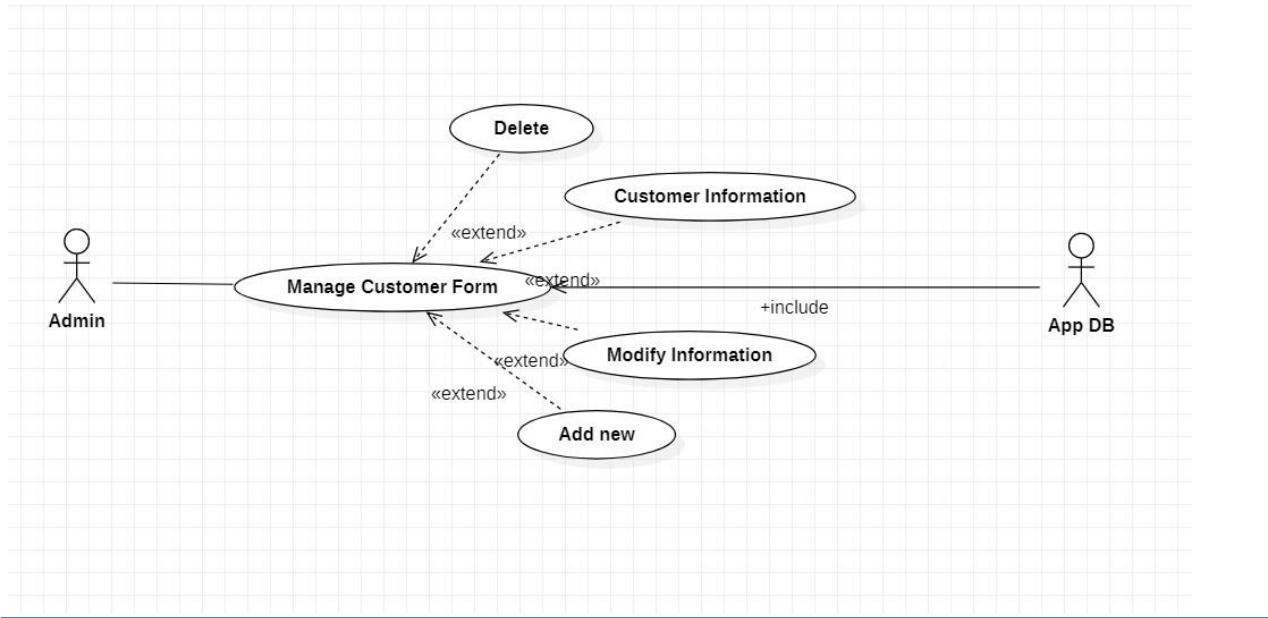
## Usercase: Admin\_Manage Shipper Information



### Admin usercase: Manage Shipper Information

<b>Actors</b>	Admin, Database
<b>Description</b>	Admin can get all of information of shipper in the database to manage them. Show out good shipper and bad shipper to have suitable solution to enhance the shipping policy. This also help admin can manage the personal information of shipper and it will necessary in some cases. For example: Contact with family when shipper get accident, send information to police if shipper steal goods from customer and hide away.
<b>Data</b>	Admin (database)
<b>Stimulus</b>	Interface including form and button
<b>Response</b>	Admin's home page
<b>Comments</b>	The admin use this form to manage shipper information with the data of shipper from DB and also can add, delete and modify

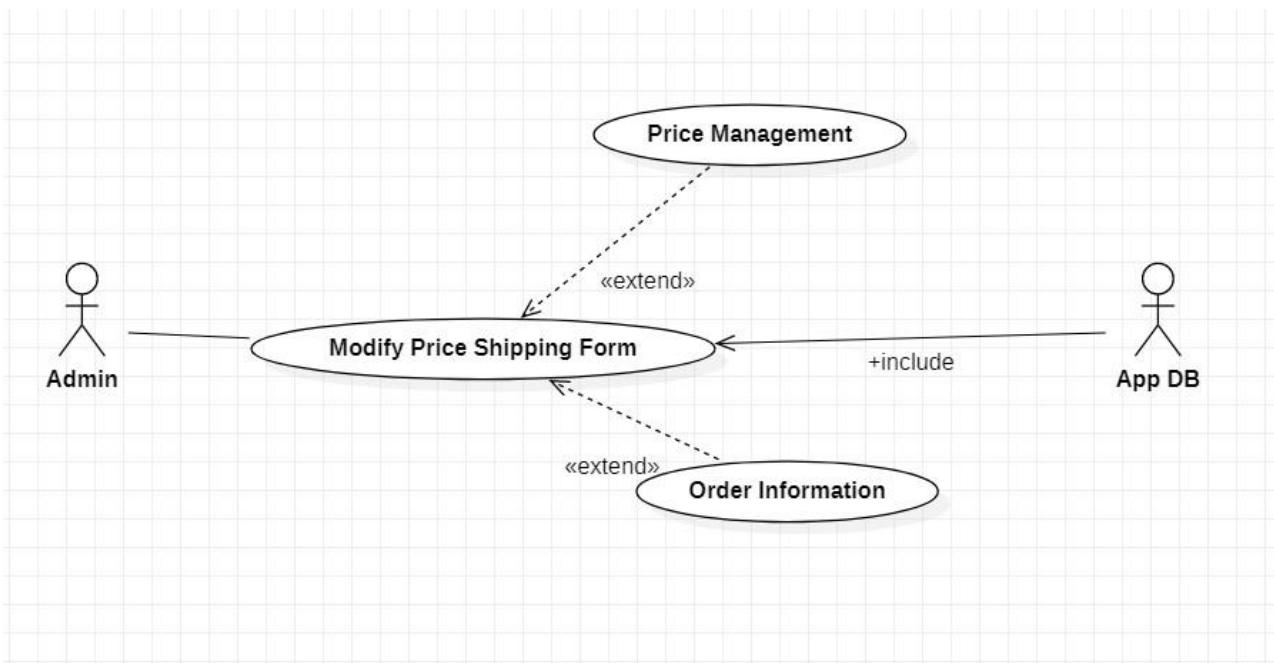
## Usercase: Admin\_Manage Customer Information



### Admin usercase: Manage Customer Information

<b>Actors</b>	Admin, Database
<b>Description</b>	Admin can get all of information of customer including: Name, address, phone number and number of using system from that customer. This is not only help admin can manage customer but also help them to find out the potential customer.
<b>Data</b>	Admin (database)
<b>Stimulus</b>	Interface including form and button
<b>Response</b>	Admin's home page
<b>Comments</b>	The admin use this form to manage customer information with the data of customer from DB and also can add, delete and modify

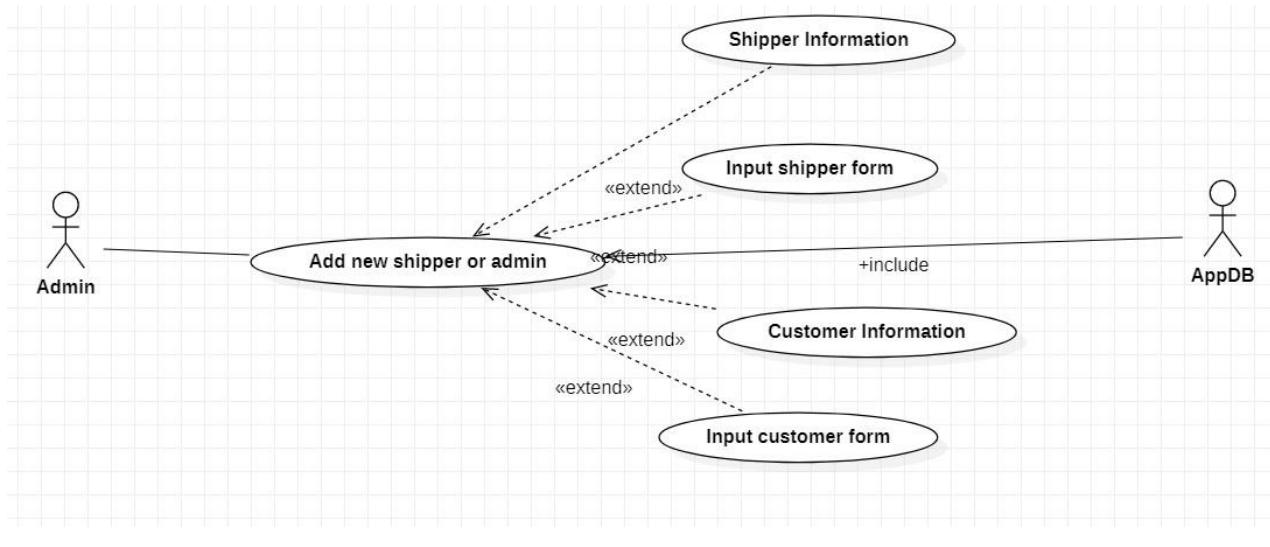
## Usercase: Admin\_Modify Shipping Price



### Admin usercase: Modify Shipping Price

<b>Actors</b>	Admin, Database
<b>Description</b>	The administrator can change the cost of shipping based on the policy of shipping and price of fuel. They can change the price per km or price per hour of shipping or the transport type.
<b>Data</b>	Admin (database)
<b>Stimulus</b>	Interface including form and button
<b>Response</b>	Admin's home page
<b>Comments</b>	The admin use this site to modify the shipping price with the data of Order and Price in Database

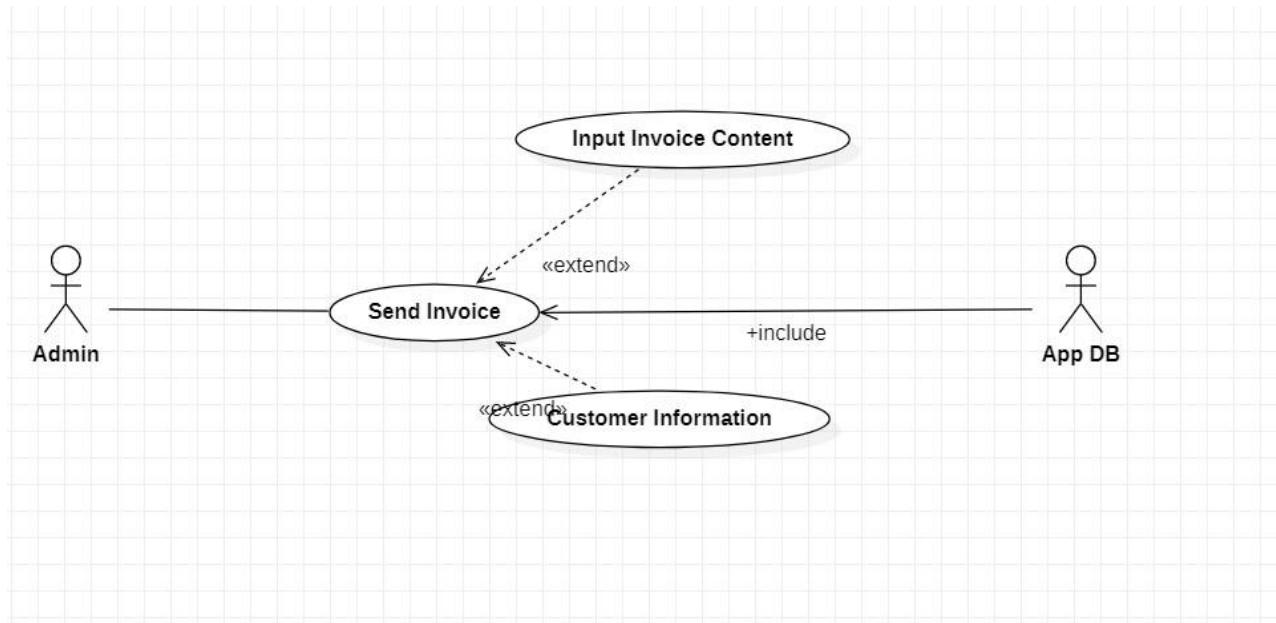
## Usercase: Admin\_Add new shipper or admin into the system



### Admin usecase: Add new shipper or admin into the system

<b>Actors</b>	Admin, Database
<b>Description</b>	Sign up function of the system can not apply for admin and shipper because of the security problem so that it need to be added by administration.
<b>Data</b>	Admin (database)
<b>Stimulus</b>	Interface including form and button
<b>Response</b>	Admin's add newbie page
<b>Comments</b>	The admin use this to add a new shipper or a new administrator into the database

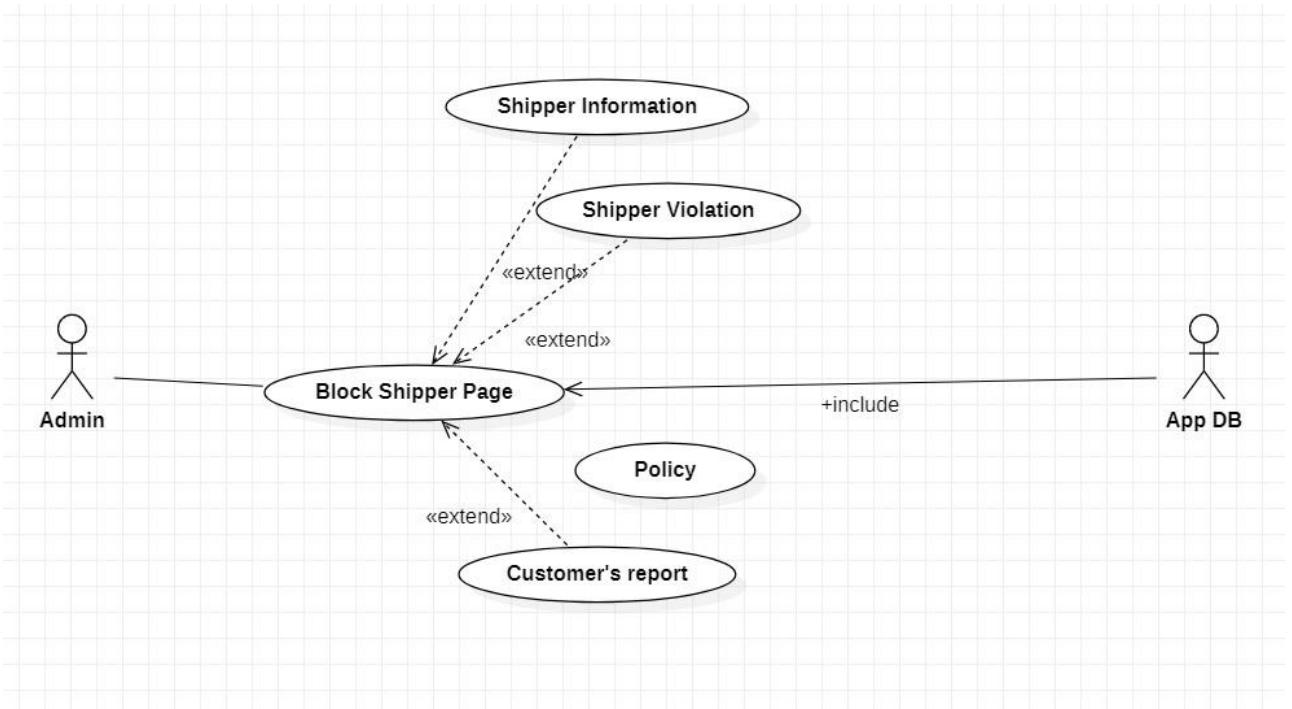
## Usercase: Admin\_Send Invoice via customer Email



### Admin usercase: Send Invoice via customer Email

<b>Actors</b>	Admin, Database
<b>Description</b>	Admin will send the invoice to customer when the order was finished
<b>Data</b>	Admin (database)
<b>Stimulus</b>	Interface including form and button
<b>Response</b>	Admin's invoice page
<b>Comments</b>	The admin use this page send invoice to the customer through Email

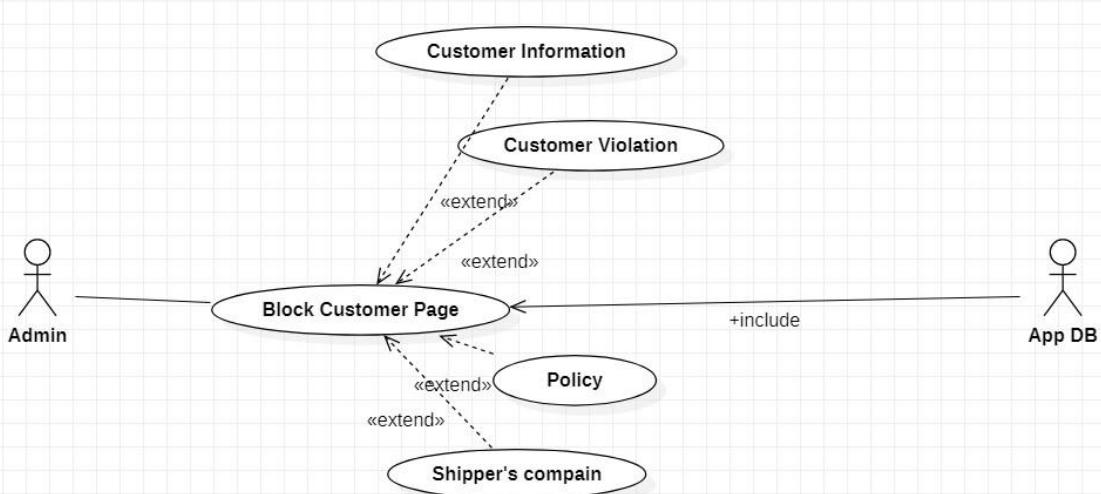
## Usercase: Admin\_Block Shipper



### Admin usecase: Block Shipper

<b>Actors</b>	Admin, Database
<b>Description</b>	If shipper infringed the policy of shipment, admin can block their account.
<b>Data</b>	Admin (database)
<b>Stimulus</b>	Interface including form and button
<b>Response</b>	Admin's block shipper page
<b>Comments</b>	The admin has to block shipper in some case, admin can use this page to do it.

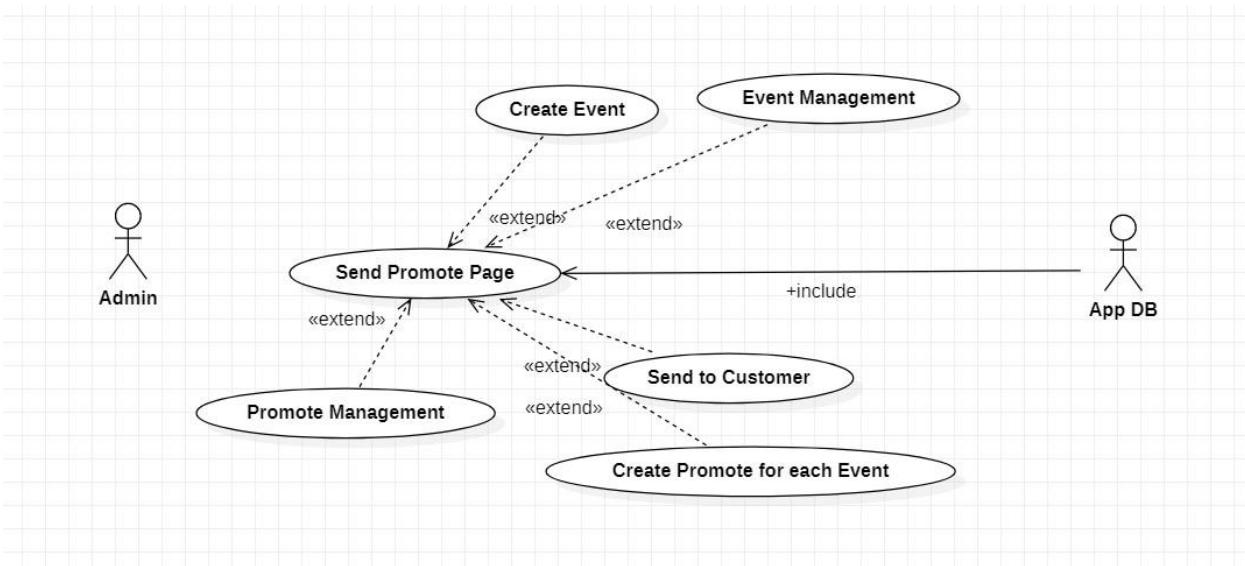
## Usercase: Admin\_BlockCustomer



### Admin usecase: BlockCustomer

<b>Actors</b>	Admin, Database
<b>Description</b>	If customer infringed the policy of shipment, admin can block their account.
<b>Data</b>	Admin (database)
<b>Stimulus</b>	Interface including form and button
<b>Response</b>	Admin's block customer page
<b>Comments</b>	The admin has to block customer in some case, admin can use this page to do it according shipper's report, ...

## Usercase: Admin\_Send Promote to Customer



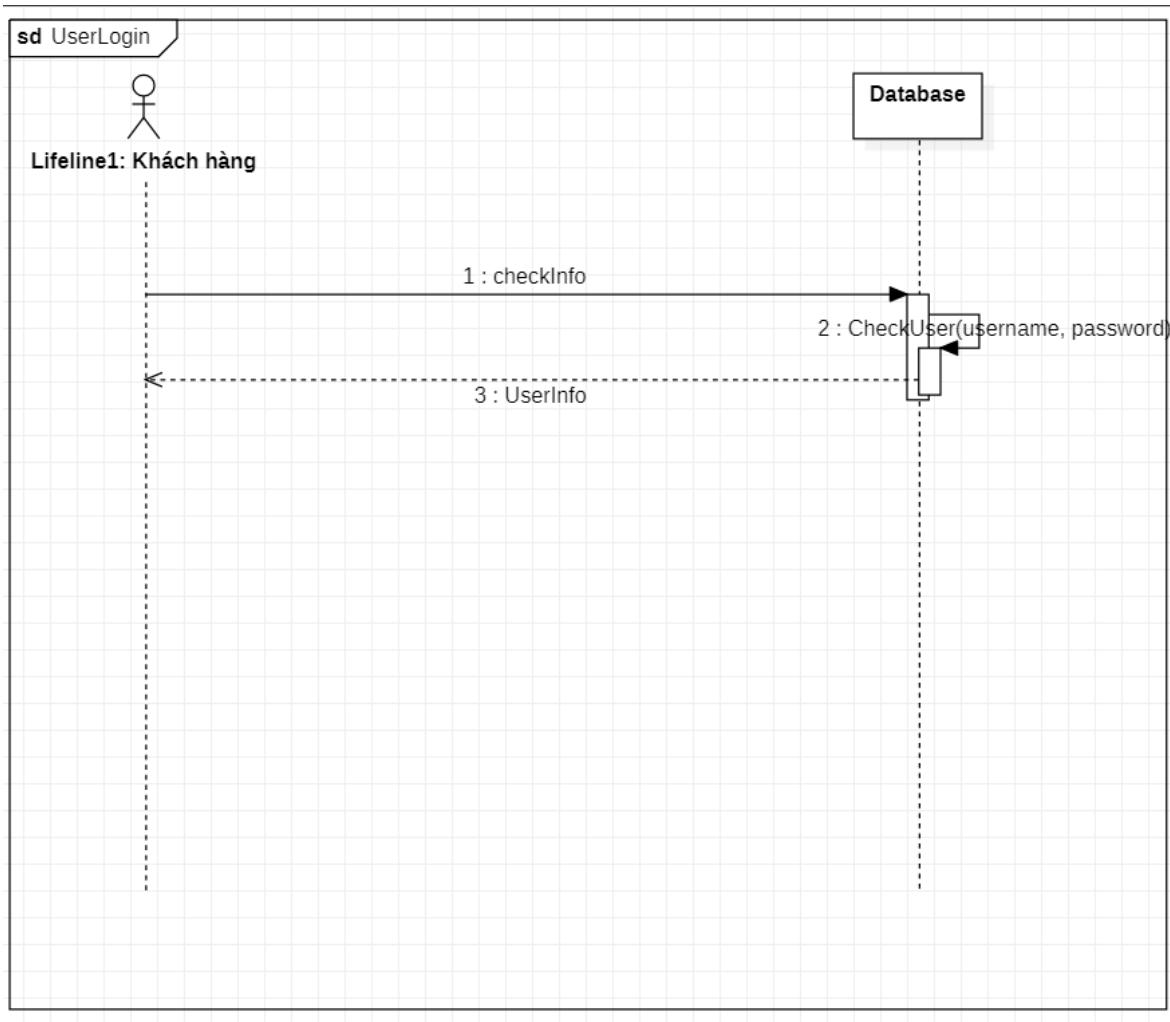
### Admin usecase: Send Promote to Cusomer

<b>Actors</b>	Admin, Database
<b>Description</b>	Admin send promotions code to customer
<b>Data</b>	Admin (database)
<b>Stimulus</b>	Interface including form and button
<b>Response</b>	Admin's promote page
<b>Comments</b>	The admin can manage promote and also add or delete the event provided promote to customer

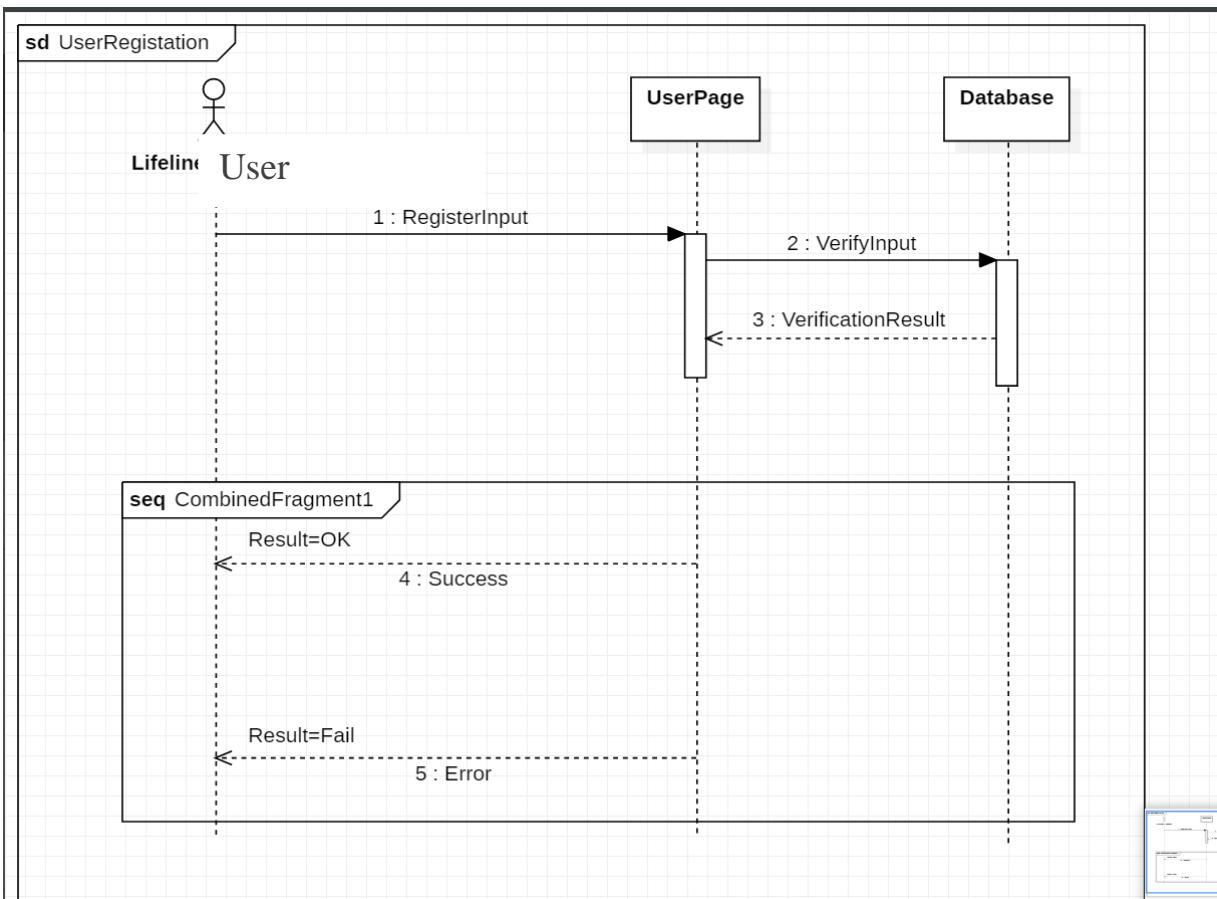
## B. SEQUENCE DIAGRAM

### CUSTOMER SIDE

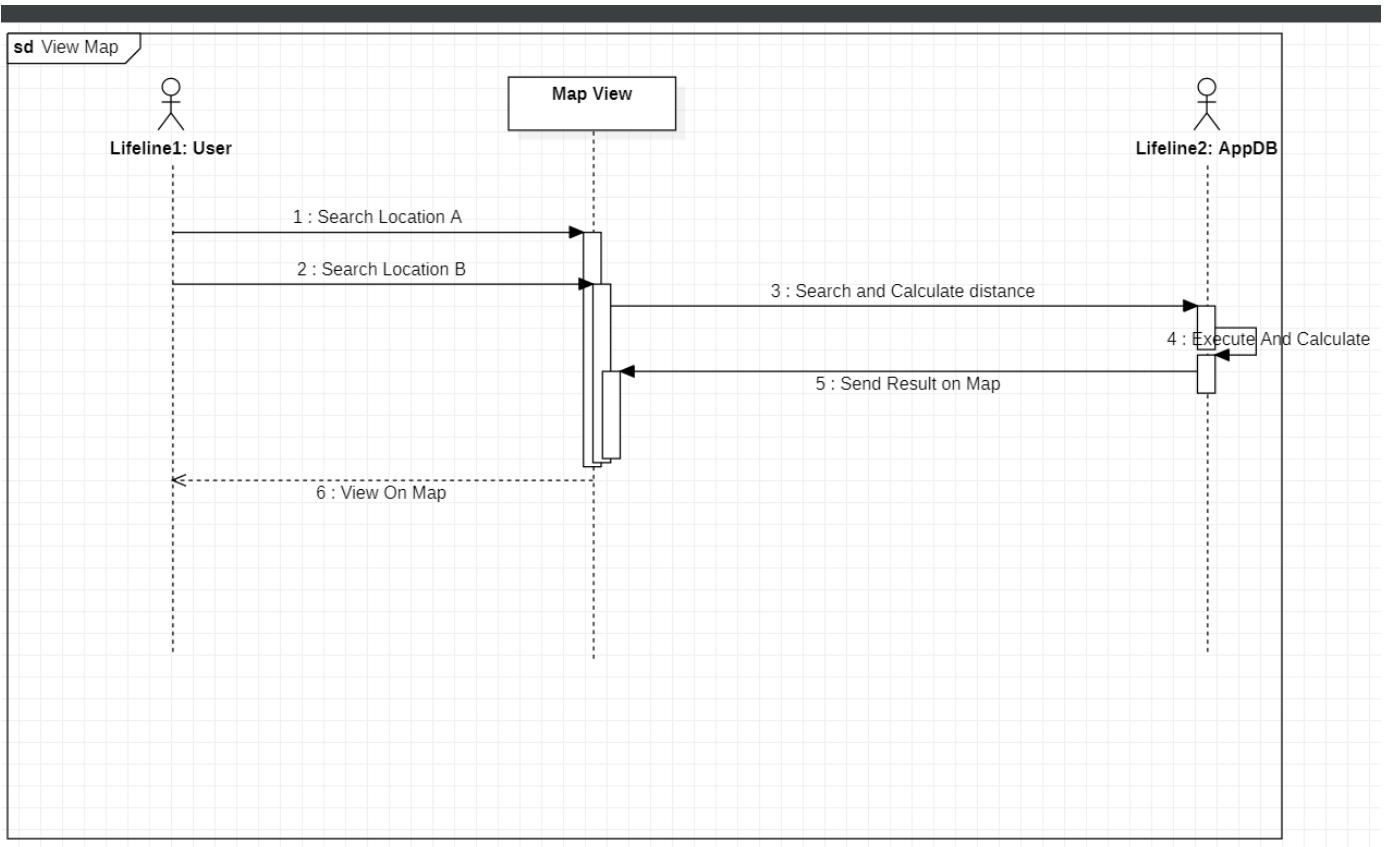
#### Sequence diagram: Login



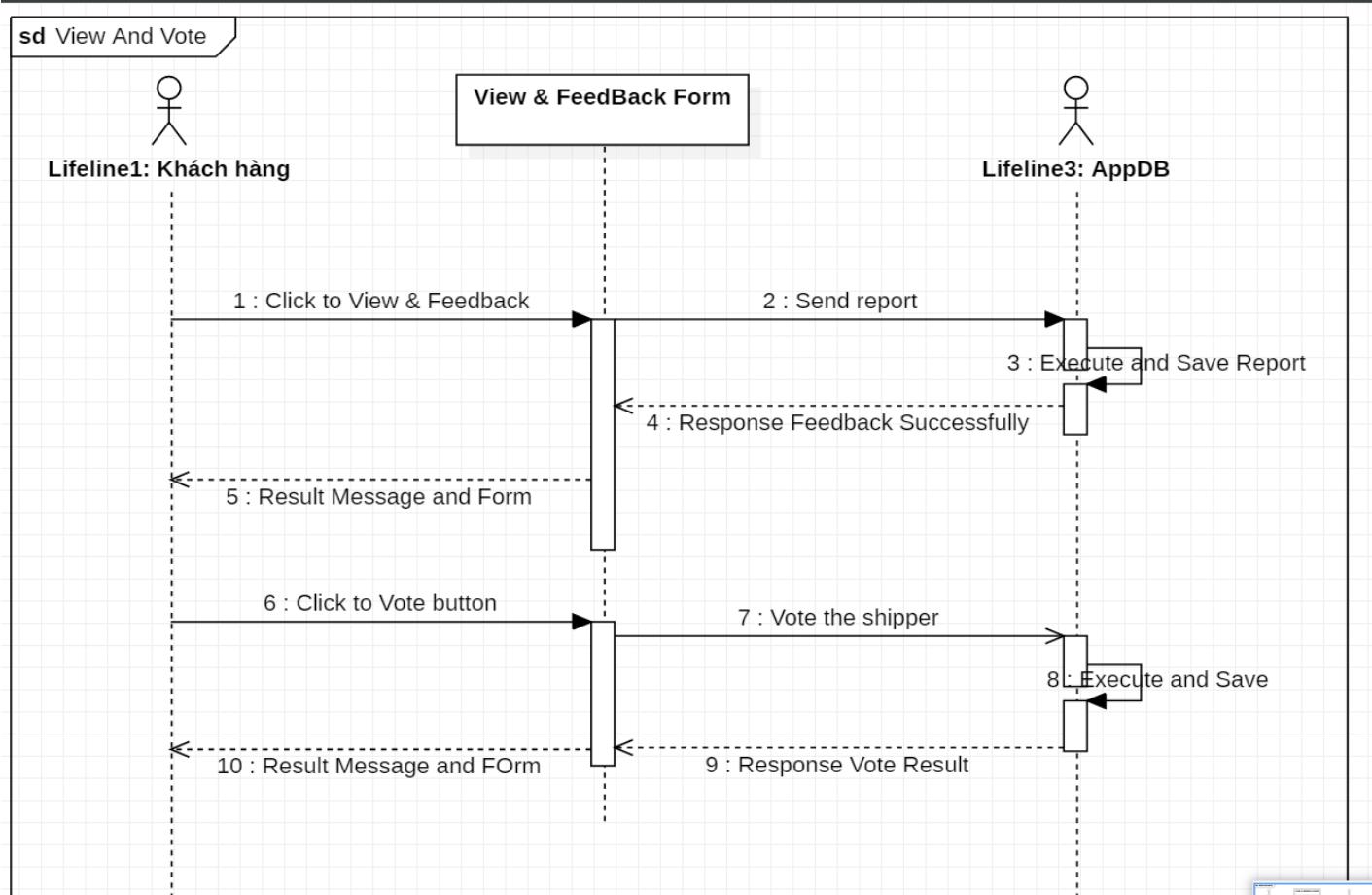
## Sequence diagram: Registration



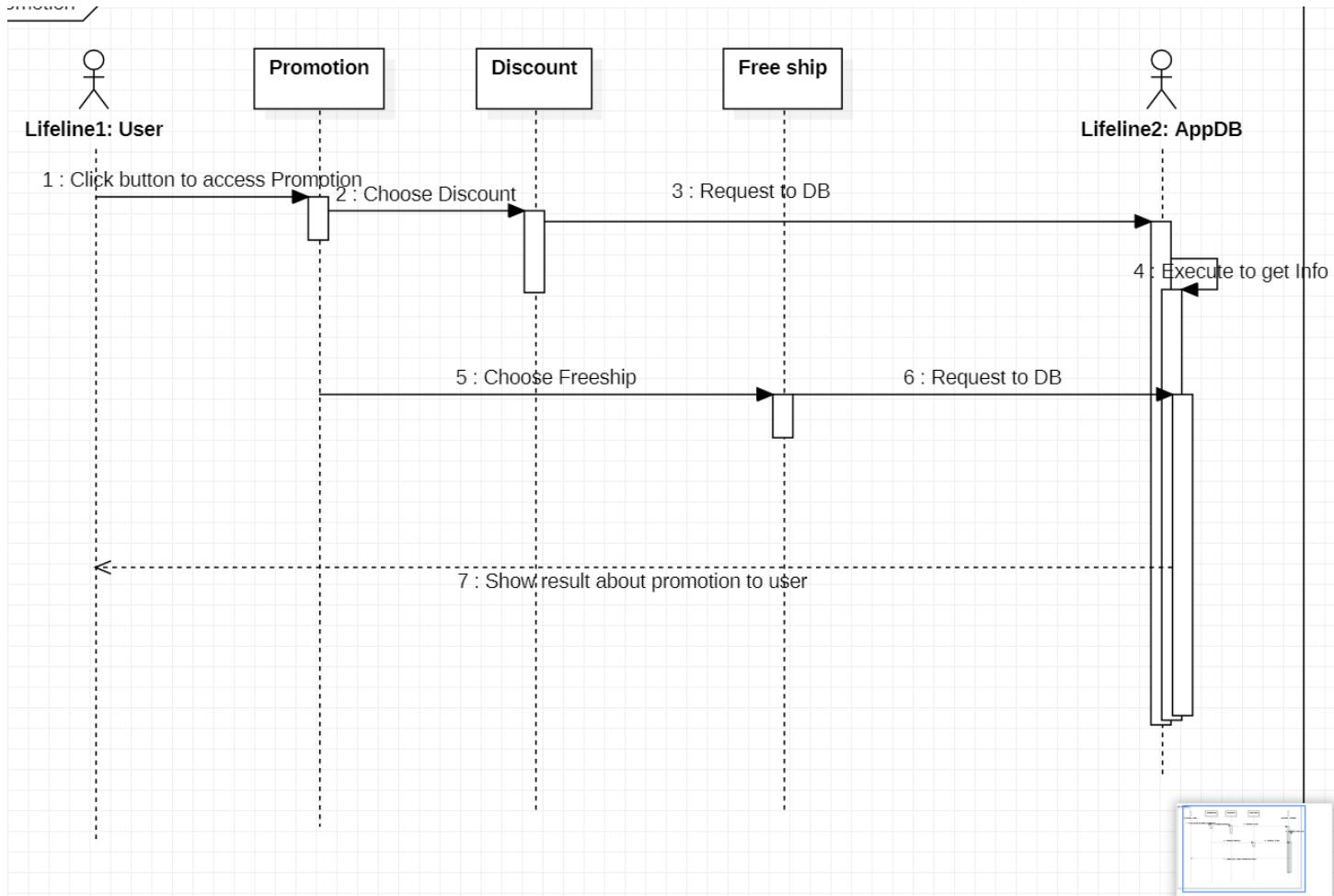
## Sequence diagram: View map



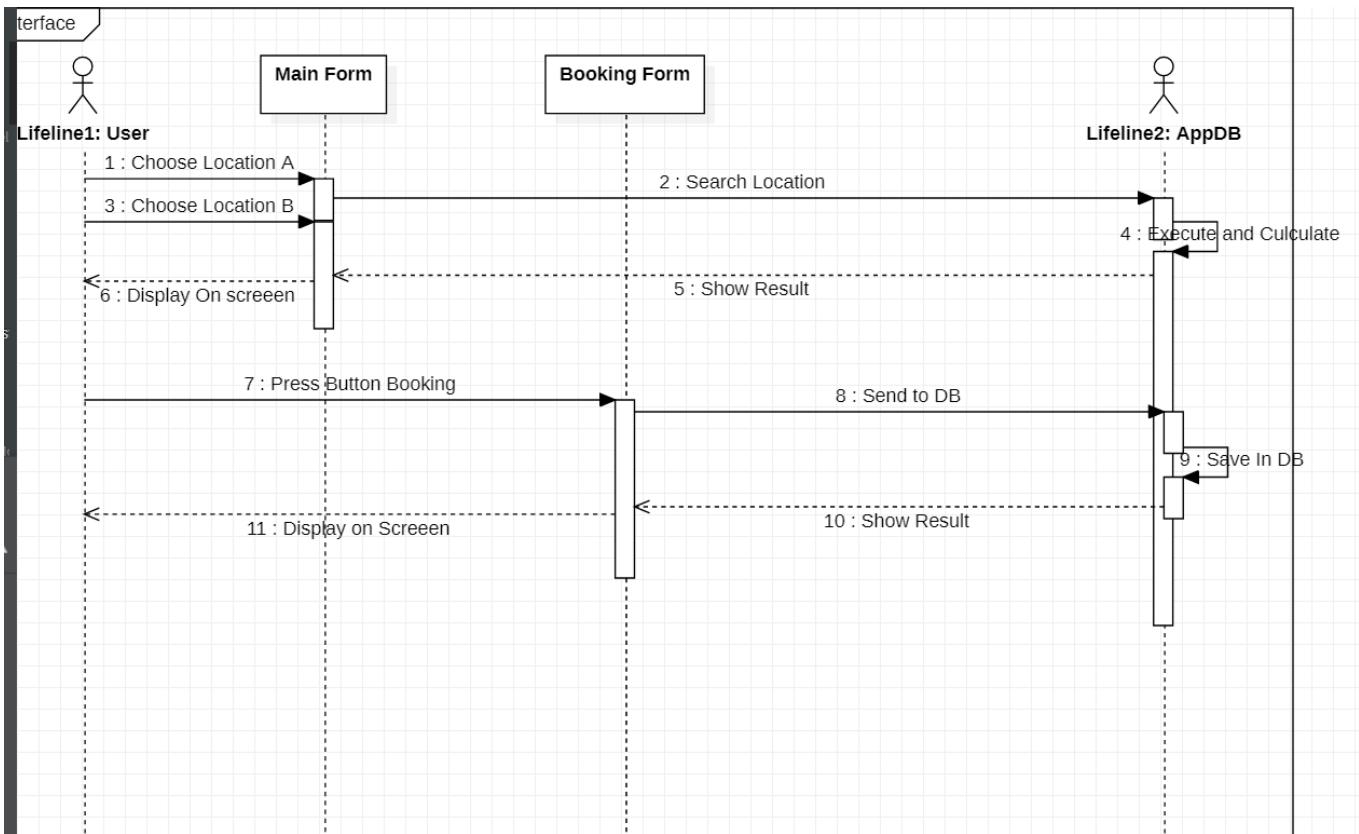
## Sequence diagram: Vote and Feedback



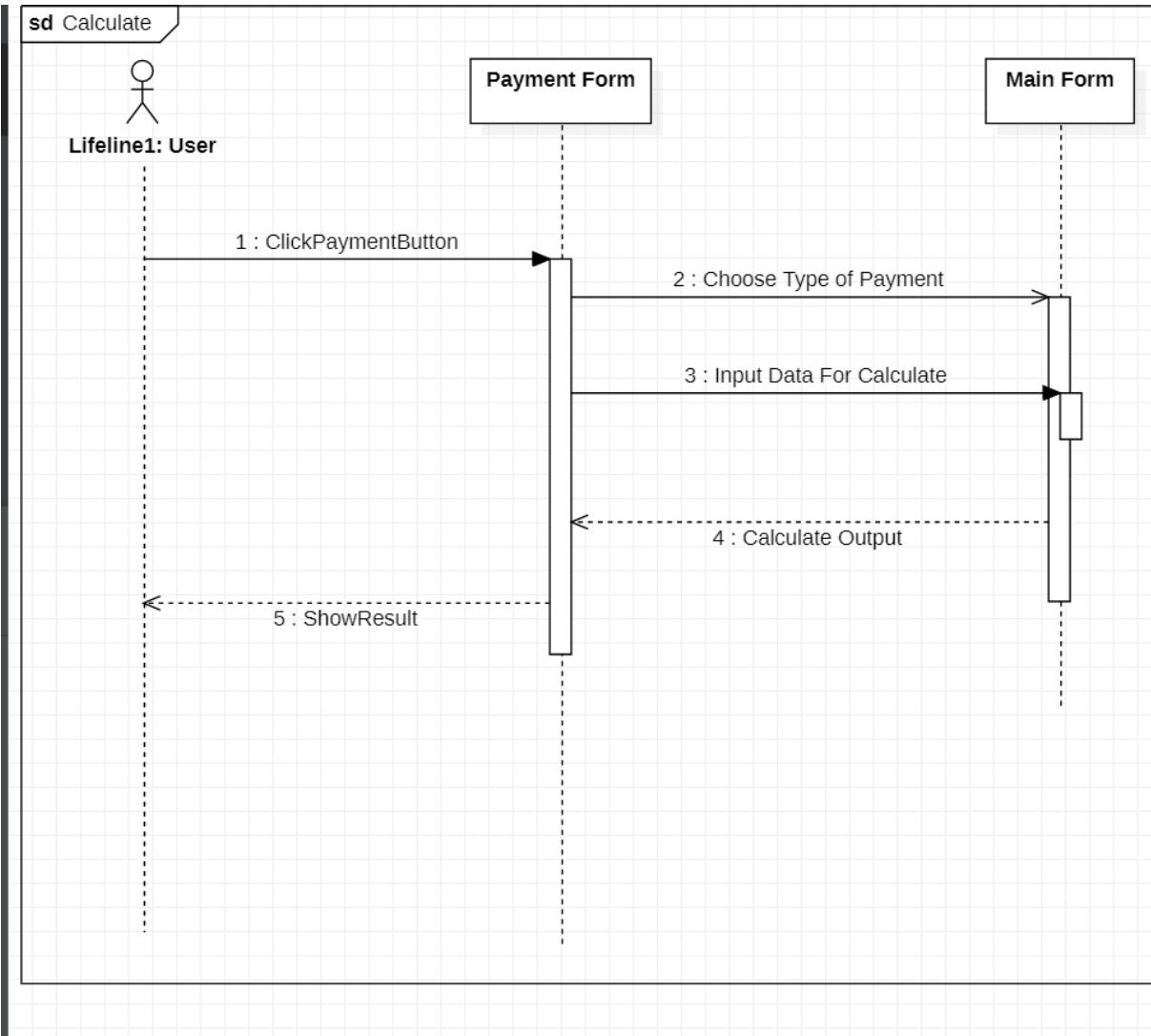
## Sequence diagram: Promotion



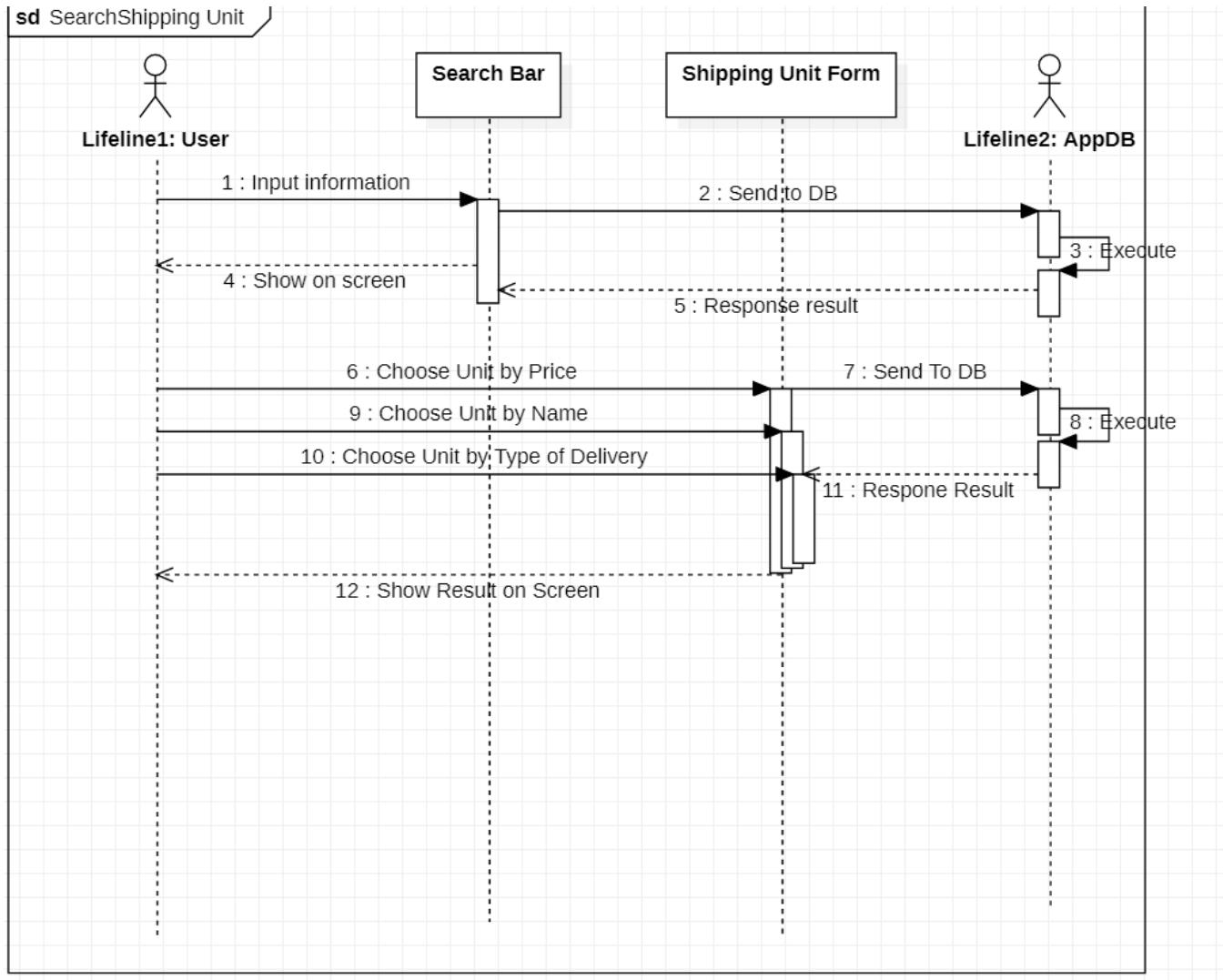
## Sequence diagram: Shipping interface



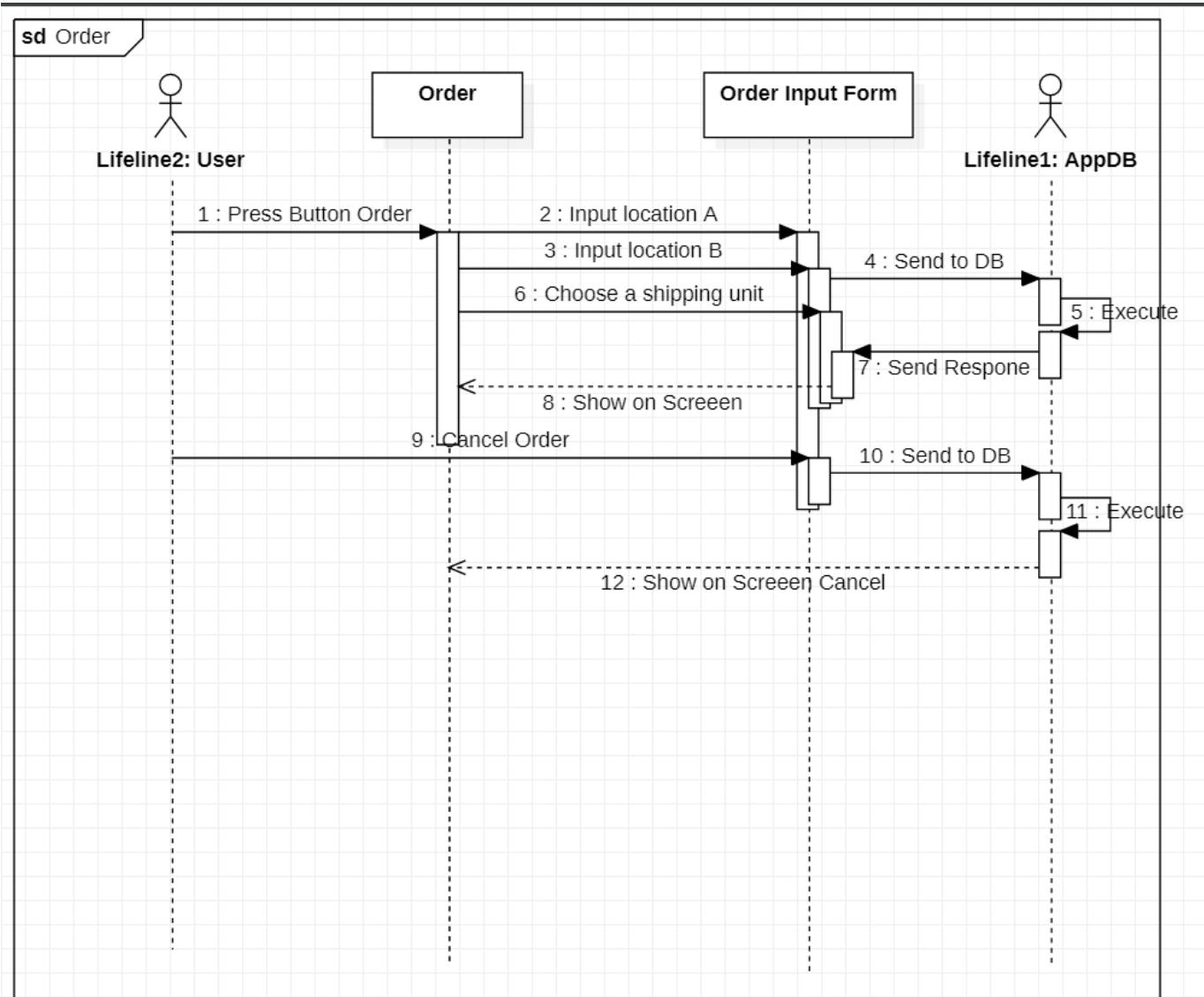
## Sequence diagram: Calculate payment



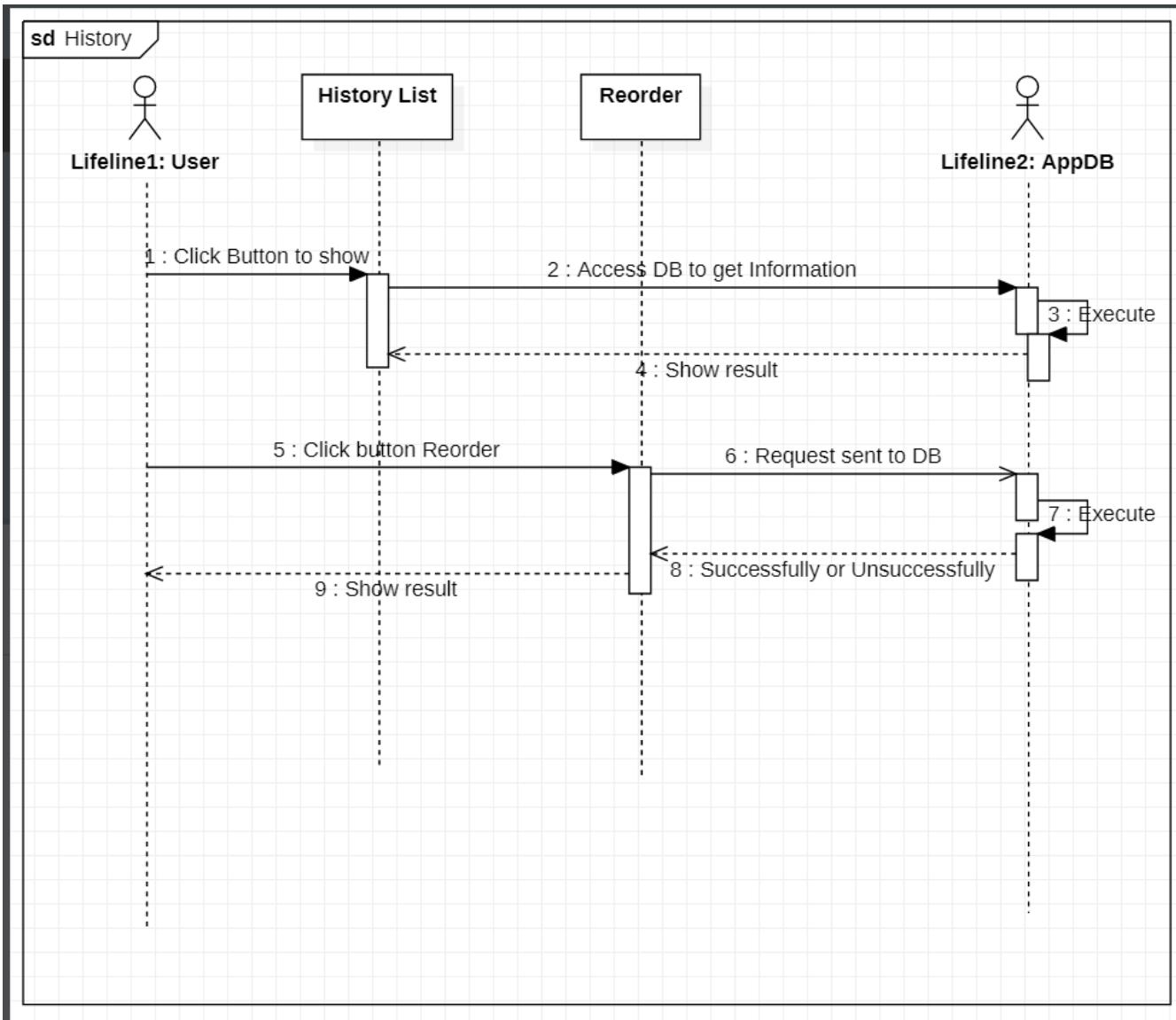
## Sequence diagram: Search shipping unit



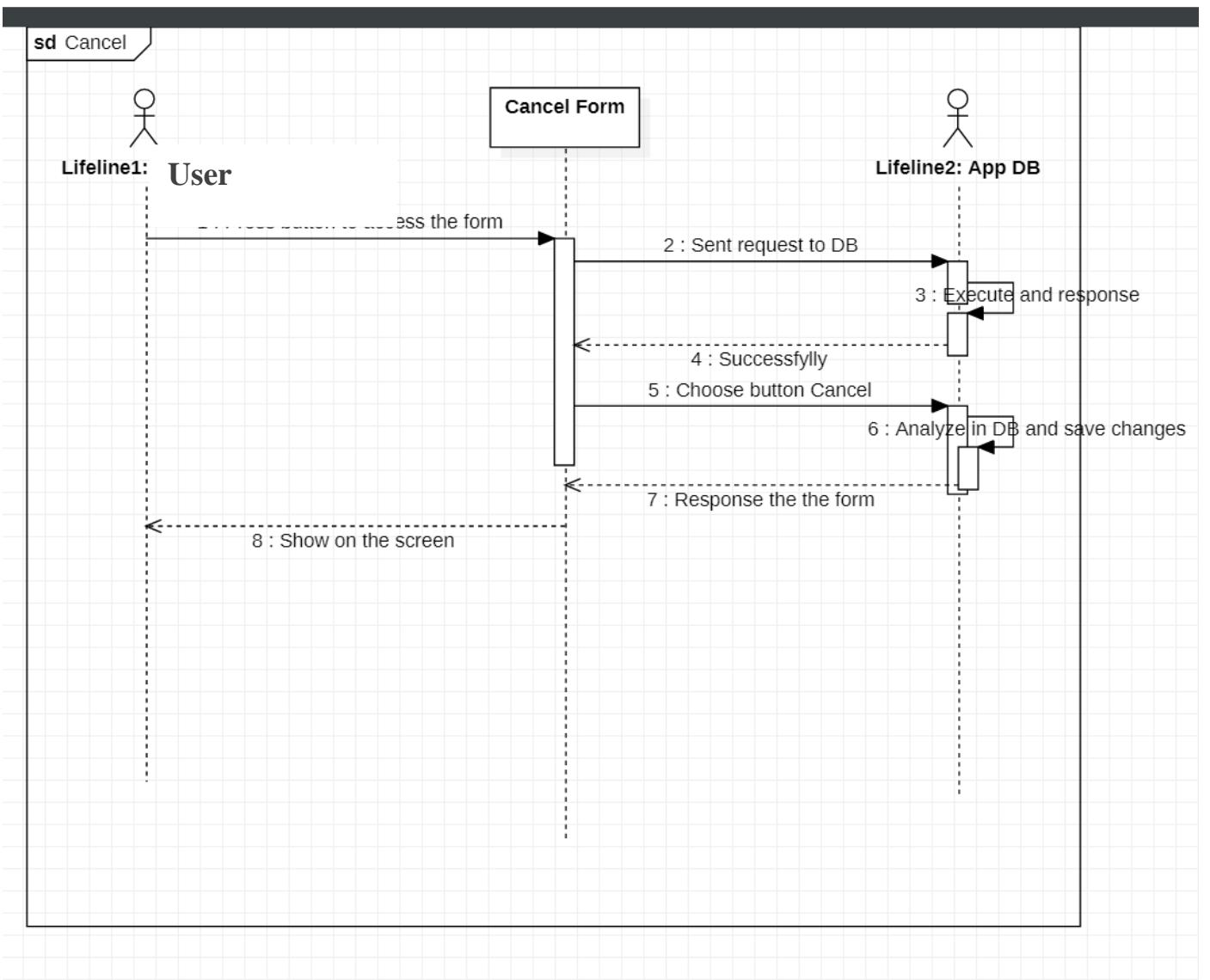
## Sequence diagram: Order



## Sequence diagram: Order's history

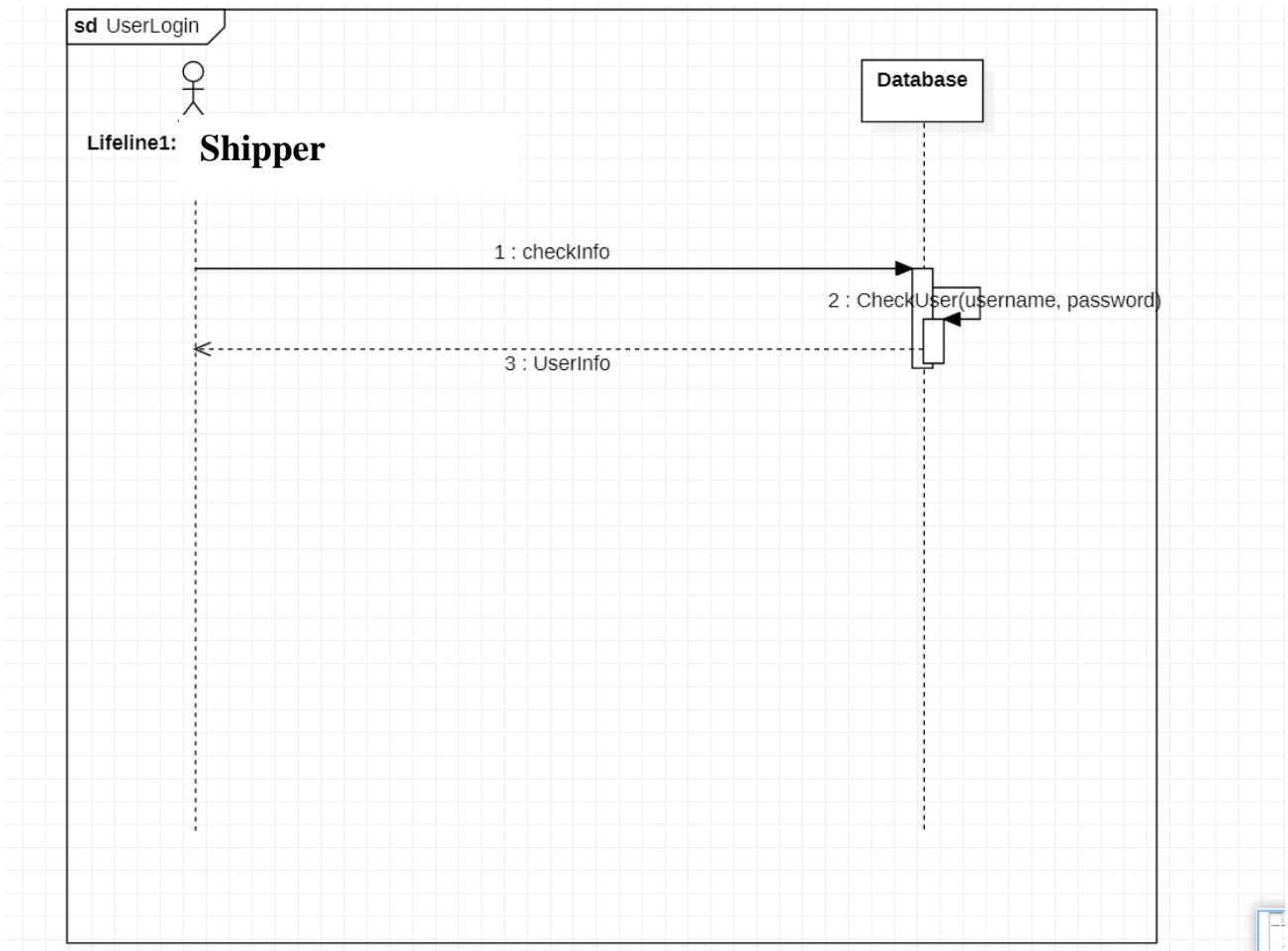


## Sequence diagram: Cancel booking

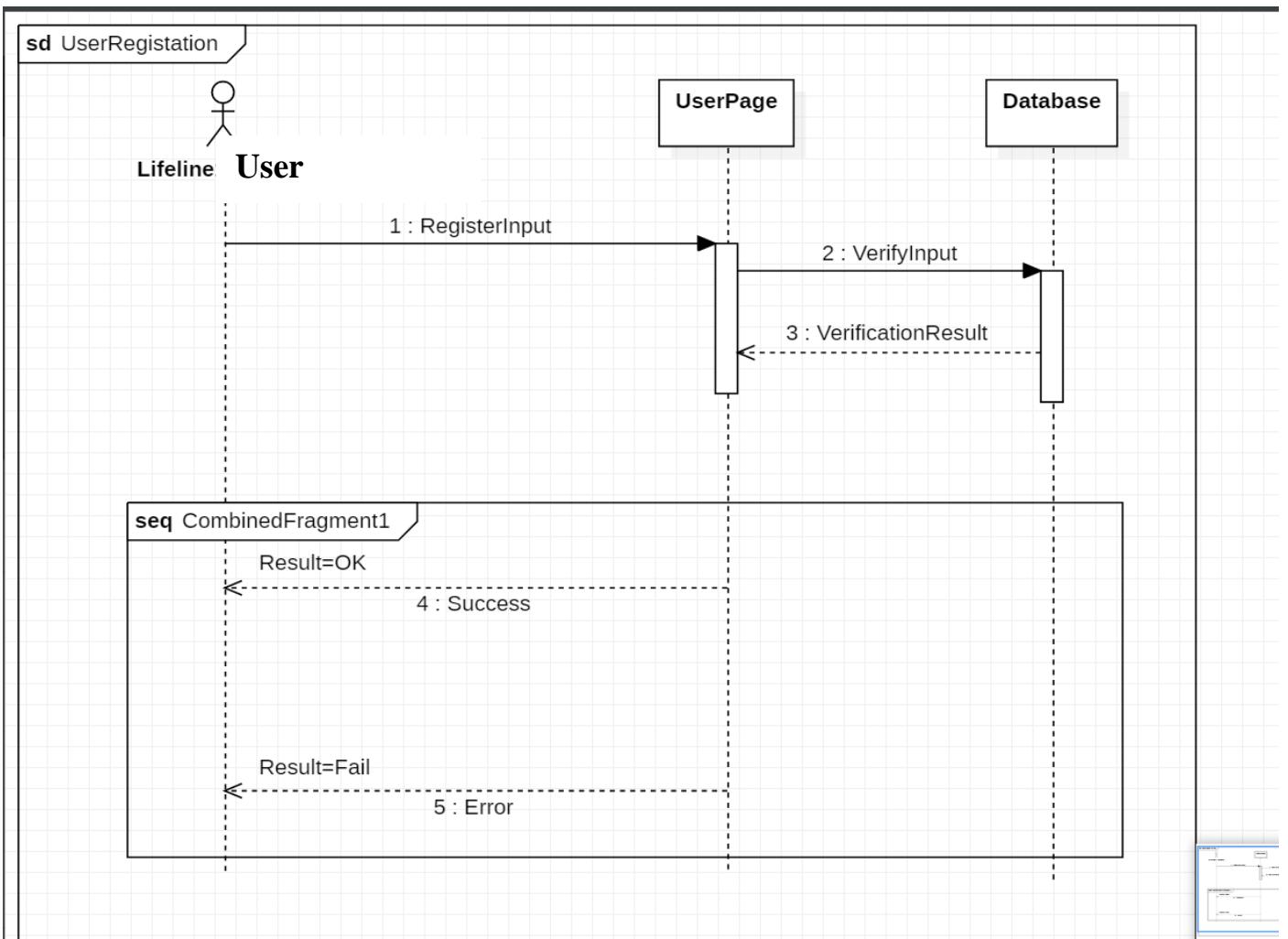


## SHIPPER SIDE

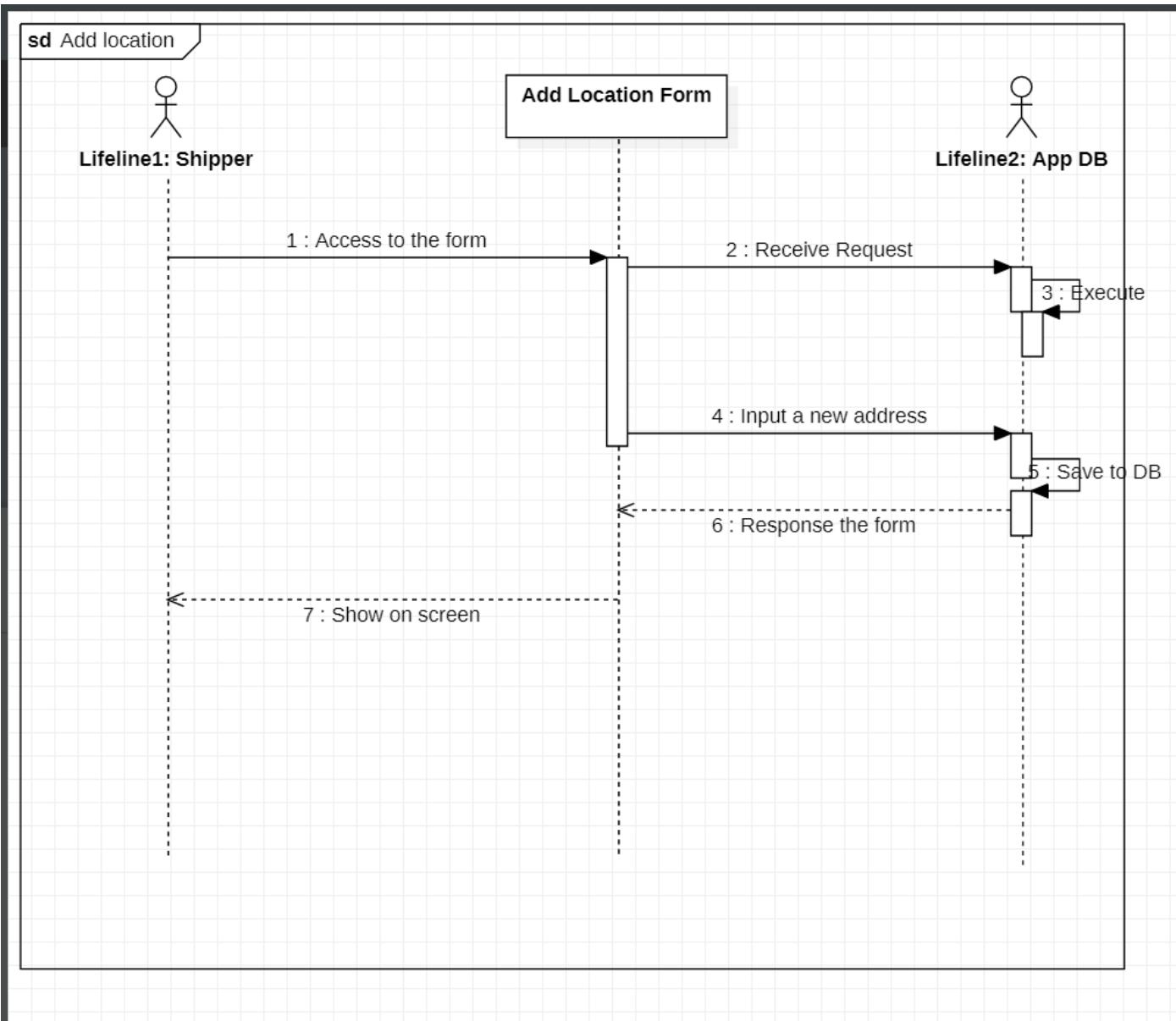
### Sequence diagram: Login



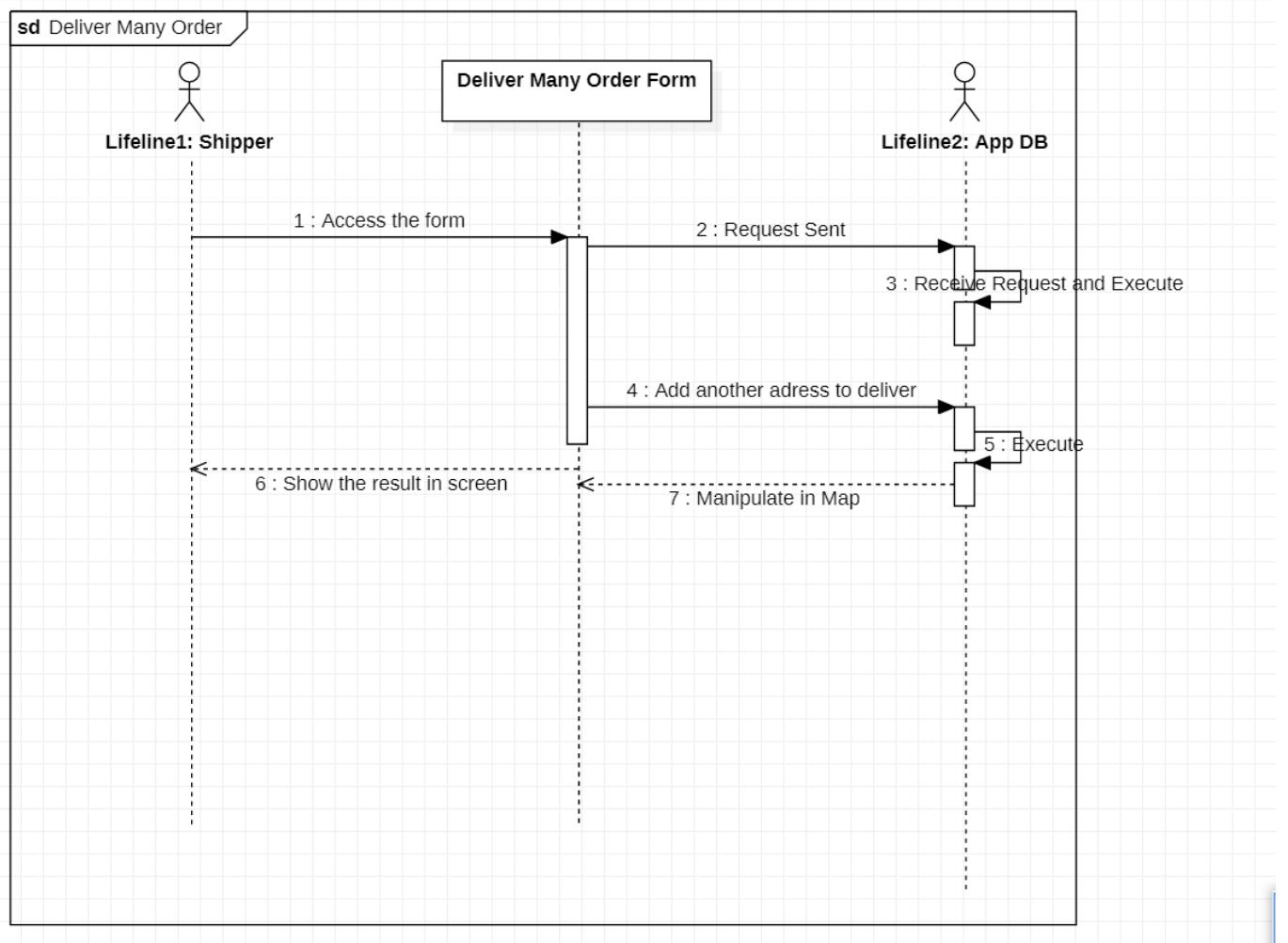
## Sequence diagram: Sign up



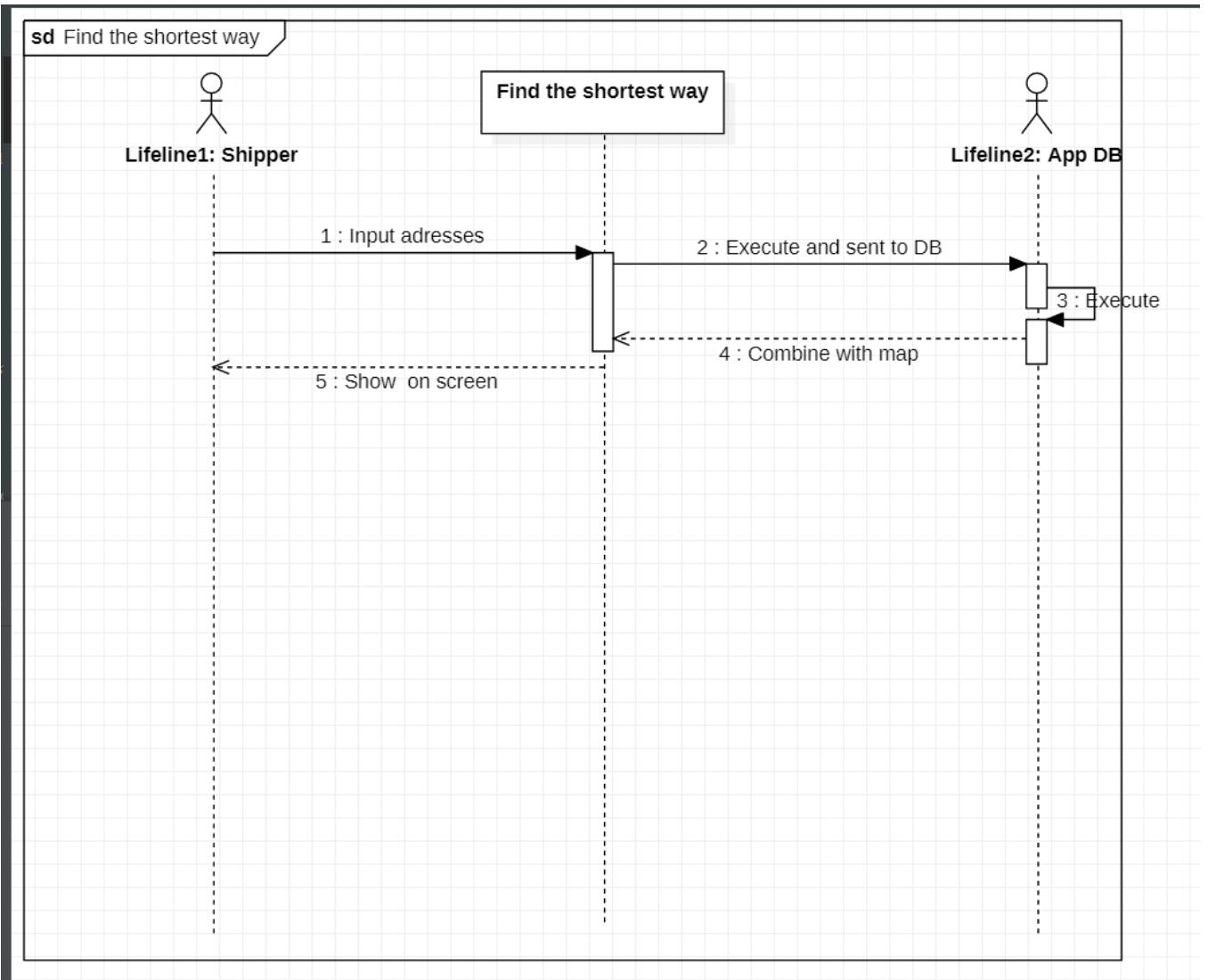
## Sequence diagram: Add location



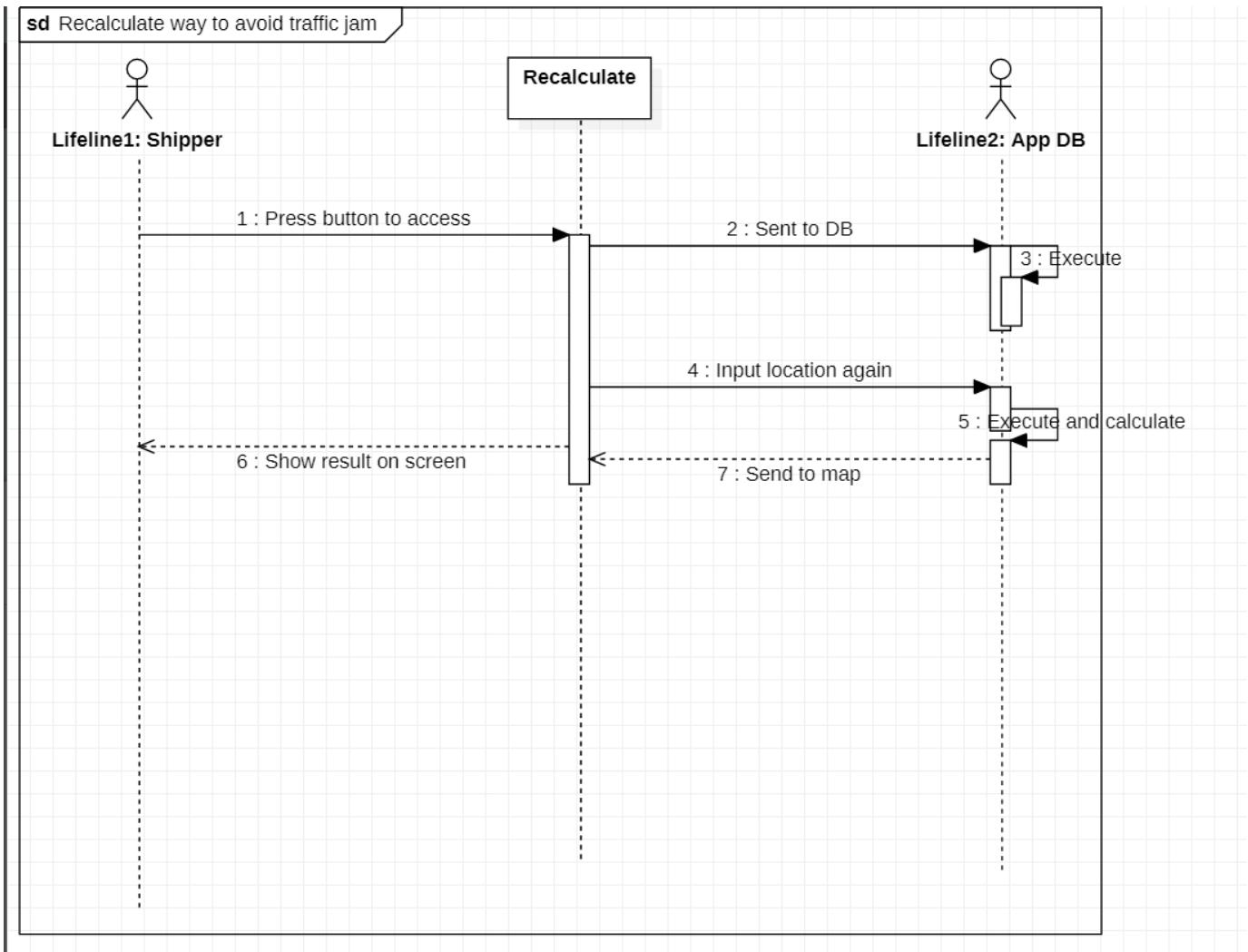
## Sequence diagram: Deliver many order



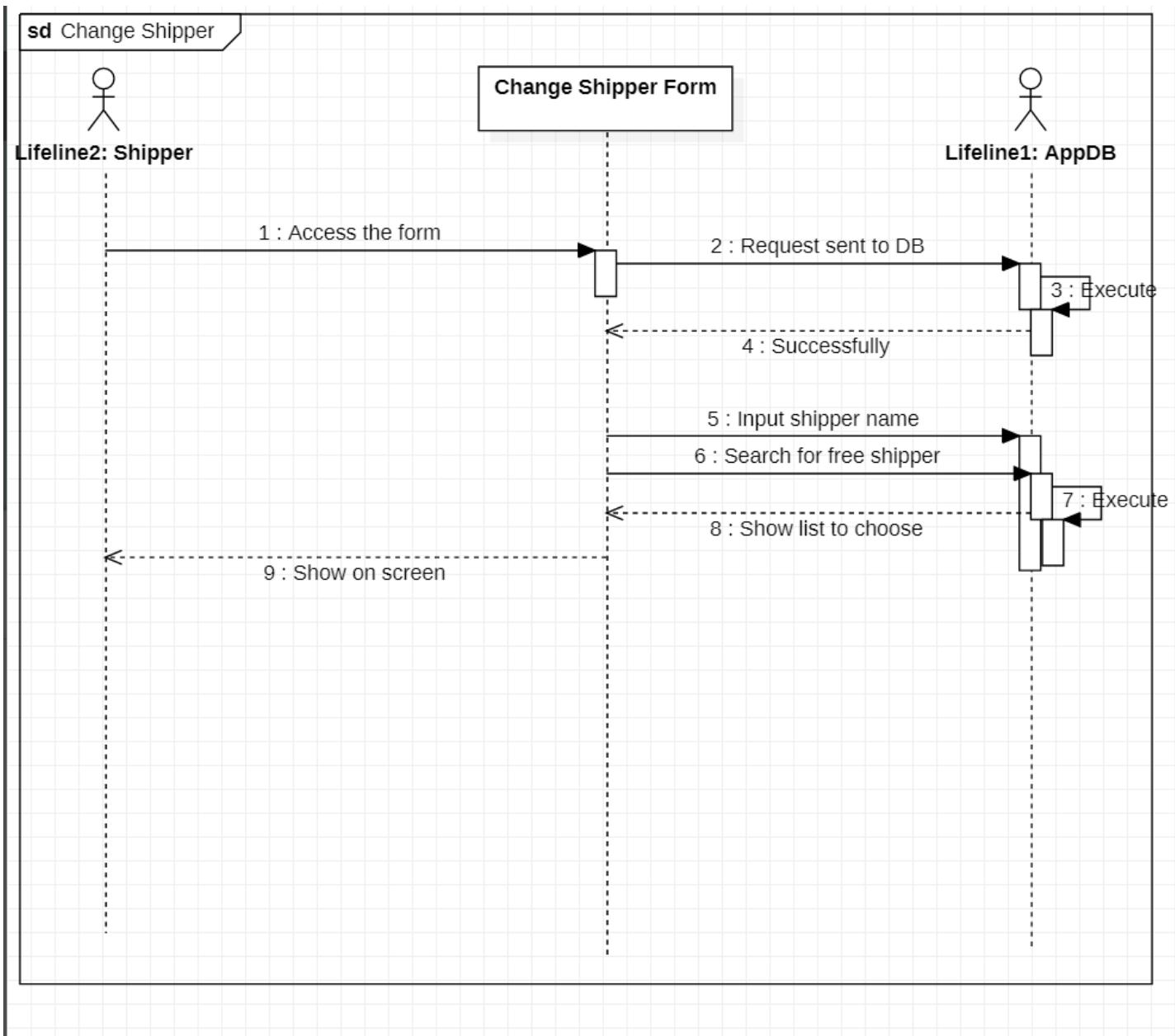
## Sequence diagram: Find the shortest way



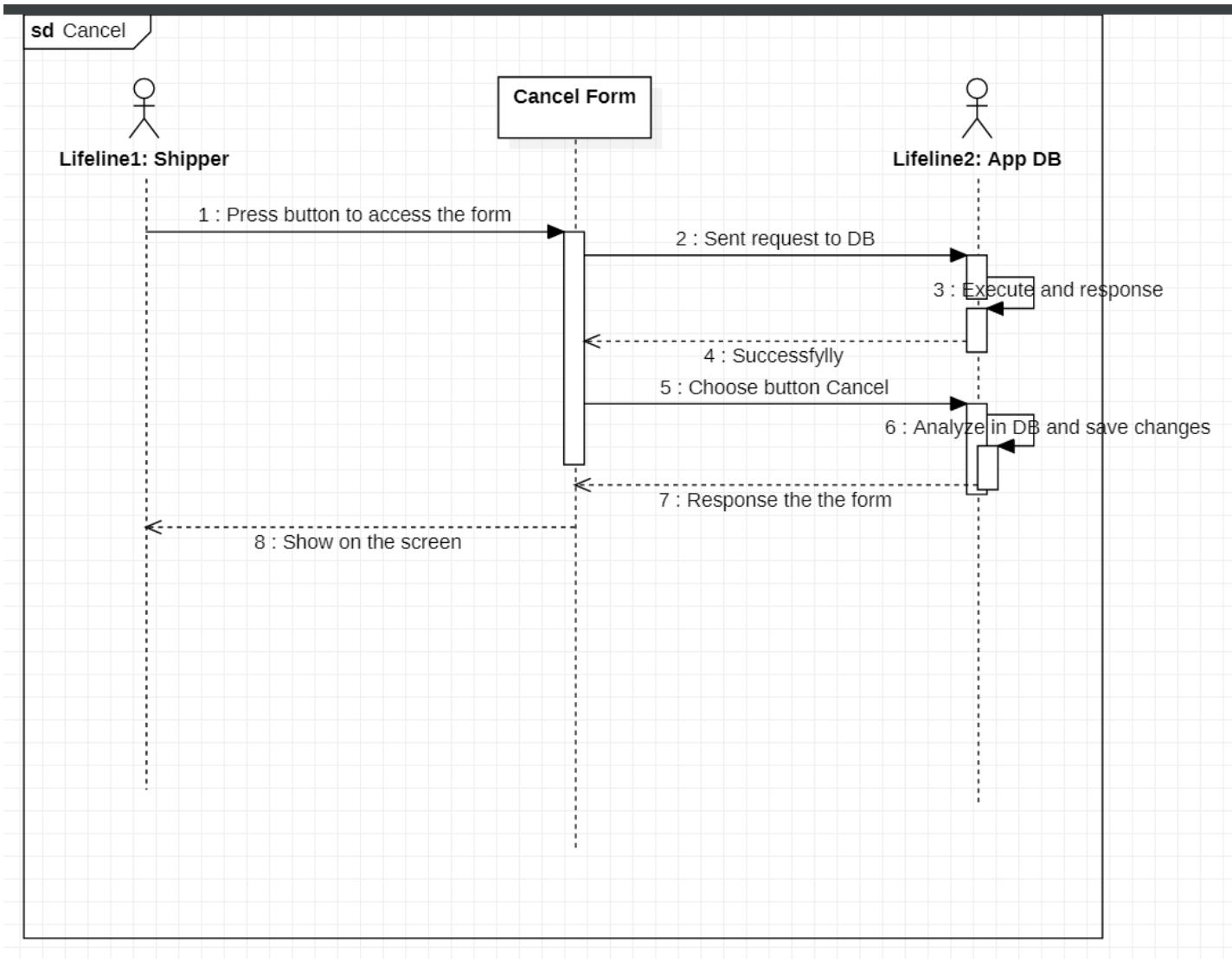
## Sequence diagram: Recalculate the way to avoid traffic jam



## Sequence diagram: Change shipper

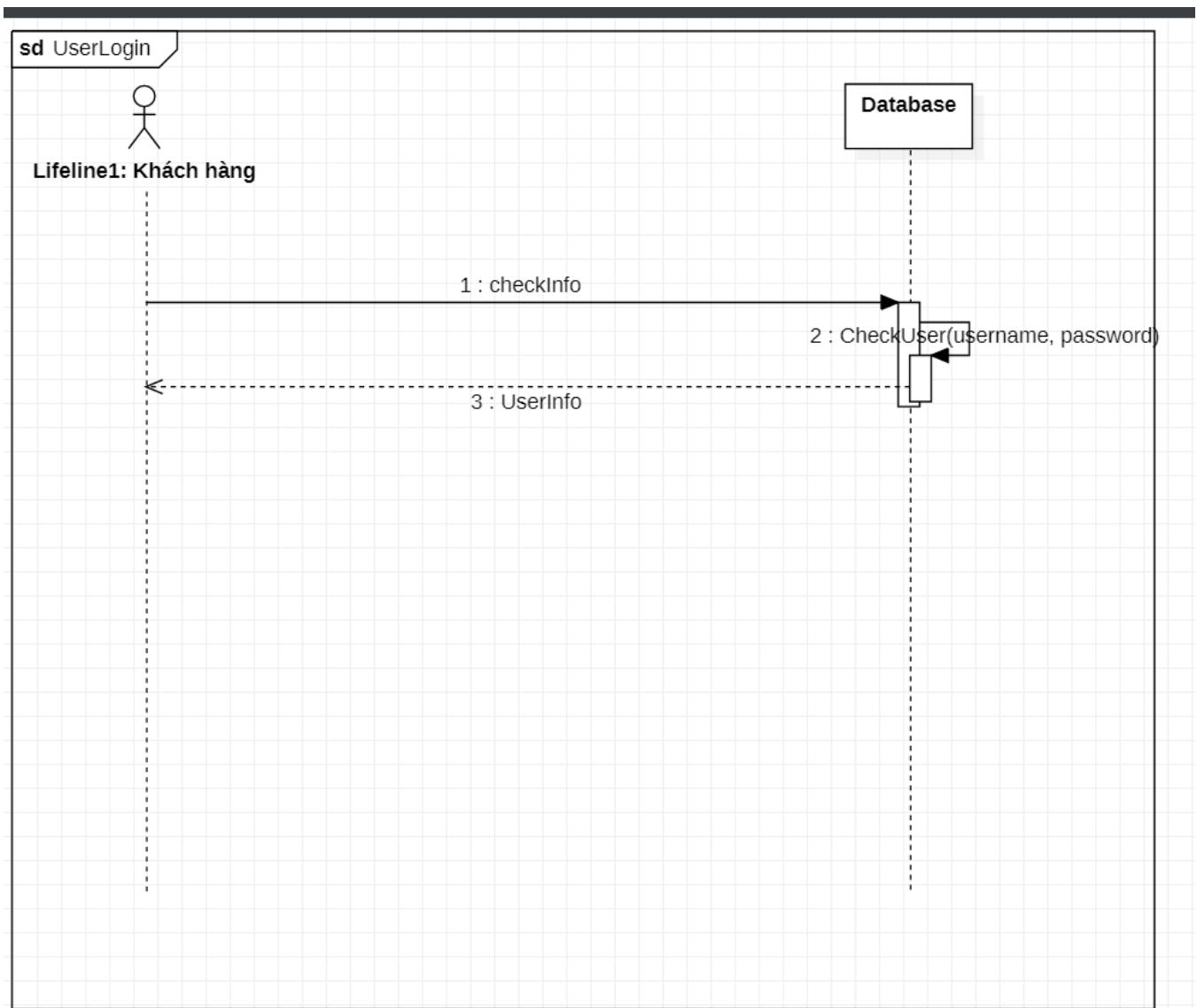


## Sequence diagram: Cancel delivery

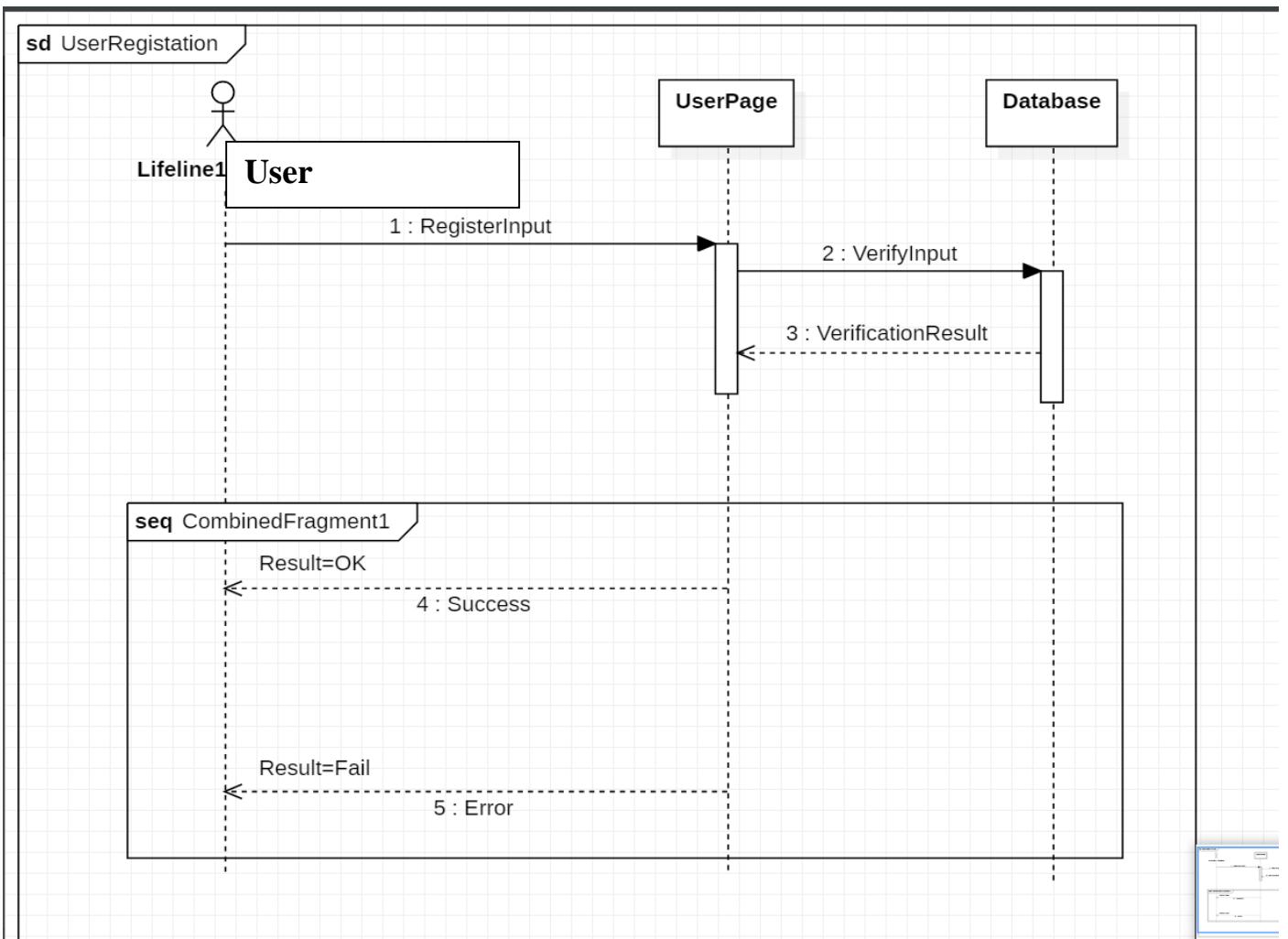


## ADMIN SIDE

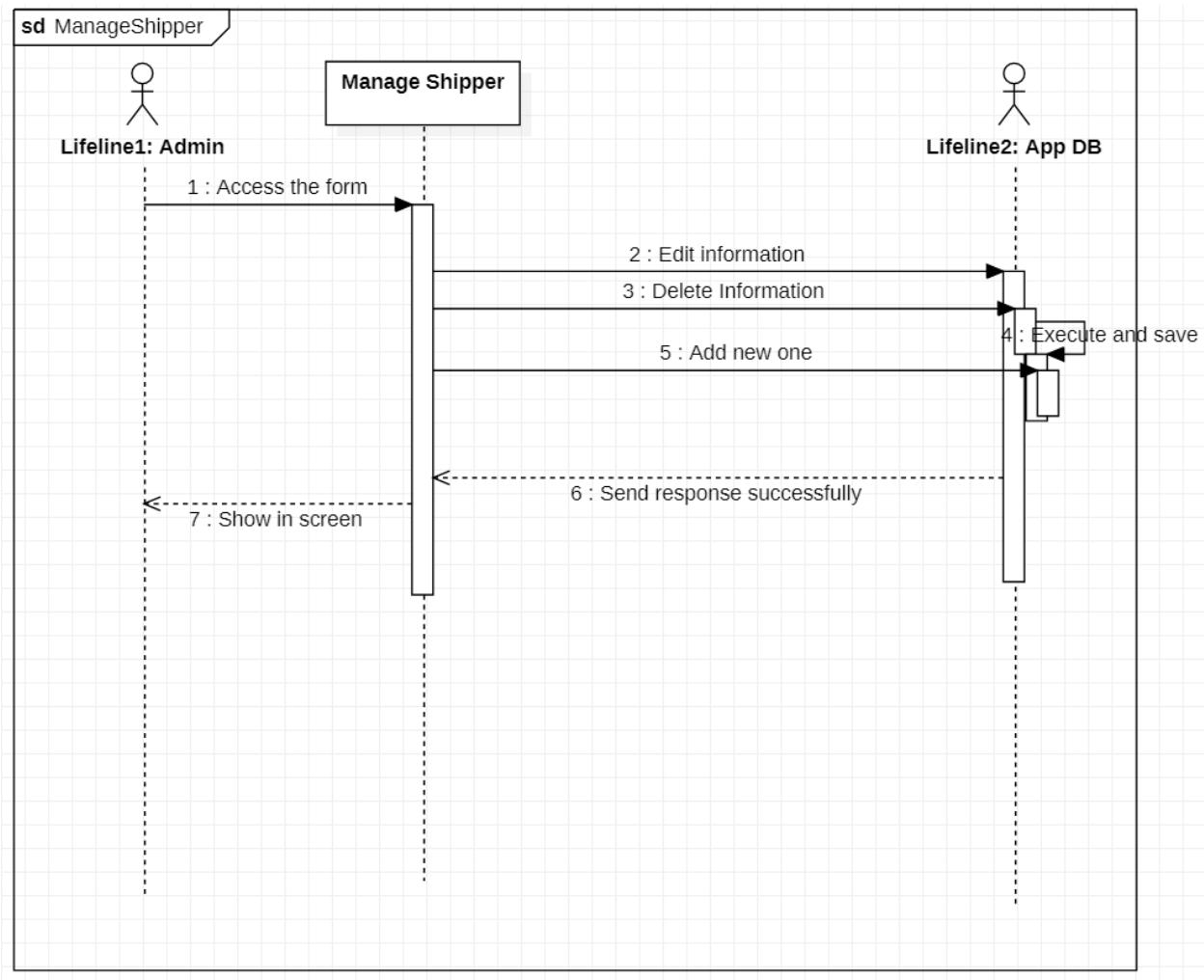
### Sequence diagram: Login



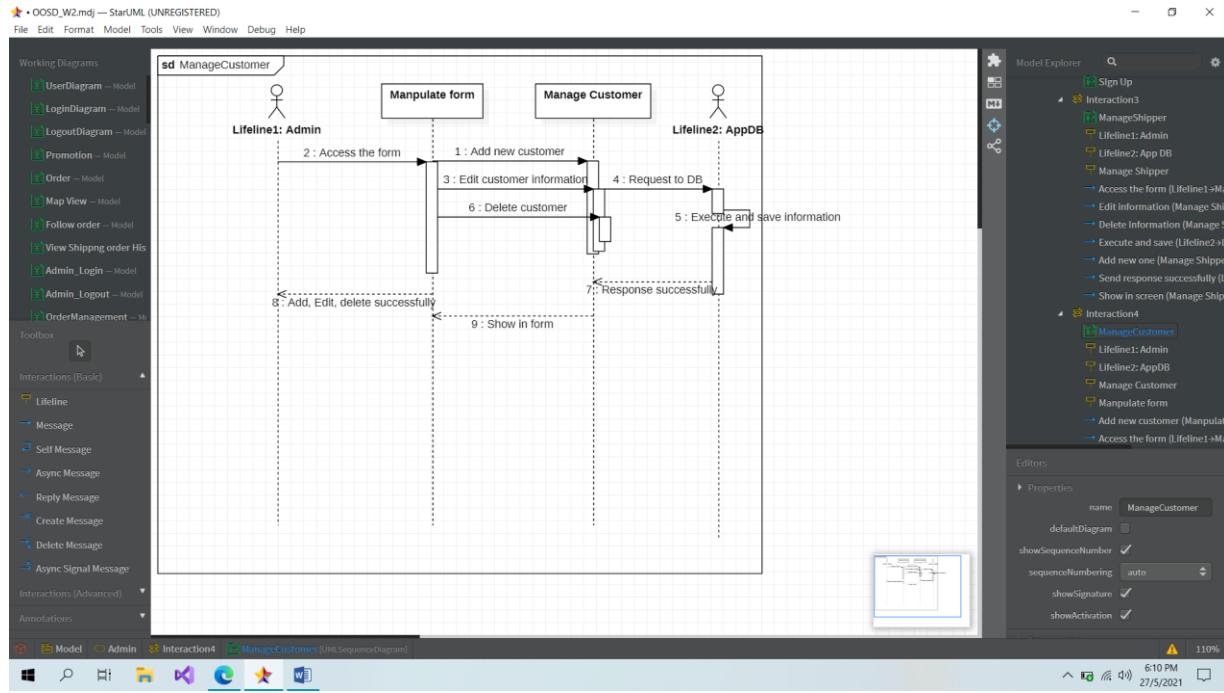
## Sequence diagram: Sign up



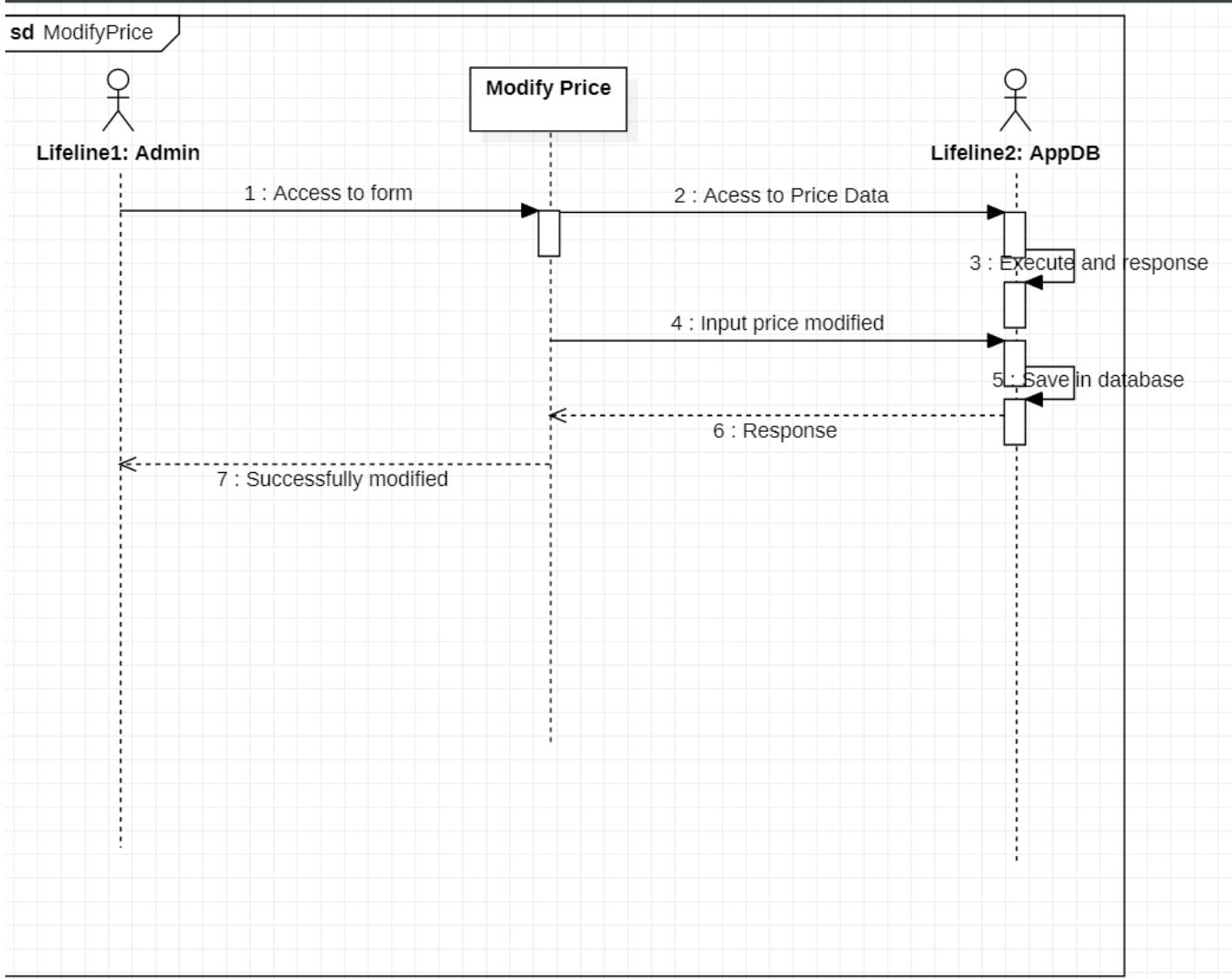
## Sequence diagram: Manage diagram



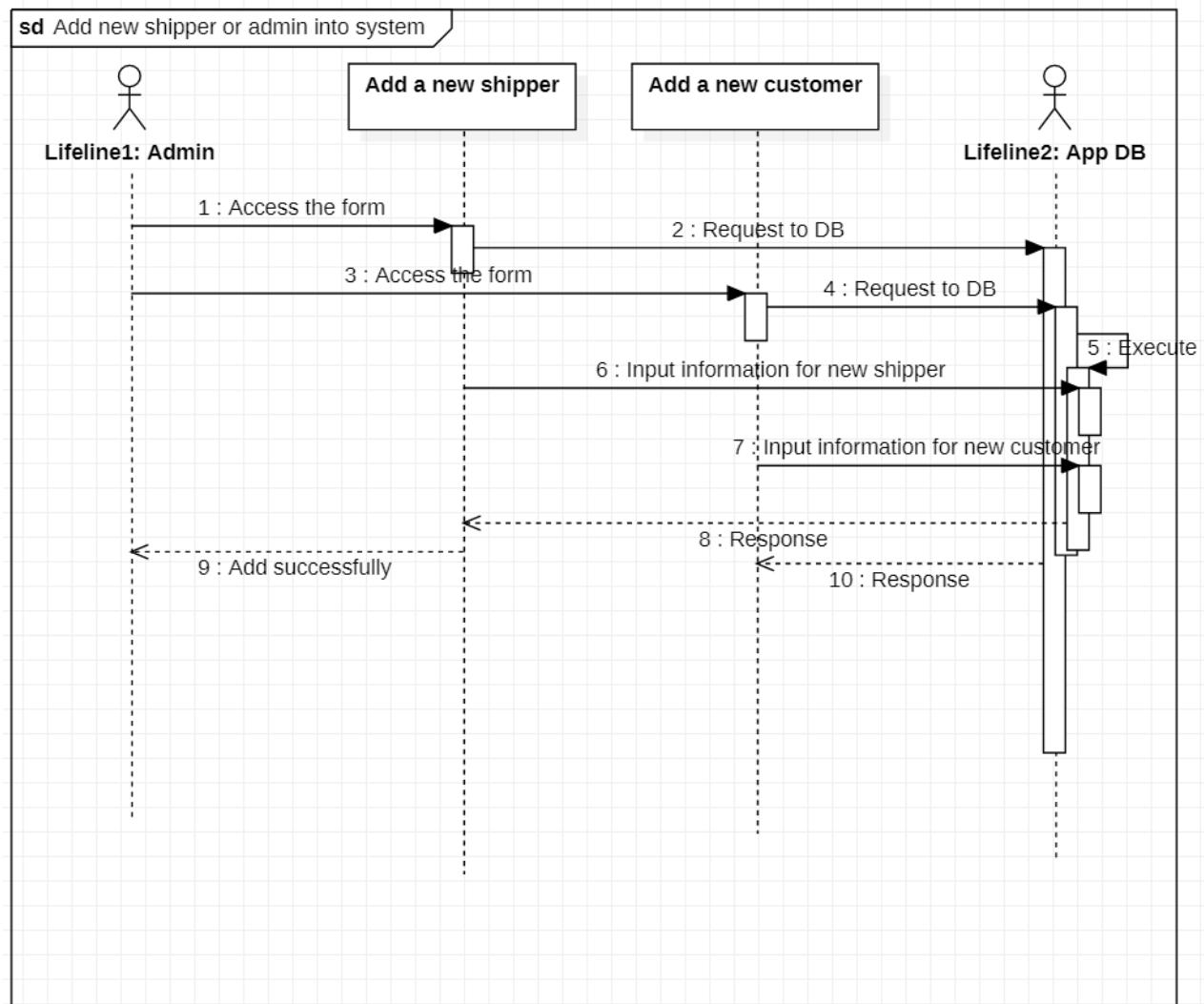
## Sequence diagram: Manage customer



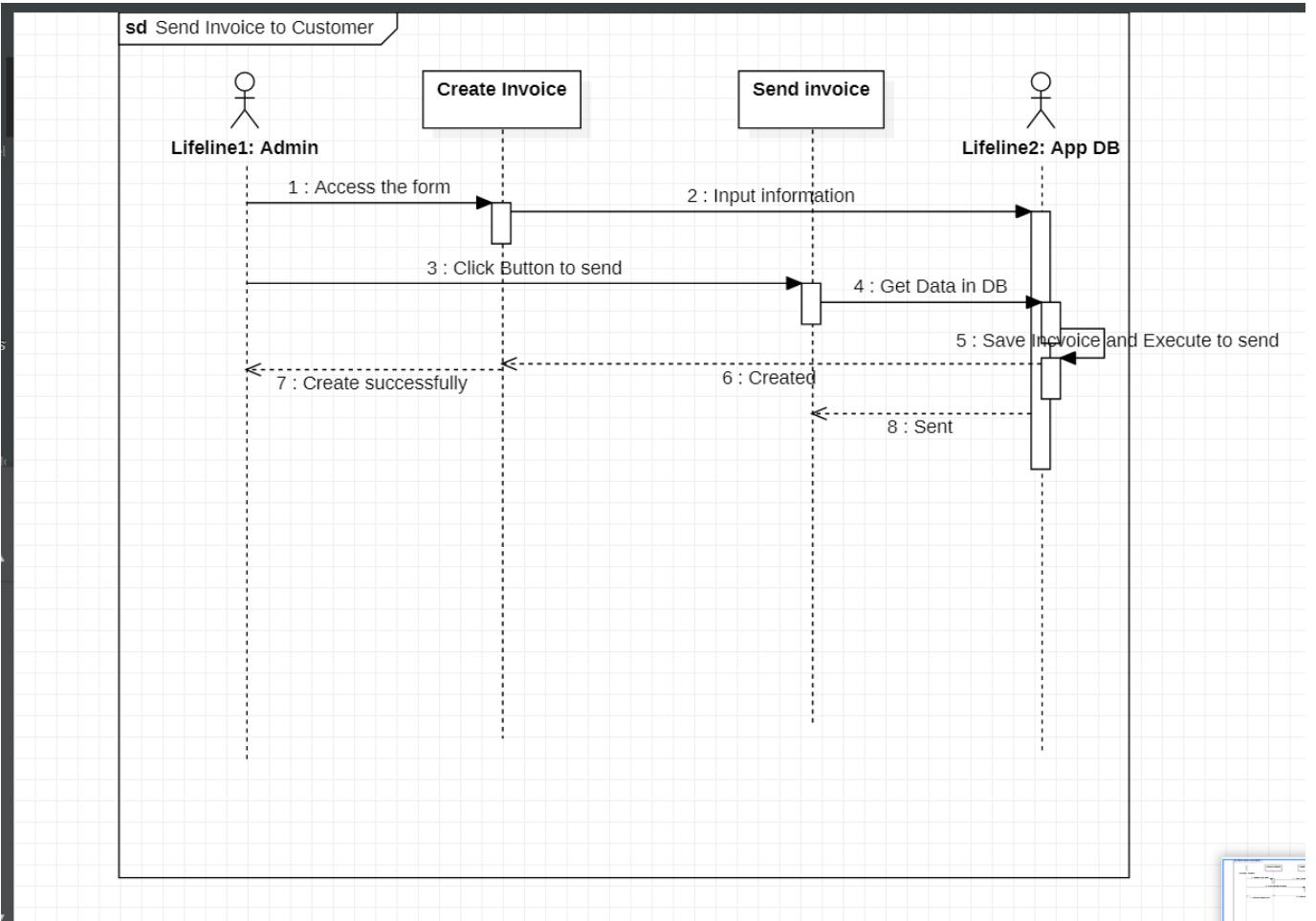
## Sequence diagram: Modify price



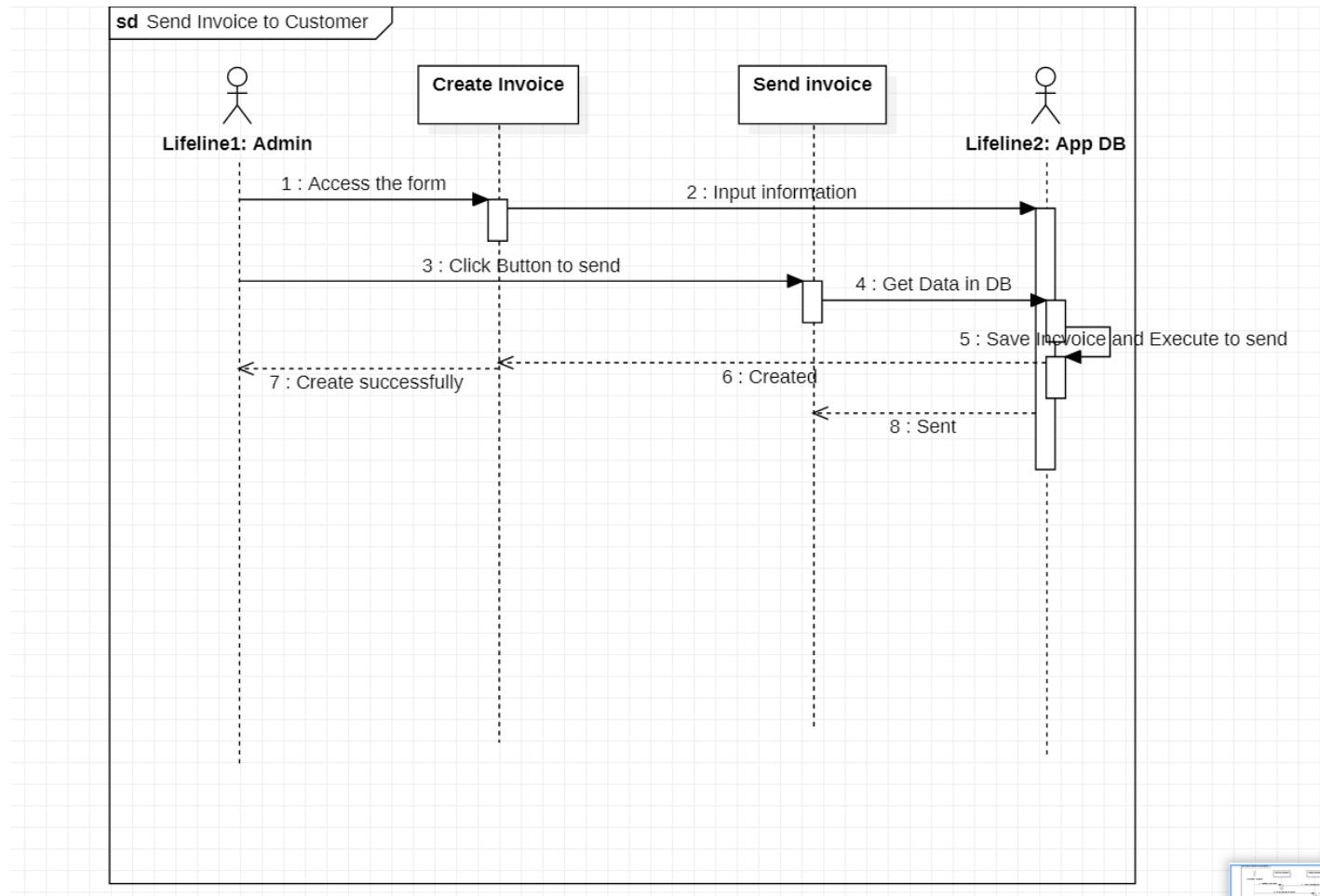
## Sequence diagram: Add new shipper or admin



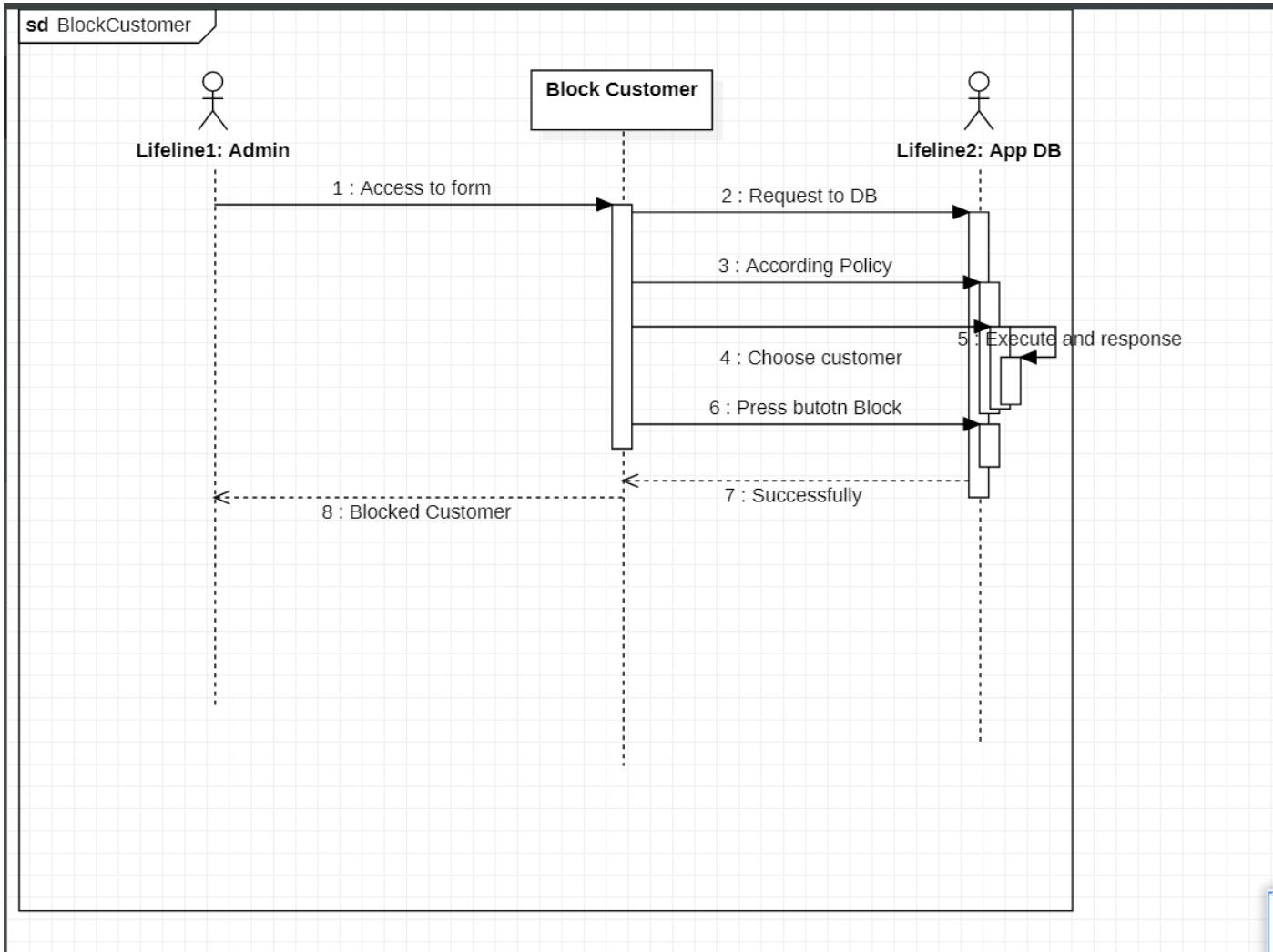
## Sequence diagram: Send invoice to customer



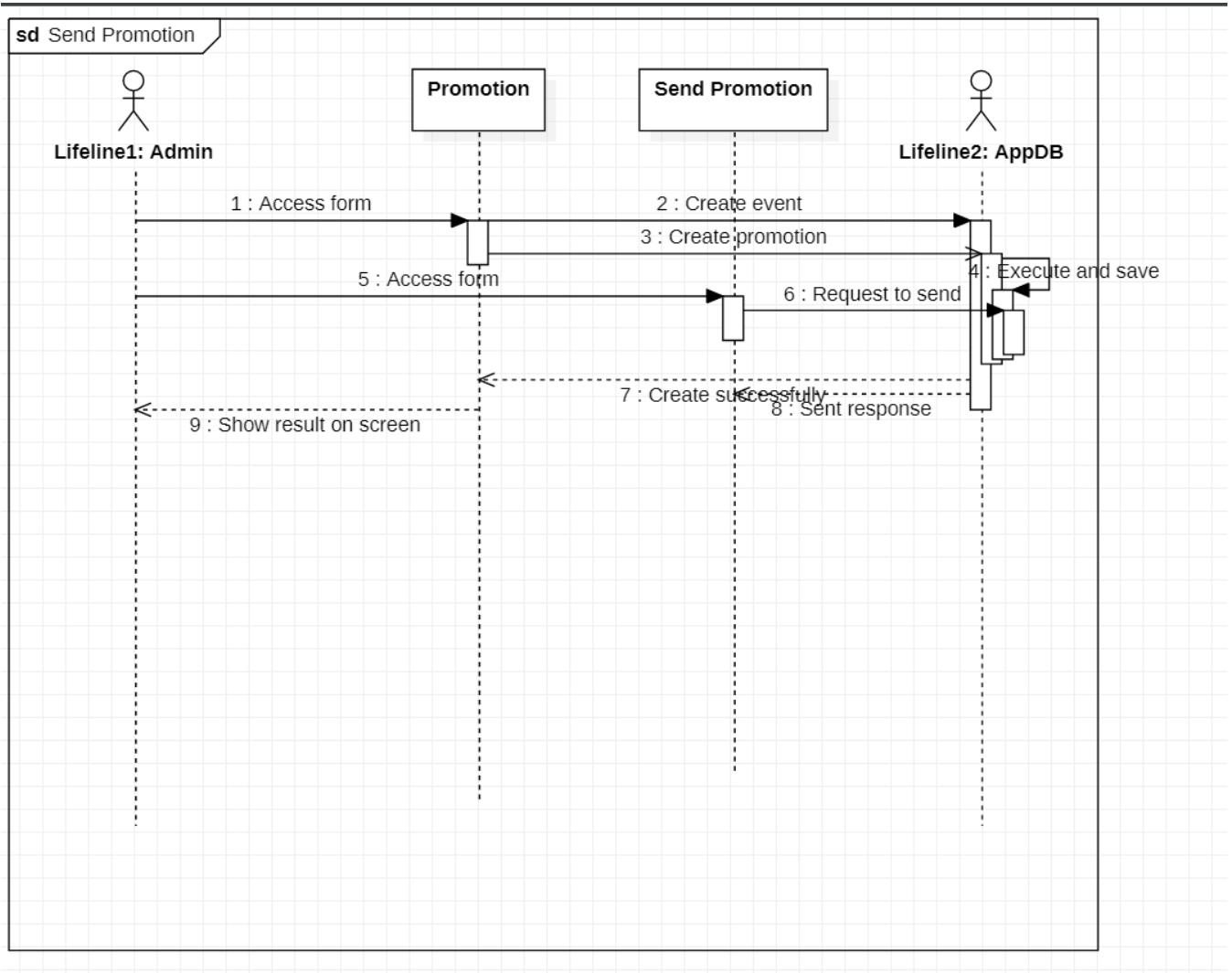
## Sequence diagram: Block shipper



## Sequence diagram: Block customer

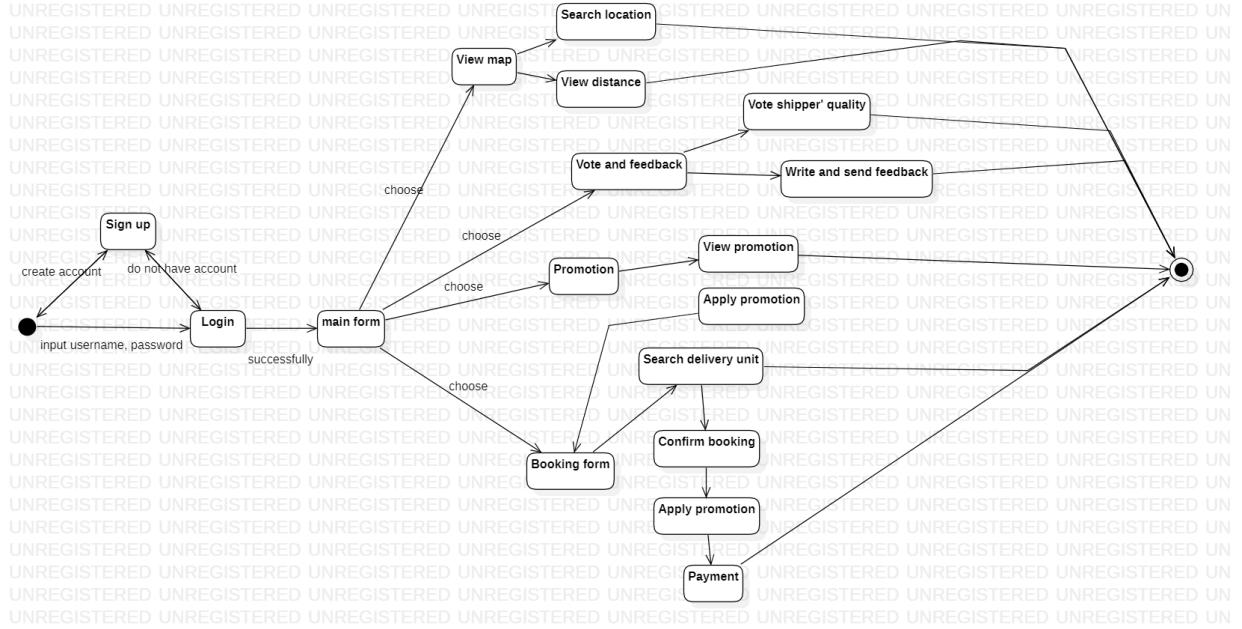


## Sequence diagram: Send promotion

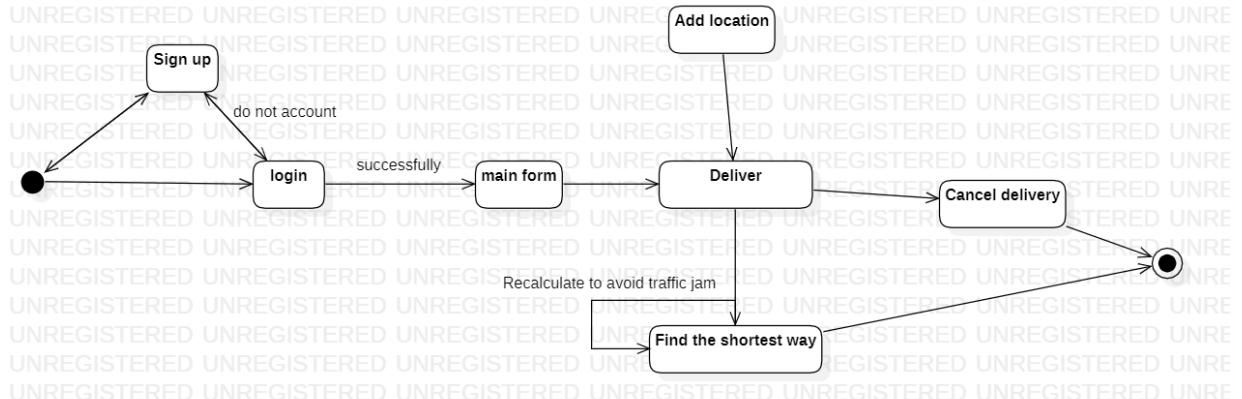


## C. STATE DIAGRAM

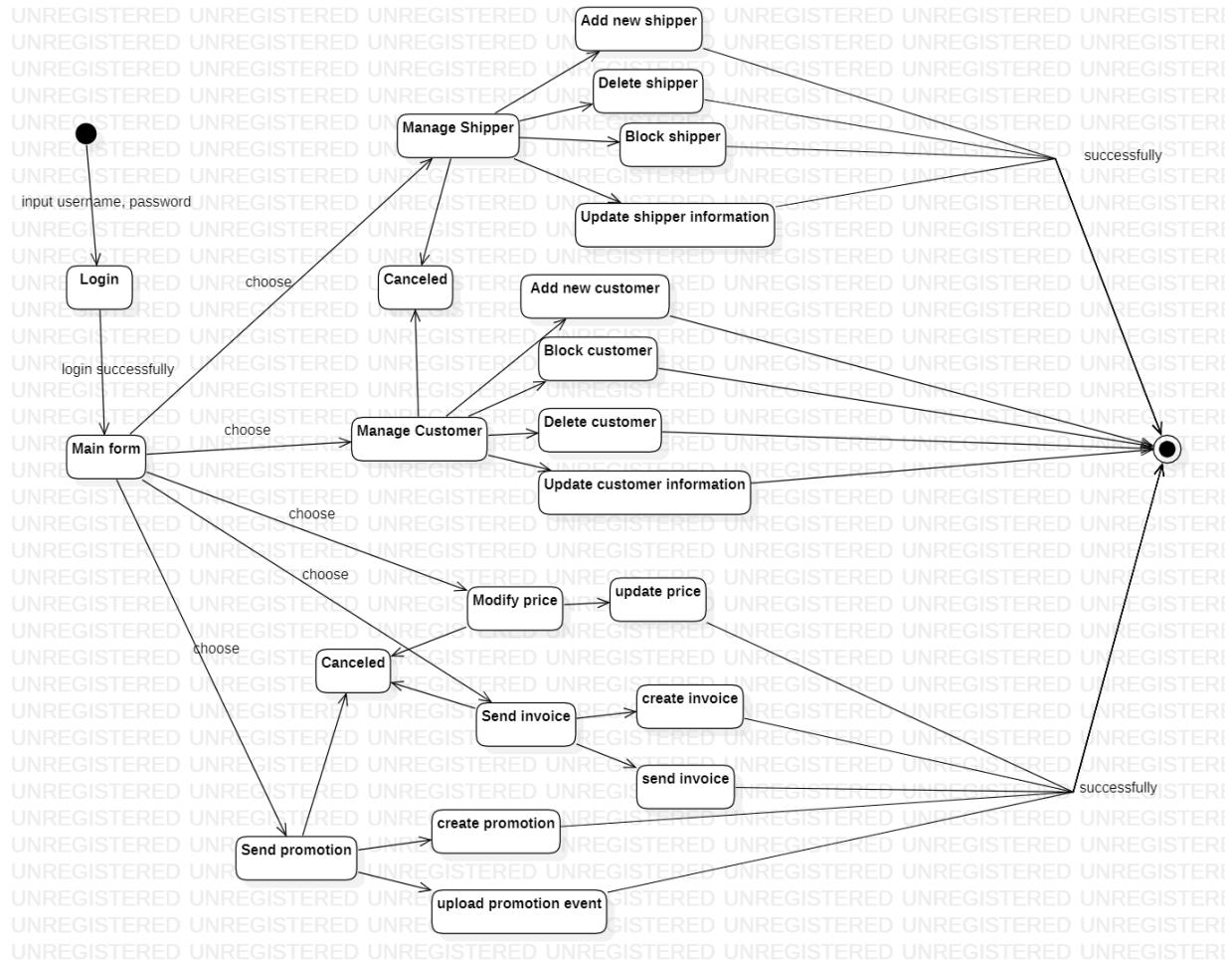
### State diagram: Customer side



### State diagram: Shipper side



## State diagram: Admin side



## CHAPTER 7: SYSTEM EVOLUTION

*Our application “FAST DELIVERY APPLICATION” designed for three user objects. We examined carefully check the operation of the app, handle a large amount of login information, goods information, orders, addresses, ... all are perfect and there is no error.*

**Grab:** The Grab app assigns taxis and private hire cars to nearby commuters through a location-sharing system. Each time the company enters a new market, they buy smartphones for drivers in countries that they expand to, allowing those drivers to pay daily installments for the phone.

**Gojek:** Gojek is Southeast Asia's leading on-demand, multi-service tech platform providing access to a wide range of services including transport, payments, food delivery, logistics, and many more.

**Beamin:** Baedal Minjok is the largest food delivery app in Korea. [1] The Korean delivery market is one of the largest in the world, valued at US\$8.6 billion in 2019. [3] In just one month of August 2020, the industry This business recorded revenue of US\$1.05 billion.

## CHAPTER 8: APPENDICES

### 8.1 Application Requirements

<b>Operating system</b>	Windows XP, Windows 7, iOS, Android
<b>Front end</b>	Microsoft Visual Basic 6.0
<b>Database</b>	MySQL 2015
<b>Connectivity</b>	ActiveX Data Object
<b>Report</b>	Data Report

### 8.2 Hardware Requirements

<b>RAM</b>	2GB or higher
<b>Hard Disk</b>	16GB or higher
<b>Computer Processor</b>	Pentium IV
<b>Clock Speed</b>	500 MHZ Processor

## **CHAPTER 9: INDEX**