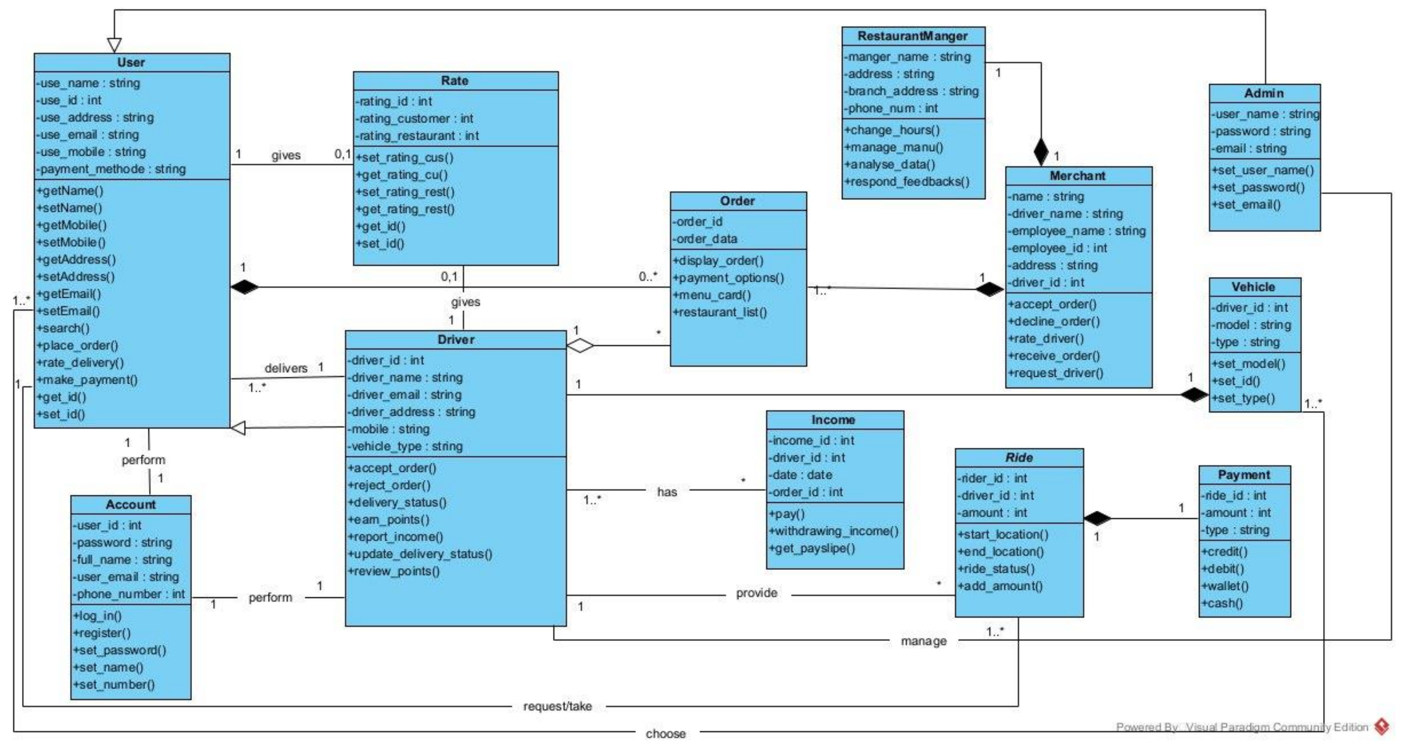
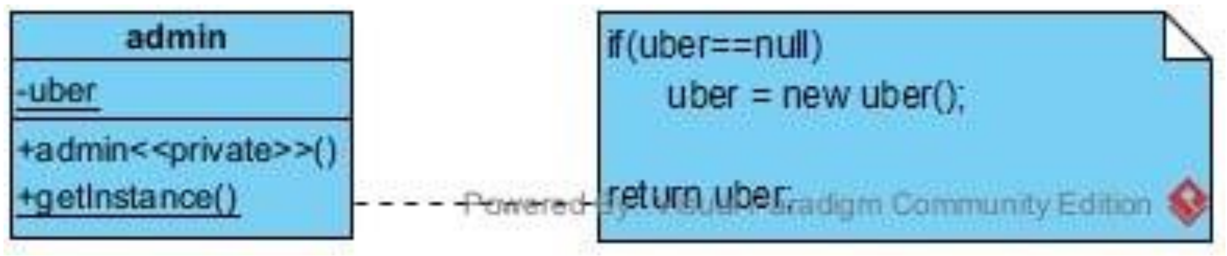
Class Diagram V2



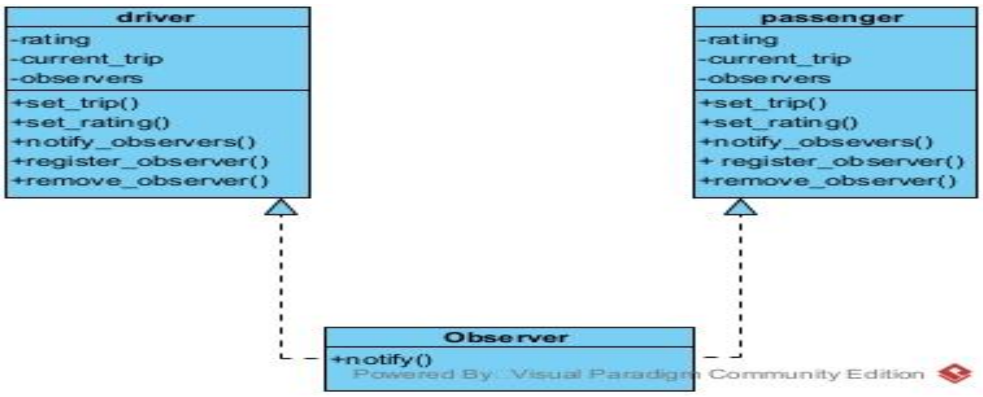
Class diagram applying the design patterns and other modifications

Name	Singleton pattern CREATIONAL DESIGN PATTERN
Context	**It is very common to find classes for which only one instance should exist (singleton)**
Problem	How do you ensure that it is never possible to create more than one instance of a singleton class. And provide a global point of access to it
Forces	**-The use of a public constructor cannot guarantee that no more than one instance will be created.** **-the singleton instance must also be accessible to all classes that require it, therefore it must often be public.**
Solution	-Have the constructor private to ensure that no other class will be able to create an instance of the class singleton. -Define a public static method, The first time this method is called , it creates the single instance of the class "singleton" and stores a reference to that object in a static private variable.

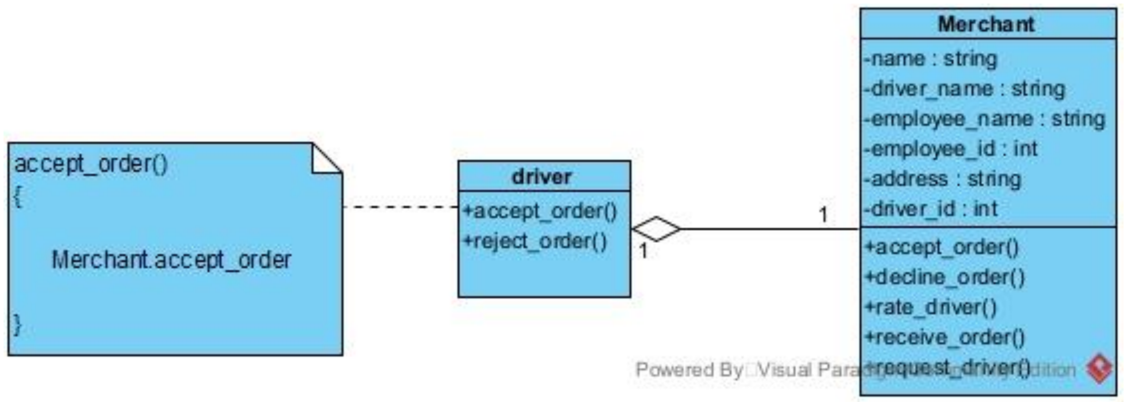
[



Name	**Observer (Publish-Subscribe) pattern** *(Behavioral Design Pattern):*
Context	When partitioning a system into individual classes you want the coupling Between then to be loose so you have the flexibility to vary them Independently. (Update class)
Problem	A mechanism is needed to ensure that when the state of an object changes related objects are updated to keep them in step.
Forces	The different parts of a system have to kept in step with one another without being too tightly coupled.
Solution	One object has the role of the subject/publisher and one or more other object the role of observers/subscribes. The observers register themselves with the subject, & if the state of the subject changes the observers are notified & can the update themselves:

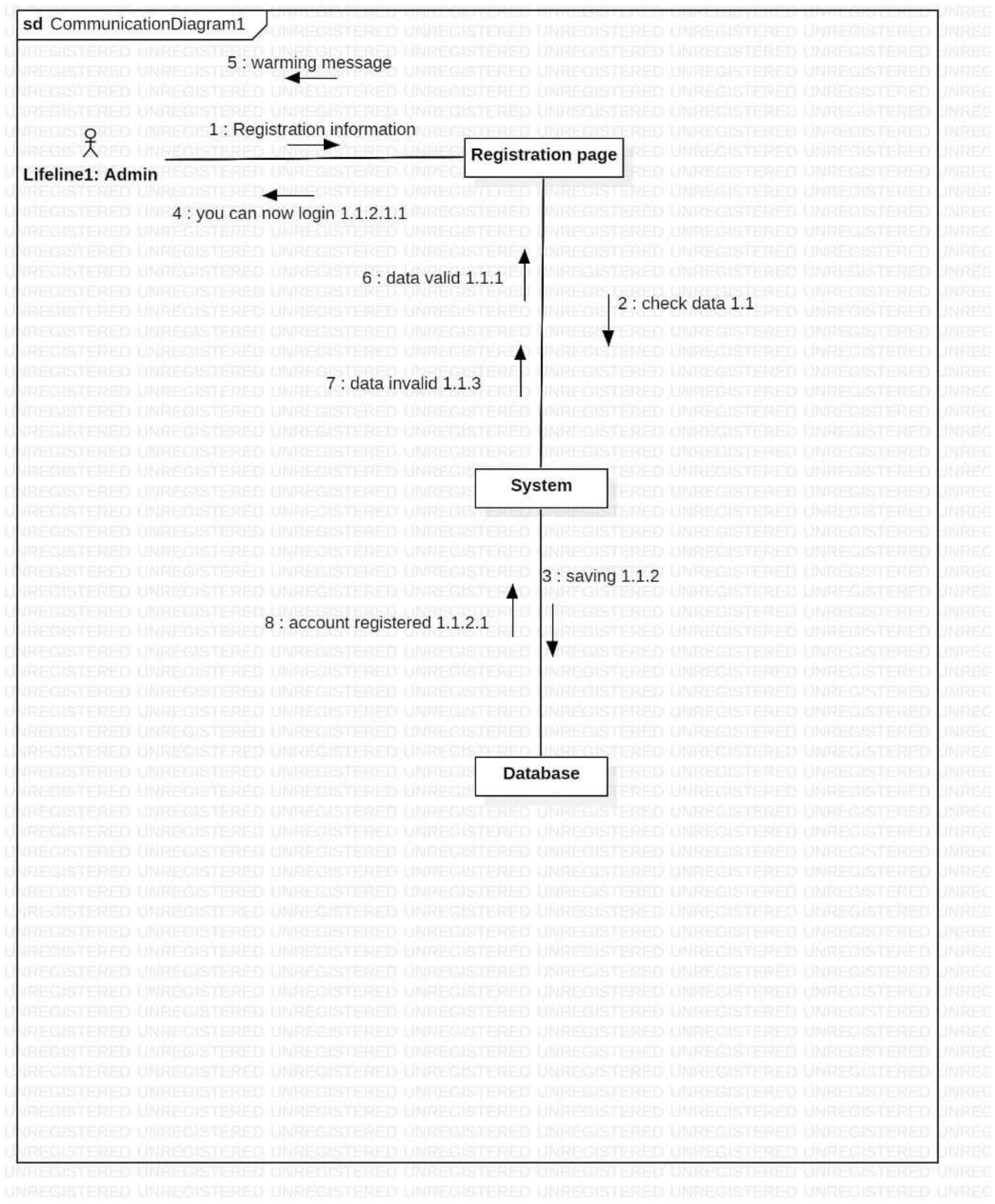


Name	**DELEGATION pattern(*Structural Design Pattern)***
Context	**-You are designing a method in a class. You realize that another class has a method which provides the required service.** **-Inheritance is not appropriate**
Problem	**•How can you most effectively make use of a method that already exists in the other class?**
Forces	**• You want to minimize development cost by reusing methods.**
Solution	**• The delegating method in the delegator class calls a method in the delegate class to perform the required task. An association must exist between the delegator and delegate classes.**

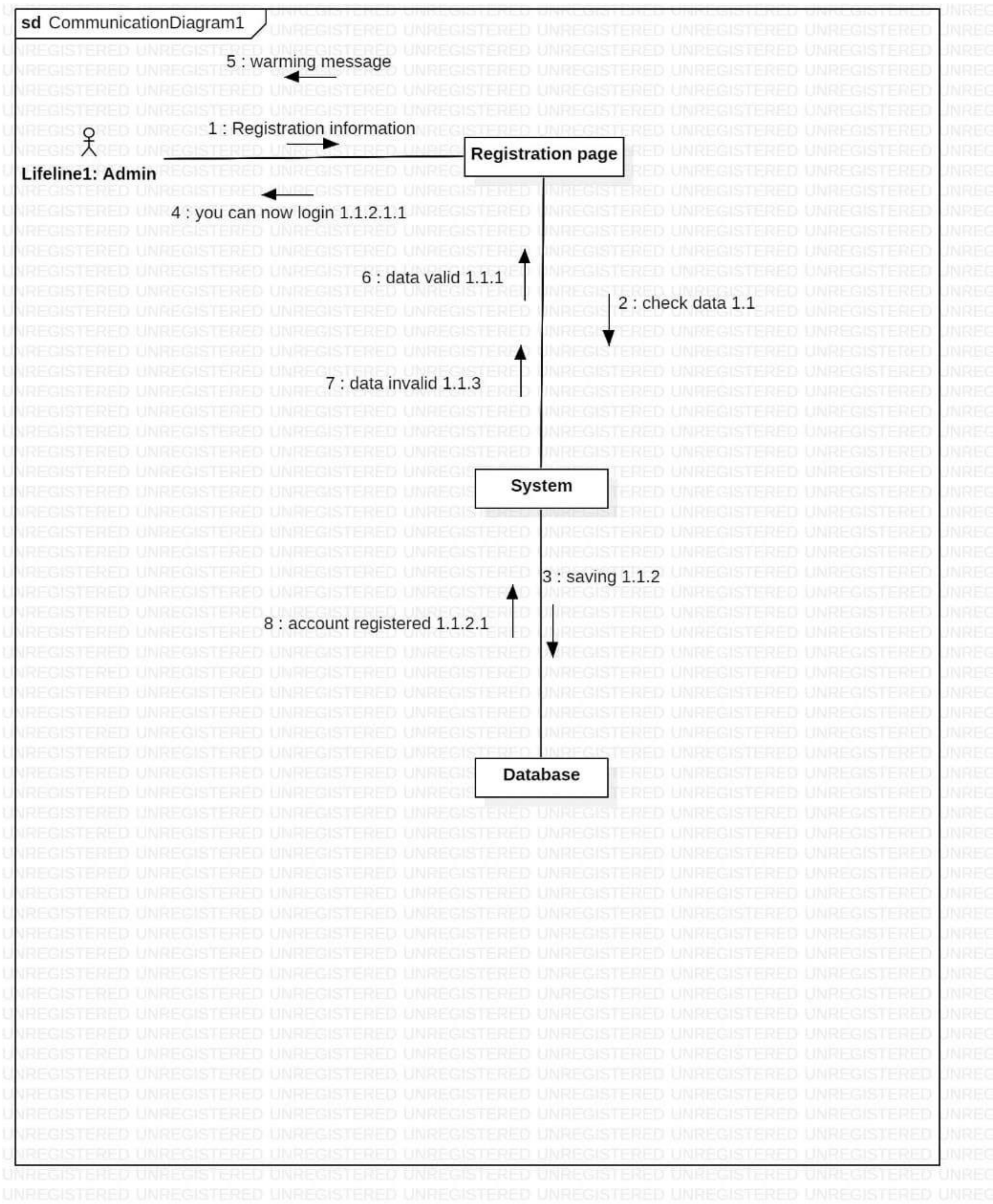


b)Callboration/Communication Daigram

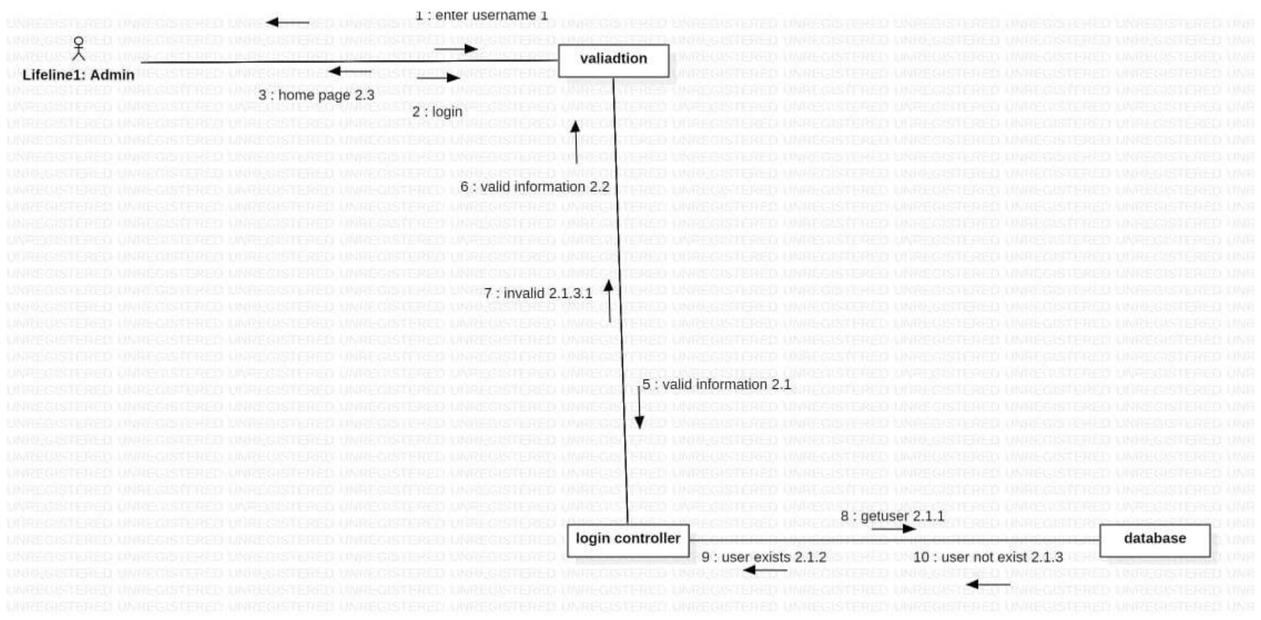
- Registration
  - Admin



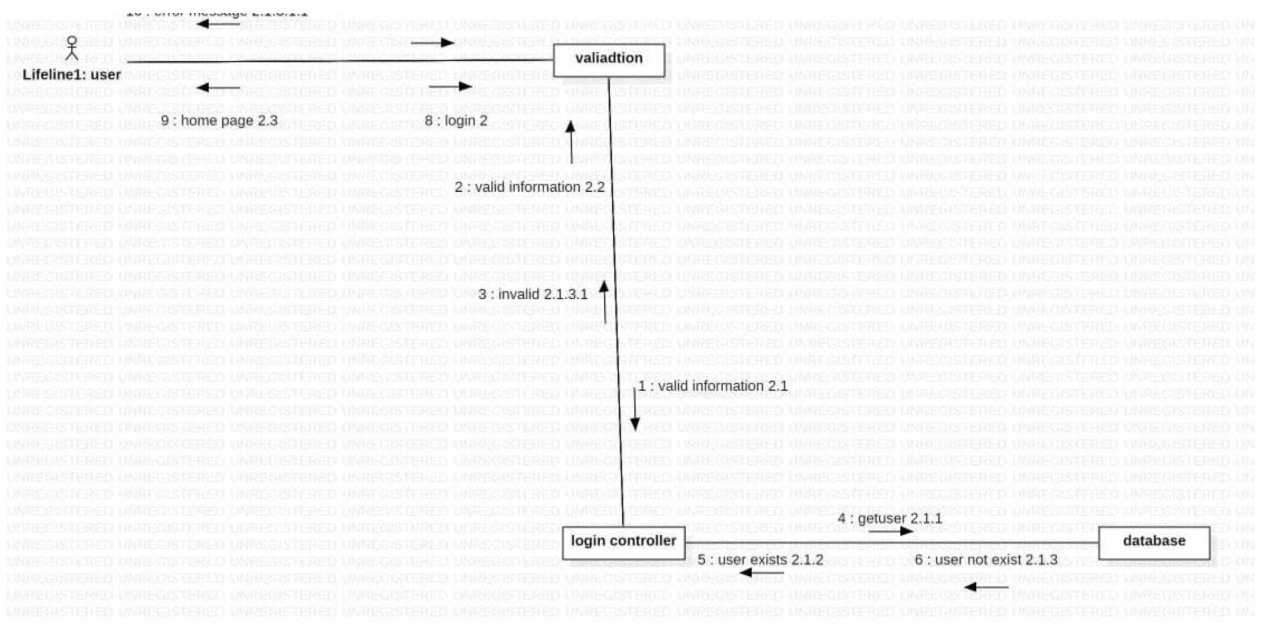
- User


sd CommunicationDiagram1
5 : warming message
1 : Registration information
Lifeline1: Admin
Registration page
4 : you can now login 1.1.2.1.1
6 : data valid 1.1.1
2 : check data 1.1
7 : data invalid 1.1.3
System
3 : saving 1.1.2
8 : account registered 1.1.2.1
Database

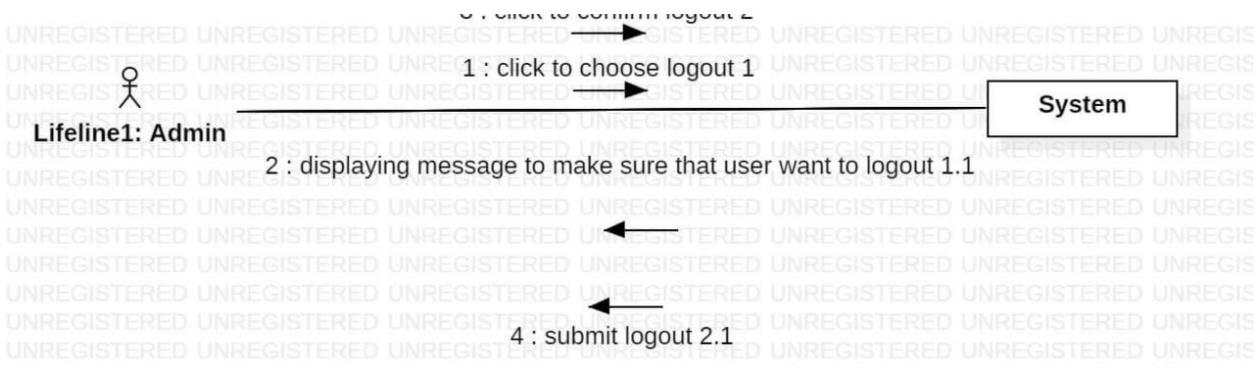
- Log in
  - Admin



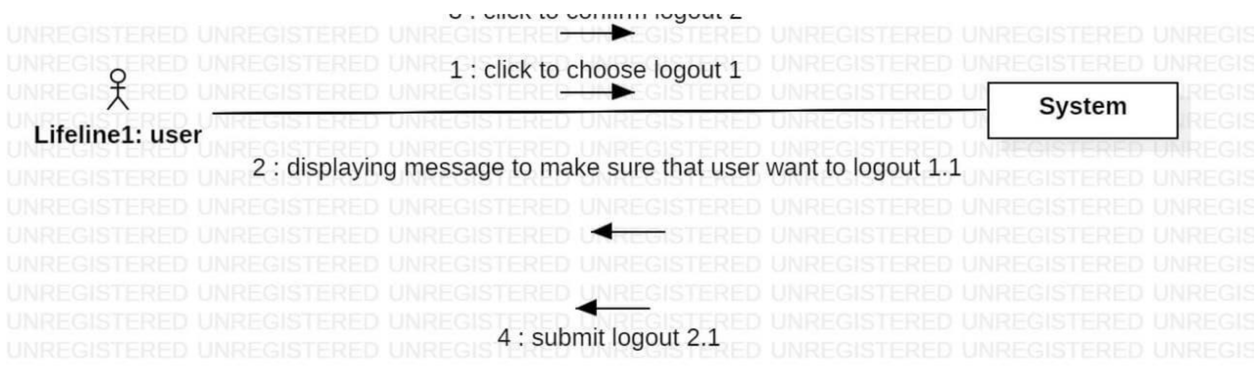
  - User



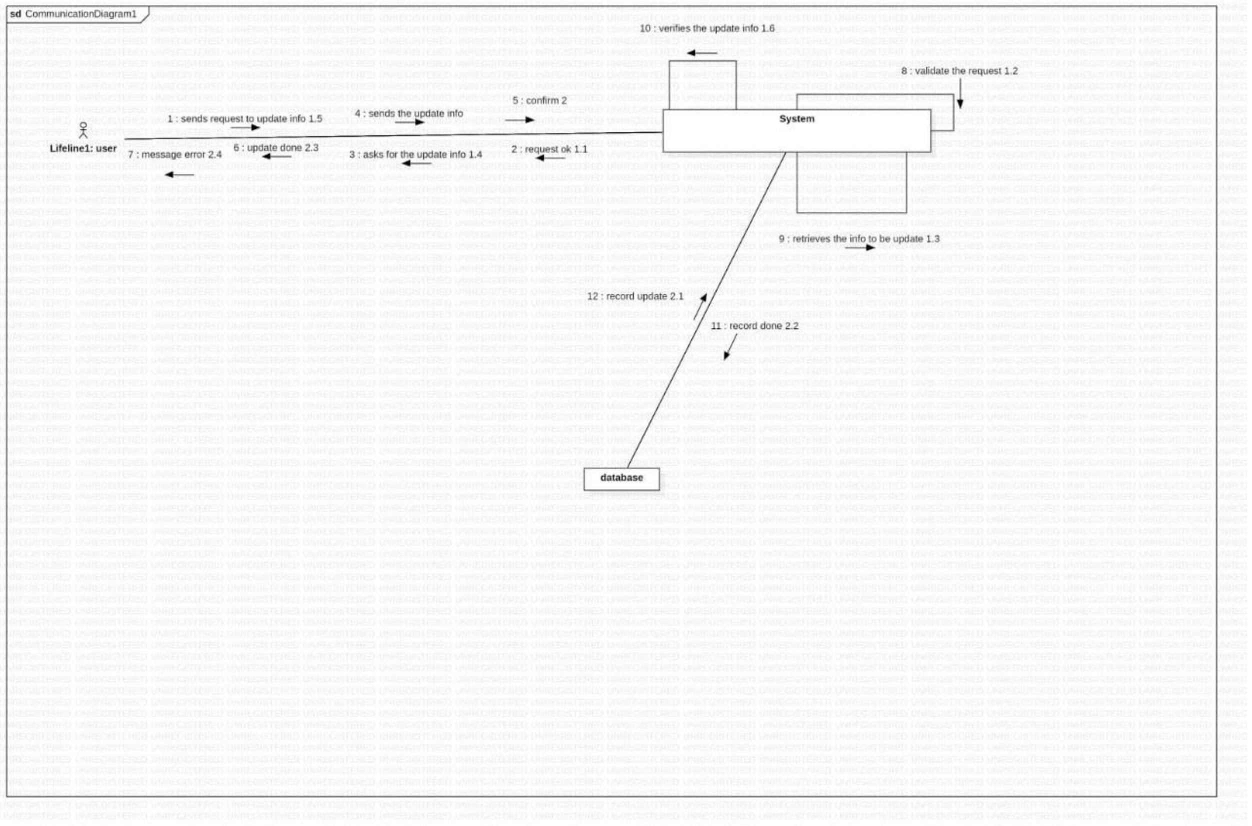
- Log out
  - Admin


1 : click to choose logout 1
Lifeline1: Admin
System
2 : displaying message to make sure that user want to logout 1.1
4 : submit logout 2.1

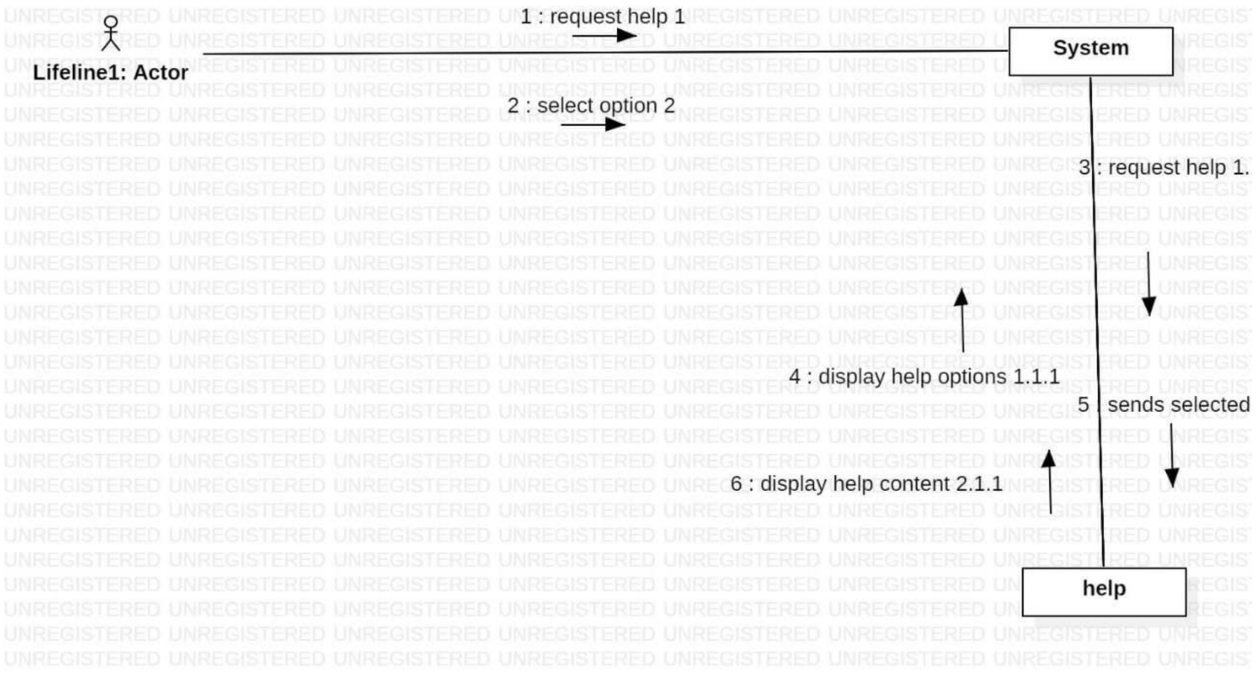
  - User


1 : click to choose logout 1
Lifeline1: user
System
2 : displaying message to make sure that user want to logout 1.1
4 : submit logout 2.1

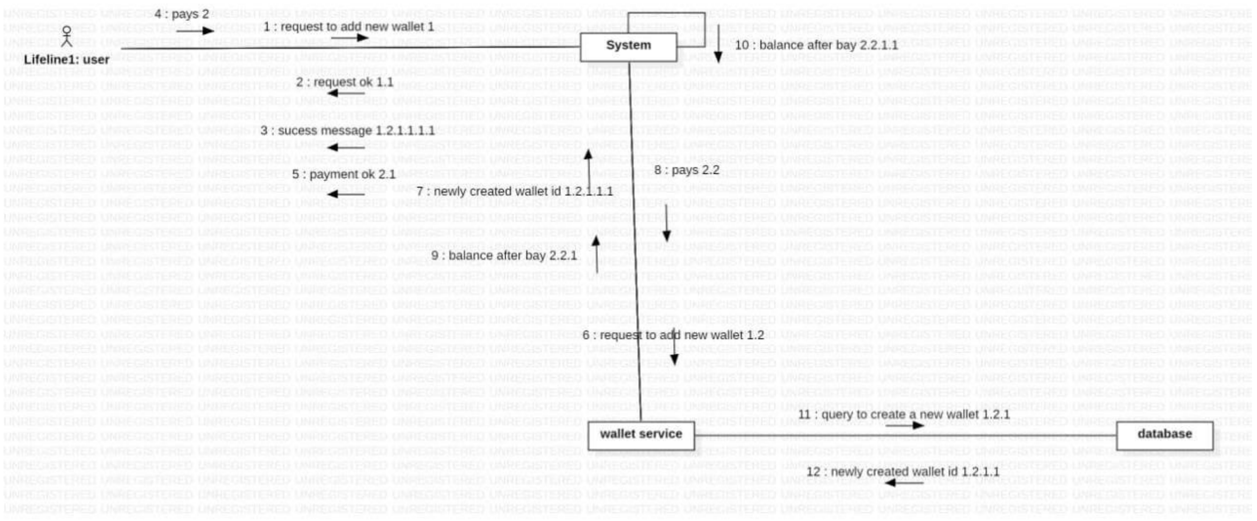
- Update Info


sd CommunicationDiagram1
10 : verifies the update info 1.6
8 : validate the request 1.2
5 : confirm 2
4 : sends the update info
1 : sends request to update info 1.5
System
Lifeline1: user
7 : message error 2.4
6 : update done 2.3
3 : asks for the update info 1.4
2 : request ok 1.1
9 : retrieves the info to be update 1.3
12 : record update 2.1
11 : record done 2.2
database

- Help

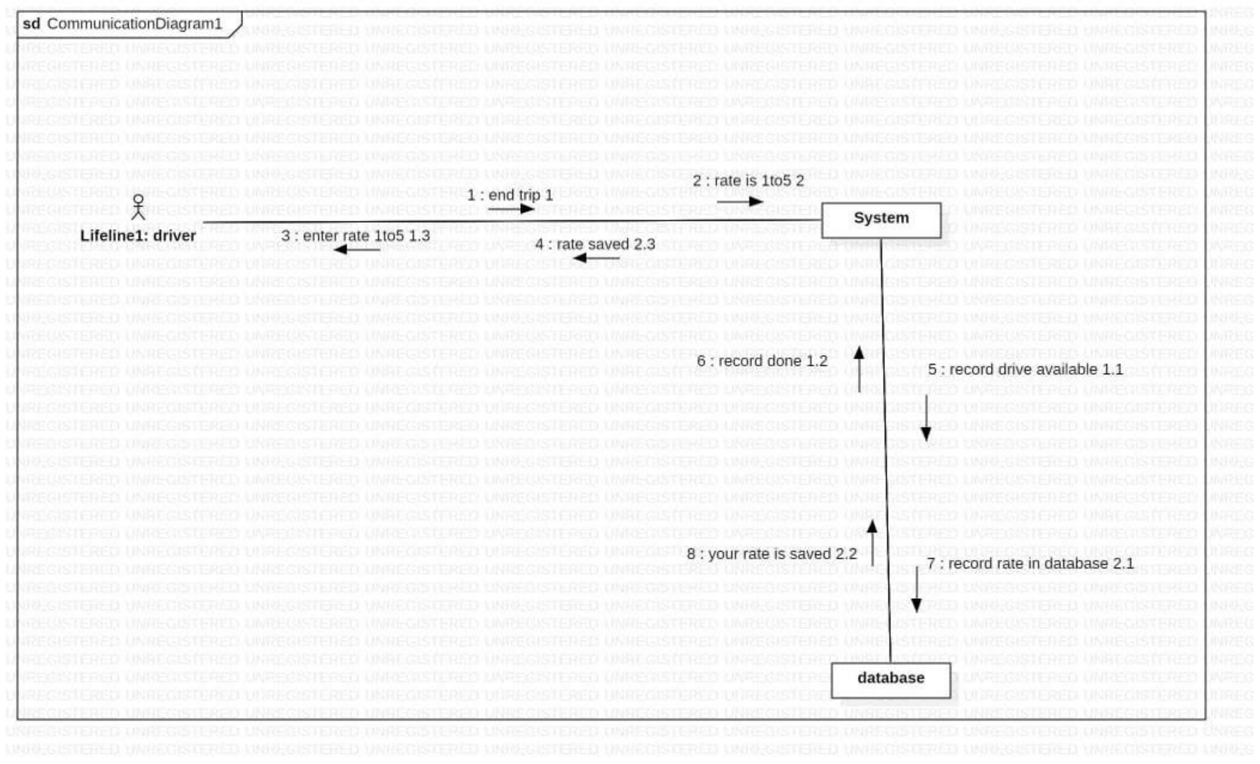


- Add Wallet

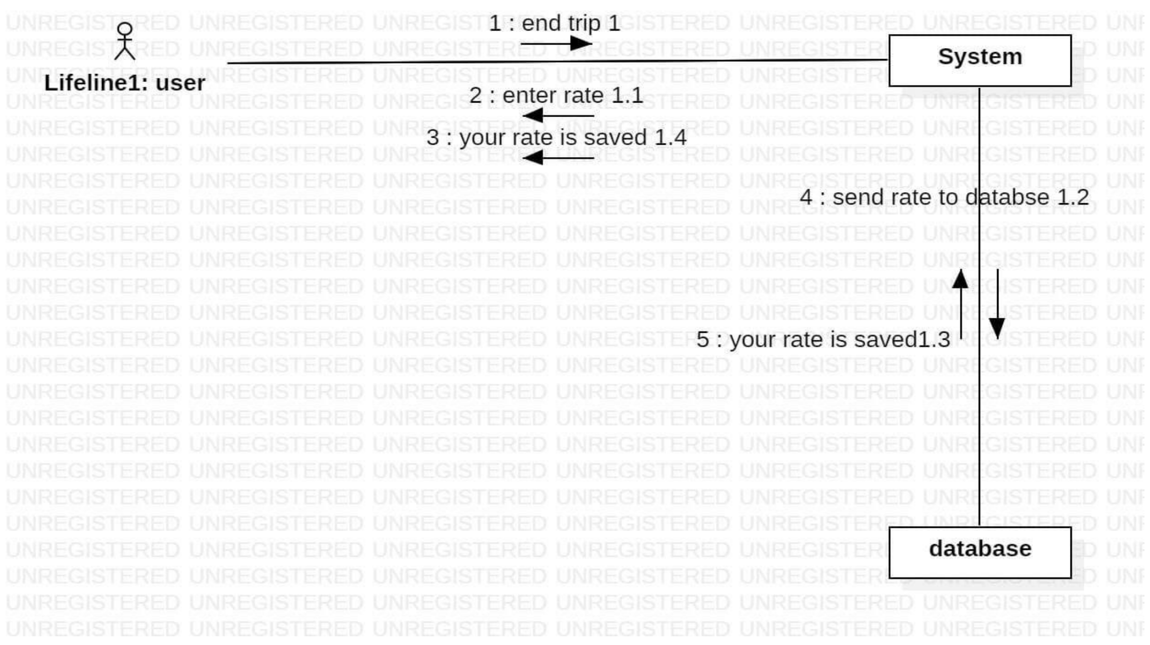

4 : pays 2
1 : request to add new wallet 1
Lifeline1: user
System
10 : balance after bay 2.2.1.1
2 : request ok 1.1
3 : sucess message 1.2.1.1.1.1
8 : pays 2.2
5 : payment ok 2.1
7 : newly created wallet id 1.2.1.1.1
9 : balance after bay 2.2.1
6 : request to add new wallet 1.2
11 : query to create a new wallet 1.2.1
wallet service
database
12 : newly created wallet id 1.2.1.1

- End Trip and Rate

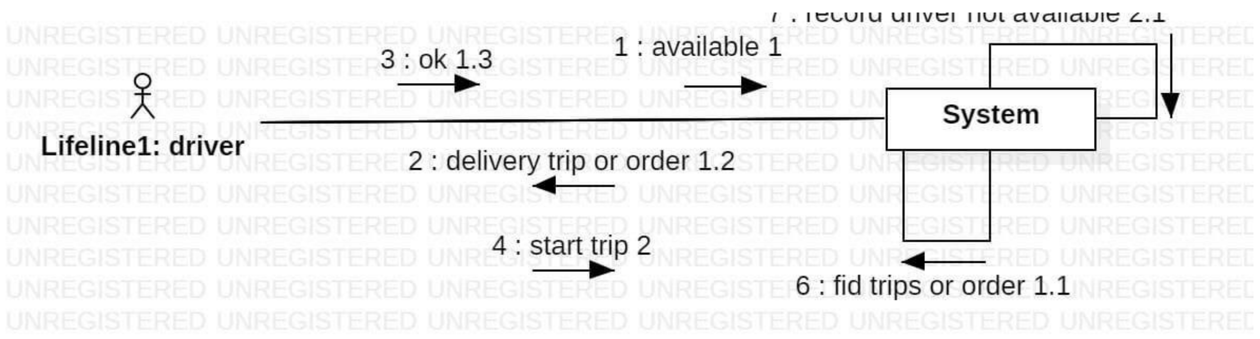
  - Driver



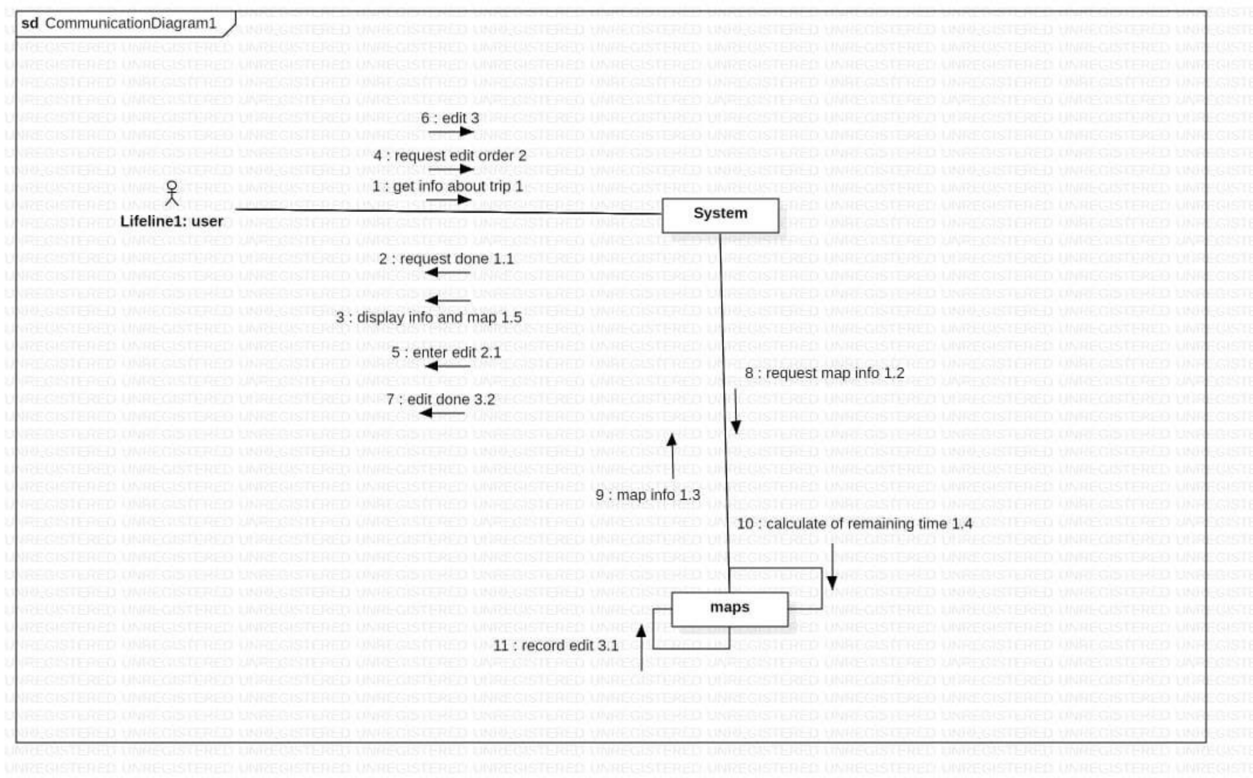
  - Customer


Lifeline1: user
1 : end trip 1
System
2 : enter rate 1.1
3 : your rate is saved 1.4
4 : send rate to database 1.2
5 : your rate is saved1.3
database

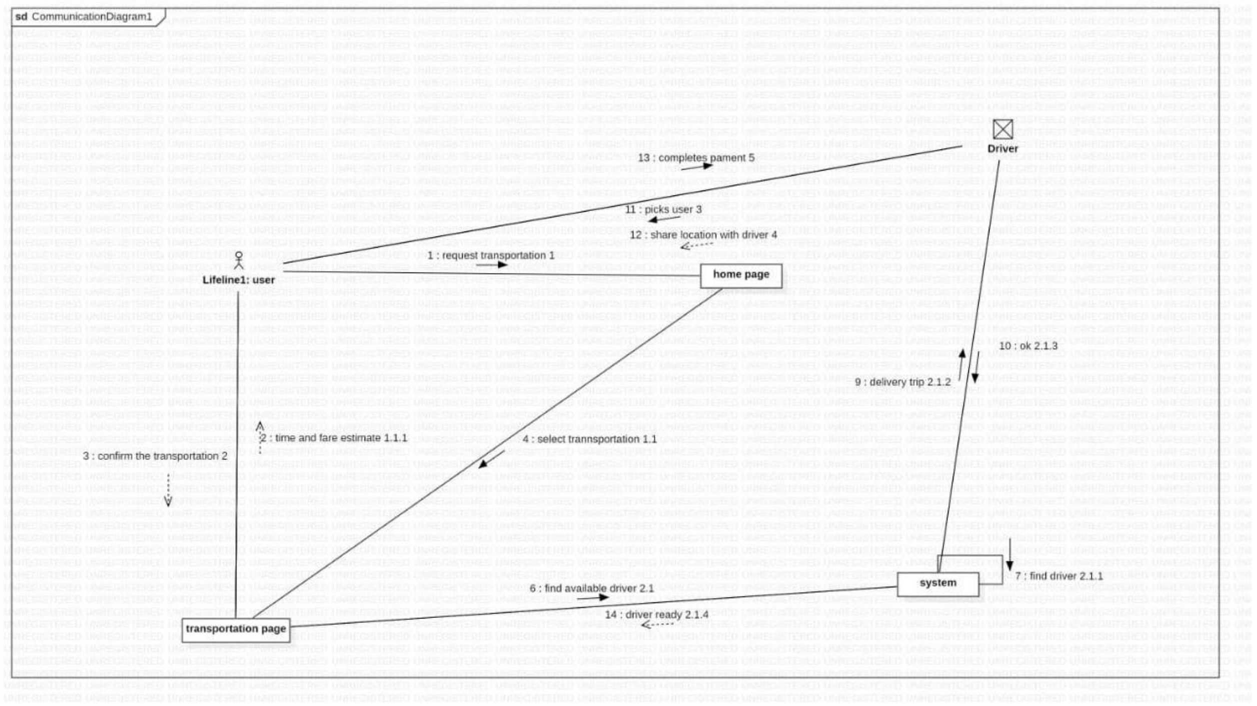
- Delivery


7 : record driver not available 2.1
1 : available 1
3 : ok 1.3
System
Lifeline1: driver
2 : delivery trip or order 1.2
4 : start trip 2
6 : fid trips or order 1.1

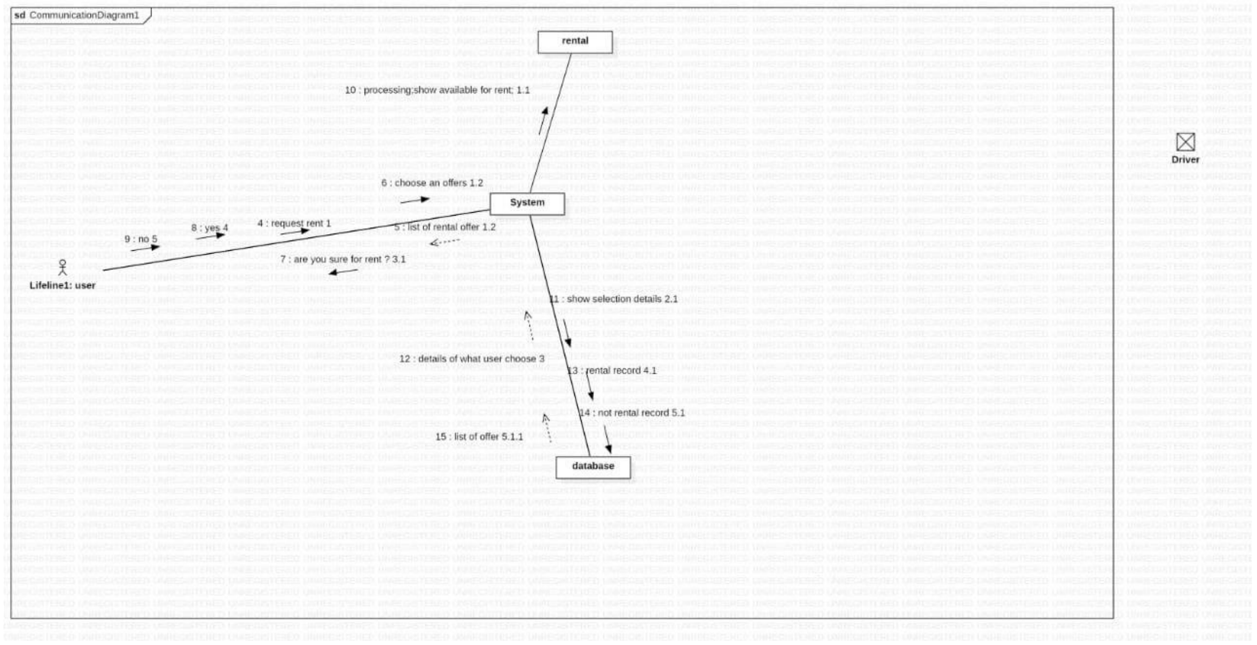
- Follow Order/Trip



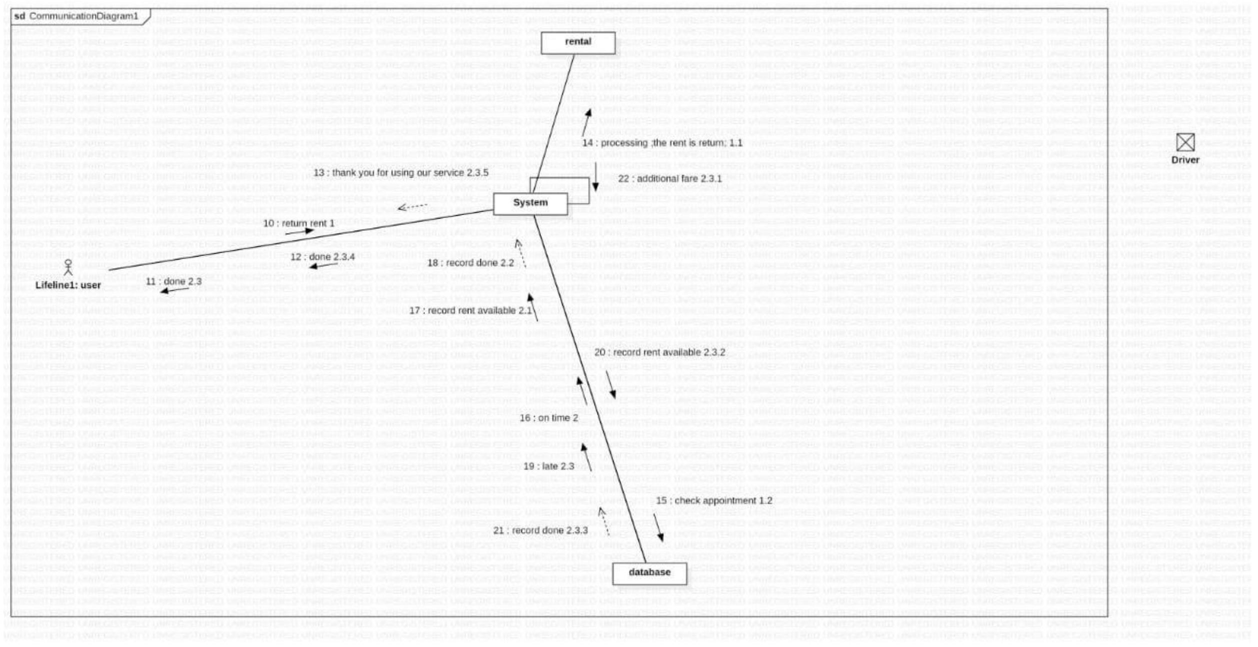
- Booking



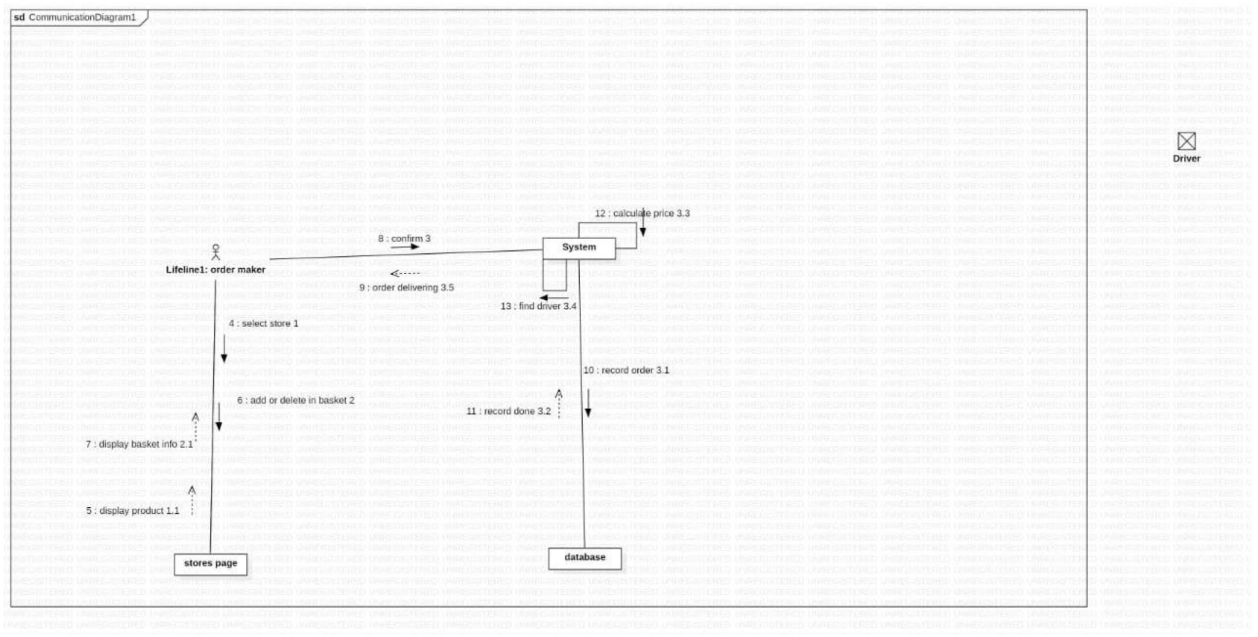
- Make Rent


sd CommunicationDiagram1
rental
10 : processing;show available for rent; 1.1
Driver
6 : choose an offers 1.2
System
8 : yes 4
4 : request rent 1
5 : list of rental offer 1.2
9 : no 5
7 : are you sure for rent ? 3.1
Lifeline1: user
11 : show selection details 2.1
12 : details of what user choose 3
13 : rental record 4.1
14 : not rental record 5.1
15 : list of offer 5.1.1
database

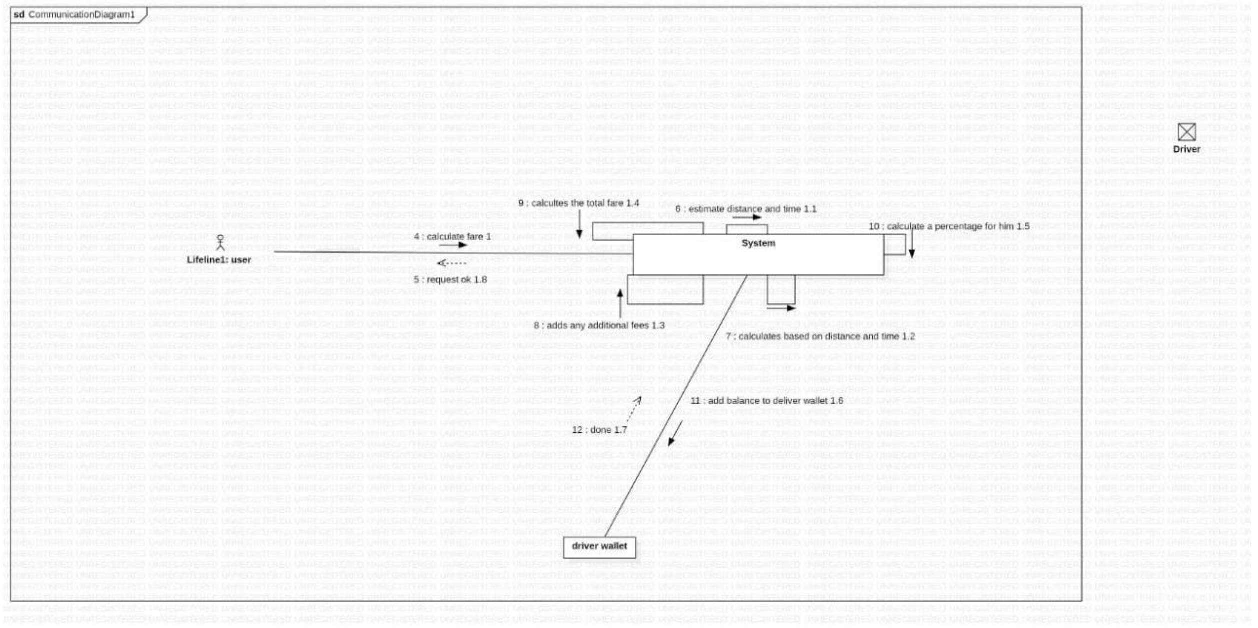
- Return Rent


sd CommunicationDiagram1
rental
14 : processing ;the rent is return; 1.1
Driver
13 : thank you for using our service 2.3.5
22 : additional fare 2.3.1
System
10 : return rent 1
12 : done 2.3.4
18 : record done 2.2
Lifeline1: user
11 : done 2.3
17 : record rent available 2.1
20 : record rent available 2.3.2
16 : on time 2
19 : late 2.3
15 : check appointment 1.2
21 : record done 2.3.3
database

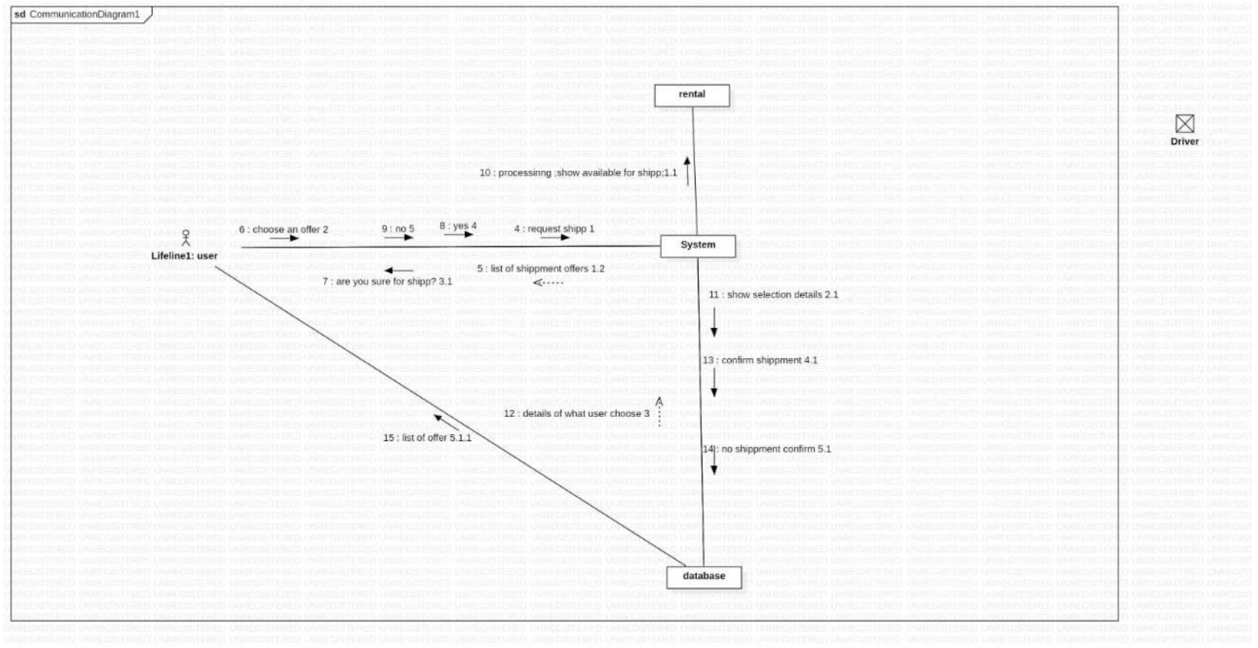
- Browse and Make Order



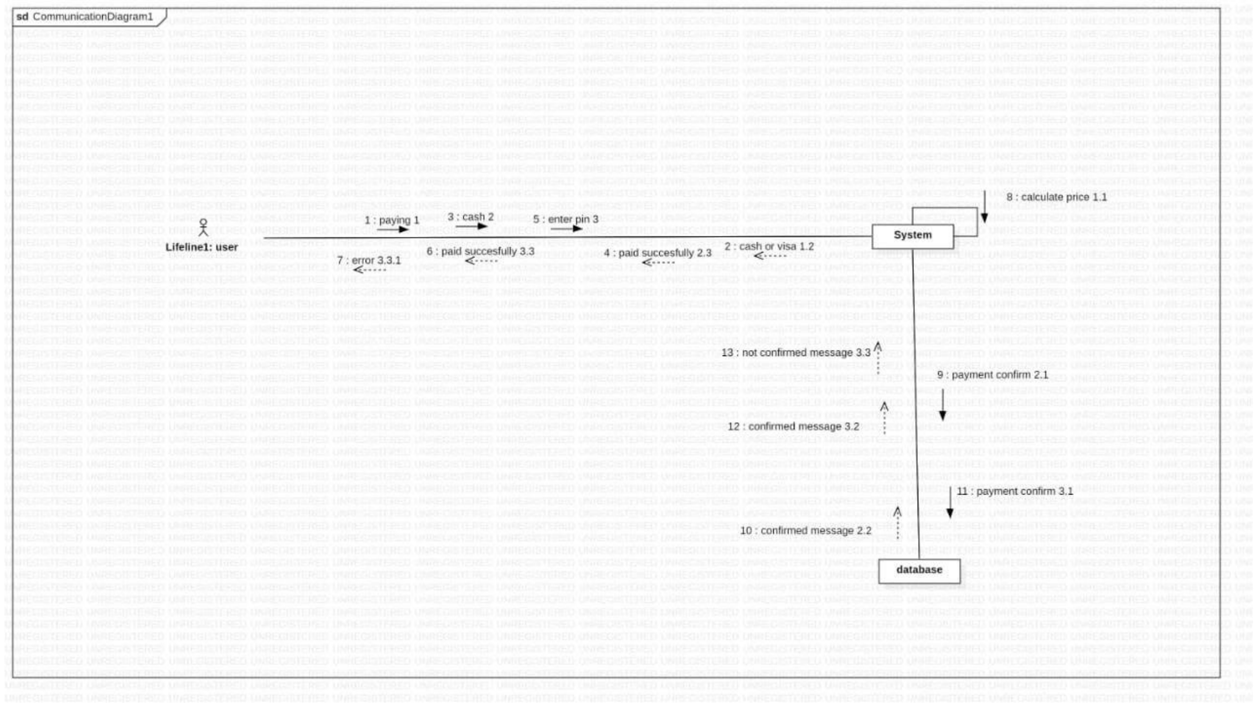
- Calculate Price


sd CommunicationDiagram1
Driver
9 : calcultes the total fare 1.4
6 : estimate distance and time 1.1
10 : calculate a percentage for him 1.5
4 : calculate fare 1
System
Lifeline1: user
5 : request ok 1.8
8 : adds any additional fees 1.3
7 : calculates based on distance and time 1.2
11 : add balance to deliver wallet 1.6
12 : done 1.7
driver wallet

- Shipment


sd CommunicationDiagram1
rental
Driver
10 : processinng ;show available for shipp;1.1
6 : choose an offer 2
9 : no 5
8 : yes 4
4 : request shipp 1
System
Lifeline1: user
5 : list of shippment offers 1.2
7 : are you sure for shipp? 3.1
11 : show selection details 2.1
13 : confirm shippment 4.1
12 : details of what user choose 3
15 : list of offer 5.1.1
14 : no shippment confirm 5.1
database

- Select Payment Method And Pay


sd CommunicationDiagram1
Lifeline1: user
System
database
1 : paying 1
3 : cash 2
5 : enter pin 3
8 : calculate price 1.1
2 : cash or visa 1.2
4 : paid succesfully 2.3
6 : paid succesfully 3.3
7 : error 3.3.1
13 : not confirmed message 3.3
9 : payment confirm 2.1
12 : confirmed message 3.2
11 : payment confirm 3.1
10 : confirmed message 2.2