Helwan University

Software Engineering

(E-commerce application)

Group Number:

Members:

- Ahmed Samir Abd-elsallam (section 1)
- Ahmed Taha Ahmed (section 1)
- Salma Mohamed Salah (section 2)

Submission date: 9/1/2022

We all worked together in each diagram explained below and we discussed all of them together.

Individual contribution:

1. **Ahmed Samir**
    - Customer and shipping entities (ER-diagram)
    - Layer 0 (Data flow diagram)
    - Product and shipping classes (Class diagram)
    - Shipping company actor, registration, login and deliver product use case (use case diagram)
    - Customer login and registration sequence (sequence diagram)
    - Functional requirement specifications (SRS and this document)

2. **Ahmed Taha**
    - Company entity and some relations (ER-diagram)
    - Layer 1 (Data flow diagram)
    - Person and company classes (class diagram)
    - Customer actor, buy, add to cart, view product, order confirmation use cases (use case diagram)
    - Company login and registration sequence (sequence diagram)
    - Customer problem statement (SRS and this document)

3. **Salma Mohamed**
    - Product entity and some relations (ER-diagram)
    - Layer 2 (Data flow diagram)
    - Buyer and cart classes (class diagram)
    - Company actor, sell, add product, modify product use cases (use case diagram)
    - Shipping sequence (sequence diagram)
    - Functional and non-functional requirements (SRS and this document)

Customer Problem Statement:

Shopping App that's glorious invention which allows people to buy things from the comfort of their homes. No more travelling to multiple stores to find the right product you need with all easy way to can choose the correct product you need with all of freedom without any human crowd or any effect; no more having to deal with over-enthusiastic sales persons how just want to buy the product without any thinking of the customer needs; no more standing in long lines at the checkout counter you just want to add your credit card information only without any lanes in shop. The shopping app has certainly changed the way we shop for the better. But, like everything else, the world of online shopping is not all roses. There are a few problems that customers still have to face while using online shop, these problems are listed below.

- **Quality issues**: The biggest problem while buying things online is that you have no guarantee of a product's quality. Reviews are not always reliable and all the research can't assure you of a product's quality, fraudulent sellers who intentionally mislead customers to increase sales are the prime reason for fake products being sold online. With the volume of goods shopping apps handle these days, it can be quite difficult for them to conduct quality checks on each and every one of the products they're selling.
- **Unprofessional Design:** Visuals always provide the first, and often most important impression, either when we see a person for the first time, or visit a website. A negative impression can often deter your customers from making a purchase, even if your store is completely legitimate otherwise. A Shopping store with an appearance that is obviously unprofessional is not inviting to customers. If your design is clearly dated, falls apart, images are missing, buttons are nowhere to

be found, customers will be reluctant to use with your app or buy anything from your app.

- **No Product suggestions:** A good App-search solution should be able to track customer behavior and adapt to it. If App behavior is not tracked, the customer will see irrelevant products in their searches, solely based on the provided keywords, not even taking popularity or current discounts into account.
- **Having Poor Search Engine:** This is probably one of the most overlooked features in the customer shopping experience. the ability to can find what the product you want to search for or to can get all the products related to what you needs to can easiest the way of shopping and find if the product have an a lot of version or type the search engine should get the product .
- **Additional Charges:** How many times has it happened that you've spotted a great deal on a product and when you're one click away from purchasing it you noticed an additional shipping charge? This is commonplace on all Shopping apps when your order amount isn't high enough to qualify for free shipping. And even when it is, sometimes these shipping charges are added on each individual product (if you're buying multiple products of course) and not the collective order.
- **Lack of Security:** Security, or more precisely the lack of it, is a major problem on the internet today. Shopping record important customer data like name, phone number, address, and bank details. If these sites don't implement strong security ways, your data is at risk of falling into the wrong hands who can then have hands on your bank account. Most of the big players in shopping apps certainly have the best security ways to protect their customer’s details, but the same can't be said about the countless smaller apps who may not have the expertise to do so.

- **Delivery:** One predicament that constantly turns up while shopping online is when the order will be delivered. While all Shopping apps have order tracking systems for their customers, they aren't always accurate. Delivery personnel often turn up at our homes when we're at work or out somewhere as there's no way to fix a particular time slot for the delivery to take place. This same issue exists while returning products. Another problem is that the vast majority of the population which lives in rural areas and slums is unable to shop online because not all Shopping apps provide delivery services to their locations.

- **Complex Checkout:** In the past years, nearly all research into cart abandonment rates confirmed that the third most common reason – after high/unexpected prices and forced account creation – people abandon their cart is that the checkout process is too complex. Most e-tailers don't take this essential fact into account: people are only willing to go through a complex process if the perceived value of the product they want to buy is very high. In the case of complex products where the customer spends a great amount of time researching, comparing prices, finding the most suitable solution, a complex check-out can be fully warranted and in itself can serve as a catalyst for the IKEA effect. But if they want to buy small appliances, food, clothes, or other relatively low-value, everyday products, they expect a very simple, very fast checkout.

- **No Returning Policy:** Just as with guarantees, the only reason for not making your return policies as flexible as they can be is if you don't trust your own product. But if you do, these policies build trust and make things much easier for those few customers who will actually return something – which means you have a better chance of retaining them as customers.

- **Unclear Product Information:** Shopping app should introduce a product to their customers, by break down the information into two basic categories: product features and product benefits. So let's start by looking at product features. Product features include all the dry information about how the product works, looks, how big it is, how fast it is, warranties, and so on. Customers are likely to abandon a purchase if they are not able to find an answer for their questions immediately.
- **Lack of Support:** If Shopping apps don't provide support for their customers, you are not only depriving help from those who already purchased something from you, but you could also be alienating would-be customers. Would-be customers have a lot of questions, especially if your product descriptions and attributes are not properly provided. If they receive poor customer service, they will just turn away and not make the purchase in the first place.

System Requirements:

1. **Functional Requirements**

REQ-x	Definition
Registration	The user should choose the reason for visiting the website either for buying products or for selling his own products or even as a shipping company. This will provide the user with a certain permissions in order to complete browsing in the website. And this considered as one of registration steps. The user can register by adding the following information: • **For Buyers:** Username, password, email, Address, phone number. • **For sellers:** Username, password, email, Address, phone number, seller id, company name. • **For shipping:** Company id, name, address, email, phone number.
Login	All users can login to the application using their username or email and password.
View products	All users can view the presented products which are divided into pages (each page contains a type like mobiles, TVs ...etc.). The user can navigate between these pages.
Search for a product	The user can search for any product by typing product's name/type inside the search bar.
Customer Cart	Every **Buyer/Customer** have his own cart that has been automatically created after he completed the registration process. The customer can add unlimited number of products he wanted to buy to his own cart. He also can remove any product from his cart in case of changing his mind.
Confirmation Process	The **Customer** should confirm buying a product by typing some payment information like: credit card number, password and expiration date. After confirmation, user's data will be sent to shipping company.

Add product	Only **Companies/Sellers** can add their own products to the application. The number of products a company can add is unlimited.
Modify product	Only **Companies/Sellers** can modify their own products by editing product's name/type/price or even remove it.

2. Non-functional Requirement:

- **Availability:**

The system is always available.

- **Security:**

No one can access the system except the **admins**.

- **Safety:**

Servers will be kept only in the vice dean office.

Stakeholders:

1- Buyer (customers).
2- Mangers (application admins).
3- Shipping companies.
4- Seller companies.
5- Application Developers

Actors and Goals:

1- **Seller companies:** the selling companies before interact to this system will sell their product only from their shops but those companies wants to increases their selling by making it easy to the customer to by their product

from home by using our application so companies will show their products on the application to be easy to the user buying it.

2- **Buyer (customers):** the customer before interact with the system customers only buying the product from companies shop or supermarket and it hard to customers to see all kinds or the products and compare between them but the customers need to make their life easy by buying all product from home, and it will help the people who having a problem to go outside the home to make shopping and the shop will be online.
3- **Shipping companies:** those companies want to increases there work by shipping the buyer product from the stoke to the customer home.

Use cases:

a- The buyers (customers):

1- Customer can register to be able to login in the next time to our application.
2- Customer can login to be able to use our functionality in our application.
3- Customer: can buy a products by adding the products in the cart then confirmation.
4- Customers can see the products from many companies and compare between them.

b- The seller (selling companies):

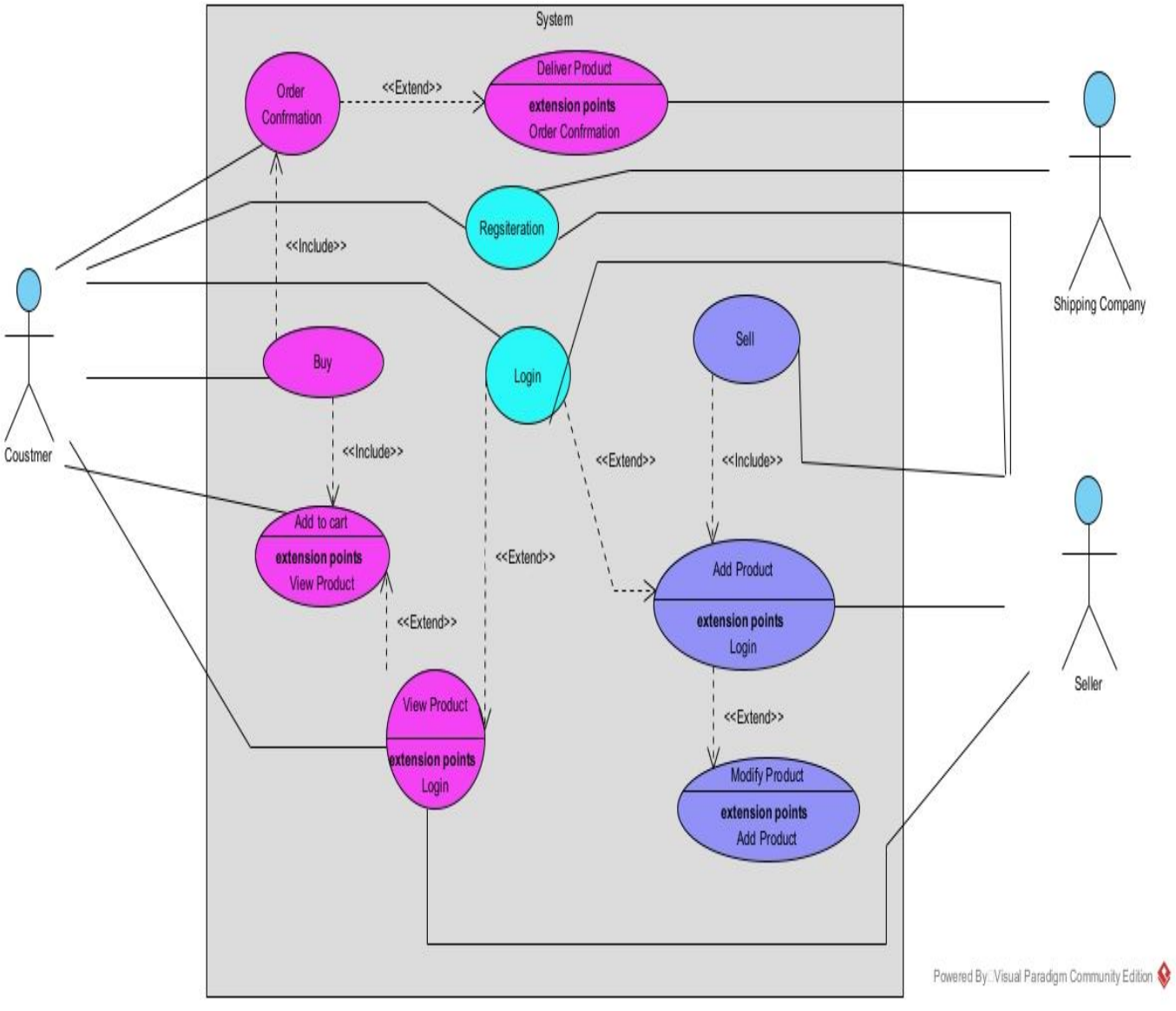
1- Selling companies can register to be able to login in the next time to our application.
2- Seller can login to be able to use our functionality in our application.
3- The selling companies can add the product to our application.

4- The selling companies can modify or delete the products and setting an offer on their products.
5- Seller can view their products that on our application.

c- Shipping companies:

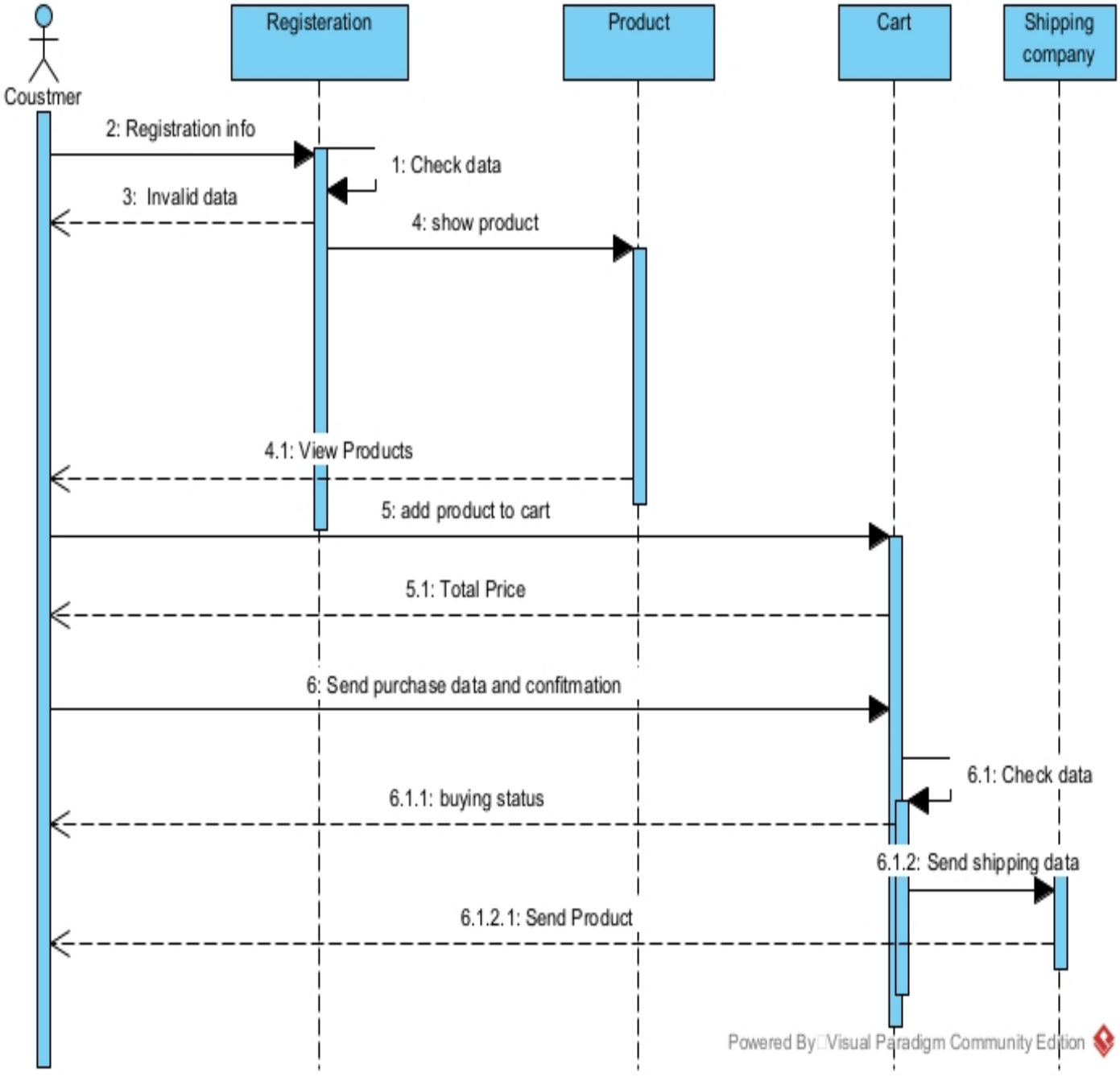
1- The shipping companies when the orders confirmed it will take it and deliver it to the customer home.

Use Case Diagram

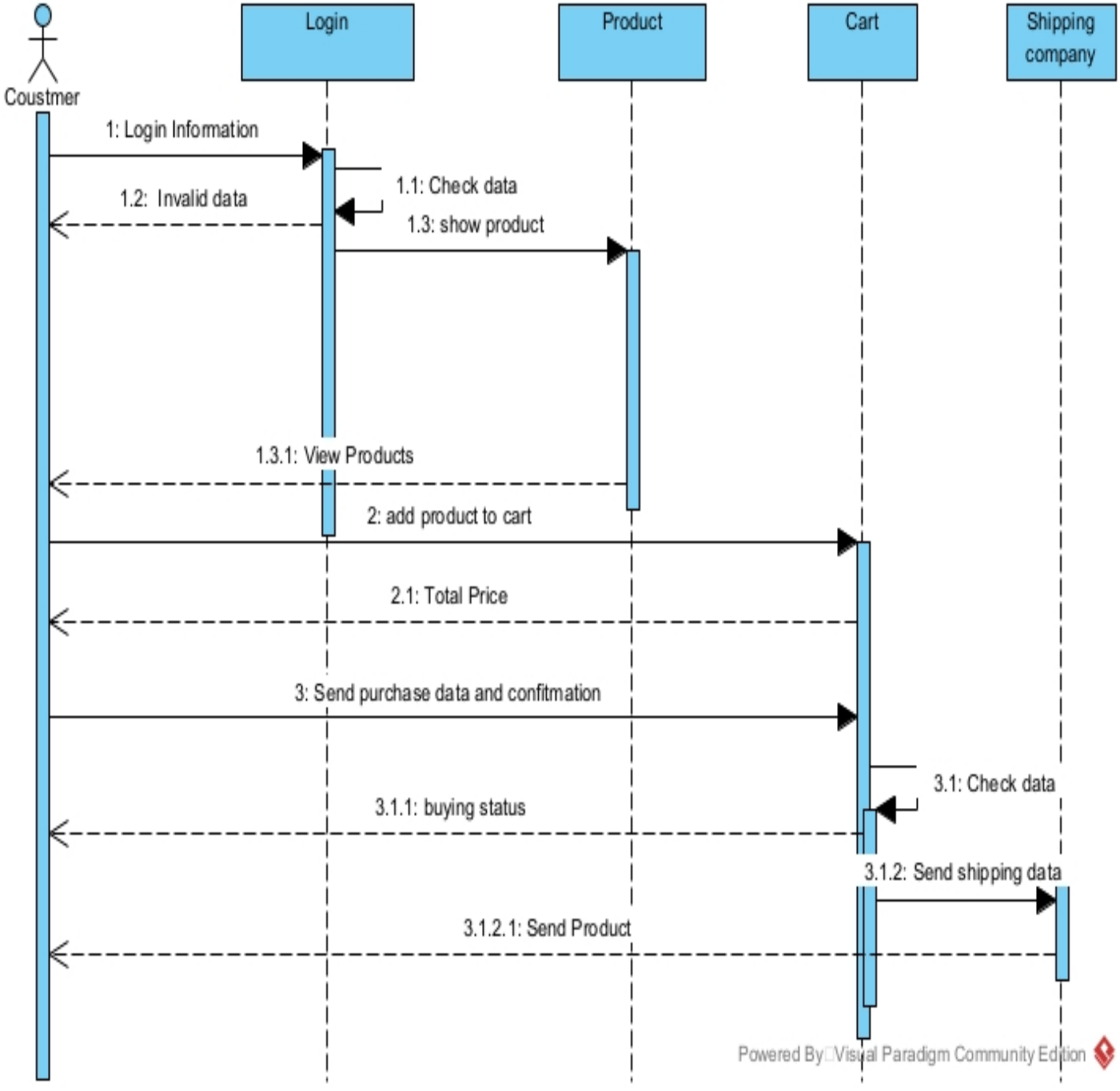


System Sequence Diagrams

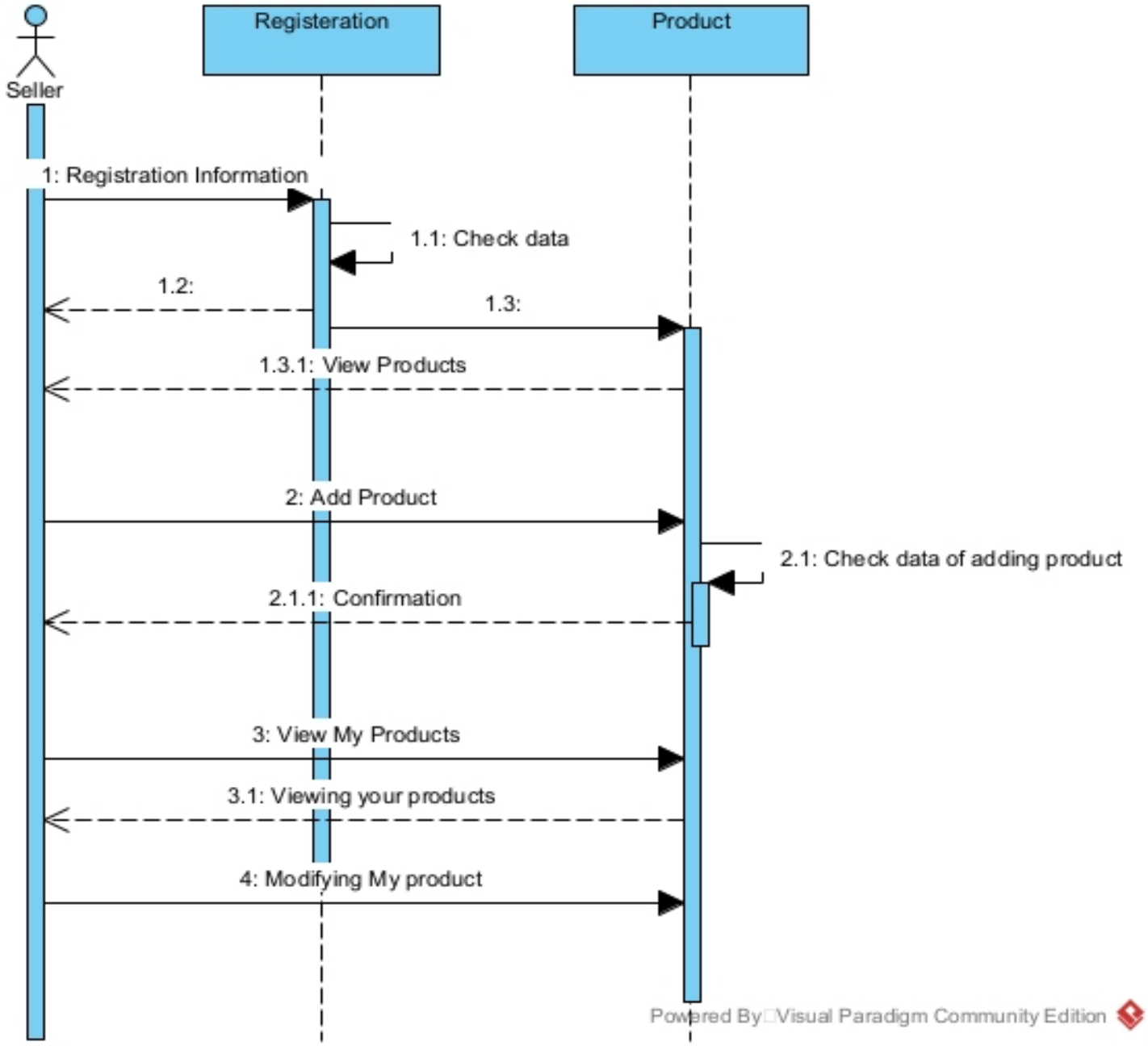
Customer registration:



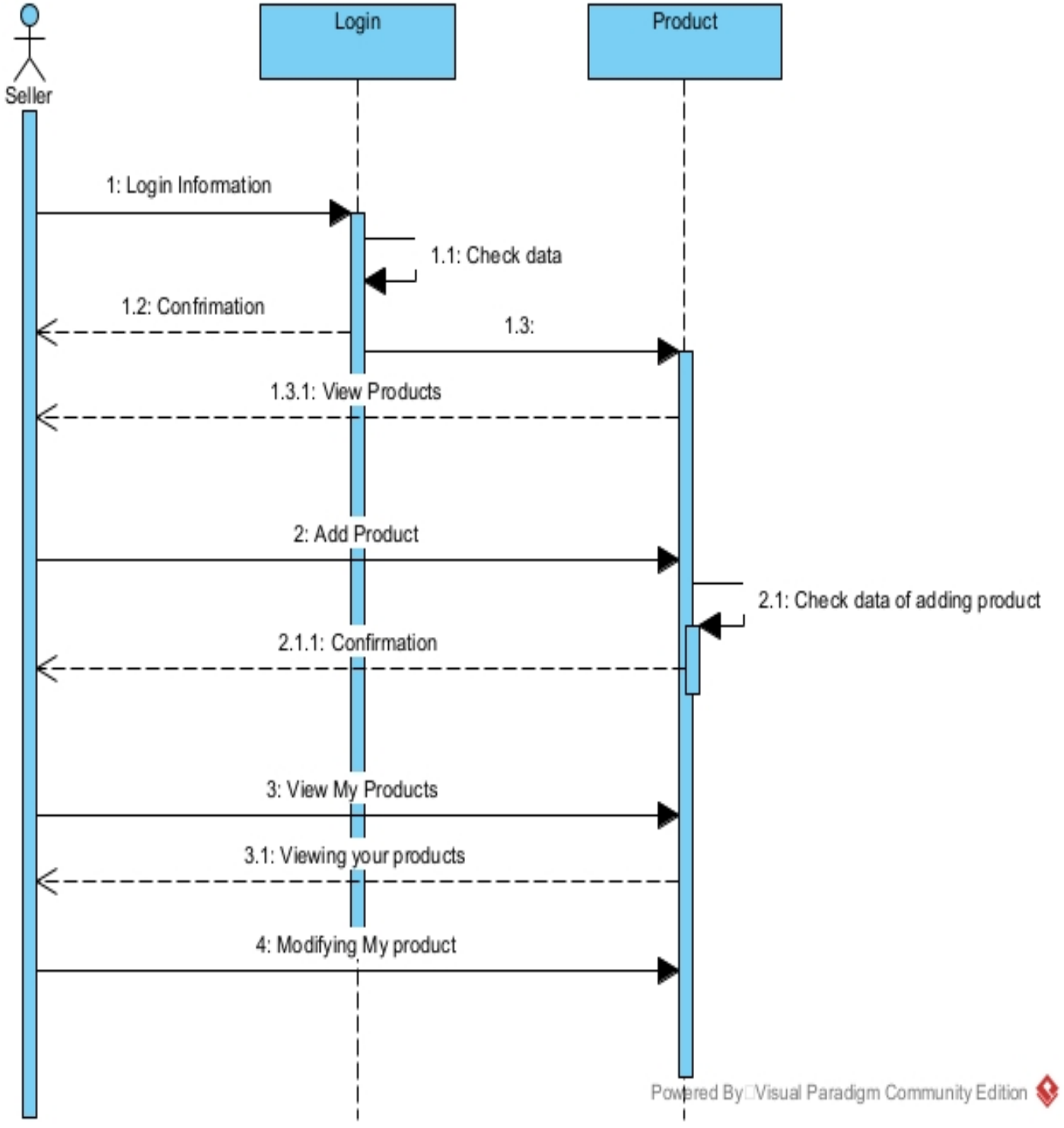
Customer login:



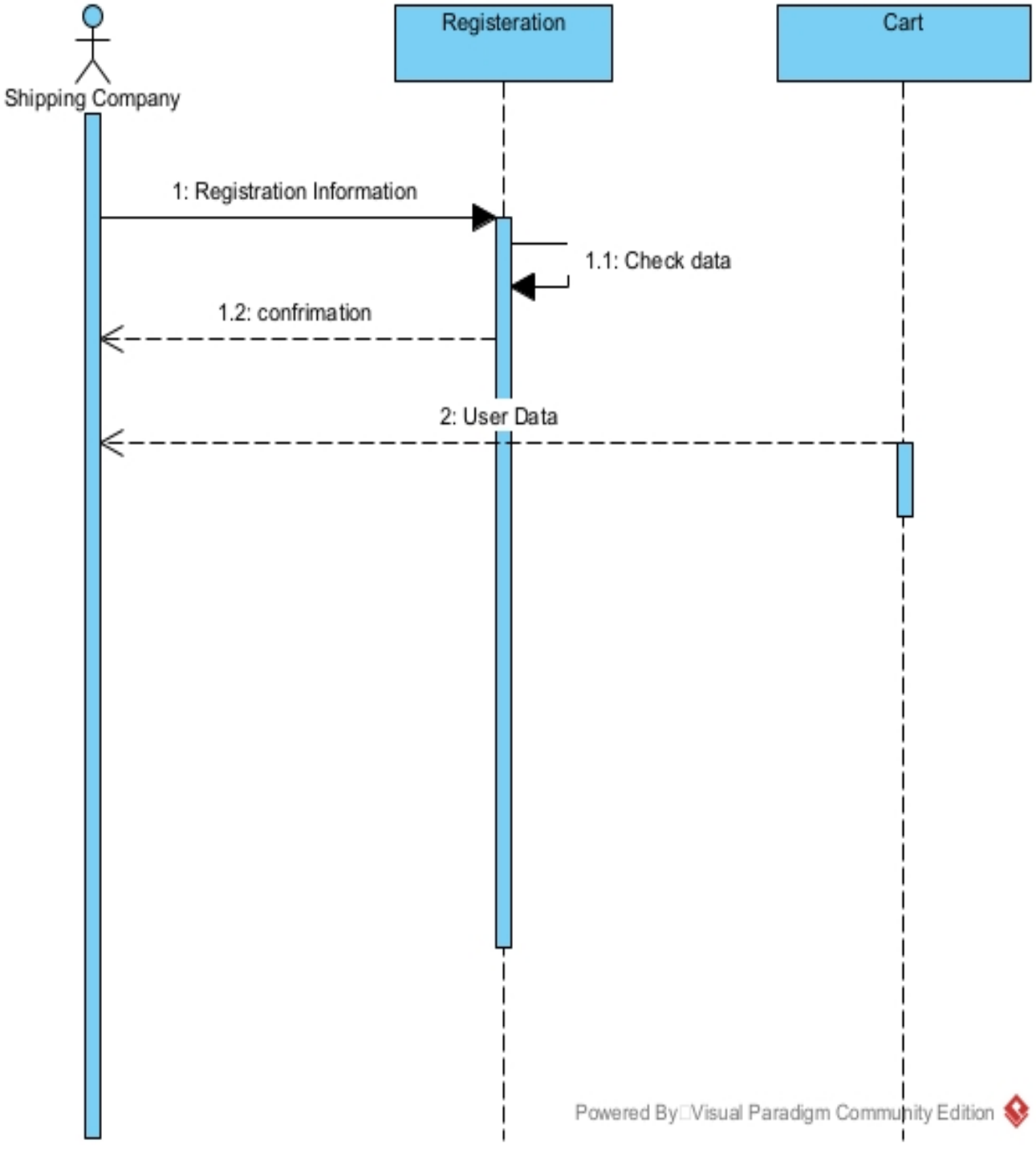
Seller registration:



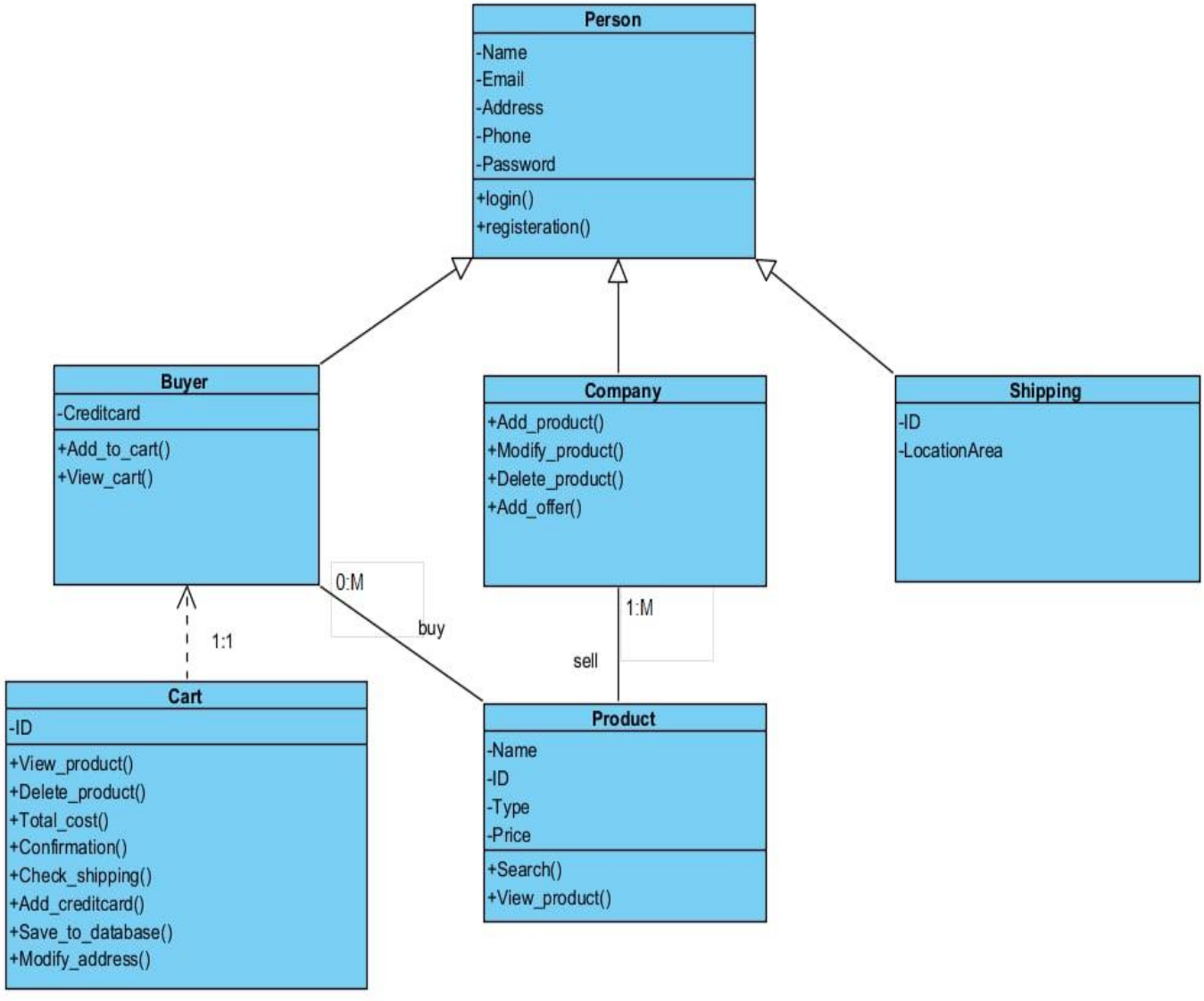
Seller login:



Shipping companies:



Class diagram:



Person class:
This class not inherit from any class it's a parent class to **buyer** class ,**company** class and **shipping company** class
Attributes:

- name its data type is string
- email it's data type is string
- Address it's data type is string
- Address Area it's data type is string
- Phone it's data type is integer
- Password it's data type is string

Operations:

- **Login operation**: This methods takes the data from the GUI class where the user interact with then put this data in our database and known what type of the user is login seller or costumer that's indication for GUI to can shows the user interface according to him
- **Register operation:** this method takes the indication from the GUI what's the type of the user how register to the system then takes the data from GUI then insert this data into our data base.

Buyer class:
This class is a child class for the person class
Attributes:

- CreaditCardNumber its data type is integer

Operations:

- **Add to cart**: This methods takes the information of the user product choose then add to the cart data base with his user name
- **View Cart:** This method View all the product in the user cart data base

---

Company class:
This class is a child class for the person class
Attributes:

- This class contains no attributes

Operations:

- **Add Product**: This method takes data from the user from GUI like the image of product, it's information's of product and it's price then add this data in our products data base
- **Modify Product:** This method View all the product in the user cart data base then the user can **delete** the product or change any of it's information or change price **offer**

Shipping Company class:
This class is a child class for the person class
Attributes:

- Id its data type is Integer
- Location Area: its data type is string

Operations:

- **Add Company data** This method takes data from the user(Shipping company) from GUI when user is register then send it to our shipping companies data base

---

Cart class:
This class is dependent class which dependent on the buyer class
Attributes:

- Id its data type is integer

Operations:

- **Delete Product**: This methods takes which the user product need to delete Which select this data in database then delete it from data base
- **Confirmation:** This method takes confirmation from the user then show him another frame to enter there credit card number data
- **Total cost :** This method calculate all the cost of all the product in the cart of the user then view it
- **View product :** This method shows the data in the cart of the user from database of the cart
- **Save to database :**This method saves any change in cart data

---

Cart class:
This class is dependent class which dependent on the buyer class
Attributes:

- Id its data type is integer
- Name it's data type is string
- Type it's data type is string
- Price it's data type is string

Operations:

- **View Product**: This method shows the products to user depend on which type the user want to show by get it from database
- **search:** This method takes any related thing to the product user search for the view it to him in GUI