



## ***Software Requirements Specification (SRS) Project***

Team members:

*Elma Sarajlić & Merima Maksumić*

## Contents

Project Proposal: .....	3
Feature description document .....	4
Functional requirements: .....	4
Non-functional requirements: .....	8
Use case diagram .....	9
Activity diagrams.....	10
Class Diagram.....	22
Object diagram.....	23
Sequence diagrams .....	24
Communication diagrams.....	32
Behavioral State Machine Diagrams.....	37
Packages Diagrams.....	39

### ***Project Proposal:***

<b>Project Title:</b> Cosmetic Pharmacy “Beauty by ME”	
<b>Start Date:</b> 07.11.2022.	<b>End Date:</b> last week of January
<b>Project Manager:</b> Merima Maksumić and Elma Sarajlić	
<b>Project Sponsor:</b> Cosmetic Pharmacy “Beauty by ME”	
<b>Purpose and business need:</b> <p>The purpose of cosmetic online pharmacy is to extend and improve already existing pharmacy system. The project will focus on skin care and makeup product placement.          The project will aim to improve the experience and satisfaction of the customers, who will be able to order all the necessary products from the comfort of their homes, as well as consult with the professional cosmetic chemists. Business need is to extend the market and increase earnings.</p>	
<b>Customers/Users:</b> <p>The cosmetic online pharmacy project is aimed toward customers of all skincare and makeup cosmetic chemists. Main users will be customers who wish to resolve or improve specific skin or hair related issue.</p>	
<b>Goals and objectives:</b> <p>The goal of the project is to improve existing business by ensuring an adequate range of products and providing more specialized assistance to the customers. The project plan is to increase number of orders by offering pharmacy services and product delivery to customers all around the country.</p>	
<b>Project priorities:</b> <p>The focus of the project will be a cosmetics pharmacy online web shop. Web shop will consist of product placement, online counseling corner, product ordering and delivery feature. Promotional offers will be offered as well.</p>	
<b>Constraints and special issues:</b> <p>Conditions that limit the actions of the project team may flow from the client's non-compliance with the contract. Also, integration with banking systems can be one of the difficulties of our team.</p>	
<b>Risks and obstacles to success:</b> <p>Potential risks of the project are exceeding the deadline as well as the lack of people in the team due to unpredicted circumstances.</p>	
<b>Technical Leads:</b> Merima Maksumić and Elma Sarajlić	

## *Feature description document*

Functional requirements:

### **1. Feature name:** Registration

**Role(s):** Customer

**Feature description/steps:**

1. When the registration is chosen, the user needs to fill in tendered fields with their personal information.
2. User enters Name and Surname, valid and unique Email address, Password, Repeat Password, phone number and address. (Password is a minimum of eight characters, combination of letters and numbers, with special symbols).
3. In addition to registration the user is going to receive a confirmation email that needs to be sent in two minutes from registration on users entered email.

**Dependencies/constraints:** All the fields must be filled in and in the right form. One username cannot be used more than once. Passwords must match in both fields. Name and Surname must be only letters.

### **2. Feature name:** Login

**Role(s):** User (Customer and Employer)

**Feature description/steps:**

1. Choosing this option will prompt the user with fields to enter username and password and to click Login.
2. Below login fields need to be the "Forgot my password" option which, when clicked, is going to provide details for login on the user's email. Email contains instructions on how to change the password.
3. Below login fields need to be the "Forgot my username" option which, when clicked, will provide details for login on the user's email. Email contains instructions on how to change username.
4. If the username or password are incorrect, the message "Username or password is incorrect" will appear.
5. After entering correct information click on button Login.

**Dependencies/constraints:** User needs to be registered first.

### **3. Feature name:** Web shop

**Role(s):** User

**Feature description/steps:**

1. After clicking on the web shop icon, which is located on the home page, it leads the user to the web shop.
2. Web shop contains a "category menu" with many categories of products, which is a part of the web shop.

3. Web shop displays all the available products with their picture, description, and option “add to cart” and “add to wish list”.
4. Filter which is an addition on web shop is placed above products.
5. Option “sorting” is available on web shop as well and is used by users to choose how they want their displayed products to be sorted.
6. Indexing of the page, with the number of products on that same page, is placed on the bottom of the web shop.
7. Option “search” is displayed at the top of the page, it is used by users to search by name one specific product.

**Dependencies/constraints:** Need to always update it. Limited source of products.

**4. Feature name:** Filter

**Role(s):** Customers

**Feature description/steps:**

1. Filter is placed on the web shop. It is an addition to it.
2. Requirement “filter” is supposed to provide users with options of descriptions of products contained on web shop. Users can filter their results by price range, by brand, by type of the product, etc.
3. Requirement “filter” needs to be also available in Categories (When a user is in a specific category, it needs to be available to filter that category as well).

**Dependencies/constraints:** Depends on web shop.

**5. Feature name:** Sorting

**Role(s):** Customers

**Feature description/steps:**

1. Option “Sorting” is placed on the web shop as well as on outlet.
2. It needs to be able to sort products by alphabet, price (lowest and highest), popularity and by last added items.
3. Requirement “Sorting” needs to be also available in Categories (When a user is in a specific category, it needs to be available to filter that category as well).

**Dependencies/constraints:** Depends on web shop.

**6. Feature name:** Outlet

**Role(s):** Users

**Feature description/steps:**

1. Additional web shop that contains soon expiring products (Product has picture, description and “add to cart” and “add to wish list” option as well).
2. Outlet has two functionalities: sorting and indexing.

**Dependencies/constraints:** Need to always update it. Limited source of products.

**7. Feature name:** Search

**Role(s):** Customers

**Feature description/steps:**

1. Screen shows all the products under the search criteria (specific string).

2. List all products that contain entered key word in their name or description.
3. If there are no products under these search criteria, then the system will give the message that there are no products found.

**Dependencies/constraints:** Search should not be case sensitive.

**8. Feature name:** Wish list

**Role(s):** Customer

**Feature description/steps:**

1. Allows shoppers to create personalized collections of products they want to buy and save them in their user account for future reference.
2. Connected with web shop and outlet.
3. Product from wish list can be forwarded to the cart.
4. Product from wish list can be deleted from the list by user.

**Dependencies/constraints:** Hold list of products until user decides to delete them.

**9. Feature name:** Cart

**Role(s):** Customer

**Feature description/steps:**

1. It stores all products that are chosen from the user with option “add to cart” and holds it until purchase.
2. Makes the sum of all chosen products prices.
3. Needs to have an option for editing a quantity of products.
4. Needs to have the option to delete the product from the cart.
5. Has the option to click “Buy” which leads users to another feature called Payment process.

**Dependencies/constraints:** List of products need to be well connected with a database of available products.

**10. Feature name:** Payment process

**Role(s):** Customer (Registered customer)

**Feature description/steps:**

1. Has implement integration system with bank.
2. Has fields which are needed to be filled in by the user with necessary information- method payment (card or payment on delivery), change or add new address, addition number, ...
3. Has autofill of already entered information about the user at the registration part (All information that he entered at the registration).
4. Needs to have field for entering valid coupon.
5. Send confirmation bill on users email when purchase is processed by system.

**Dependencies/constraints:** Dependent on third party system (bank).

**11. Feature name:** Counselling corner

**Role(s):** Customer, Employer

**Feature description/steps:**

1. Counselling corner represent direct communication between user and employer via messages.
2. Needs to have an automatic reply to messages.

**Dependencies/constraints:** Not being able to work 24/7.

**12. Feature name:** Log out

**Role(s):** User

**Feature description/steps:**

1. Needs to return the customer to log in again.
2. Automatic timeout after 10 minutes of inactivity.

**Dependencies/constraints:** The user needs to be logged in to log out.

**13. Feature name:** Add product

**Role(s):** Employer

**Feature description/steps:**

1. Can add products with all necessary info about them in web shop and outlet.

**Dependencies/constraints:** Chosen product might be unavailable in stock at that moment.

**14. Feature name:** Delete product

**Role(s):** Employer

**Feature description/steps:**

1. Can delete products from the web shop and outlet.

**Dependencies/constraints:** To be able to delete a product, that product first needs to exist on a web shop.

**15. Feature name:** Defining users of system

**Role(s):** Admin

**Feature description/steps:**

1. Should be able to create a user as an Employer and unlock specific features for that employer.
2. Should be able to restrict irresponsible users.

**Dependencies/constraints:** To restrict account, that account first needs to exist in account database.

Non-functional requirements:

- The app should be fast, all requests need to be handled under 2s.
- All the information about users need to be highly protected and only be available to admins.
- Orders shall be processed within 2 minutes.
- The application software should be stable 99% of time.
- The system needs to be able to process 5000 users at the same time.
- Mobile-Friendliness (should be adapted to the screens of the mobiles that are 4 years old and below).
- Header and footer must be the same on every feature (Header -provides menu with all features and footer provides basic information)
- Suitable for all browsers
- The database needs to be able to contain 1 million registered users
- The user can choose between two languages. The website can be translated to English and Bosnian language.
- Update of the information needs to be done on the daily basis (every hour).



## Use Case Diagram

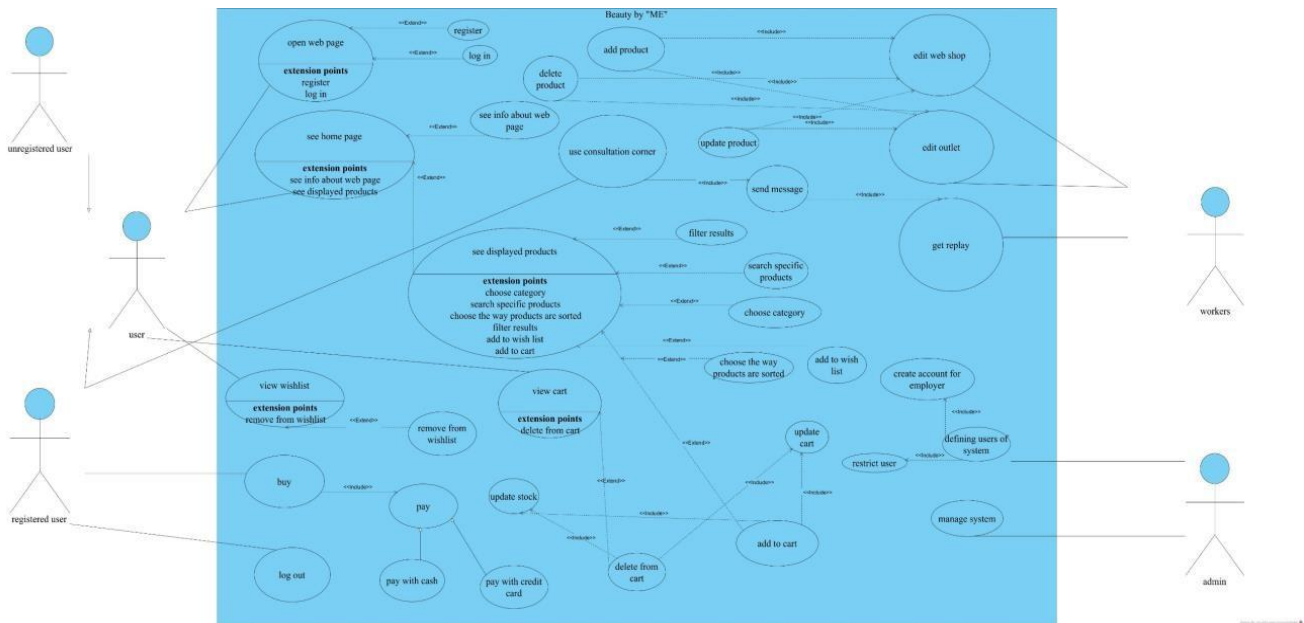
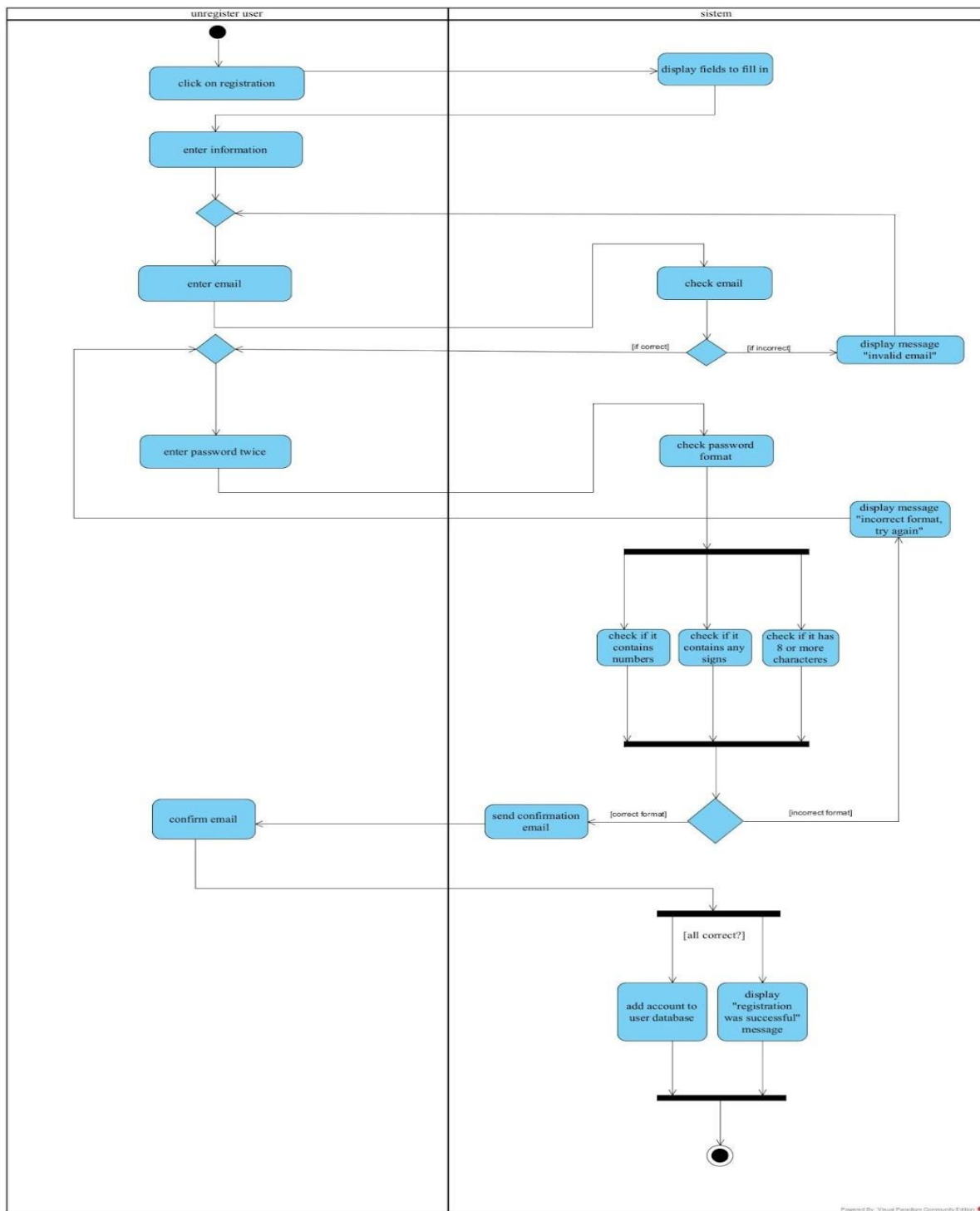


Diagram 1: Use Case Diagram

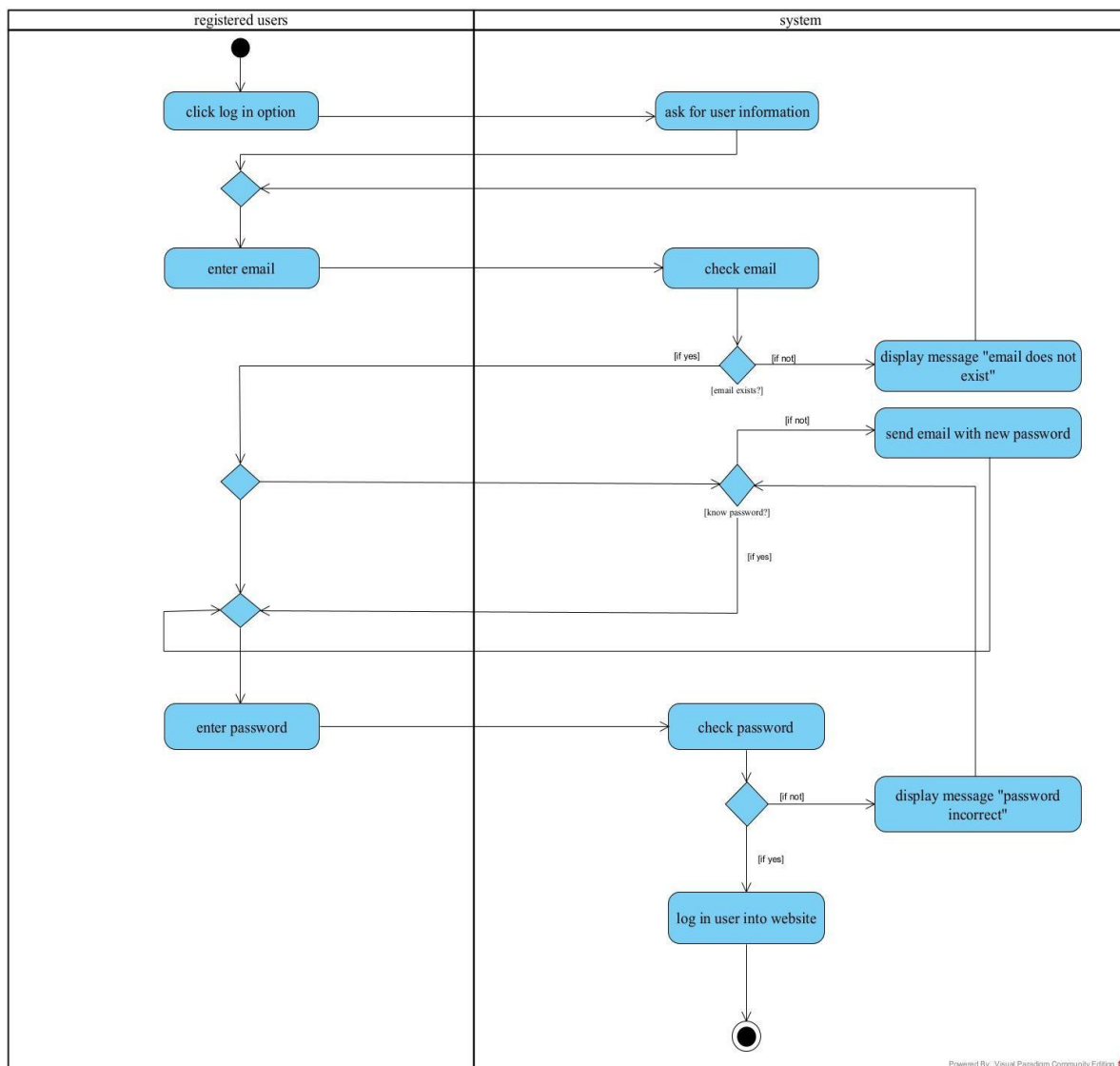
## Activity Diagrams



Activity diagram 1

When users decide to register to our web page system is first going to display fields that include entering users name, surname, age, etc. after which user is going to insert email

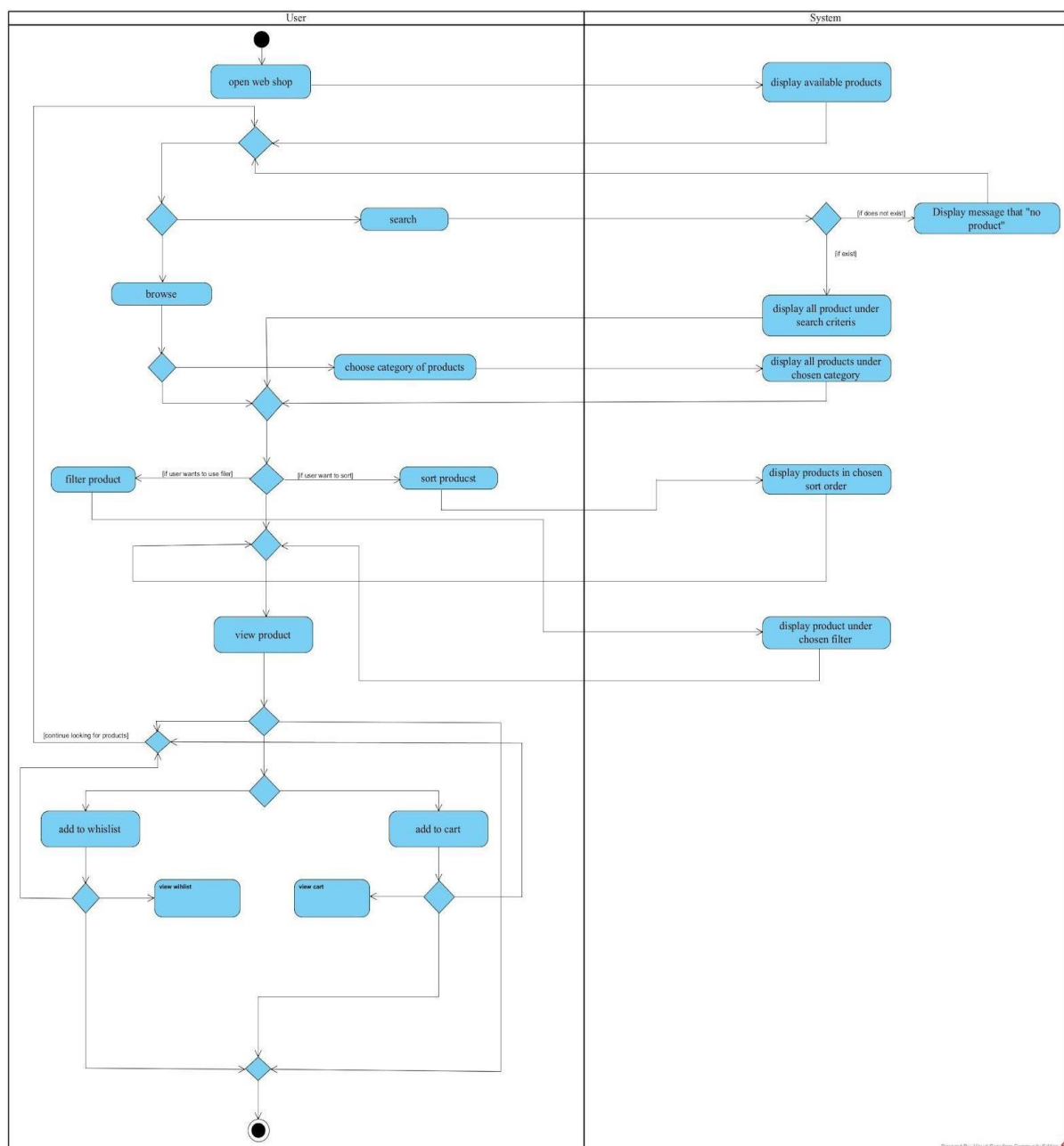
address, that needs to be checked by the system whether that email address exists or not (whether it is correct or not). Next step is password, our website requires users to enter their chosen password twice, just to confirm it, and it is supposed to contain at least one number, one sign and it is supposed to have at least eight characters. After all of this is correct the system sends a so-called confirmation email which the user is supposed to use to confirm making this account. After the confirmation system just adds an account to the database. If any of the required information is not correct, the system returns the user to the incorrect field until the user enters valid and correct information (email and password).



Activity diagram 2

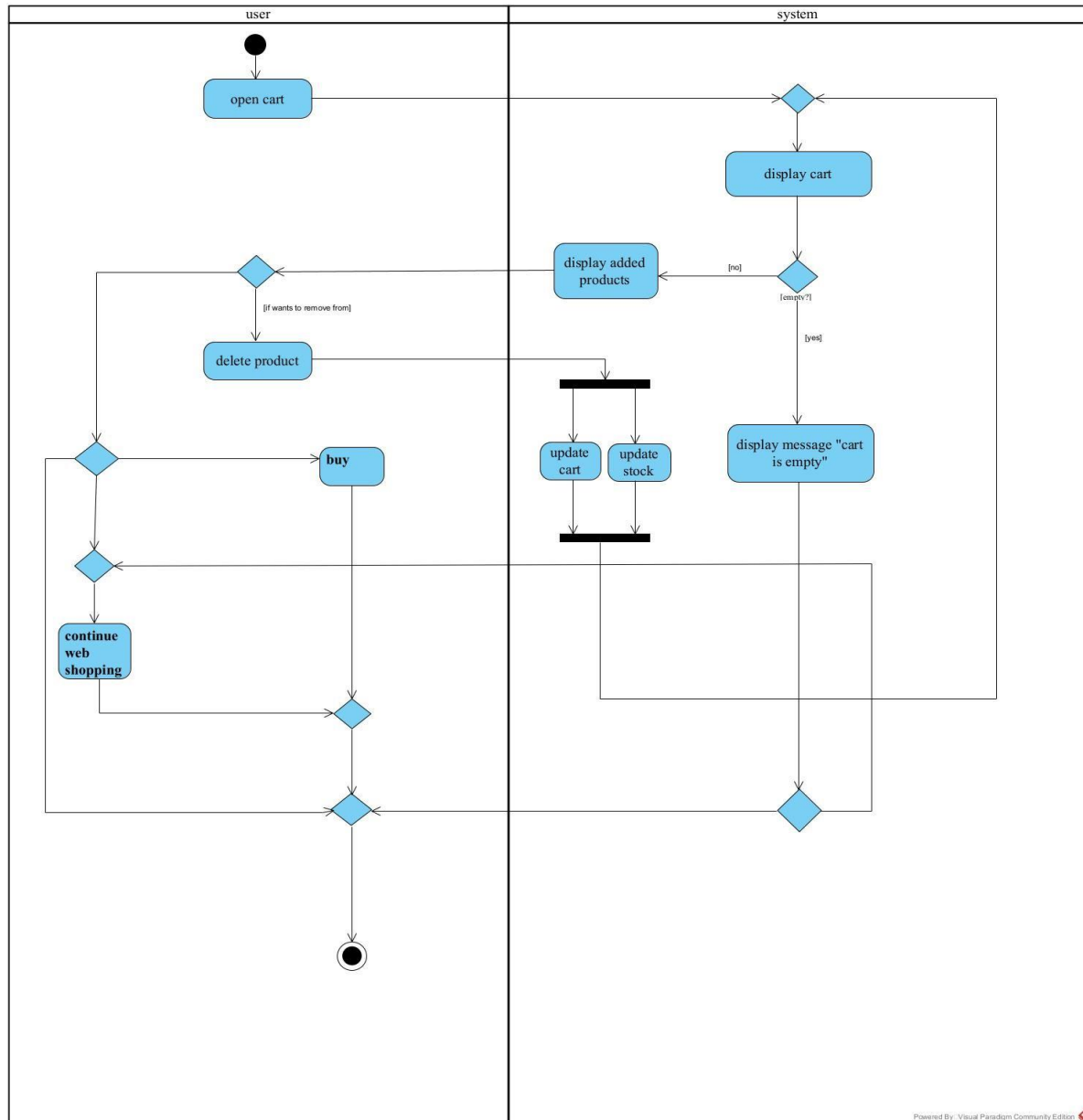
Log in is only for registered users and employers. The principle is the same, except that users must register by themselves, and employers get registered by the admin of the website and get their login information from admin. After choosing the login requirement, the user is asked by the system for their email and password. User first enters the email address which

is then needed to be checked by the system whether it exists or not, if it doesn't, the system returns user to the field where it is required to enter it again until it is correct, after the email entered is correct it transfers to entering password, but before entering password user is asked whether it has forgotten the password or if it wants to go directly to entering it. If the user says that he had forgotten the password, the system sends a new one to the previously entered email which the user then enters. After action, the "enter password", system checks whether the entered password is correct or not, if not it goes back to the if node asking the user whether he knows the password or not, previously explained. If the password is correct, the system logs user into the website.



Activity diagram 3

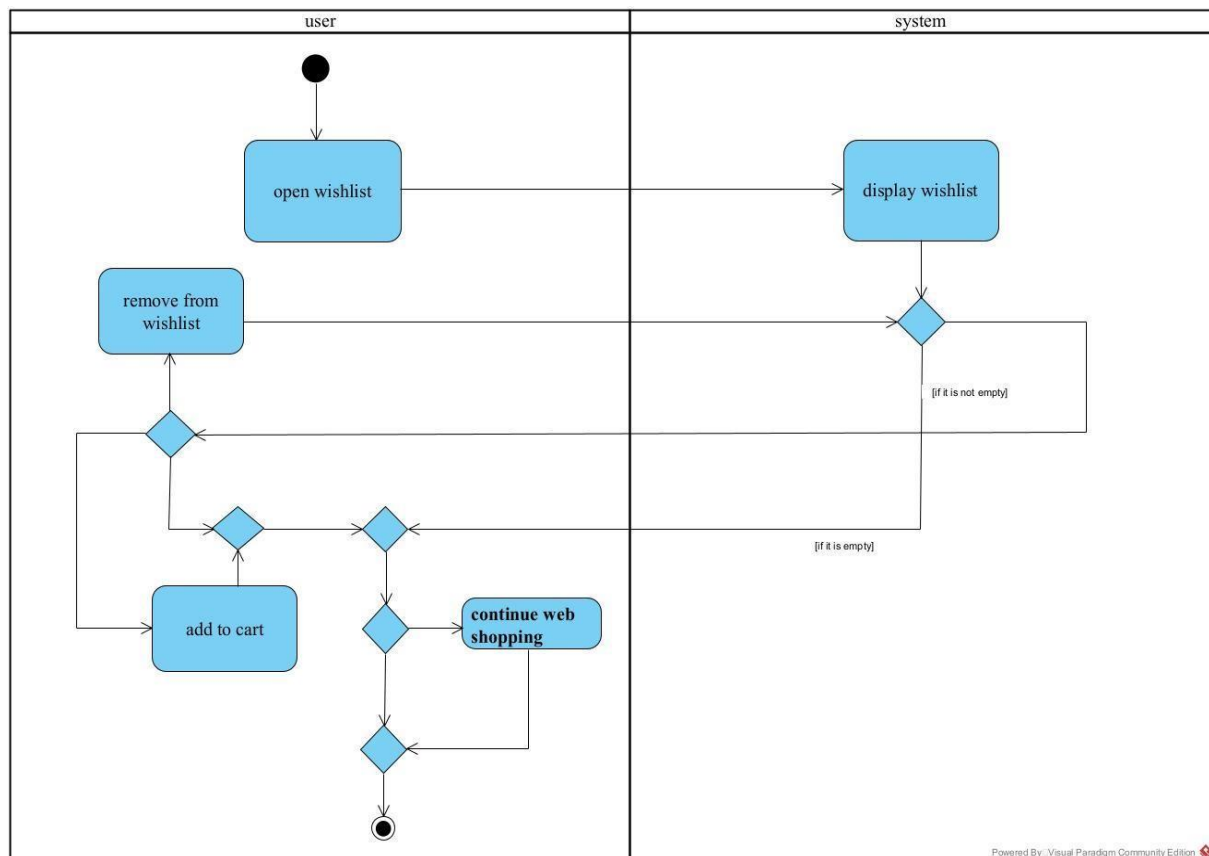
Functional requirement web shopping starts after the user opens the website and chooses the option “web shop”. What the system does after the user enters the web shop is display all the available products after which the user is provided with two first actions, the user can either browse (scroll through the pages which contain all the products) or search for one specific product or group of, entered word related, products. What can also happen after searching is that the product(s) under entered word does not exist and the system displays message “no product”, after which the user is returned to choose again if it wants to search something again, or it can just continue browsing. But if the product(s) did exist under search criteria, the system displayed all related products. If we had chosen the option browse, we would have the option to choose the category of the products (example: body products, face creams,..), after which the system would have also displayed all available products, but under that category, but if we had skipped the “choose category” action, the system would stay at the initial display of all products. So displayed products under search criteria, displayed products under chosen category and initial display of all products available at the website meet at the one merge node (because the same next actions can be used at all of those “different” displays. Those displayed products then can be filtered (example: by price from 1KM do 18 KM) and sorted (example: by price from the most expensive to the least expensive). After choosing the filtering or sorting system displayed products by the chosen filter or by chosen sorting order. But users can decide to do neither, and they all again come back to one merge node, which leads us to action “view product”. View product represents choosing any single product from offered displayed mentioned before. After opening any single product, we have few options what to do from it. We can go back to the beginning and search/browse products more, we can finish our action directly there or we can add that product to either Wishlist or Cart. After adding the product to the Wishlist, we get the option “view Wishlist”, which is activity, which if chosen leads us directly to the “View Wishlist” diagram. Also, after adding the product to the cart, we get the option “View Cart”, which is also activity, which if chosen leads us directly to the “View Cart” diagram, and our action in web shopping diagram is over. But another option that is given either after adding product to the Wishlist or cart is to go back to the beginning to continue searching/browsing. But also, we can choose neither, we can just finish our activity after adding a product to Wishlist/Cart.



Activity diagram 4

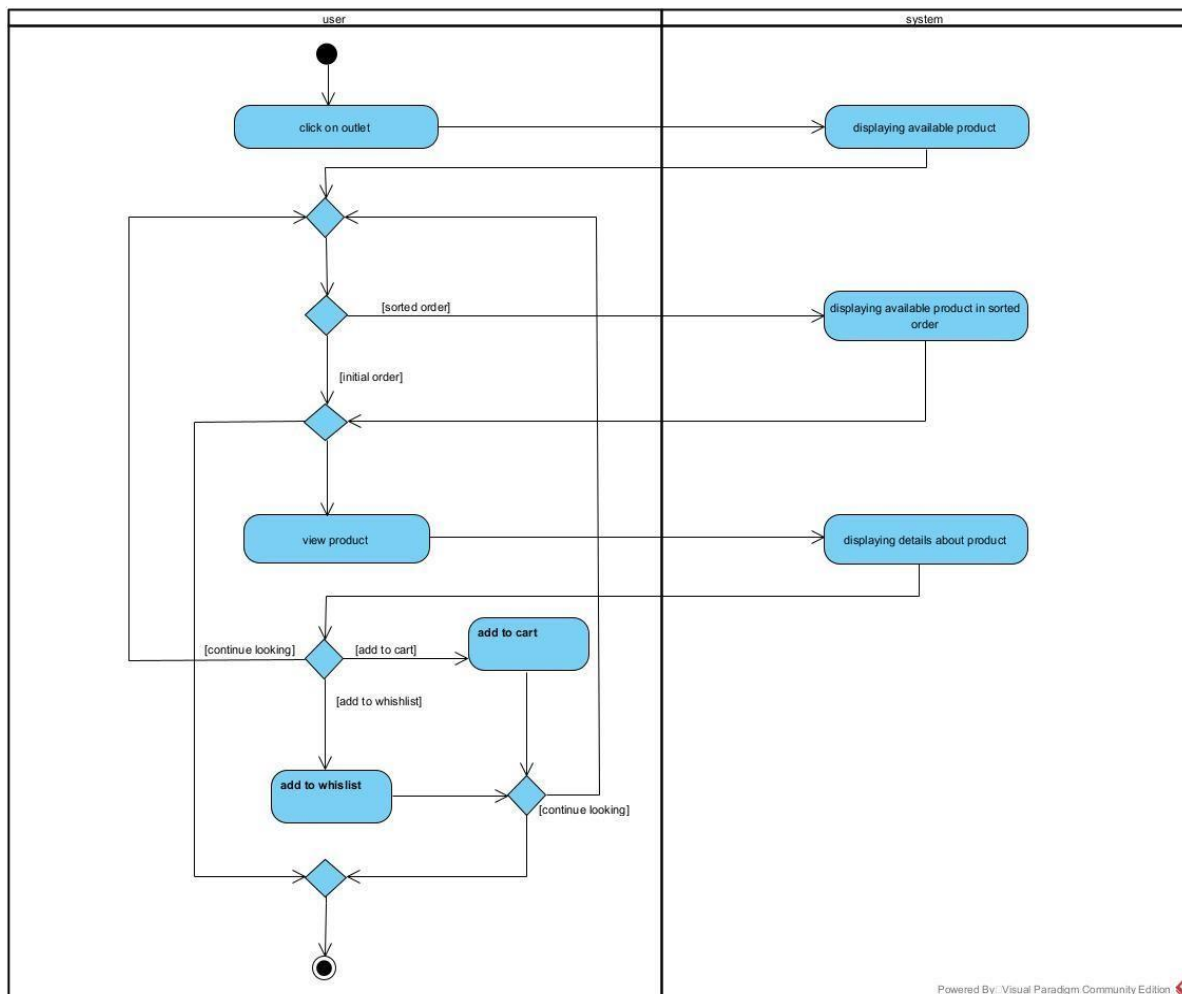
Requirement "View Cart": after opening the cart, the system displays the cart with all its information. From there have a node that means that the system checks whether it is empty or not. If it is empty, the system displays the message "Cart is empty", and from there we can choose to either continue shopping or to finish activity completely. But if the cart is not empty, system displayed products we added to it and we have two allowed actions, to either delete some or all products from cart, after action deleting, the system has to update cart (price and number of products) and stock (products available in the web shop), after that the system has to checks whether the cart is again empty or not, if it is empty, it does the

actions I explained above, but if it is not empty, cart again displays products cart contains. Again, we have the option to either delete more or continue, if we choose to continue, the next node gives us three options, to either buy, finish our activity there or to go back to web shopping. If we choose the option to buy (which we have defined as activity), we finish our activity at the cart, and go to the buying activity diagram. If we choose the activity to “Continue web shopping” we also finish our activity at the cart.



Activity diagram 5

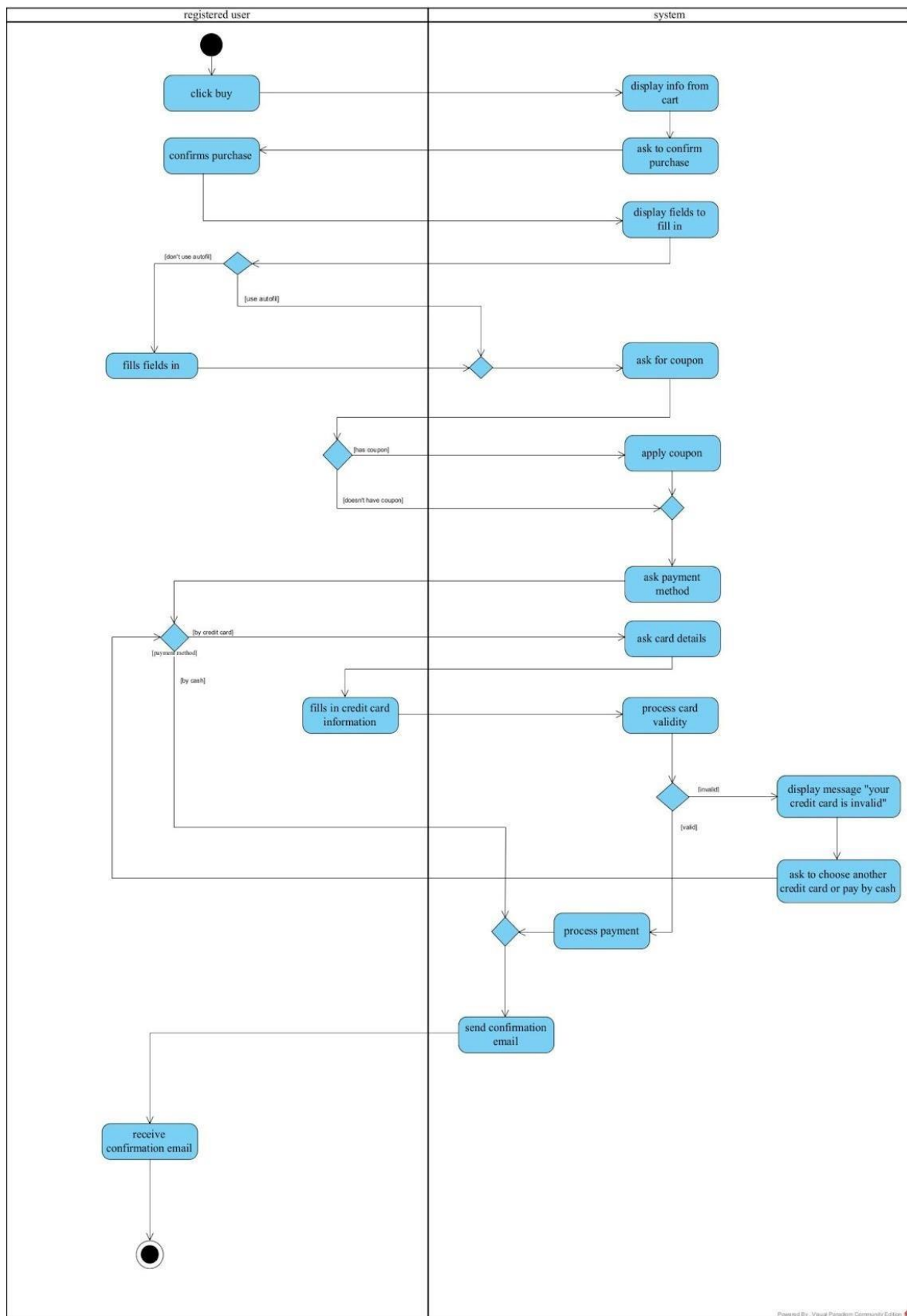
Activity view Wishlist starts after we open Wishlist, next action that happens is that the system displays Wishlist with its allowed actions and added products. System checks whether the Wishlist is empty or not, if it is, the user has the activity option to either continue web shopping or to just finish their activity there. But if it is not empty the user has the option to remove products from the Wishlist, after which the system checks again if it is empty and if it is, the user has previously explained options. If it is not, the user again can choose whether to delete products from the Wishlist, to add to cart, to continue web shopping, or to just finish their activity there. After choosing the option to add to cart, the user again returns to options to either continue web shopping or to finish their activity there. But user can directly choose, after establishing that the Wishlist isn't empty, to continue web shopping or to just finish their activity there.



Activity diagram 6

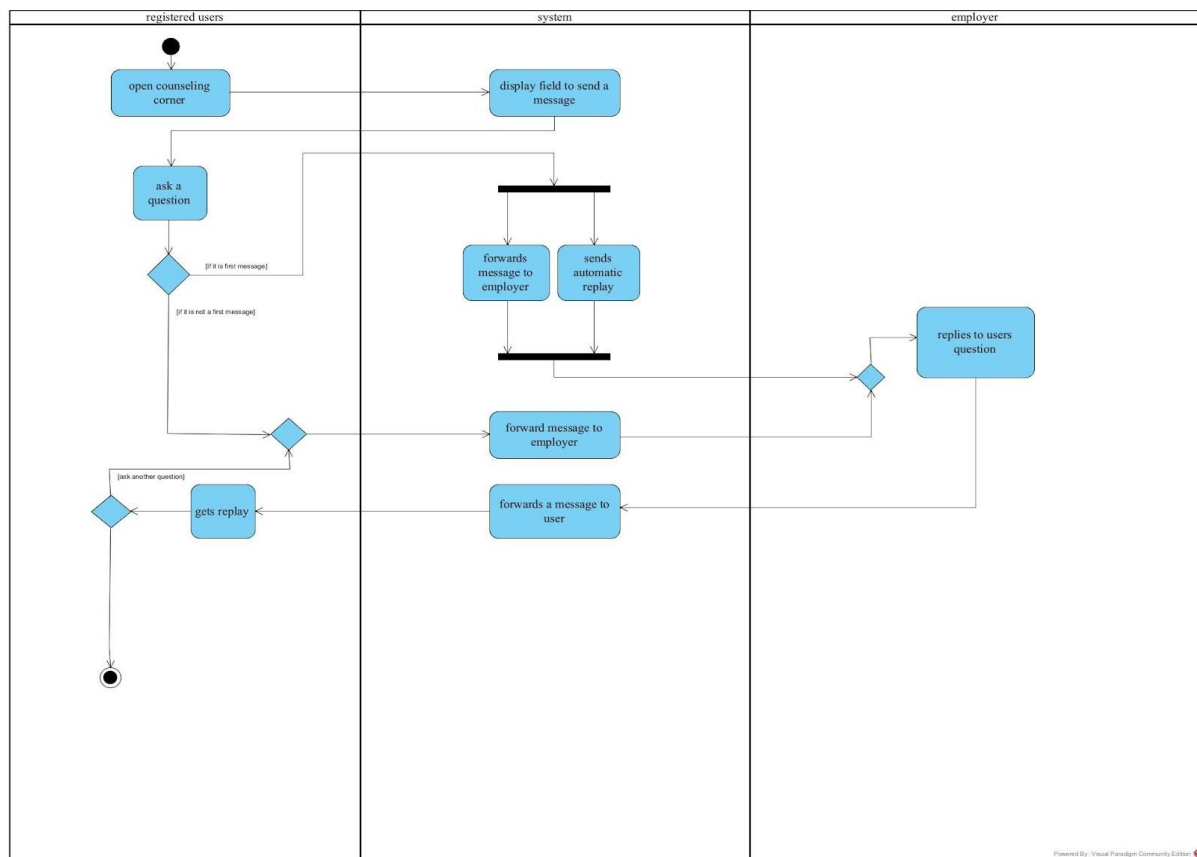
Outlet shopping requirement is very similar to web shopping, but it is simpler, because we are talking about fewer products that are soon to expire and are sold at a lower price. User opens outlet shopping activity, and the system displays available products. In outlet shopping activity users can only choose the way products are sorted, after which the system processes it and displays products the way it was chosen to be sorted, but doesn't have to, can choose to stay at the initial order. After choosing the way products are sorted and after the system displays the products, users can choose to view the product (see one single product), after choosing this action, the system displays details about the chosen product. Instead of viewing products, users could have chosen to just finish activity there. After viewing the products user can either add product to cart, add product to Wishlist (after both options, user can choose to go back at the beginning of outlet and continue looking through products or to finish activity there), or to choose to just go back to the beginning of outlet to look at products more.





Activity diagram 7

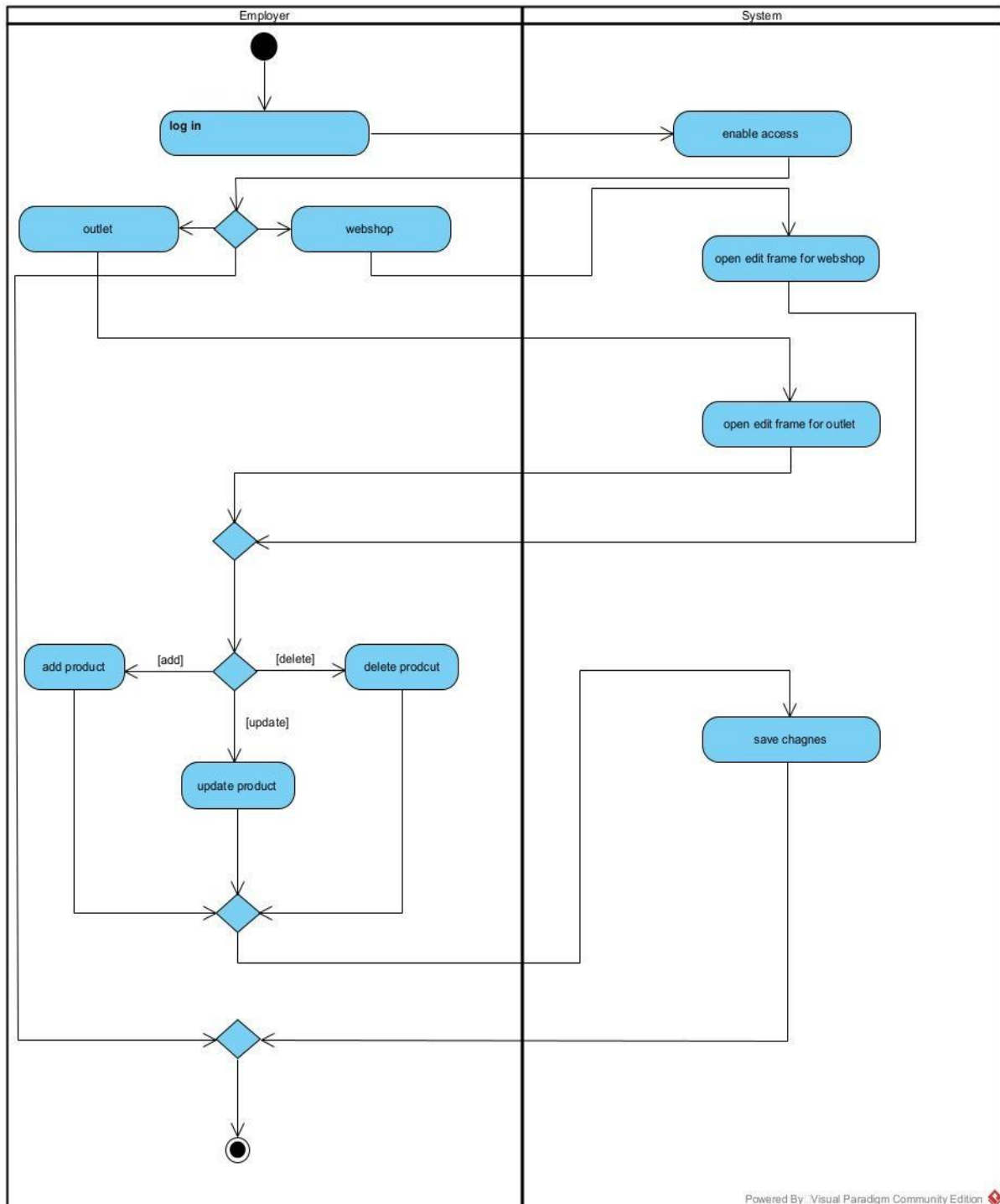
After choosing the option to buy products from the cart, the user goes to the activity “Buy”. System first displays options from the cart and asks to confirm purchase, the user confirms purchase. System then displayed fields required to fill in to buy. Users have options to autofill fields by system, or to enter a new one. These connect into a merge node, after which the system asks for coupons. If it has a coupon, the system asks to apply for it and after asking for a payment method, if the user doesn't have it, it goes directly to the system asking for a payment method. If the user chooses a credit card, it asks for credit card details, users enter details, user checks if it is a valid credit card, if it is valid, system processes payment and sends confirmation email and ends activity. If the card was invalid, the system displays a message saying it is invalid and asks again for the card or to pay by cash, if the card is again chosen, the same process as the first time happens. If payment by cash is chosen, system just sends a confirmation email and ends the activity.



Activity diagram 8

User opens a counselling corner, and the system displays fields in order to send messages. User asks a question and sends it, if it is the first one the system will forward it to the employer and reply with an automatic reply. After the employer sees a message, the employer replies, the system forwards it to the user, the user gets a reply, after which the user can either end activity there or send another message, which in this case only gets forwarder directly to the employer, because the user stayed in the counselling corner

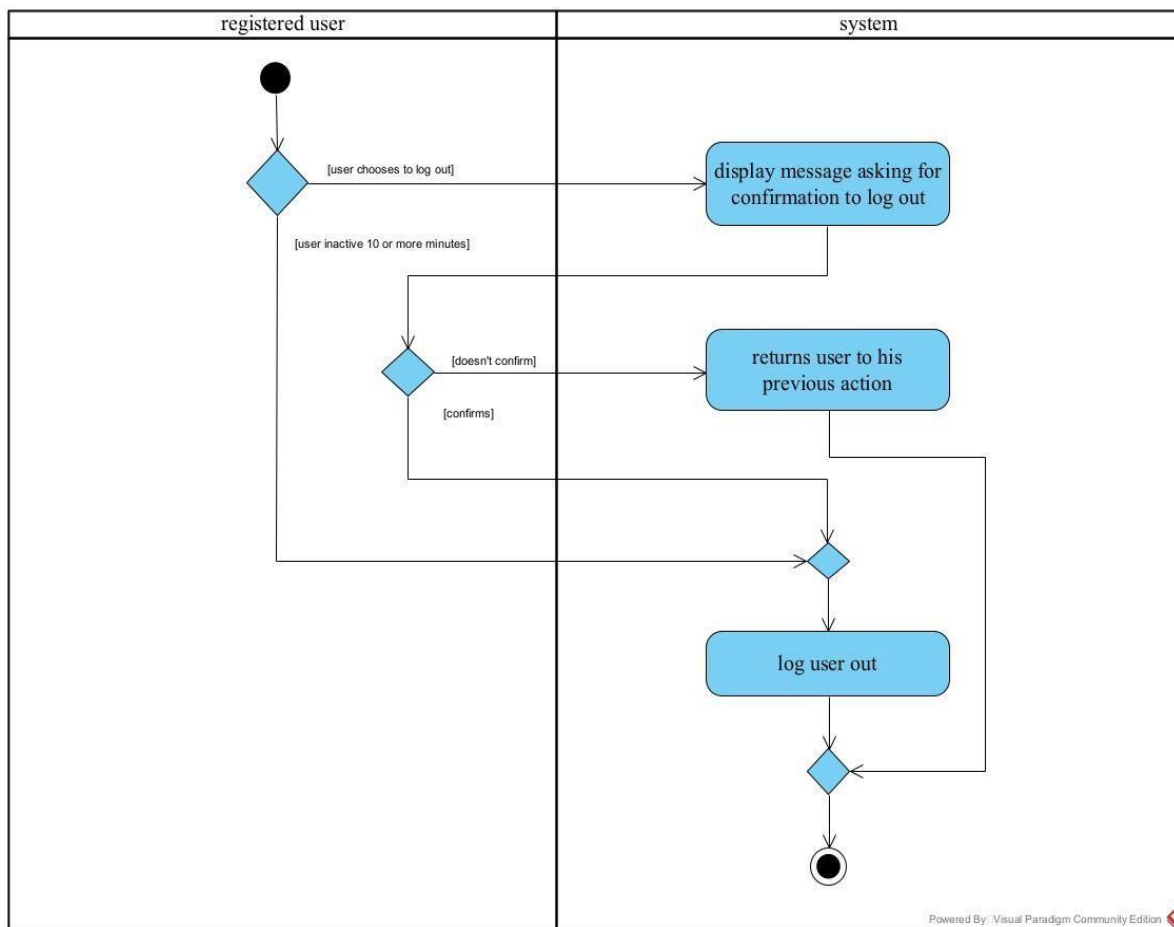
activity. If the user finishes activity, and for example tomorrow, enters the activity again, the system will check from the beginning if it is the first message or not.



Activity diagram 9

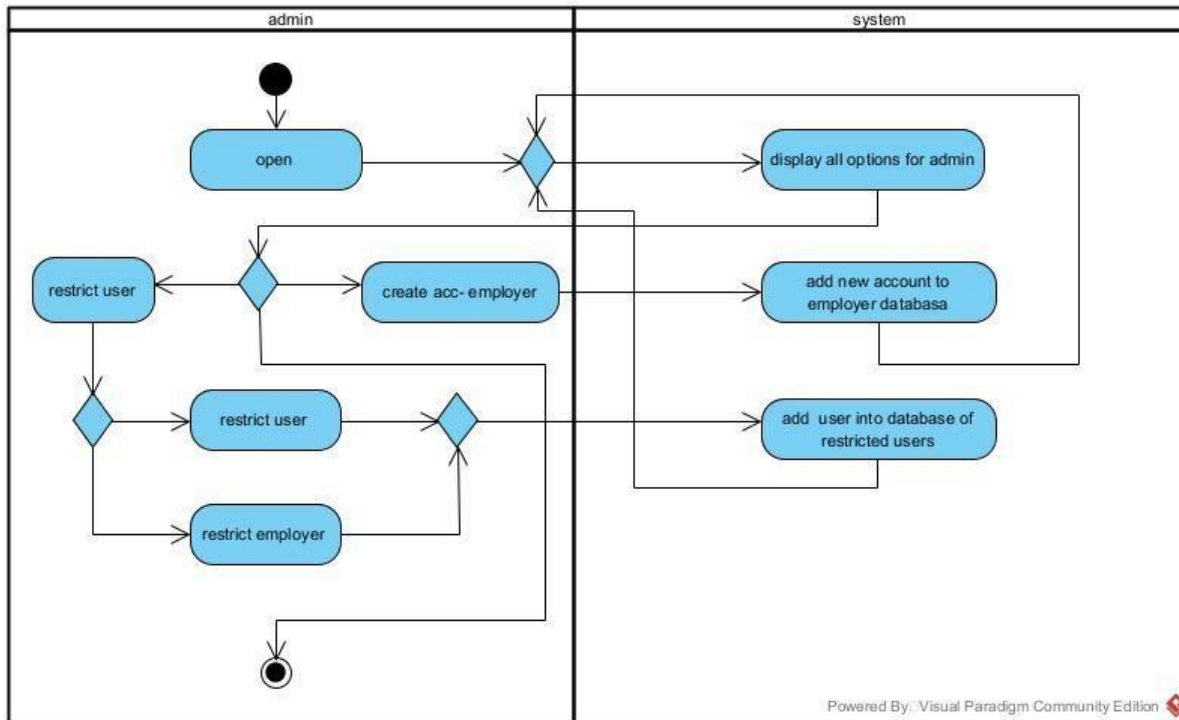
When it comes to employers' role in a web shop. Employers can either delete, add, or update products. First employer must log in with a given email and password, and the system enables that process. After an employer can choose to add, delete, or update

product from either web shop or outlet shop or can just end activity. System opens an edit frame for the one chosen, after which the employer decides which of the offered actions it wants to do. System saves changes and ends activity.



Activity diagram 10

Activity Log out starts after the user clicks on the option to log out or the user was inactive for 10 or more minutes. If the user chooses to log out, the system will ask for confirmation, if the user does not confirm, the system will return the user to his previous action and finish log out activity. If the user confirms, the system will log out the user and finish log out activity. And if the user was inactive for 10 or more minutes, the system will automatically log out the user and finish log out activity.



Activity diagram 11

Admin's role is to define users of the system. Admin opens the web page, and the system displays options that belong to admin. After which admin can choose to either create an account for the employer or to restrict some users, or just finish activity there. If the admin wants to restrict users, admin can choose between users (meaning regular users/customers) or employers. Admin chooses the users he wants to be restricted, and system adds them to the database of the restricted users and goes back if the admin wants to choose another option. If the admin decides to create an account for the employer, the system just adds that account to the employer database, and again returns to the beginning so the admin can choose another option or finish the activity.

## Class Diagram

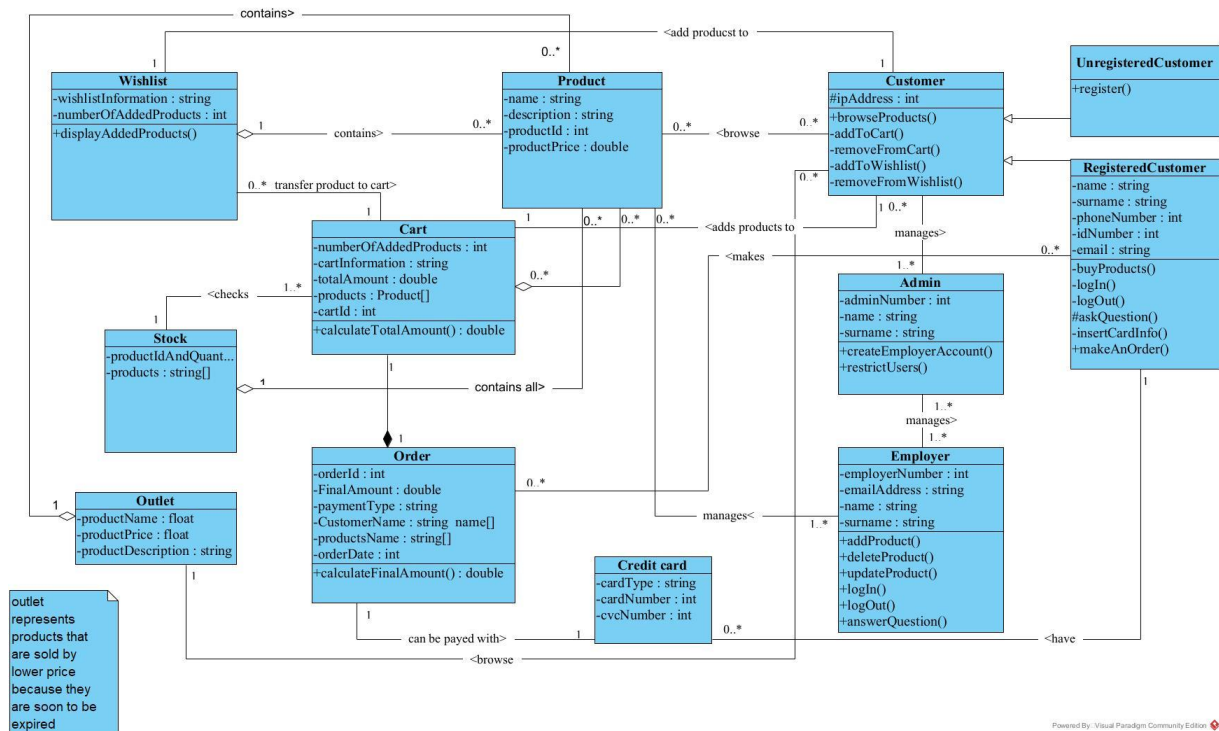


Diagram 1 Class diagram

## Object diagram

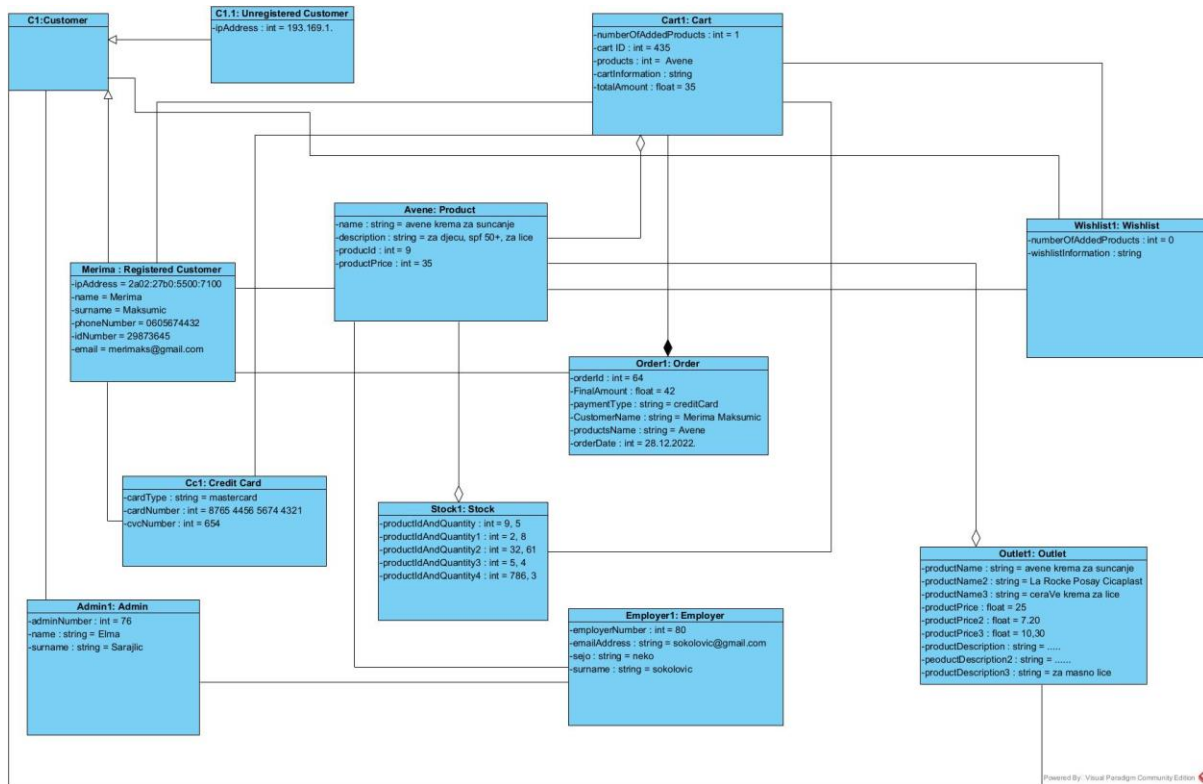
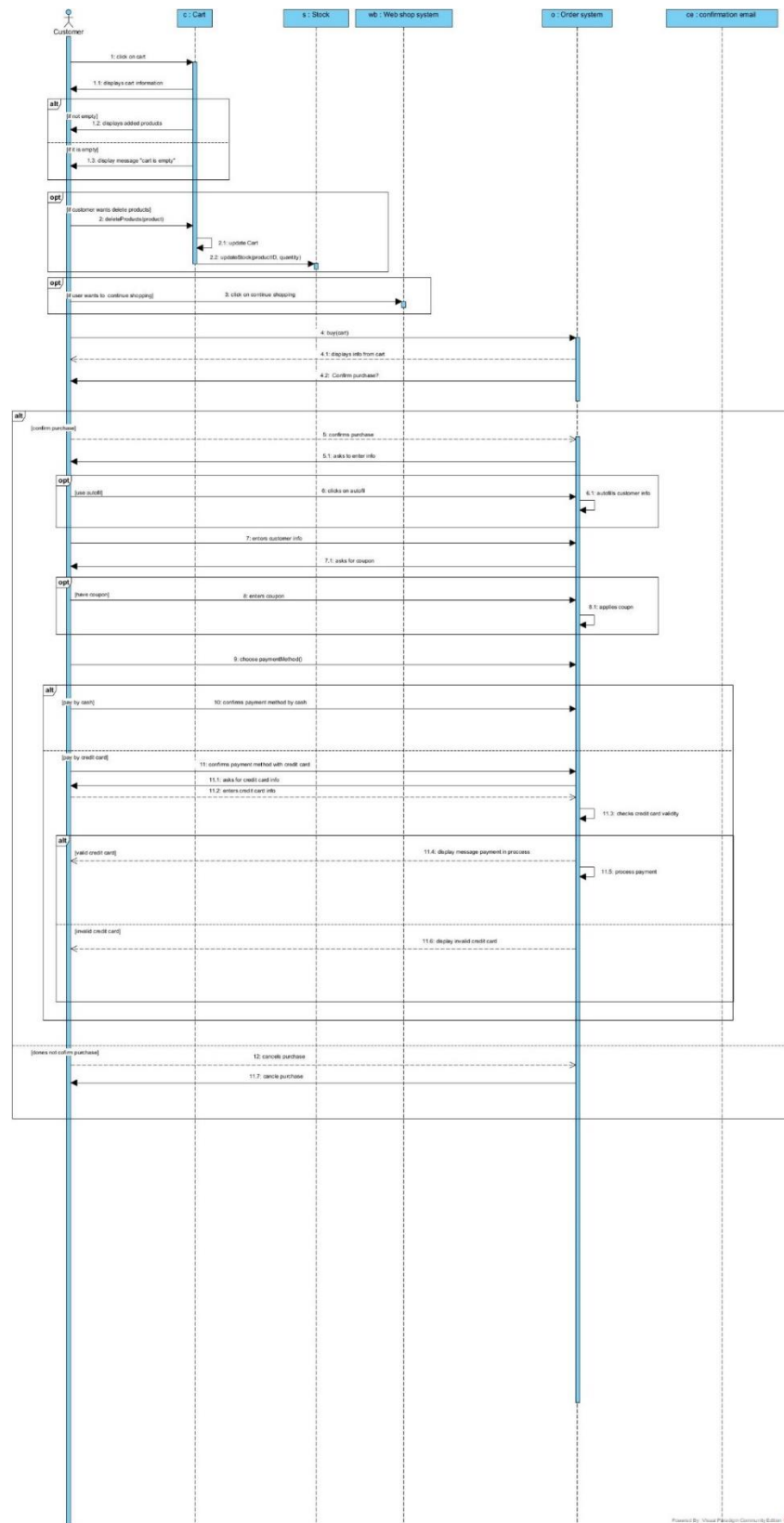


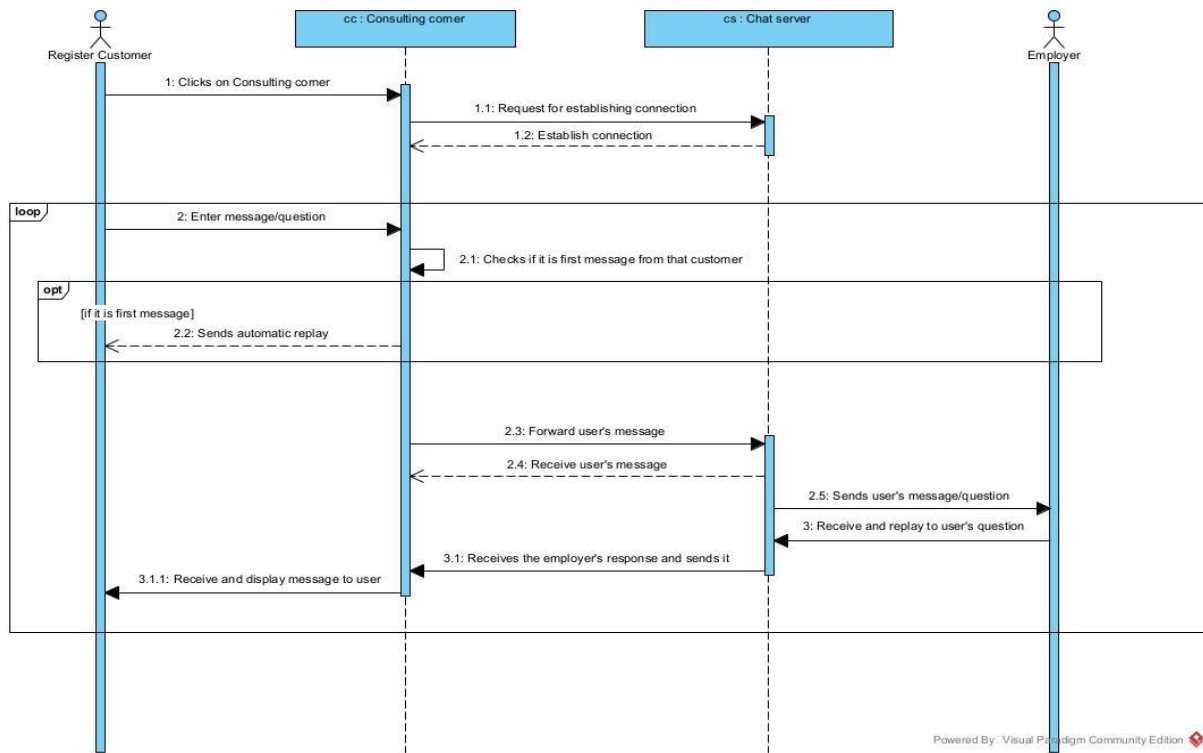
Diagram 1 Object diagram

## Sequence diagrams

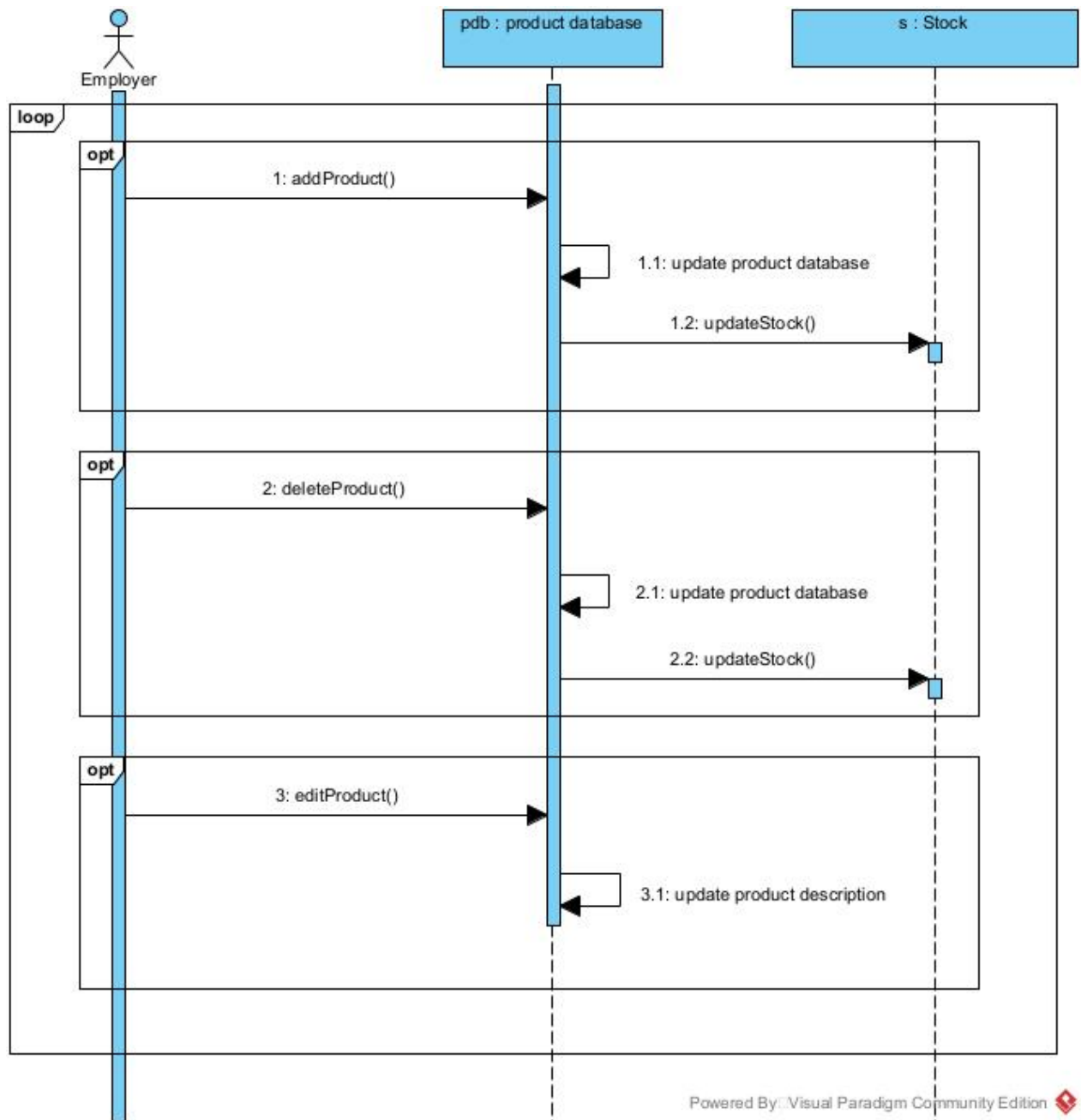


Sequence diagram 1



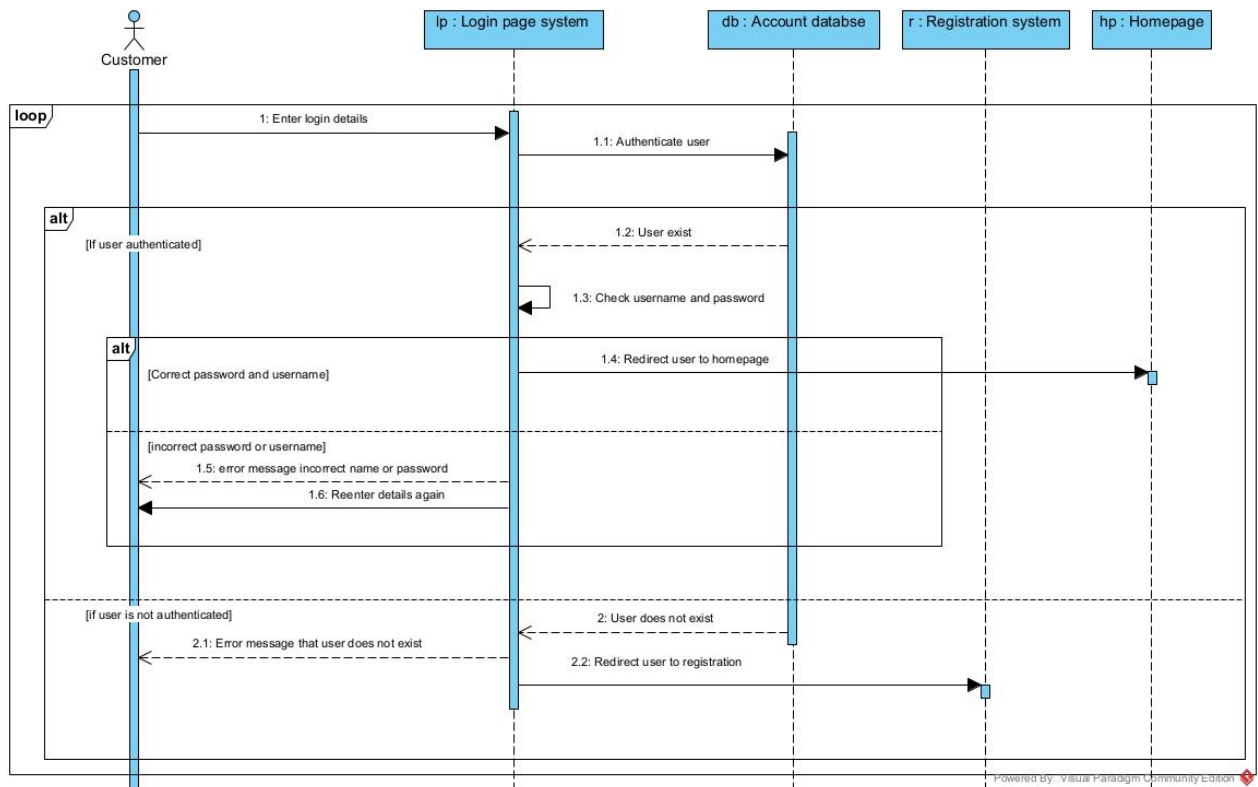


Sequence diagram 2

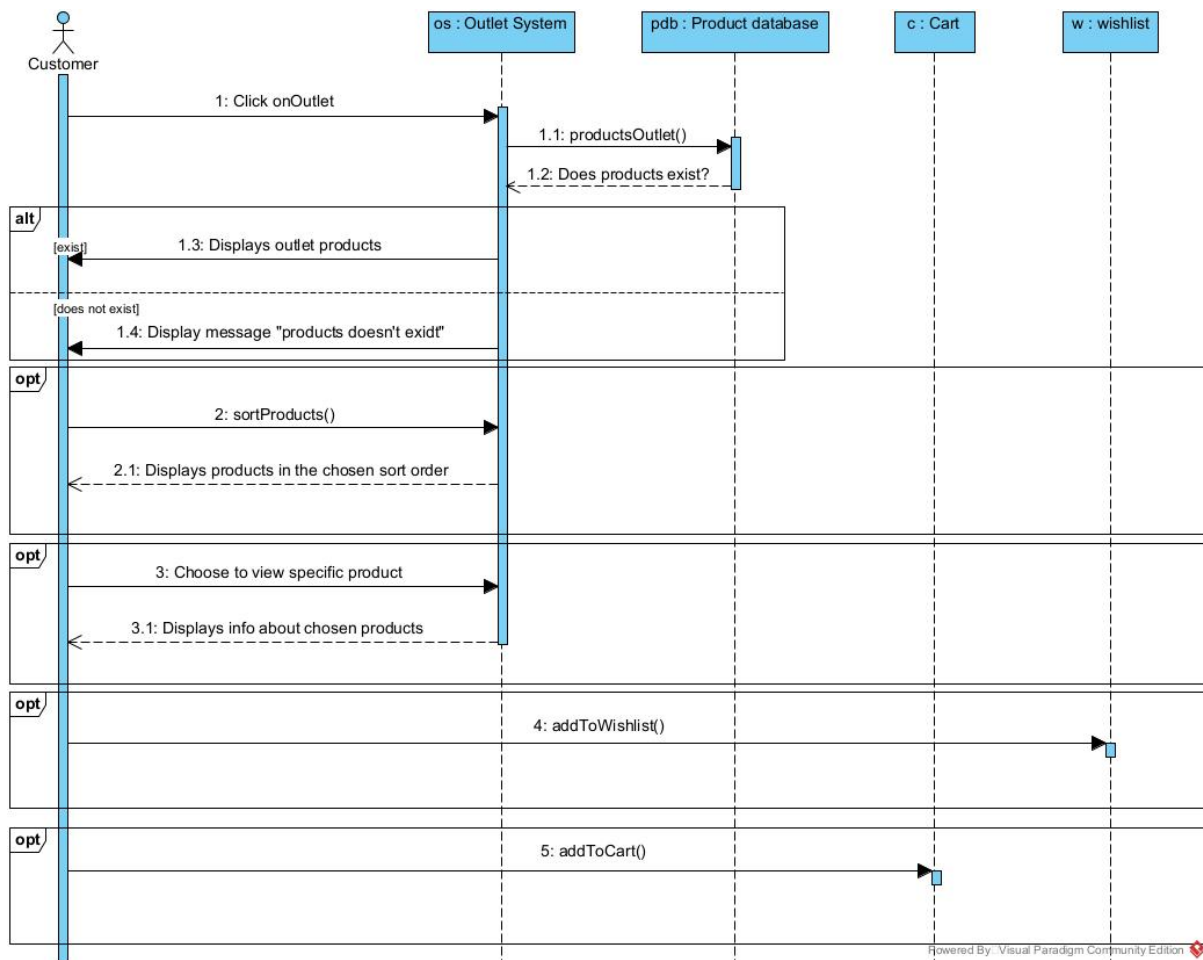


Powered By: Visual Paradigm Community Edition

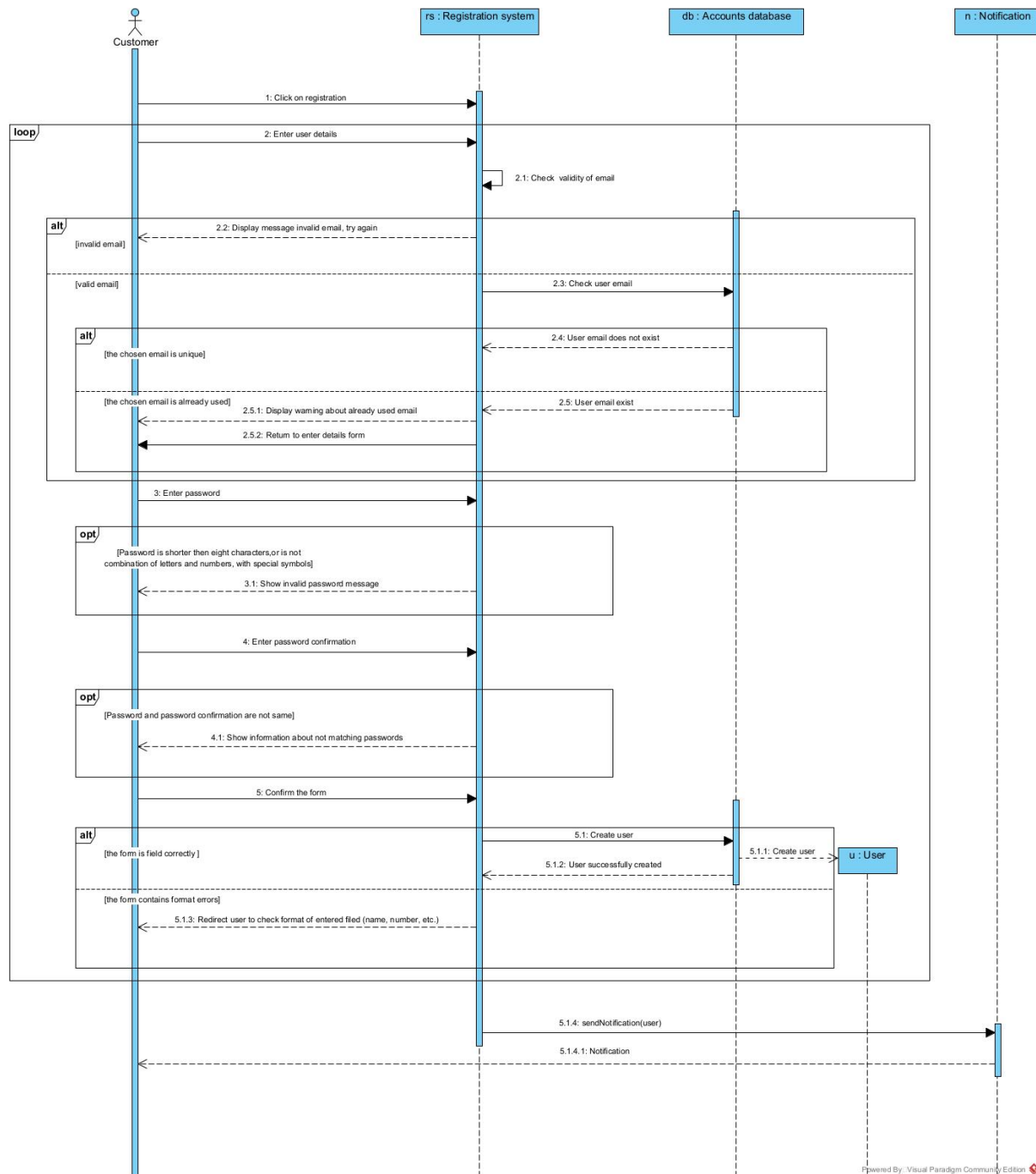
Sequence diagram 3



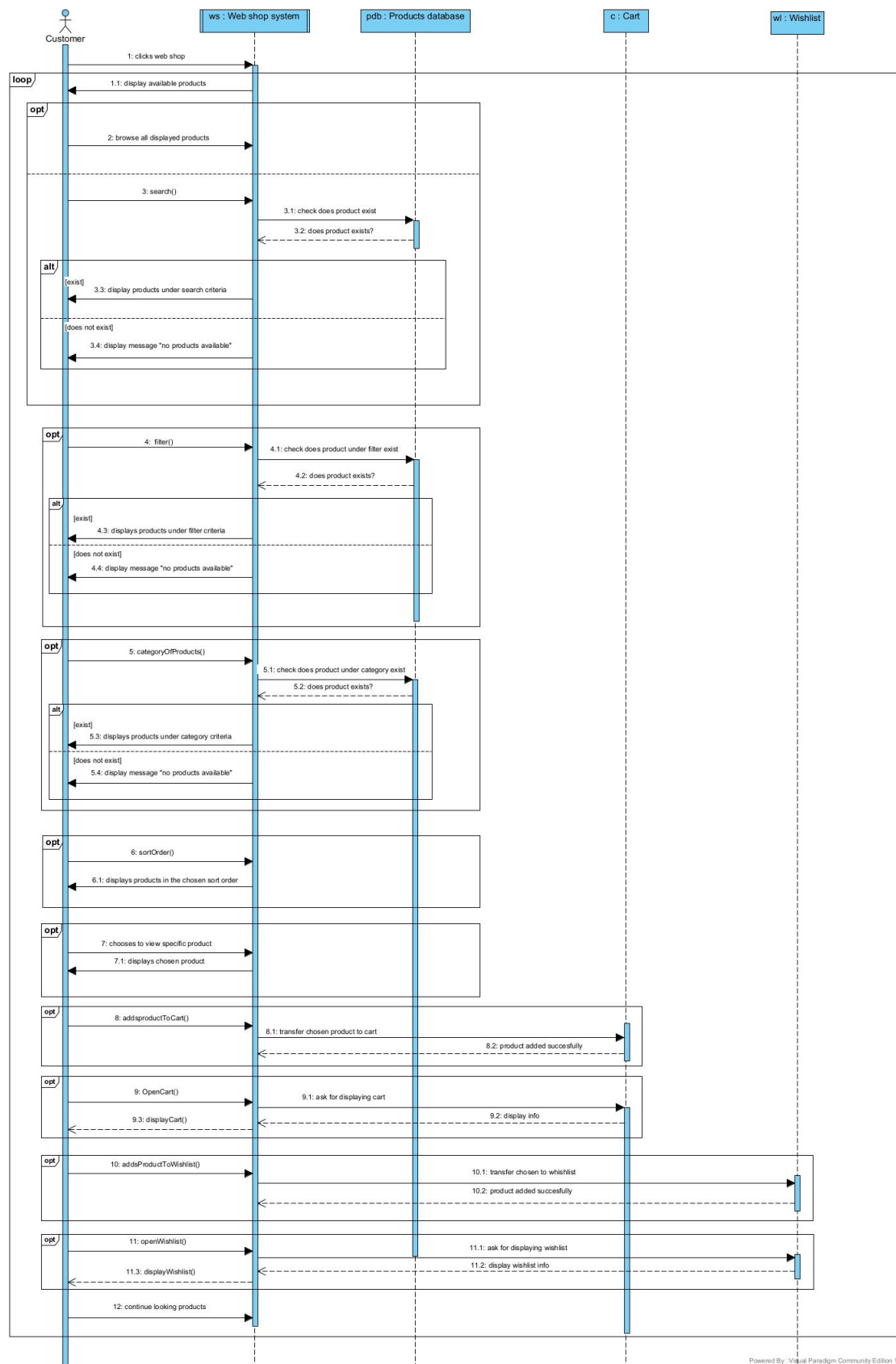
Sequence diagram 4



Sequence diagram 5

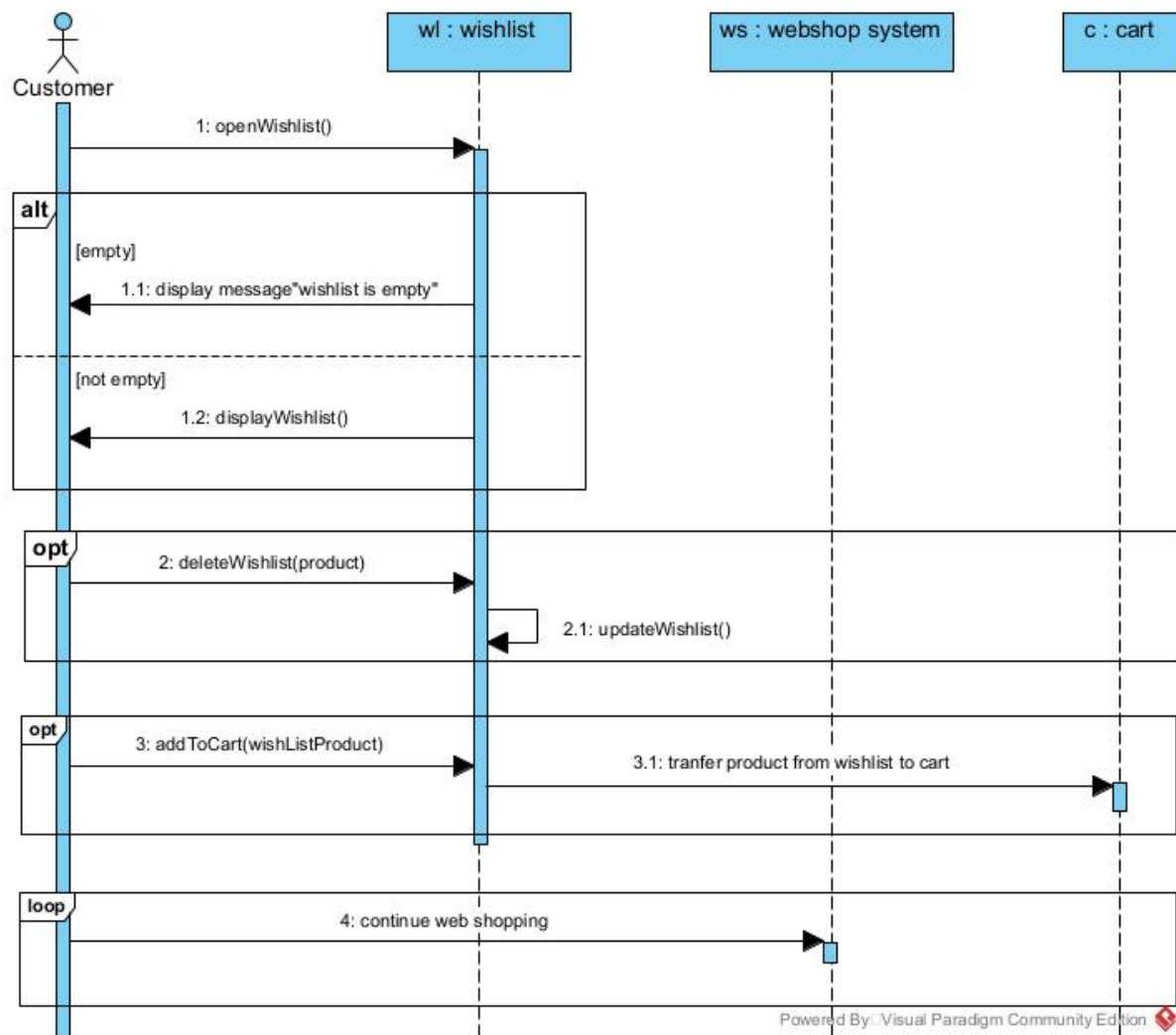


Sequence diagram 6



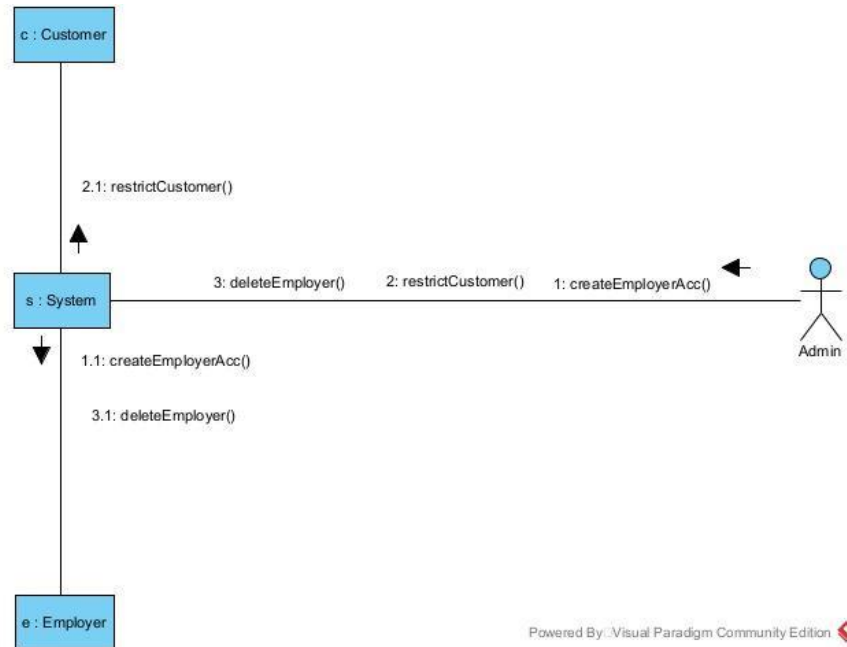
Powered By: Visual Paradigm Community Edition

Sequence diagram 7

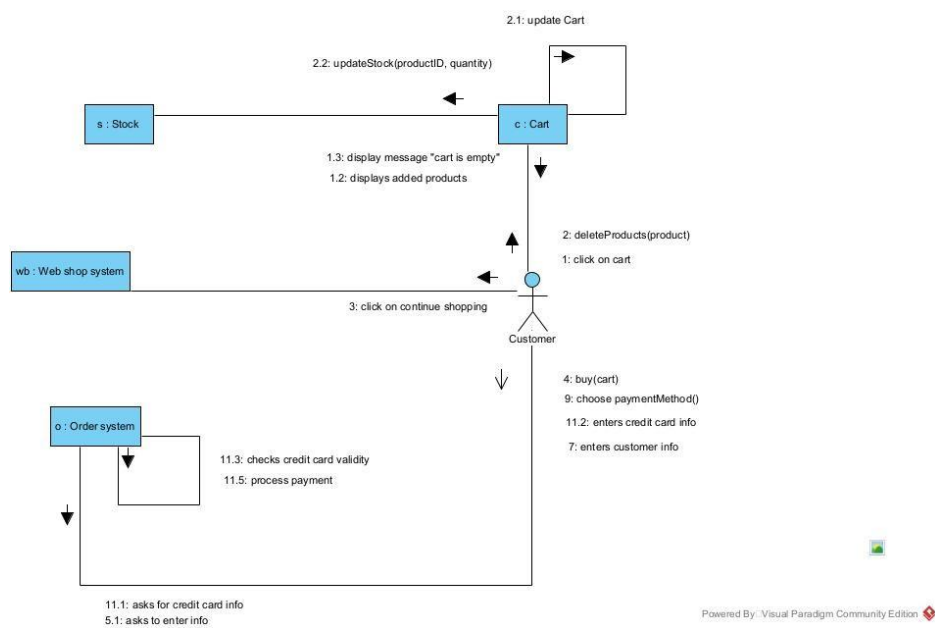


Sequence diagram 9

## Communication diagrams

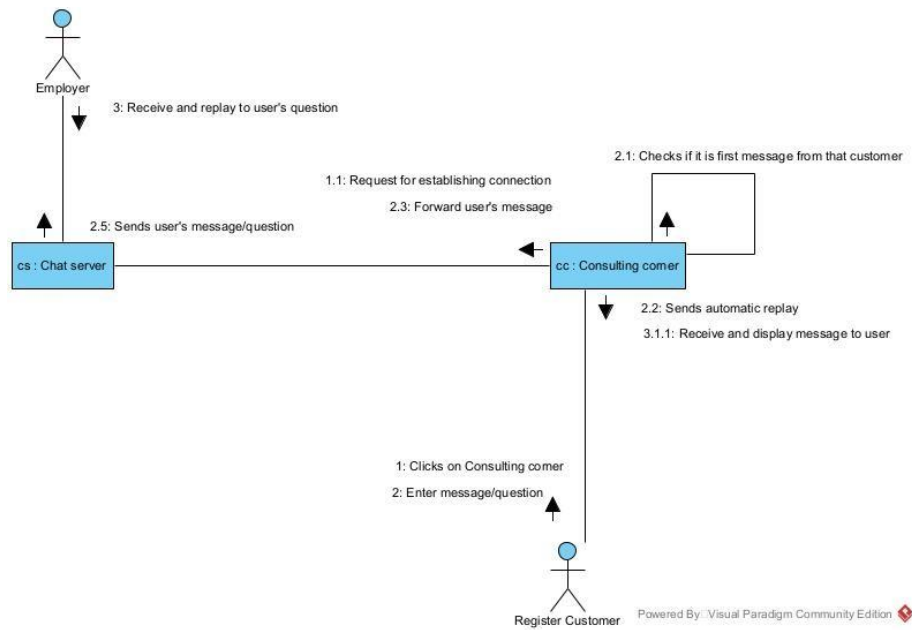


Communication diagram 1

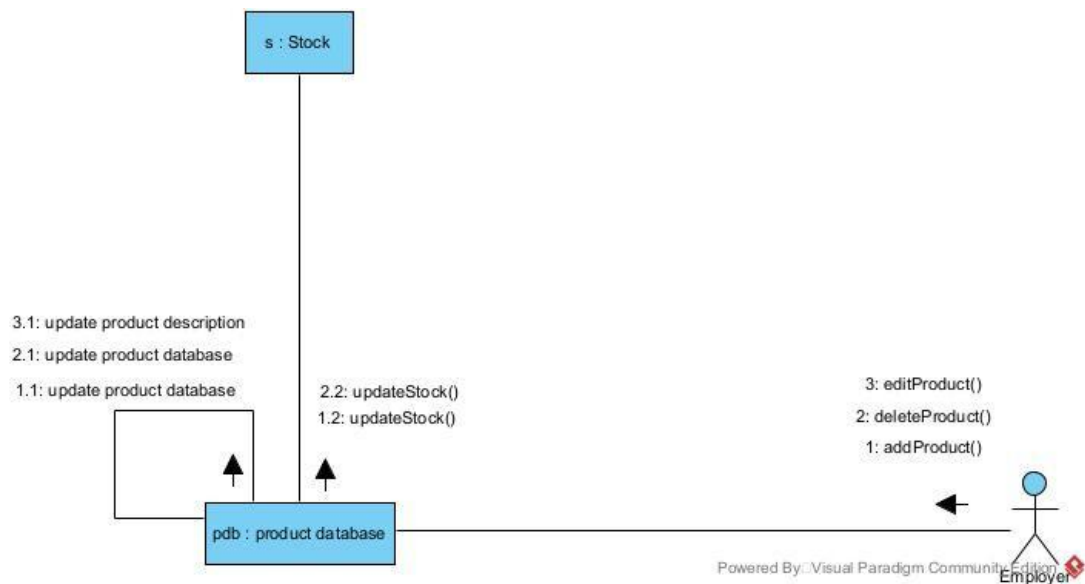


Communication diagram 2

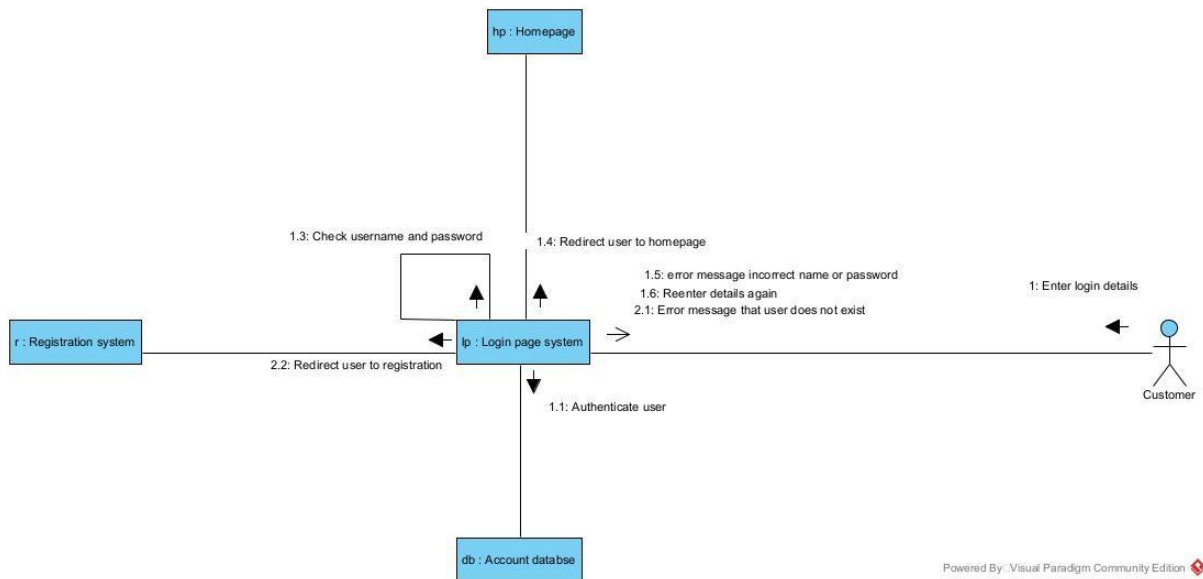




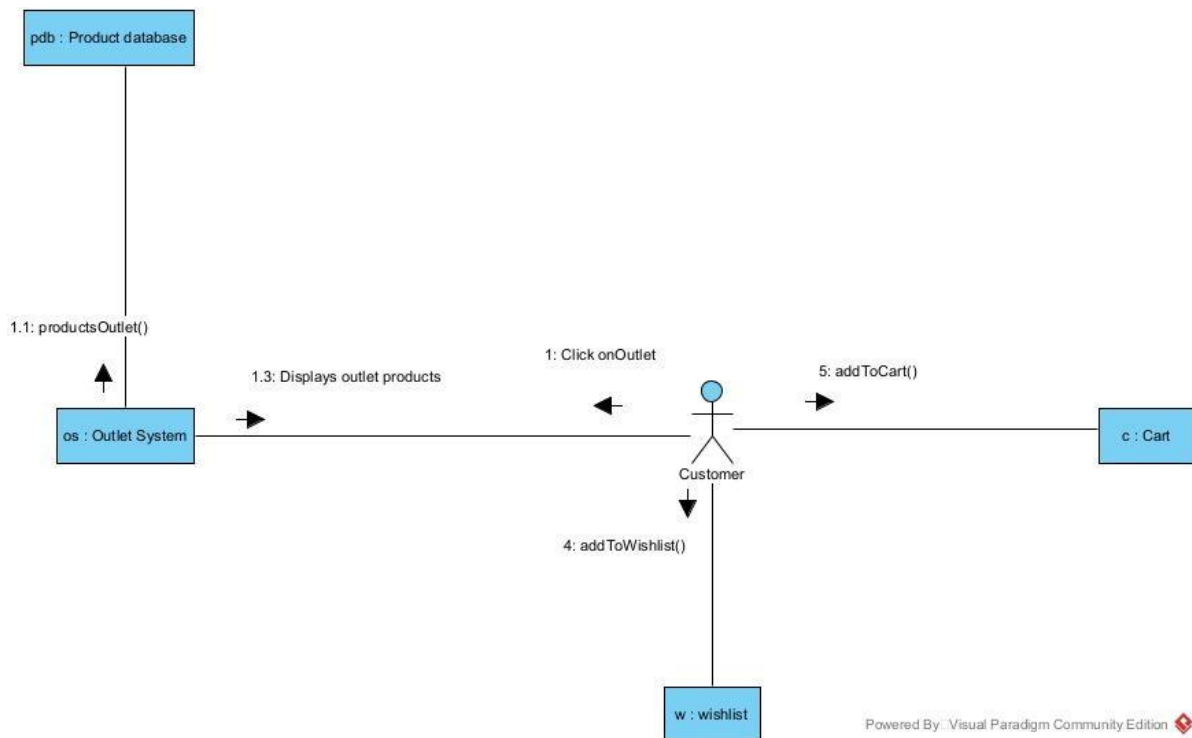
Communication diagram 3



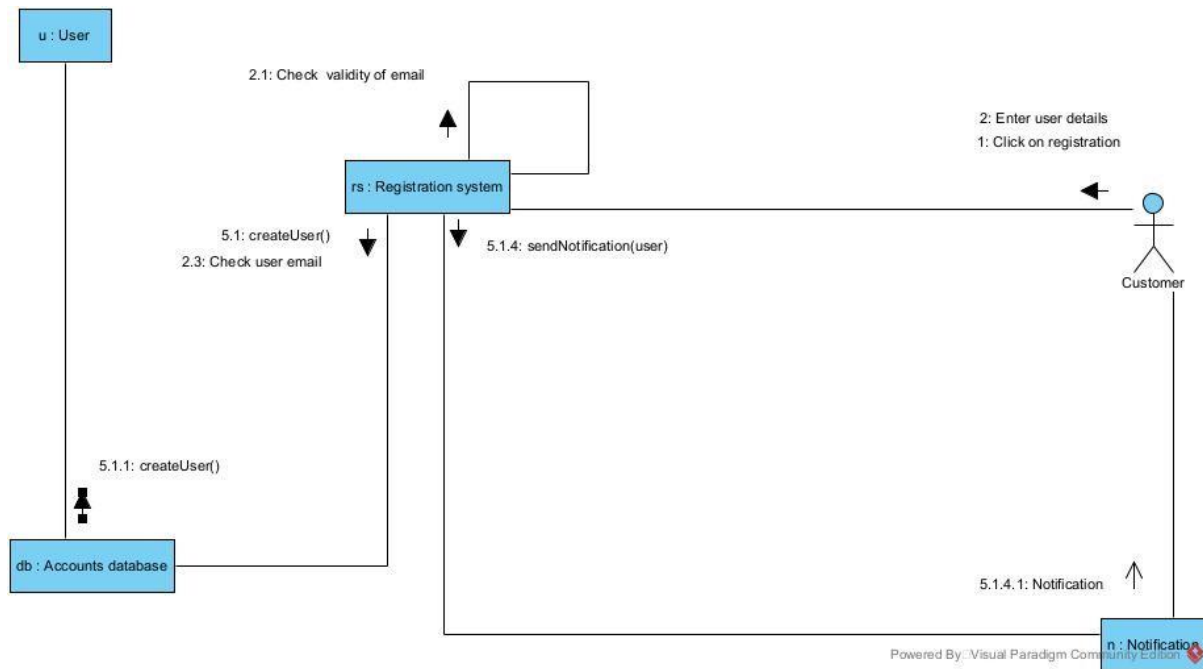
Communication diagram 4



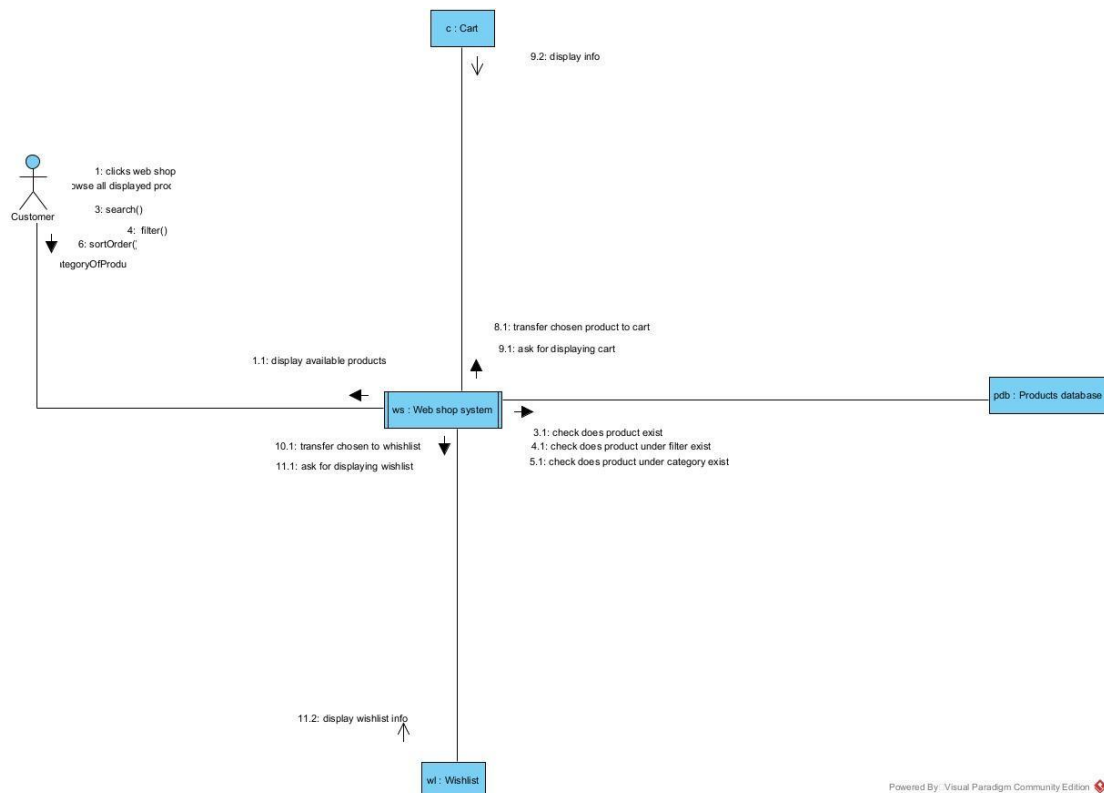
Communication diagram 5



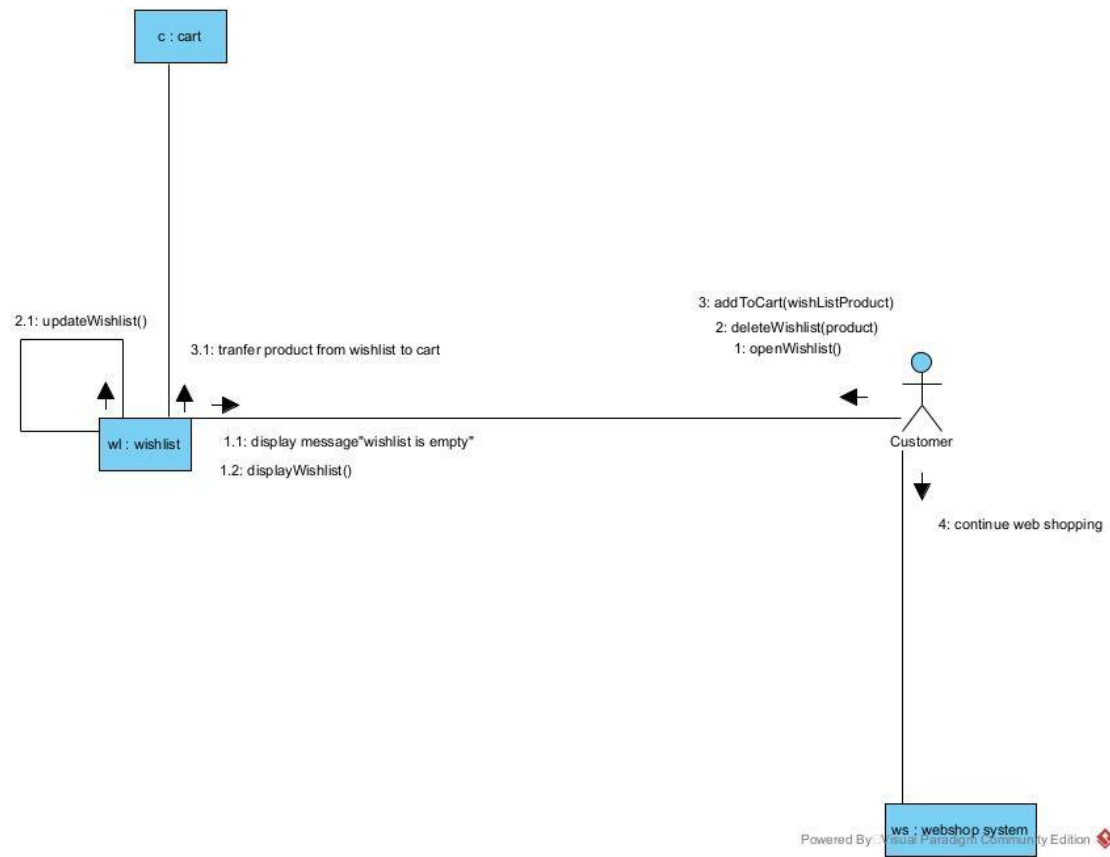
Communication diagram 6



Communication diagram 7

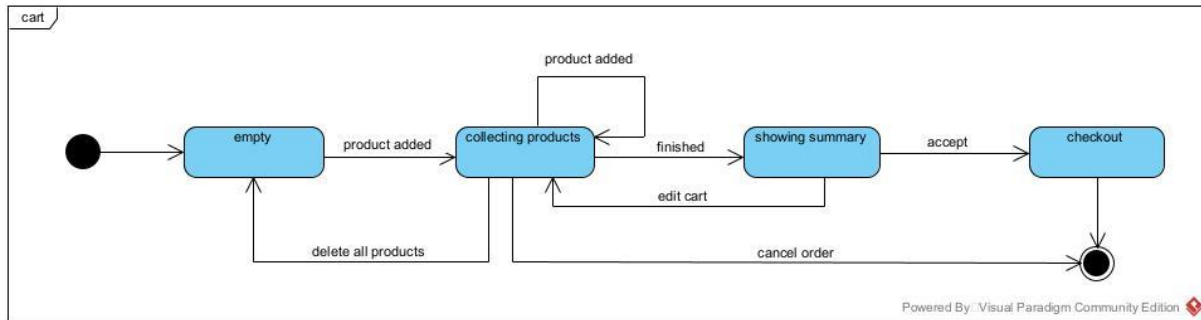


Communication diagram 8

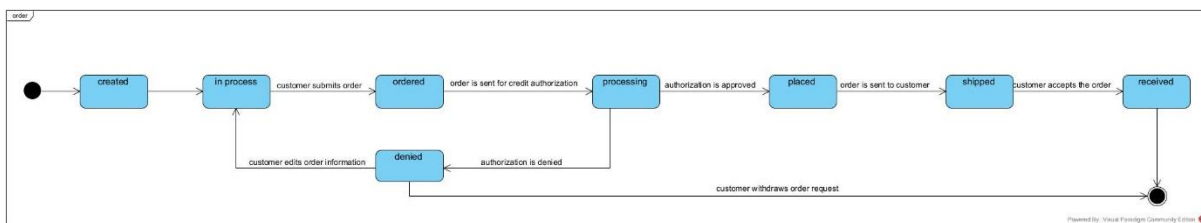


Communication diagram 9

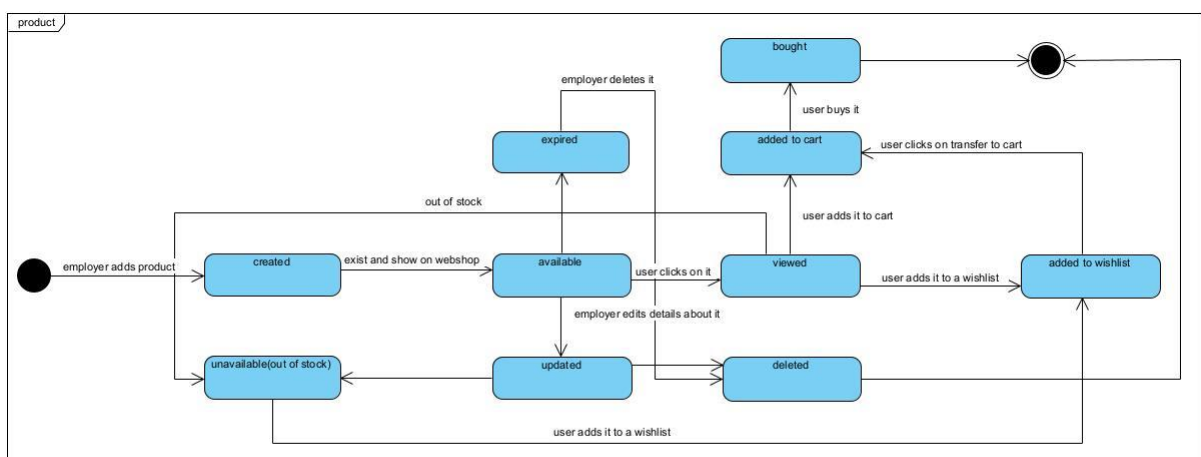
## Behavioural State Machine Diagrams



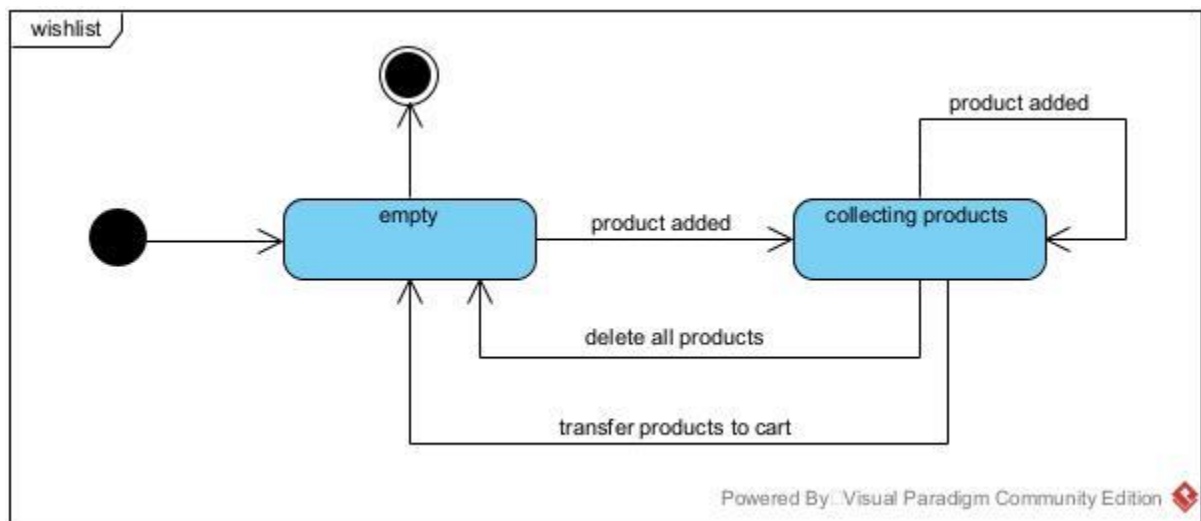
Behavioral State Machine diagram 1



Behavioral State Machine diagram 2



Behavioral State Machine diagram 3



Behavioral State Machine diagram 4

## Packages Diagrams



### Package diagram 1



### Pacage diagram 2 (show relationships)