KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

T120B029 Programų sistemų analizės ir projektavimo įrankiai
Projekto ataskaita nr. 1

Tema: Barbora su receptais

Studentų komanda:
Nerijus Dulkė IFF-6/11
Arūnas Bendoraitis IFF-6/11
Evaldas Kušlevič IFF-6/1
Gediminas Jakovlevas IFF-6/1

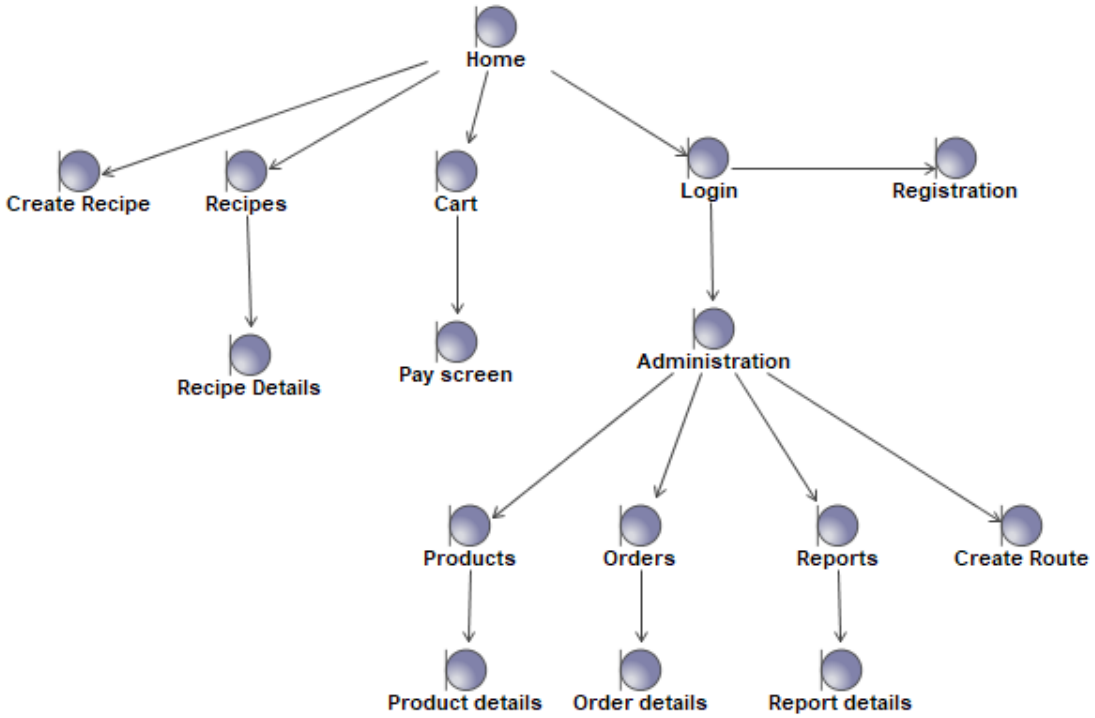
Dėstytojai:
Ryselis Karolis
doc. ČEPONIENĖ Lina
lekt. Jurgelaitis Mantas
Magylaitė Kristina

Kaunas 2019

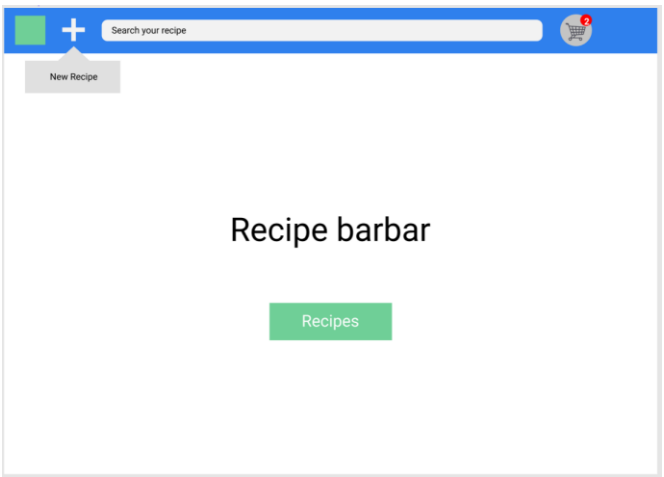
1. System description

System under development is a digital food item store, where people can pick up products for given recipes. The list of dishes are created by the users, then it is possible to choose some recipes, put them in the cart and order the products to be delivered to home.

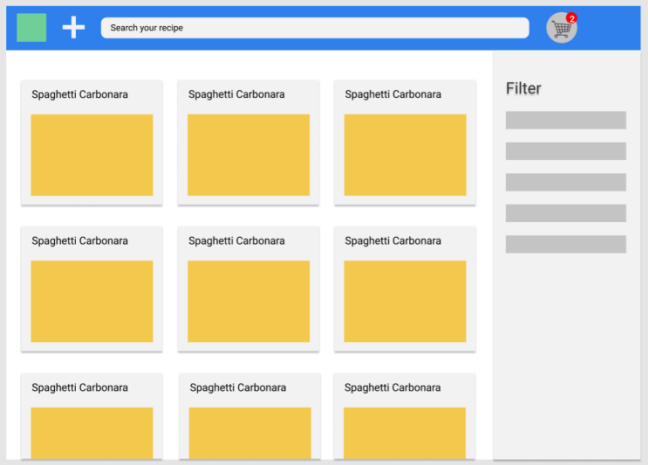
2. User interface prototype



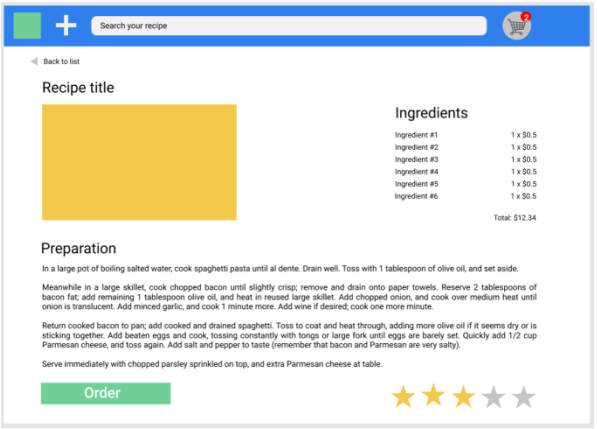
The navigation contains these windows.



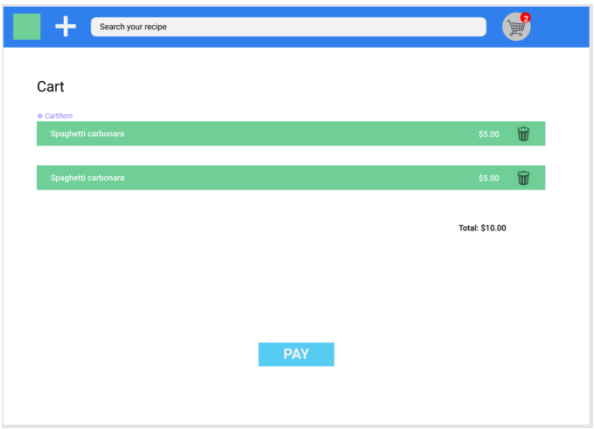
Main window. In here there's main navigation elements - logo, new item creation dropdown, search, cart with an indicator of how many things are added to the cart.
Clicking on logo will redirect to this page and from here recipe list can be accessed easily.



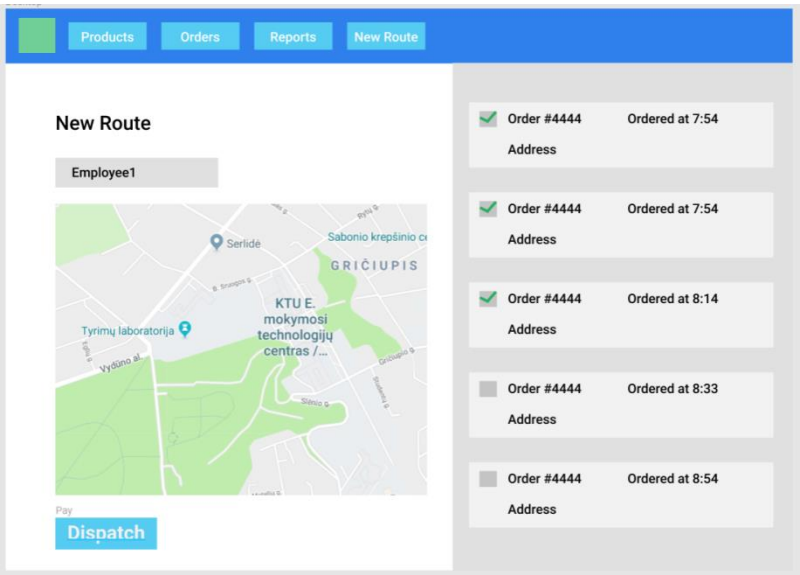
Inside the recipe list it is possible to see many different recipes that a user can buy. The list can be filtered by various features of dishes like price, user rating. By clicking on a card, user is redirected to details view.



Details view displays data about recipe ingredients, price for ingredients and the instructions on how to prepare the dish. It is also possible to see how other people rate this recipe and the user can provide his or her own feedback. By clicking order, the item goes to a cart and user is redirected back to the list.



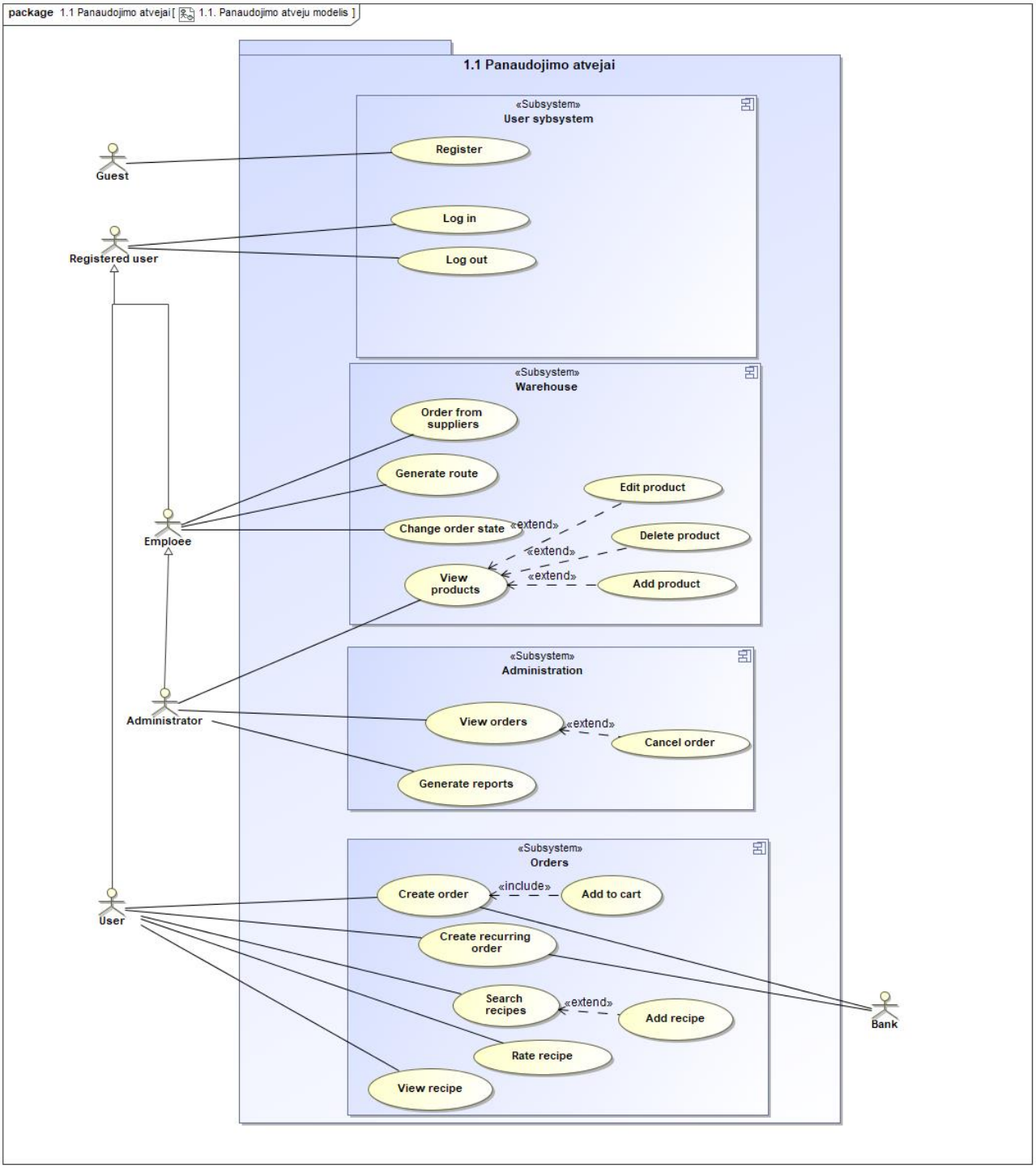
Once all of the items are in the cart, by clicking on the top left cart icon user is redirected to checkout page where the details of the order are presented, with the estimated cost and ability to pay for the service.



In administration mode there are different menu items to choose from which have a similar pattern of a list and detail views. One of the more interesting menu items is creating a new route. In the dropdown it is possible to select the employee which will get dispatched and on the right there is a list of orders sorted by time which, when selected, will appear on the map.

3. System requirements specification

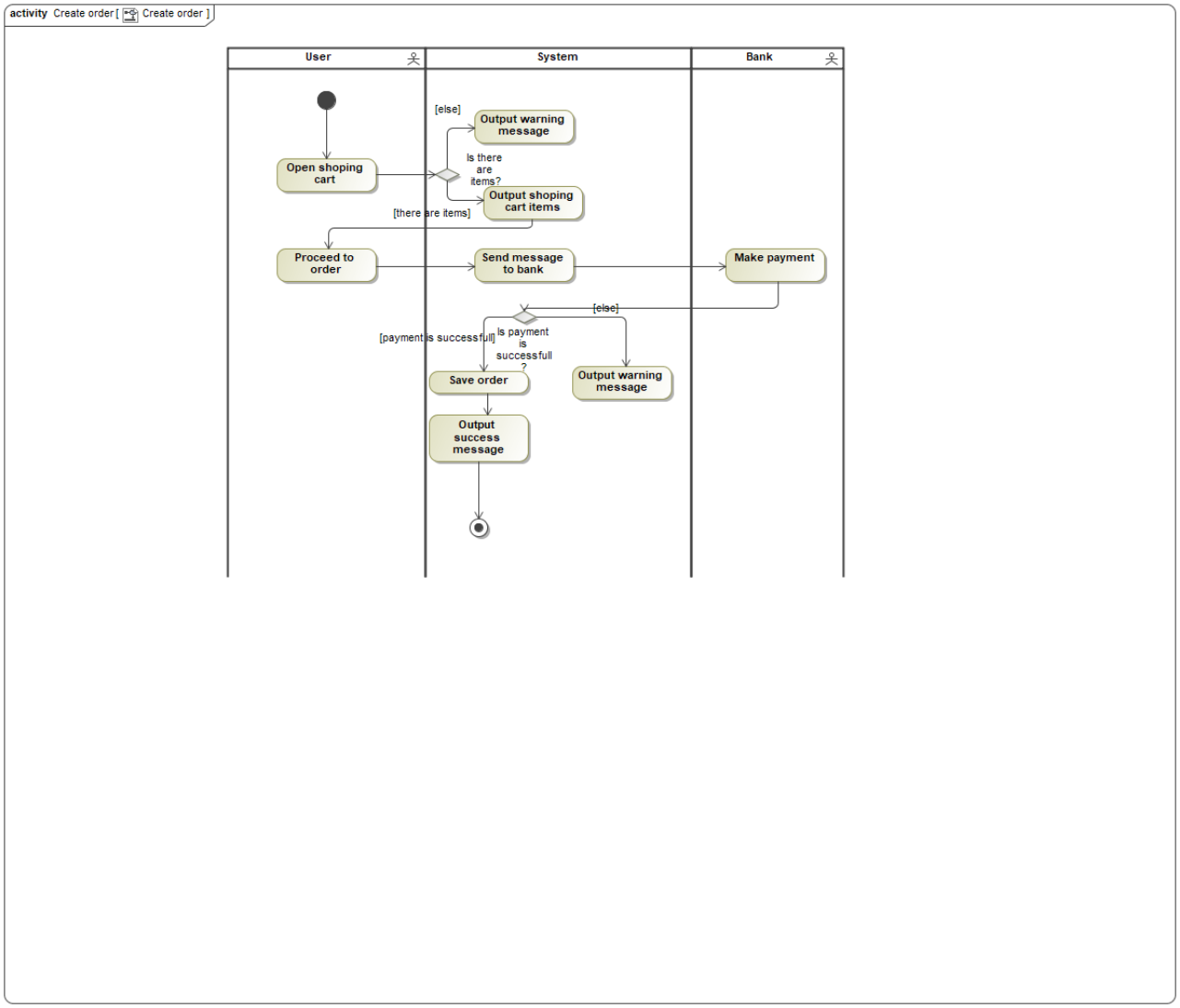
3.1. Use case model



“1.1. Panaudojimo atveju modelis” panaudojimo atvejų diagrama

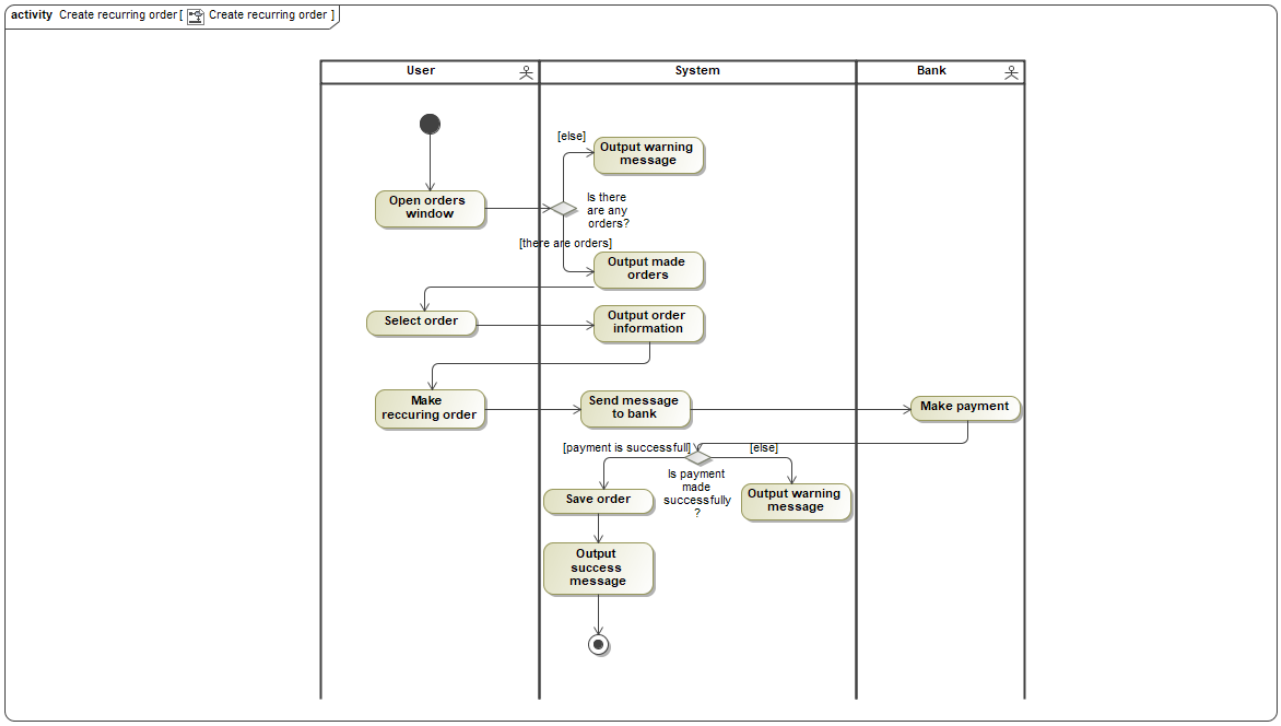
Panaudojimo atvejis		“Log out”	
Tikslas.			
Aprašymas.			
Prieš sąlyga			
Aktoriai		Registered user	
Susiję panaudojimo atvejai	**Apimami PA**		
	Išplečiantys PA		
	Specializuojami PA		
Po sąlyga			

Panaudojimo atvejis		“Create order”	Evaldas
Tikslas. Order created			
Aprašymas.			
Prieš sąlyga		User should be logged in	
Aktoriai		Bank User	
Susiję panaudojimo atvejai	**Apimami PA**		
	Išplečiantys PA		
	Specializuojami PA		
Po sąlyga		Recipes should be in shopping cart	



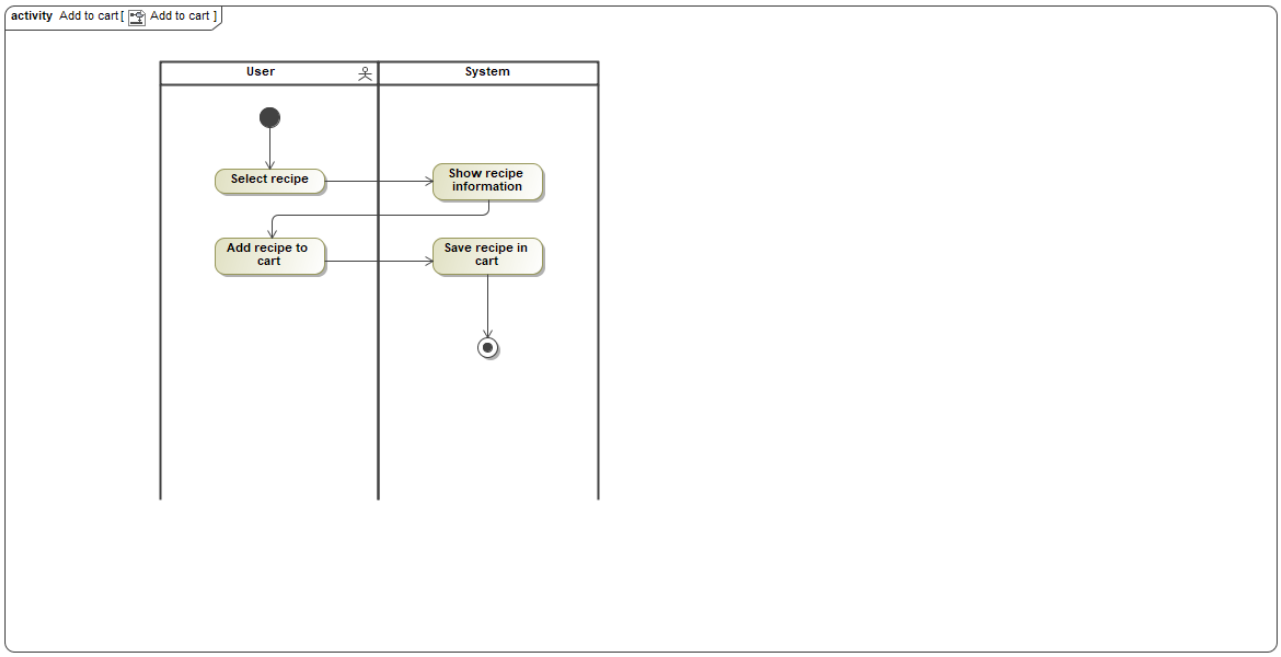
Panaudojimo atvejo “Create order” scenarijus

Panaudojimo atvejis		“Create recurring order”	Evaldas
Tikslas. can create recurring order			
Aprašymas.			
Prieš sąlyga		User must be logged in	
Aktoriai		Bank User	
Susiję panaudojimo atvejai	**Apimami PA**		
	Išplečiantys PA		
	Specializuojami PA		
Po sąlyga		Orders should be made before	



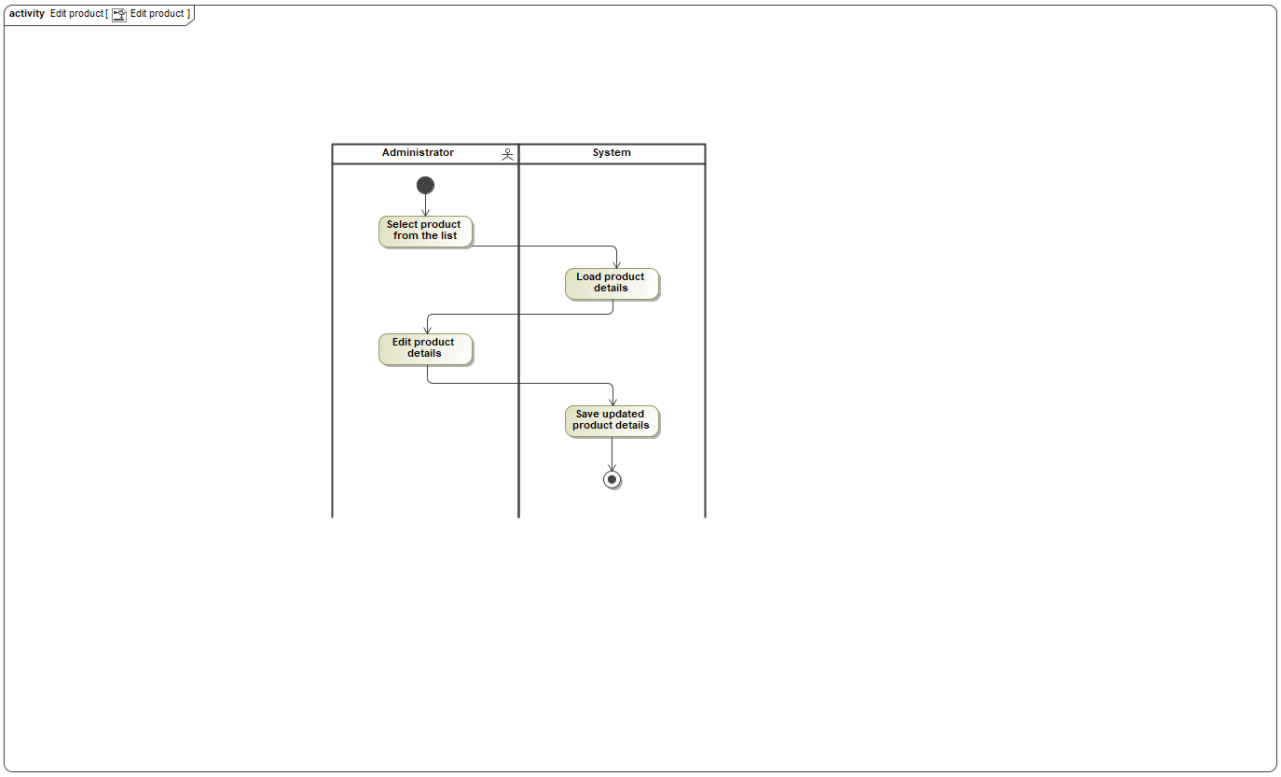
Panaudojimo atvejo “Create recurring order” scenarijus

Panaudojimo atvejis		“Add to cart”	Evaldas
Tikslas. User can add items to cart			
Aprašymas.			
Prieš sąlyga		User must be logged in	
Aktoriai			
Susiję panaudojimo atvejai	**Apimami PA**	“Create order”	
	Išplečiantys PA		
	Specializuojami PA		
Po sąlyga		Recipe should be chosen	



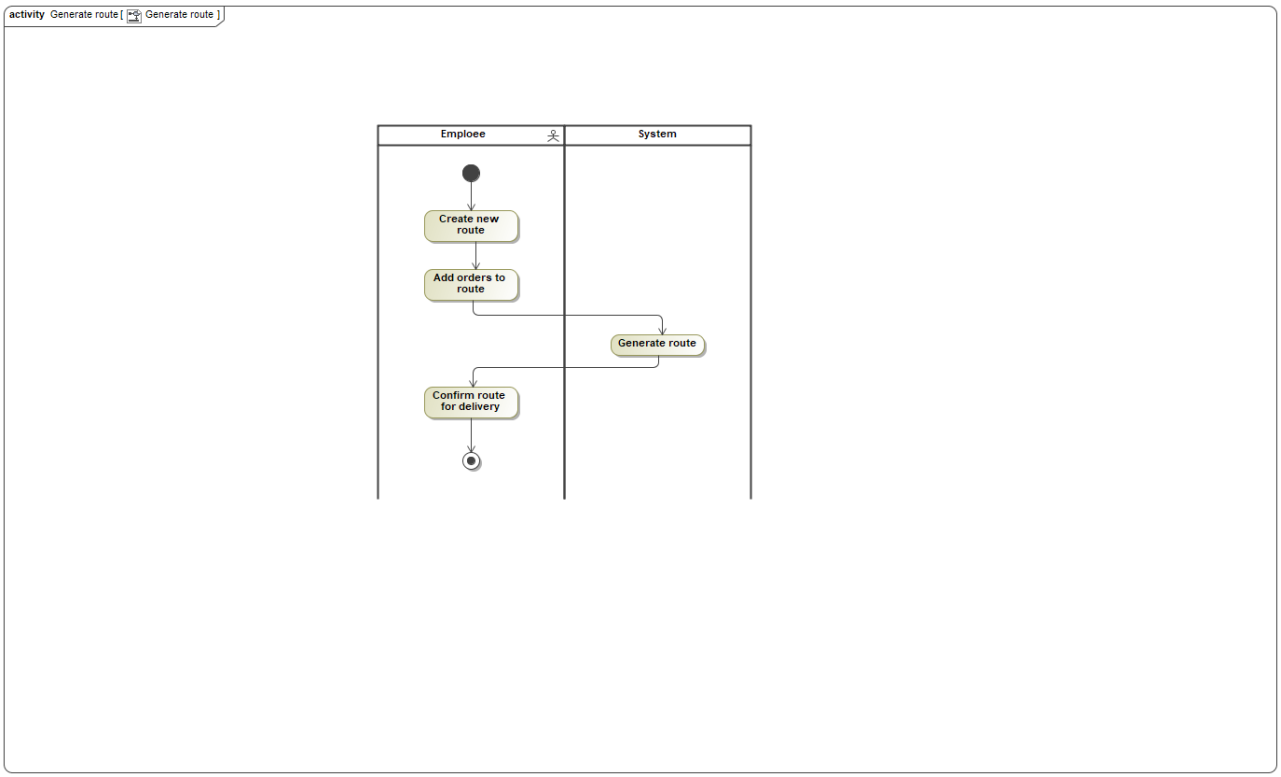
Panaudojimo atvejo “Add to cart” scenarijus

Panaudojimo atvejis		“Edit product”	Nerijus
Tikslas. Product can be edited			
Aprašymas. Product details can be changed			
Prieš sąlyga		User must be logged in as an admin	
Aktoriai			
Susiję panaudojimo atvejai	**Apimami PA**		
	Išplečiantys PA		
	Specializuojami PA		
Po sąlyga		Product details are updated	



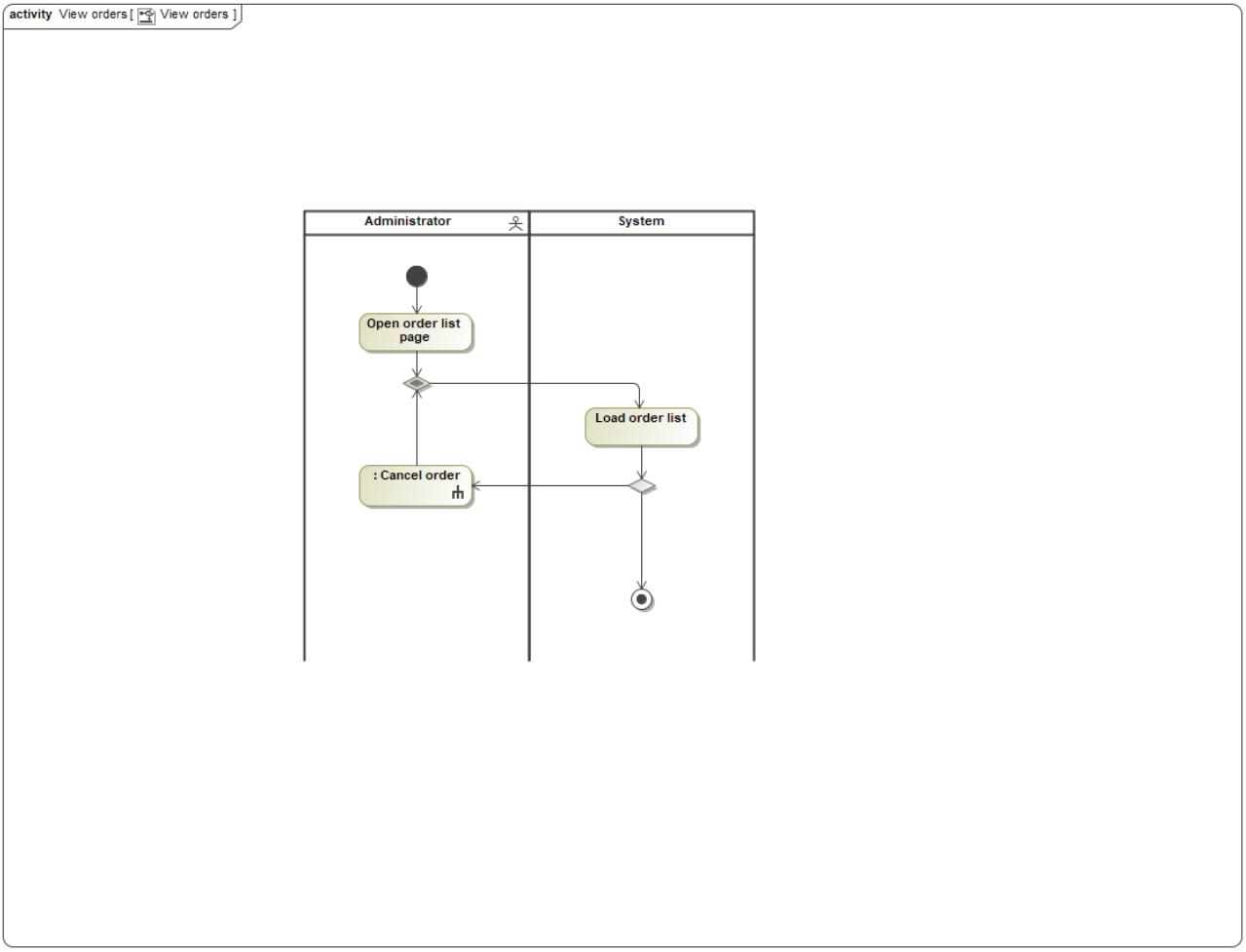
Panaudojimo atvejo “Edit product” scenarijus

Panaudojimo atvejis		“Generate route”	Nerijus
Tikslas. Should generate new optimal route to deliver multiple orders			
Aprašymas. Should generate new optimal route to deliver multiple orders			
Prieš sąlyga		User must be logged in as an admin and not closed order list should contain at least 1 order	
Aktoriai		Emploee	
Susiję panaudojimo atvejai	**Apimami PA**		
	Išplečiantys PA		
	Specializuojami PA		
Po sąlyga		New route is generated	



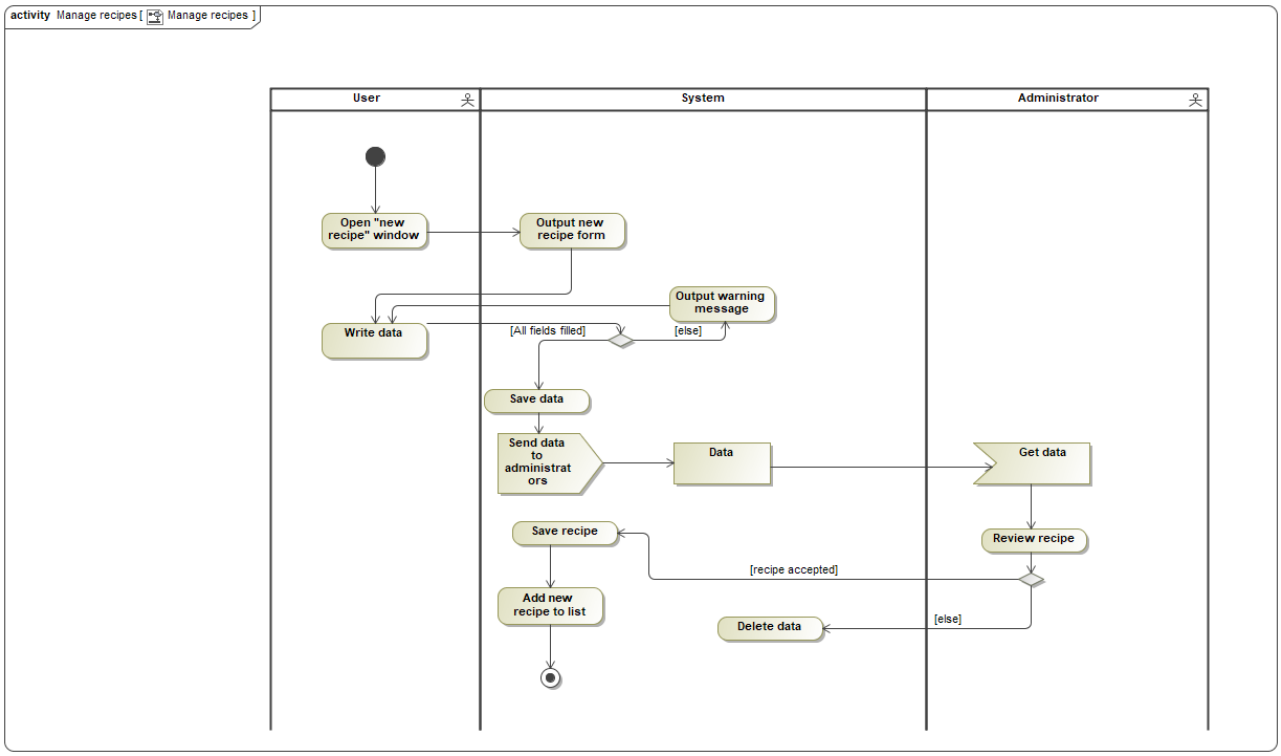
Panaudojimo atvejo “Generate route” scenarijus

Panaudojimo atvejis		“View orders”	Arunas
Tikslas. Allow the administrator to manage orders			
Aprašymas.			
Prieš sąlyga		User must be logged in as administrator	
Aktoriai		Administrator	
Susiję panaudojimo atvejai	**Apimami PA**		
	Išplečiantys PA	“Cancel order”	
	Specializuojami PA		
Po sąlyga		List of users orders should be shown	



Panaudojimo atvejo “View orders” scenarijus

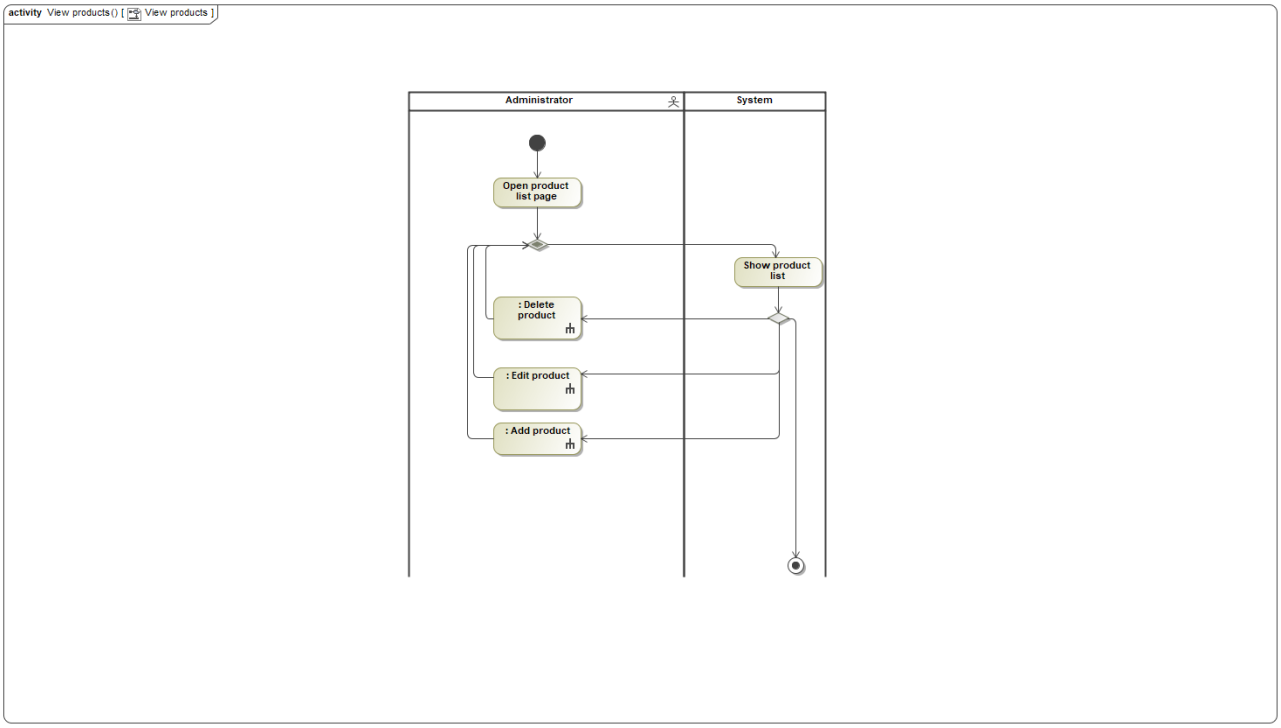
Panaudojimo atvejis		“Add recipe”	Evaldas
Tikslas. Can add new recipe			
Aprašymas.			
Prieš sąlyga		User must be logged in	
Aktoriai			
Susiję panaudojimo atvejai	**Apimami PA**		
	Išplečiantys PA		
	Specializuojami PA		
Po sąlyga		Should chose to add new recipe	



Panaudojimo atvejo “Manage recipes” scenarijus

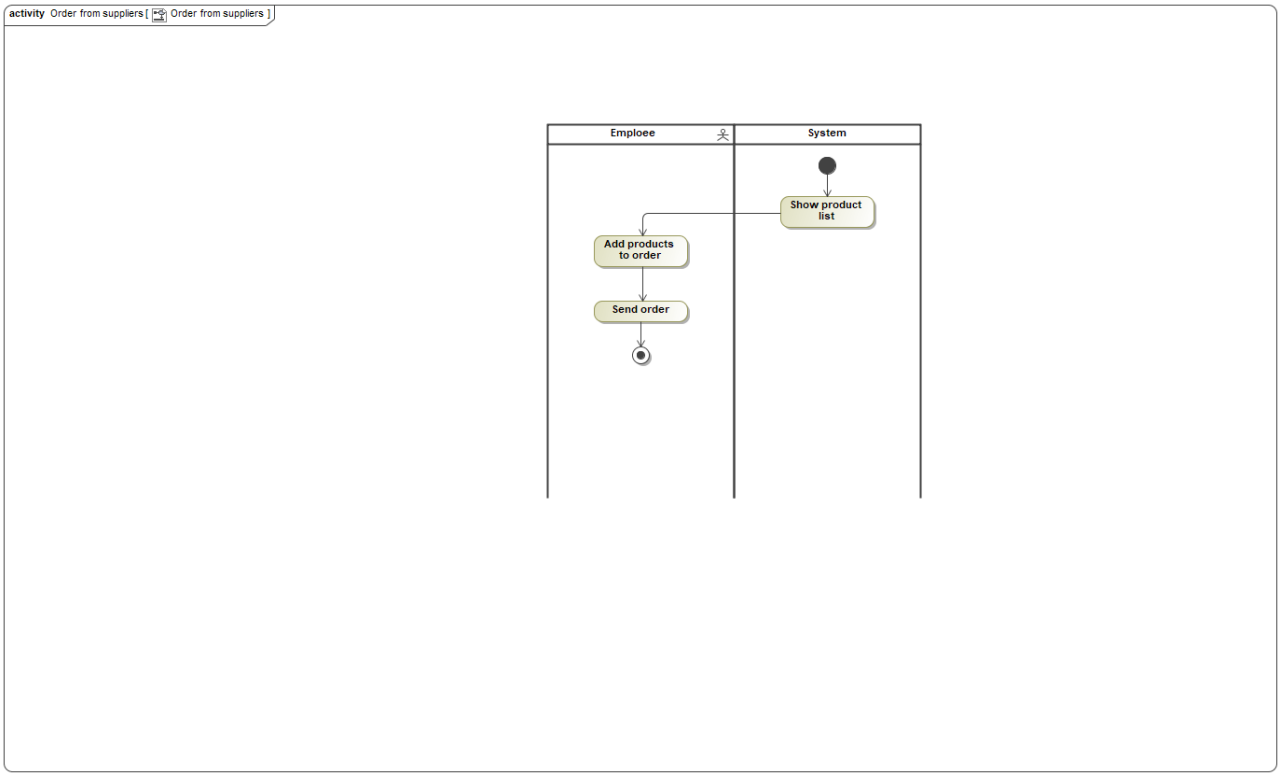
Panaudojimo atvejis		“Log in”	
Tikslas.			
Aprašymas.			
Prieš sąlyga			
Aktoriai		Registered user	
Susiję panaudojimo atvejai	**Apimami PA**		
	Išplečiantys PA		
	Specializuojami PA		
Po sąlyga			

Panaudojimo atvejis		“View products”	Nerijus
Tikslas. Allow to view prodict list			
Aprašymas. Allows an admin to add, edit and delete products			
Prieš sąlyga		User must be logged in as an admin	
Aktoriai		Administrator	
Susiję panaudojimo atvejai	**Apimami PA**		
	Išplečiantys PA	“Add product” “Delete product” “Edit product”	
	Specializuojami PA		
Po sąlyga		Product list must be shown	



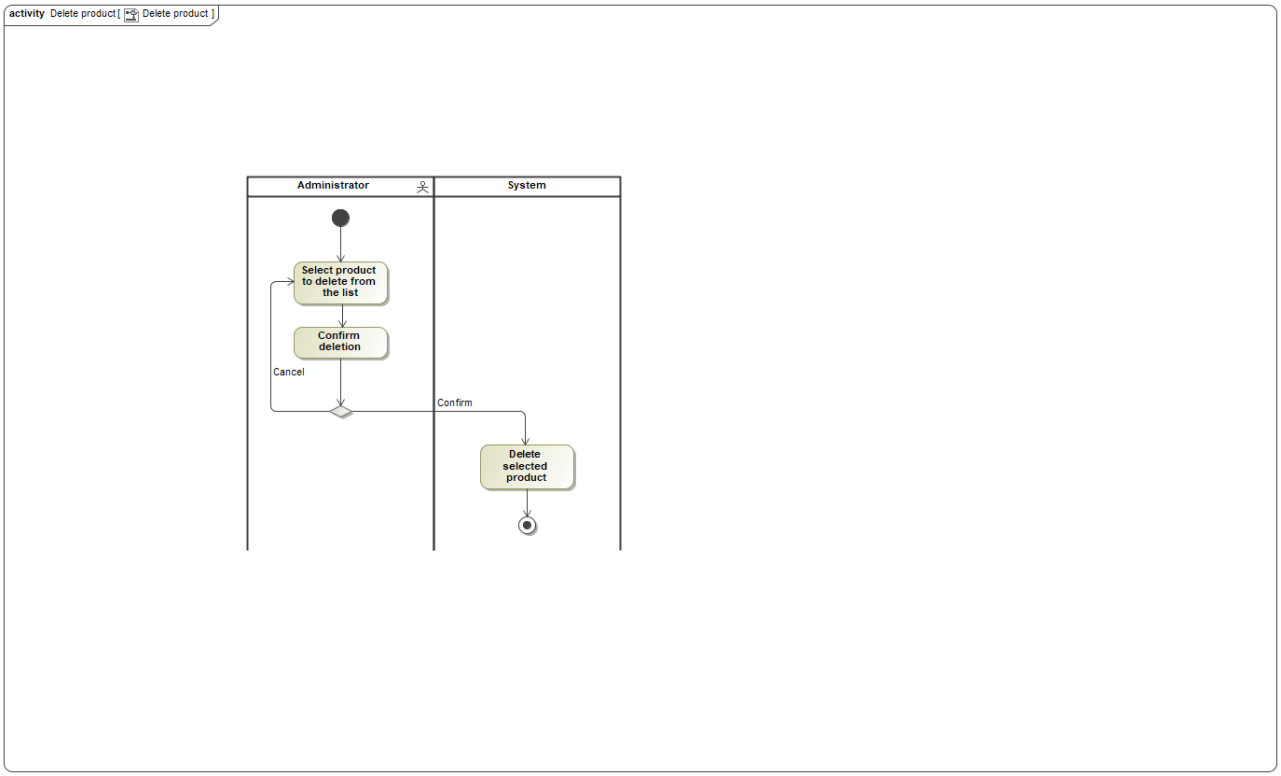
Panaudojimo atvejo “View products” scenarijus

Panaudojimo atvejis		“Order from suppliers”	Nerijus
Tikslas. Allow to order products to warehouse			
Aprašymas. Allows an admin to order needed product from suppliers to warehous			
Prieš sąlyga		User must be logged in as an admin	
Aktoriai		Emploee	
Susiję panaudojimo atvejai	**Apimami PA**		
	Išplečiantys PA		
	Specializuojami PA		
Po sąlyga		New order should be placed for supplier	



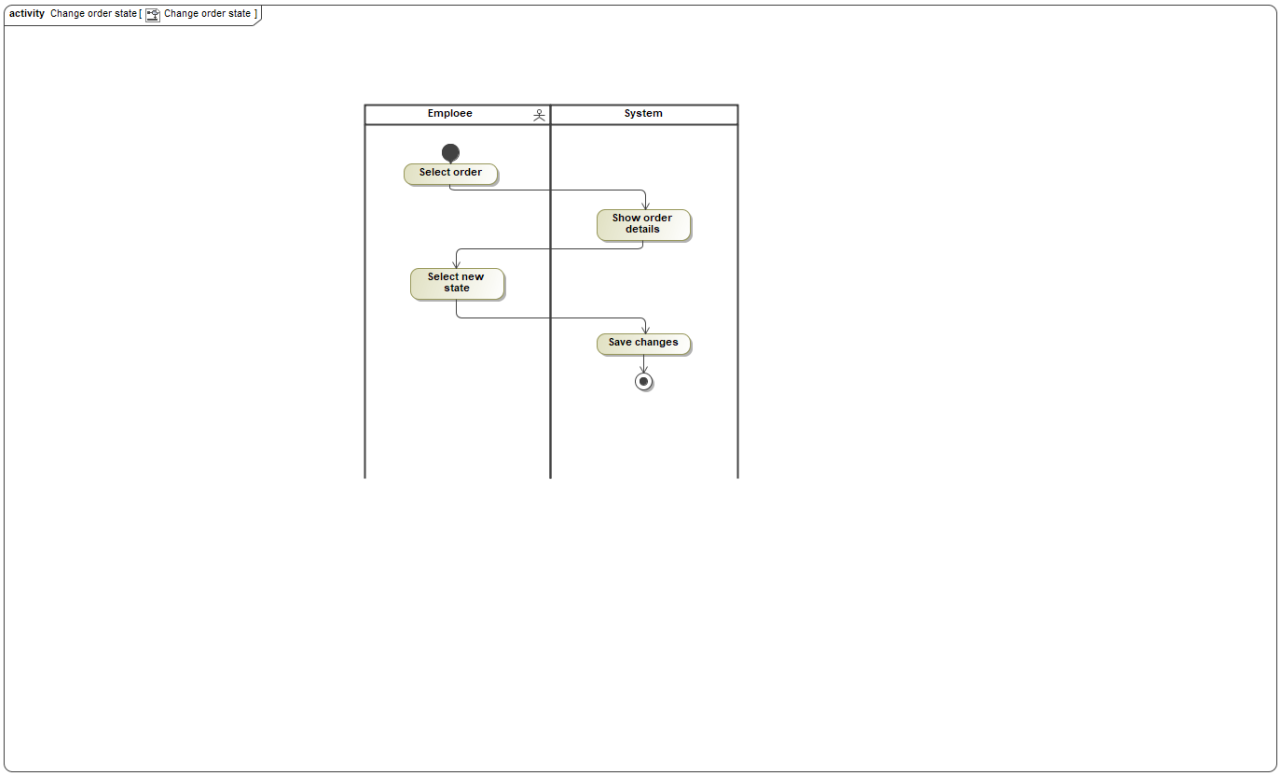
Panaudojimo atvejo “Order from suppliers” scenarijus

Panaudojimo atvejis		“Delete product”	
Tikslas. Product is deleted from the list			
Aprašymas. Product can be deleted from all products list			
Prieš sąlyga		User must be logged in as an admin	
Aktoriai			
Susiję panaudojimo atvejai	**Apimami PA**		
	Išplečiantys PA		
	Specializuojami PA		
Po sąlyga		Product is deleted from the list	



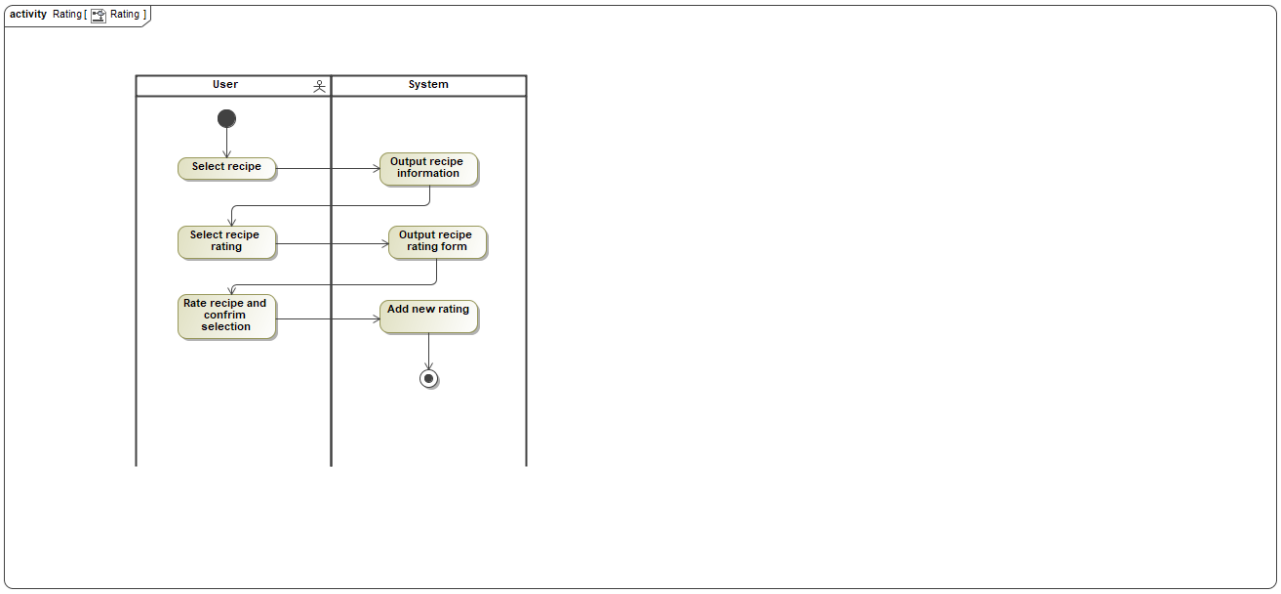
Panaudojimo atvejo “Delete product” scenarijus

Panaudojimo atvejis		“Change order state”	Nerijus
Tikslas. Employee can change order state			
Aprašymas. Employee can change order state			
Prieš sąlyga		User must be logged in as an employee and not closed order list should have orders	
Aktoriai		Emploee	
Susiję panaudojimo atvejai	**Apimami PA**		
	Išplečiantys PA		
	Specializuojami PA		
Po sąlyga		Order state is changed	



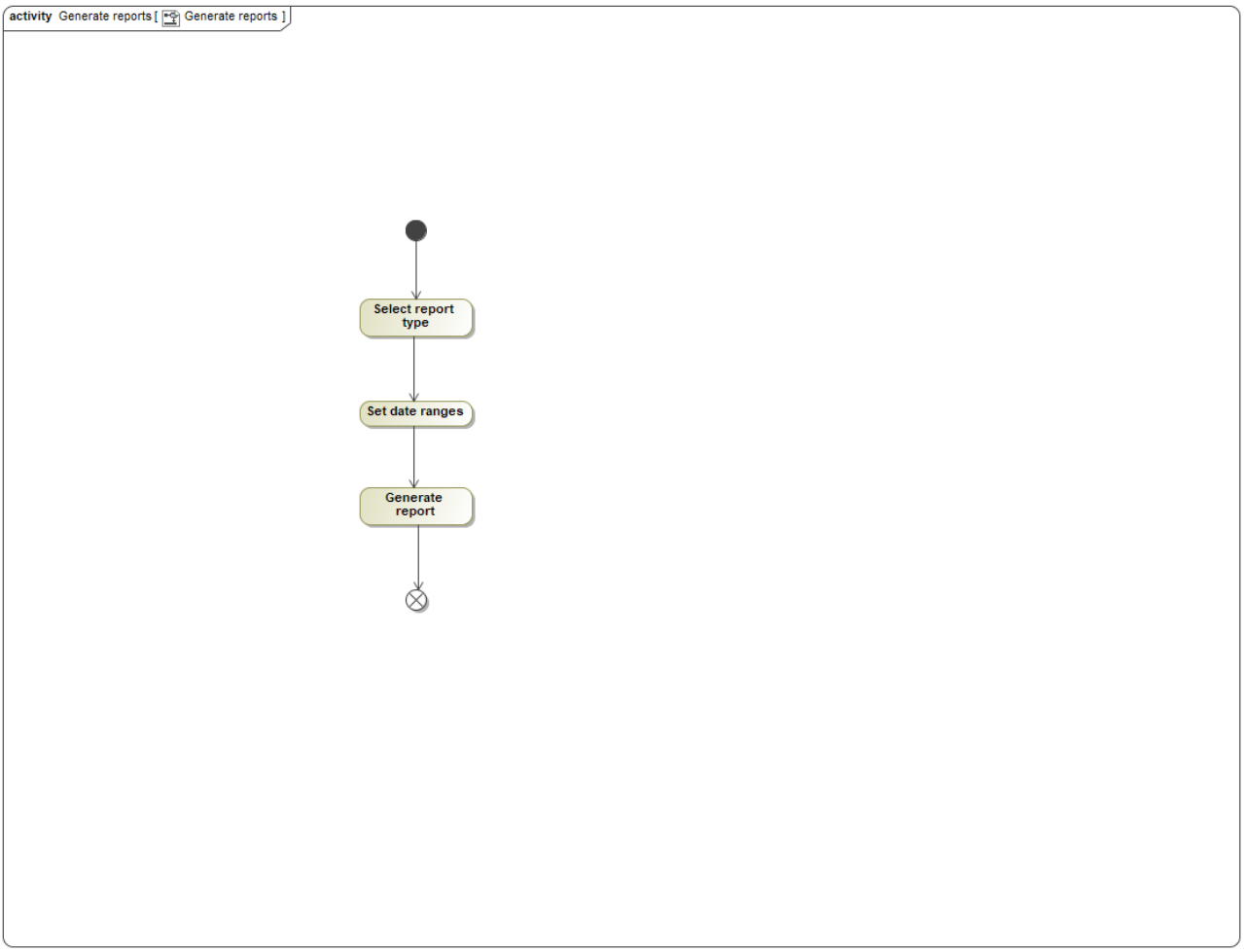
Panaudojimo atvejo “Change order state” scenarijus

Panaudojimo atvejis		“Rate recipe”	Arunas
Tikslas. To allow user to find the best recipes			
Aprašymas.			
Prieš sąlyga		User is logged in	
Aktoriai		User	
Susiję panaudojimo atvejai	**Apimami PA**		
	Išplečiantys PA		
	Specializuojami PA		
Po sąlyga		Recipe rating is updated	

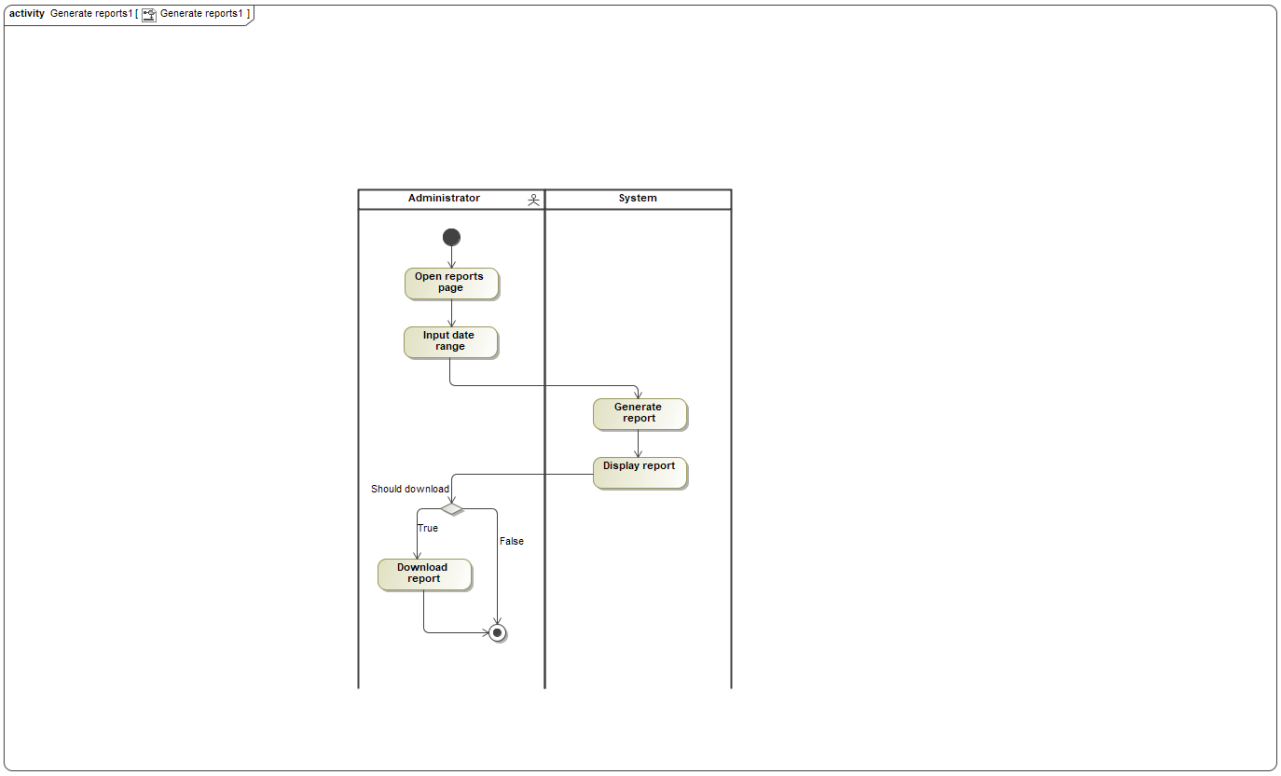


Panaudojimo atvejo “Rating” scenarijus

Panaudojimo atvejis		“Generate reports”	Arūnas
Tikslas. Generate a report			
Aprašymas. User should be able to generate different reports for specified date ranges.			
Prieš sąlyga		User must be logged in as an admin	
Aktoriai		Administrator	
Susiję panaudojimo atvejai	**Apimami PA**		
	Išplečiantys PA		
	Specializuojami PA		
Po sąlyga		Report should be generated for specified date range	



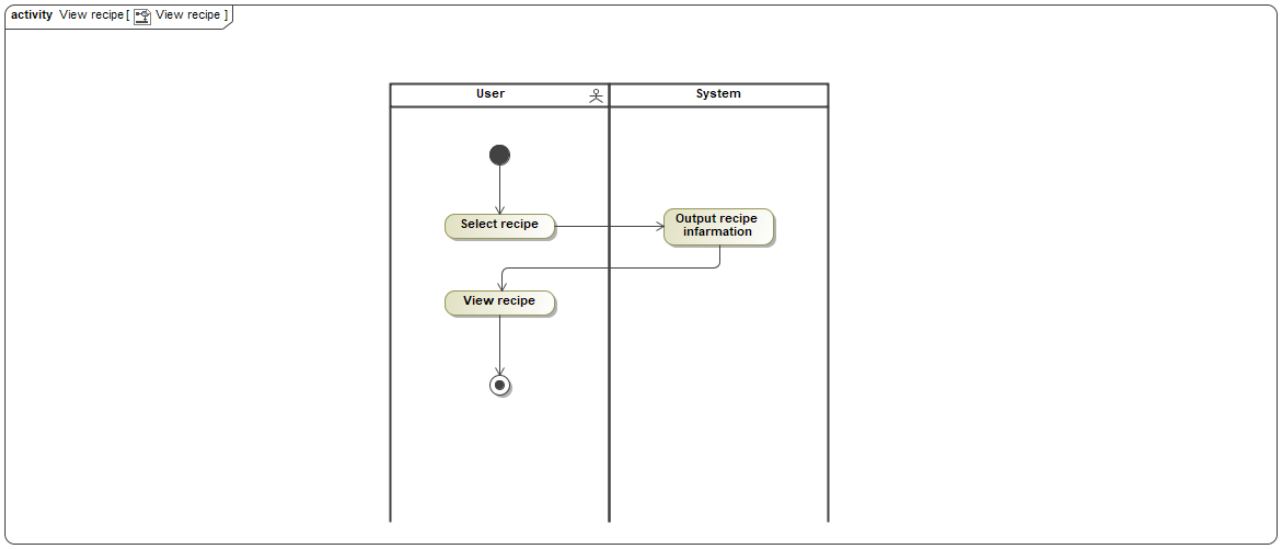
Panaudojimo atvejo “Generate reports” scenarijus



Panaudojimo atvejo “Generate reports1” scenarijus

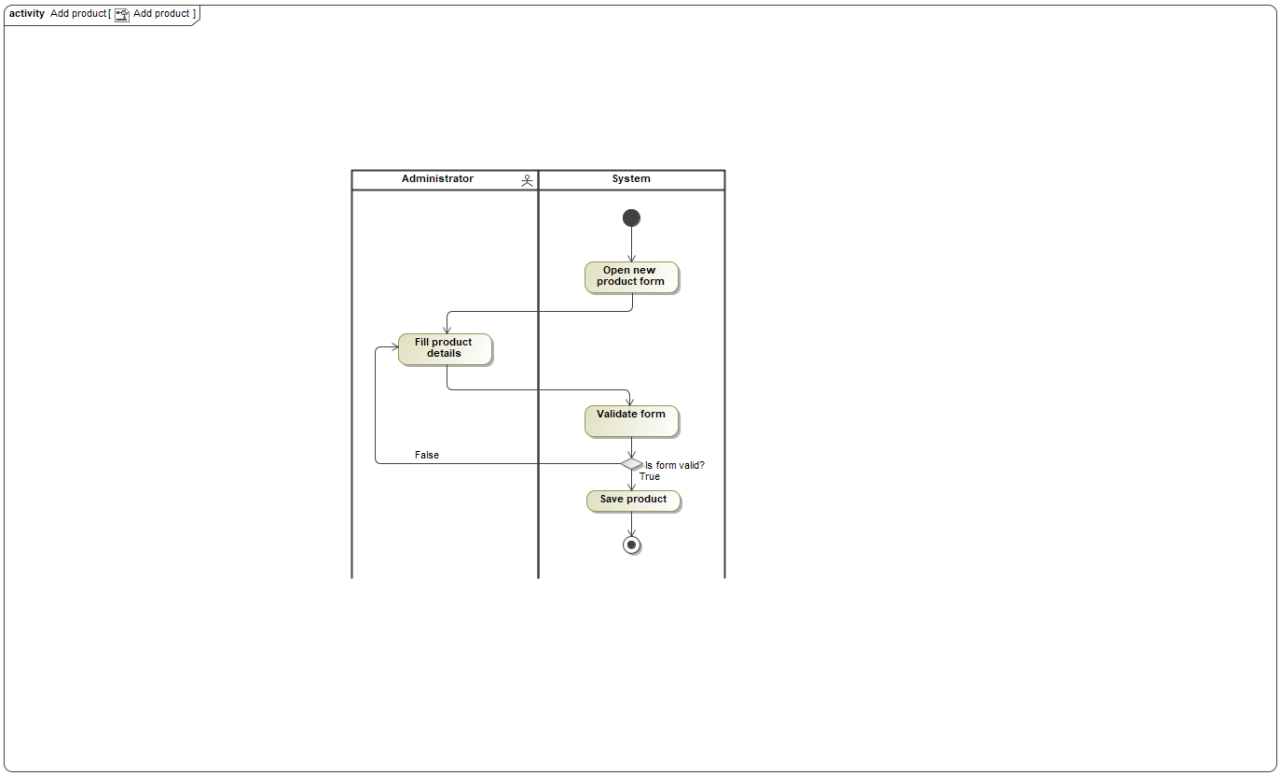
Panaudojimo atvejis		“Register”	
Tikslas.			
Aprašymas.			
Prieš sąlyga			
Aktoriai		Guest	
Susiję panaudojimo atvejai	**Apimami PA**		
	Išplečiantys PA		
	Specializuojami PA		
Po sąlyga			

Panaudojimo atvejis		“View recipe”	Evaldas
Tikslas. User can view recipes			
Aprašymas.			
Prieš sąlyga		User must be logged in	
Aktoriai		User	
Susiję panaudojimo atvejai	**Apimami PA**		
	Išplečiantys PA		
	Specializuojami PA		
Po sąlyga		Recipes are found	



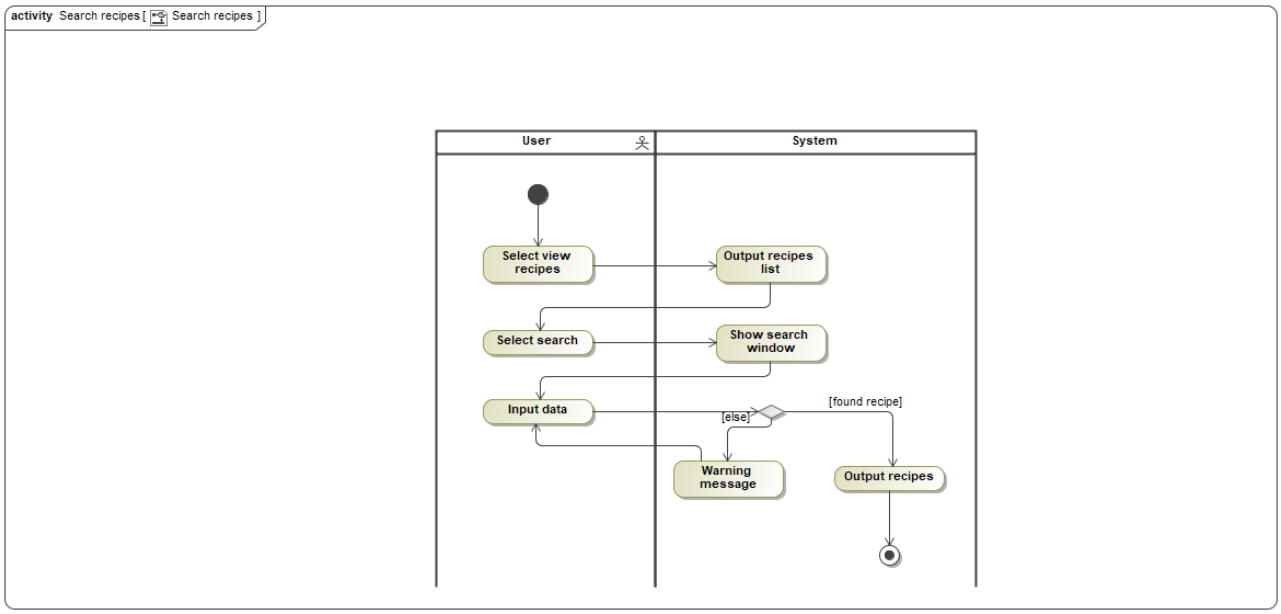
Panaudojimo atvejo “View recipe” scenarijus

Panaudojimo atvejis		“Add product”	Nerijus
Tikslas. New product is added			
Aprašymas. New product can be added			
Prieš sąlyga		User must be logged in as an admin	
Aktoriai			
Susiję panaudojimo atvejai	**Apimami PA**		
	Išplečiantys PA		
	Specializuojami PA		
Po sąlyga		New product is added to all products list	



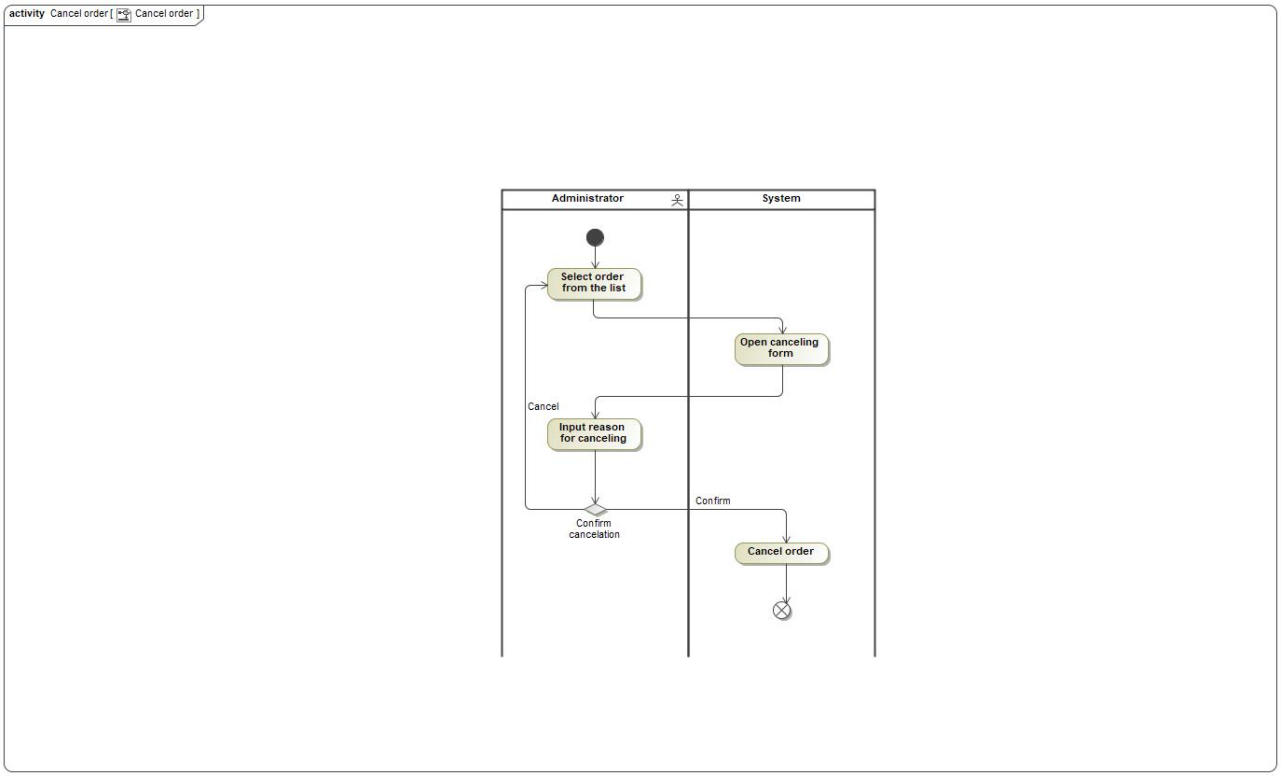
Panaudojimo atvejo “Add product” scenarijus

Panaudojimo atvejis		“Search recipes”	Evaldas
Tikslas. User can find recipes			
Aprašymas.			
Prieš sąlyga		User must be logged in	
Aktoriai		User	
Susiję panaudojimo atvejai	**Apimami PA**		
	Išplečiantys PA	“Add recipe”	
	Specializuojami PA		
Po sąlyga		Recipe should be chosen	



Panaudojimo atvejo “Search recipes” scenarijus

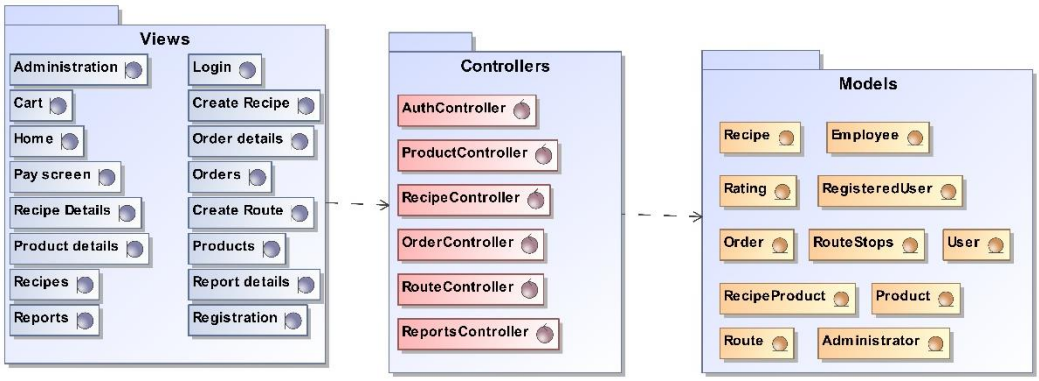
Panaudojimo atvejis		“Cancel order”	Arunas
Tikslas. Order is canceled			
Aprašymas. Users order can be canceled by an admin			
Prieš sąlyga		User must be logged in as an admin	
Aktoriai			
Susiję panaudojimo atvejai	**Apimami PA**		
	Išplečiantys PA		
	Specializuojami PA		
Po sąlyga		Client can see that his order was canceled by an adm in	



Panaudojimo atvejo “Cancel order” scenarijus

4. Sistemos projekto modelis

4.1. Sistemos loginė architektūra



4.2 Use case sequence diagrams

This section provides diagrams of the project phase sequences for each use case

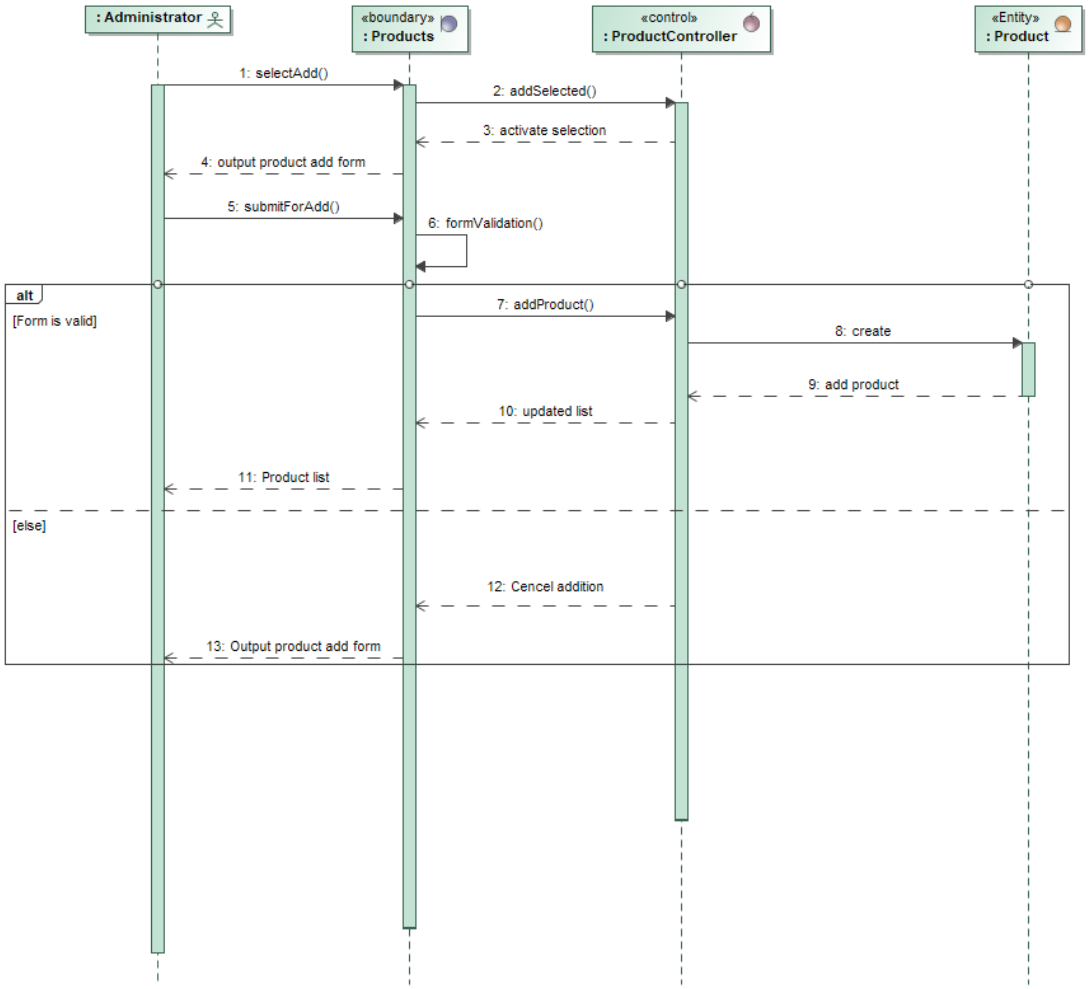


Diagram 1: Add product

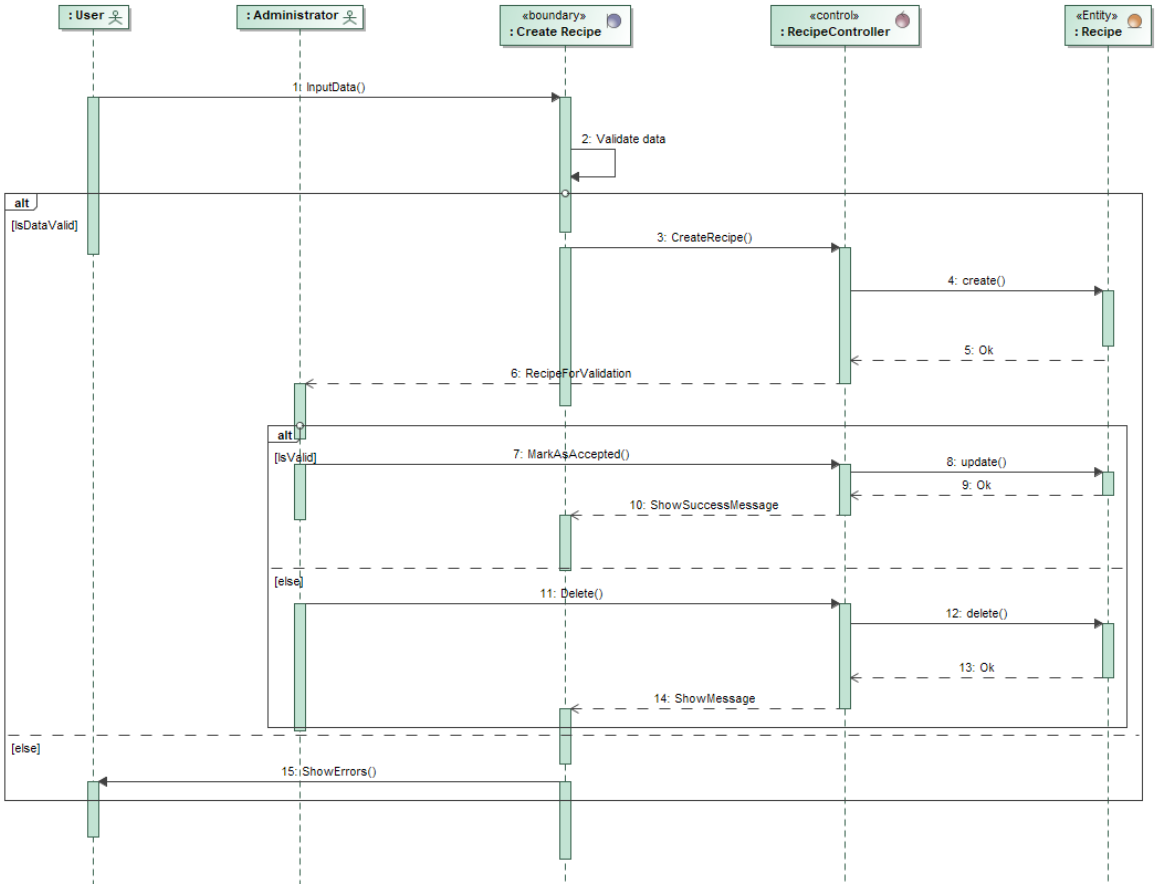


Diagram 2: Add recipe

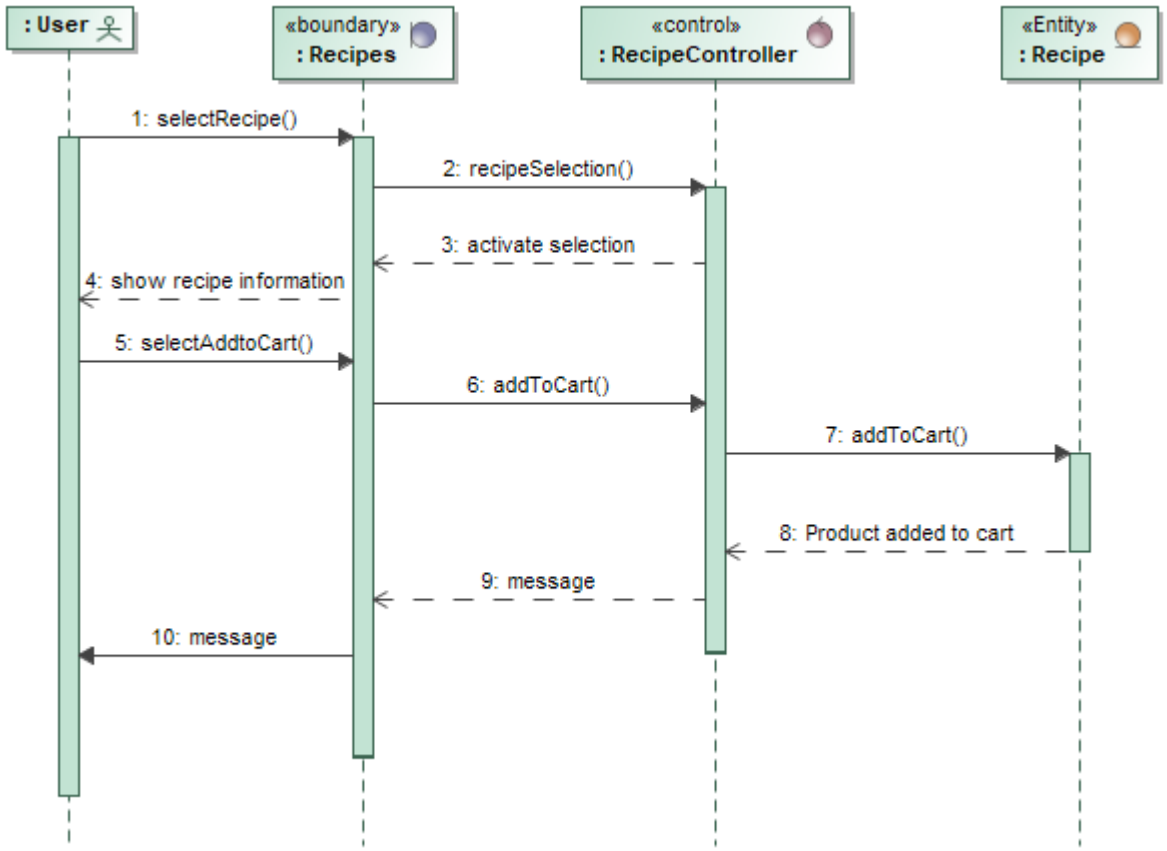


Diagram 3: Add to cart

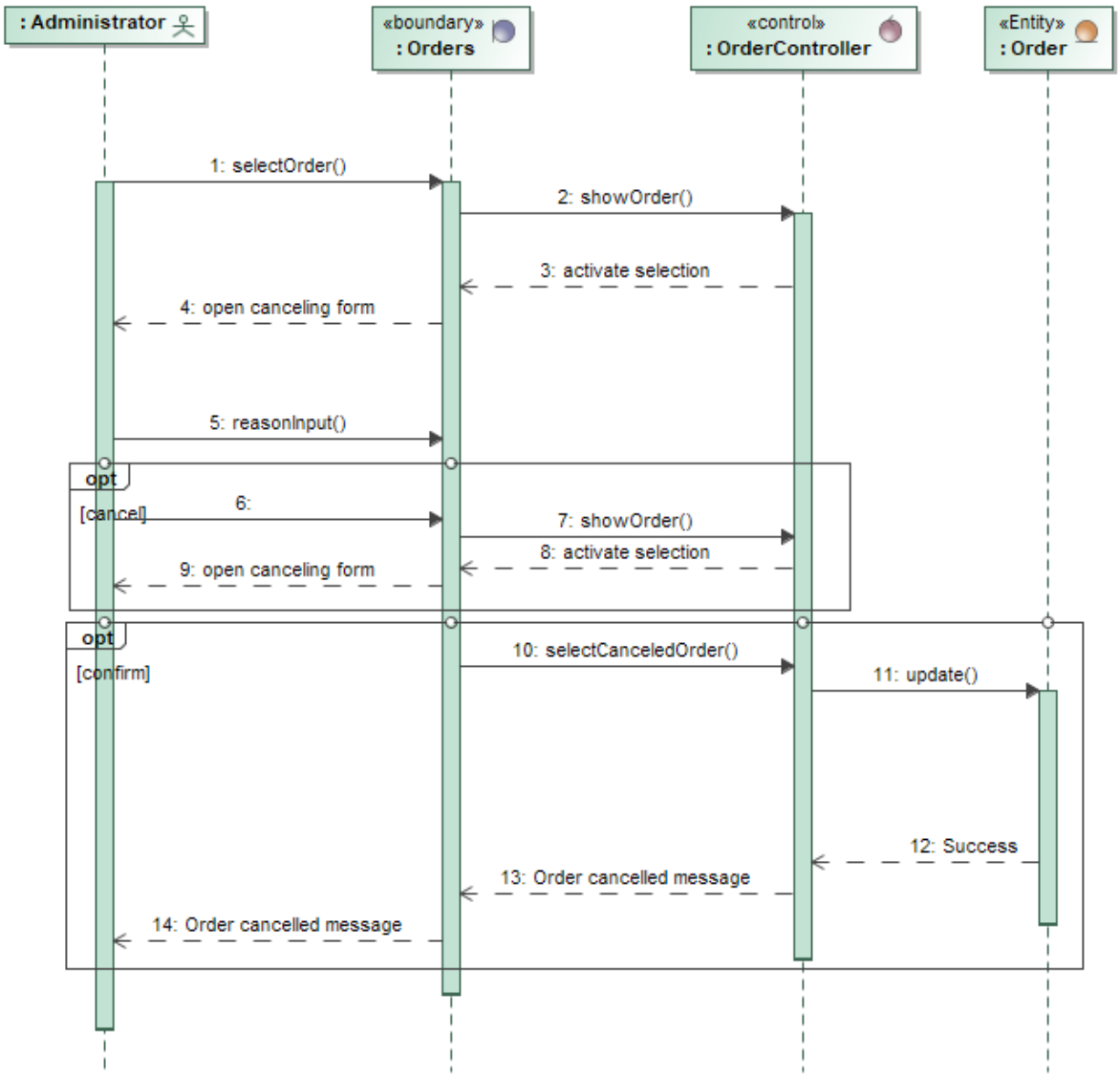


Diagram 4: Cancel order

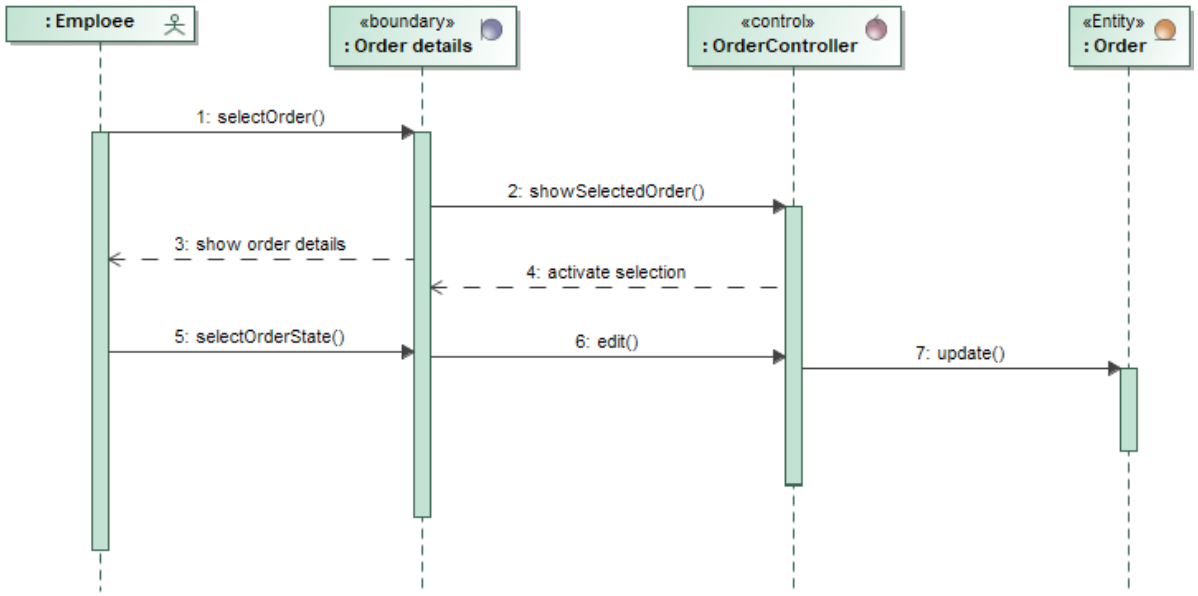


Diagram 5: Change order state

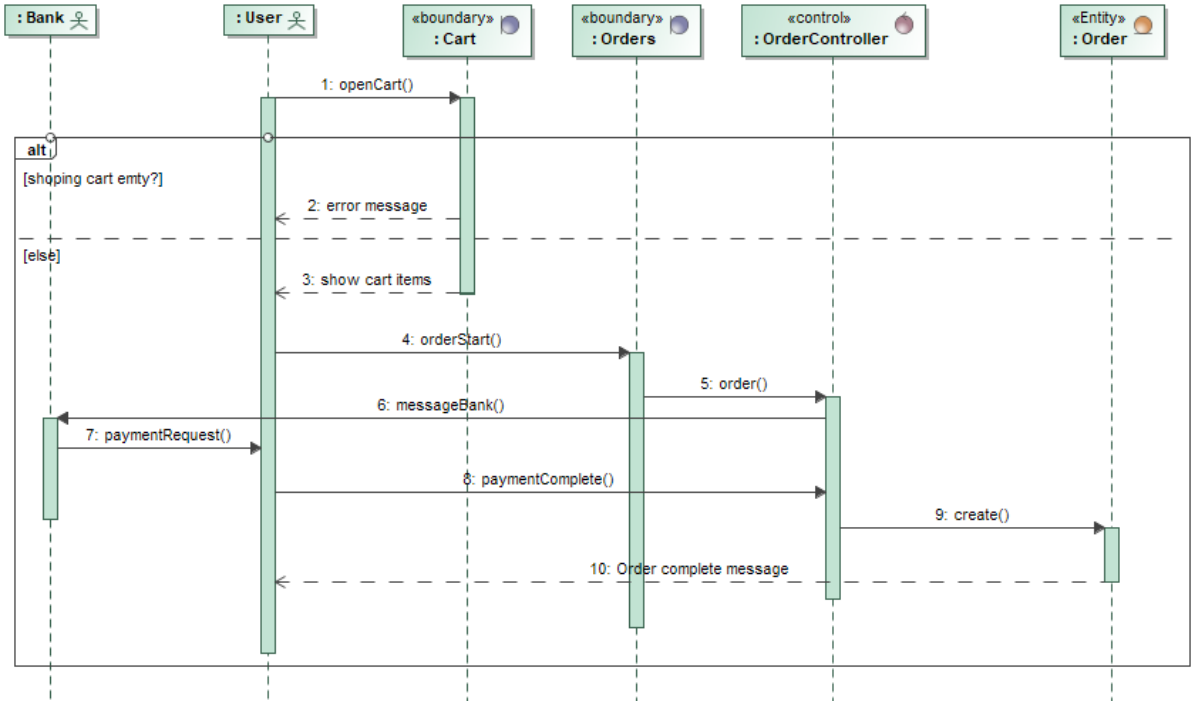


Diagram 6: Create order

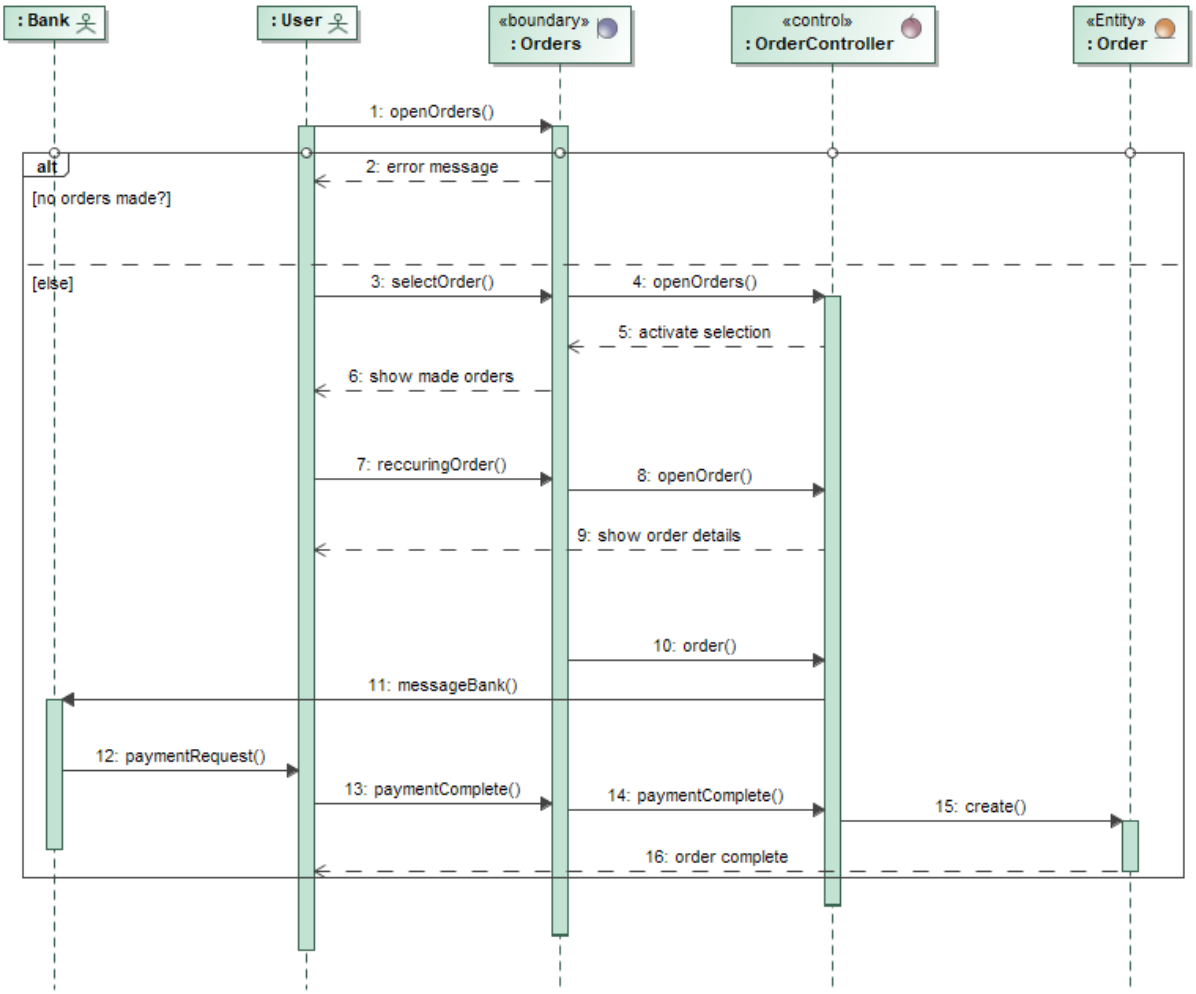


Diagram 7: Create recurring order

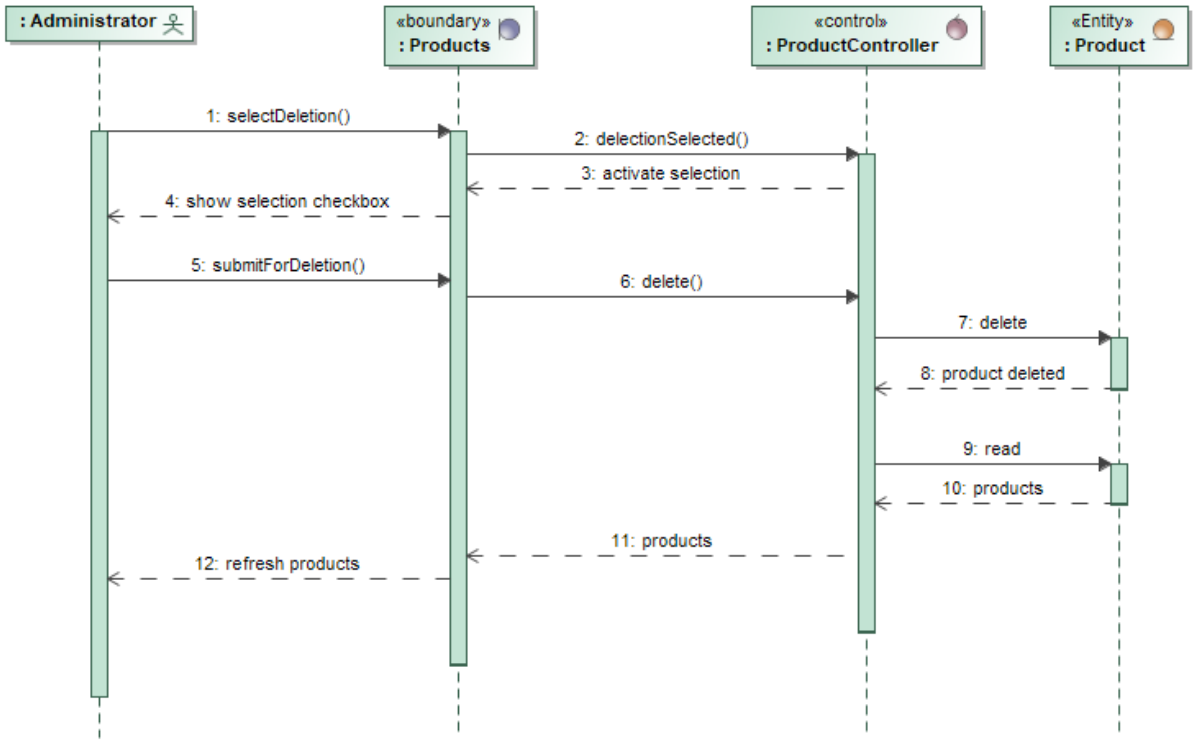


Diagram 8: Delete product

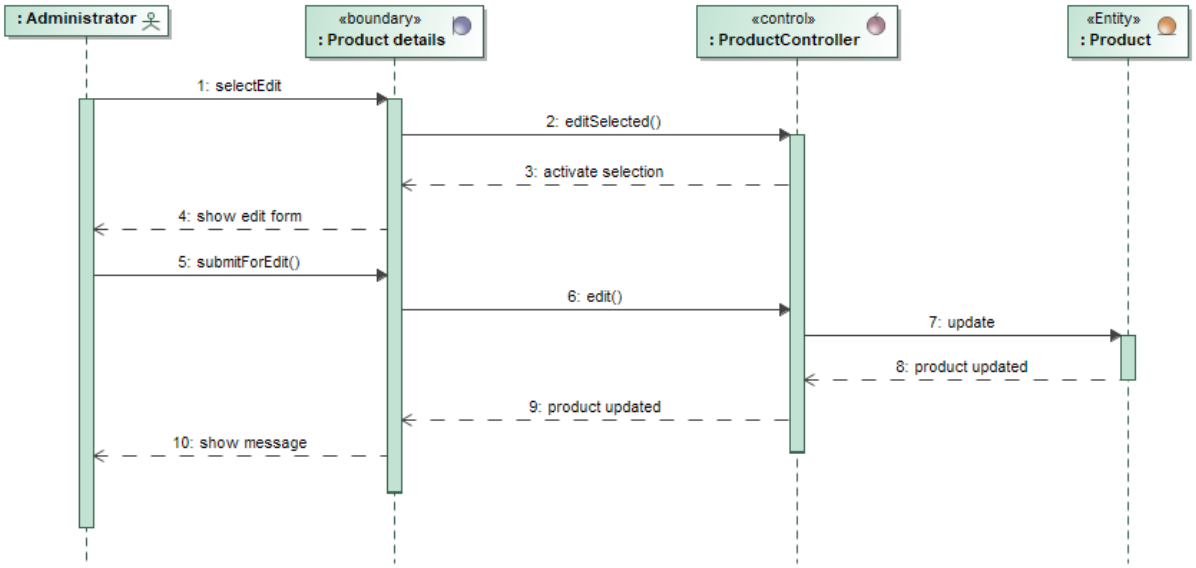


Diagram 9: Edit product

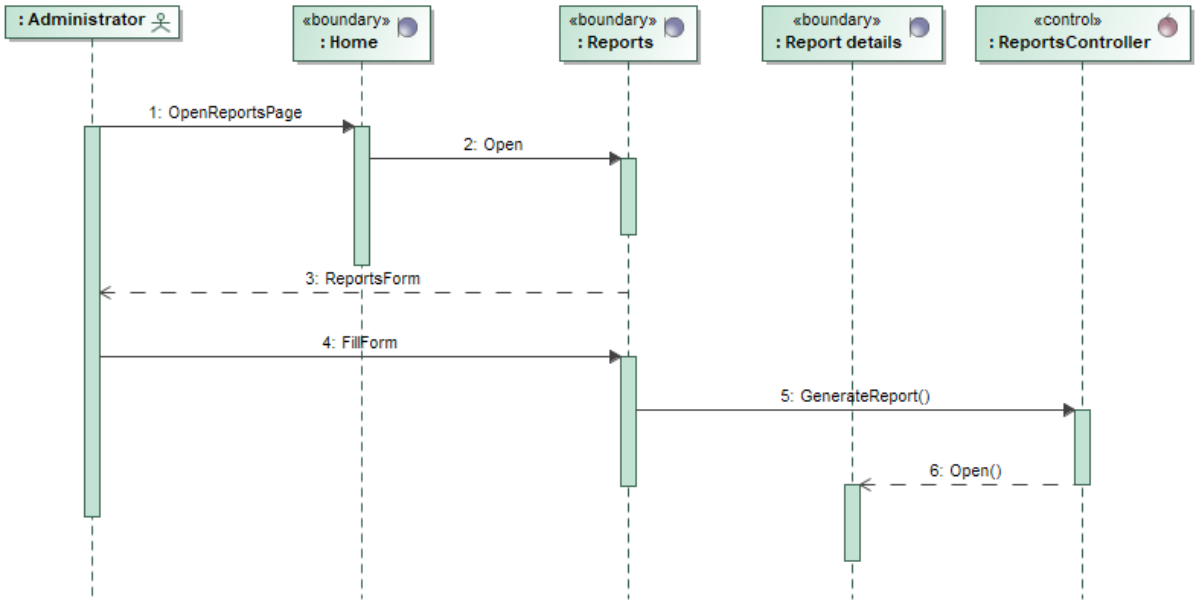


Diagram 10: Generate reports

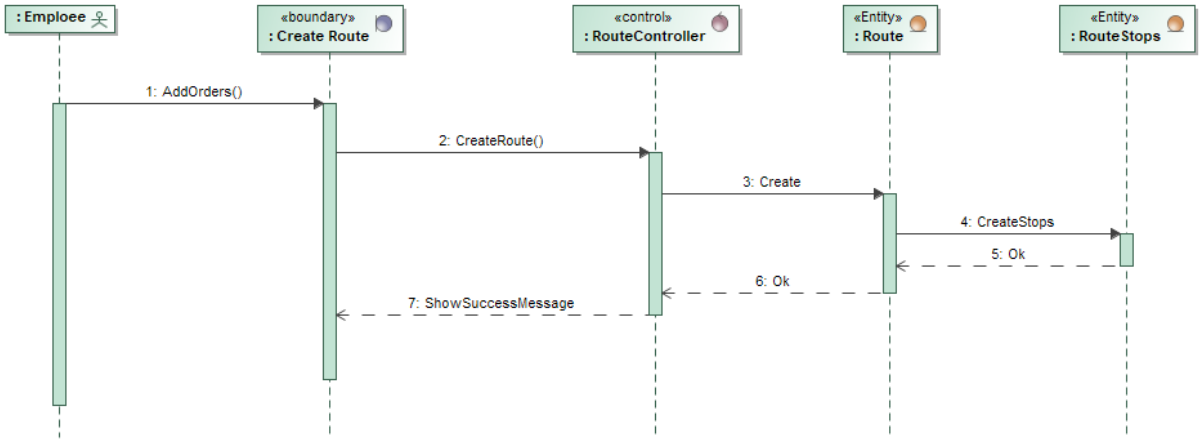


Diagram 11: Generate route

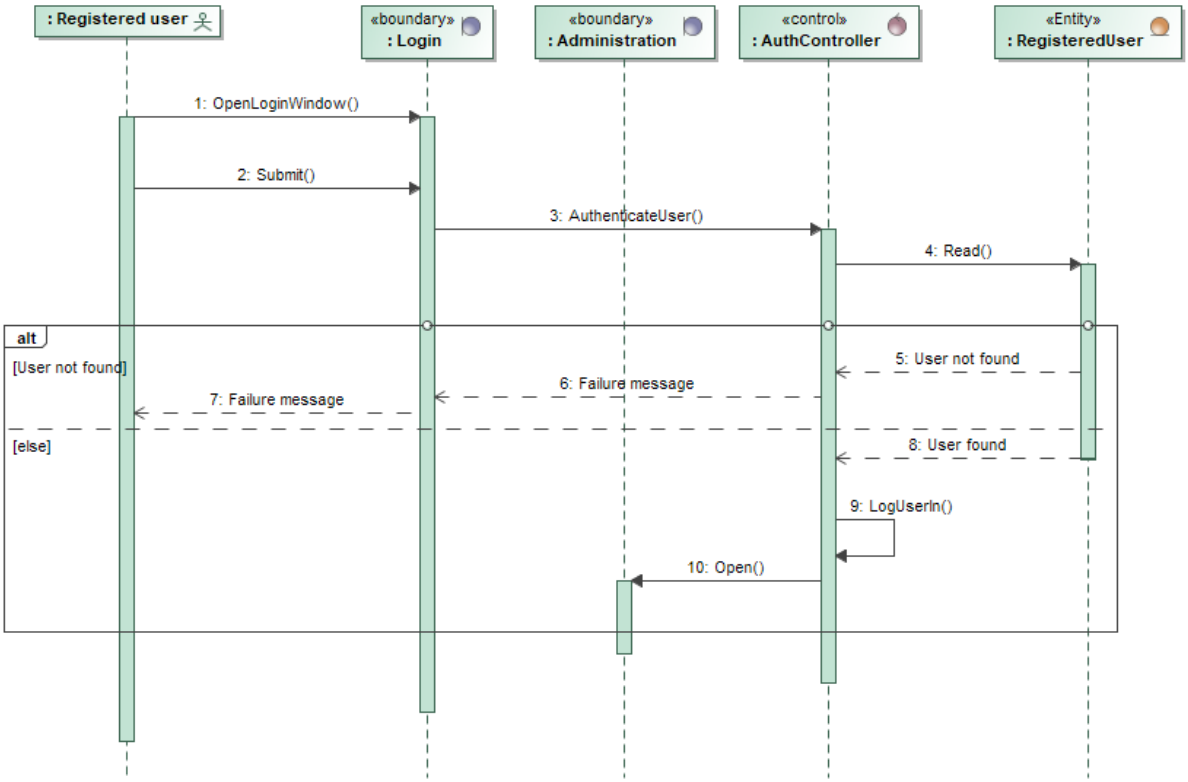


Diagram 12: Log in

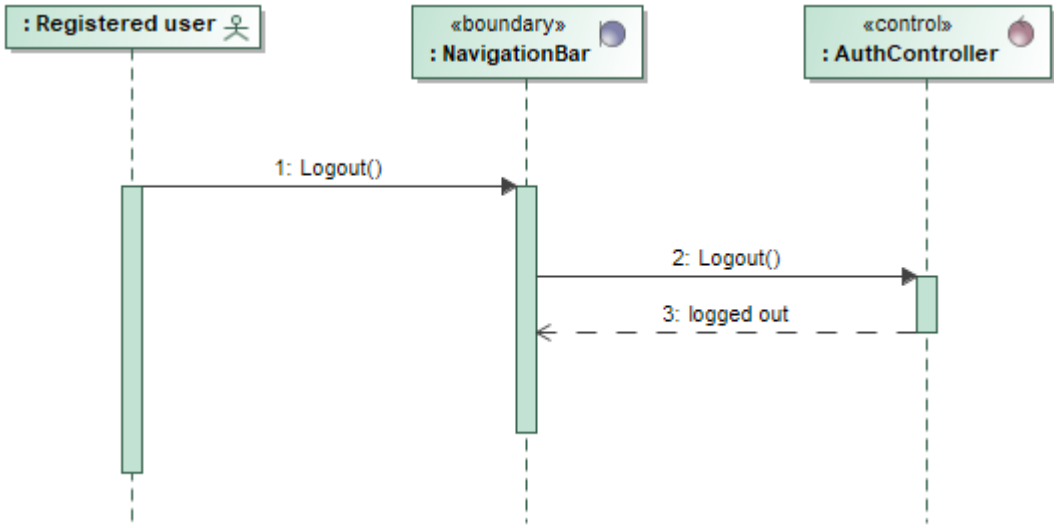


Diagram 13: Log out

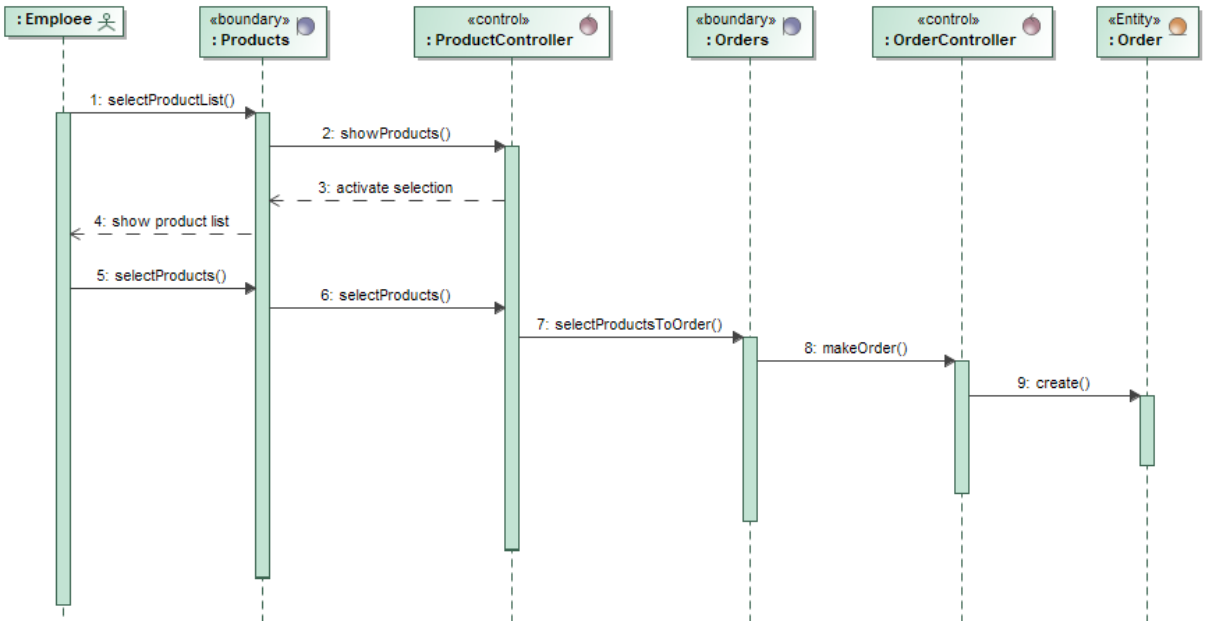


Diagram 14: Order from suppliers

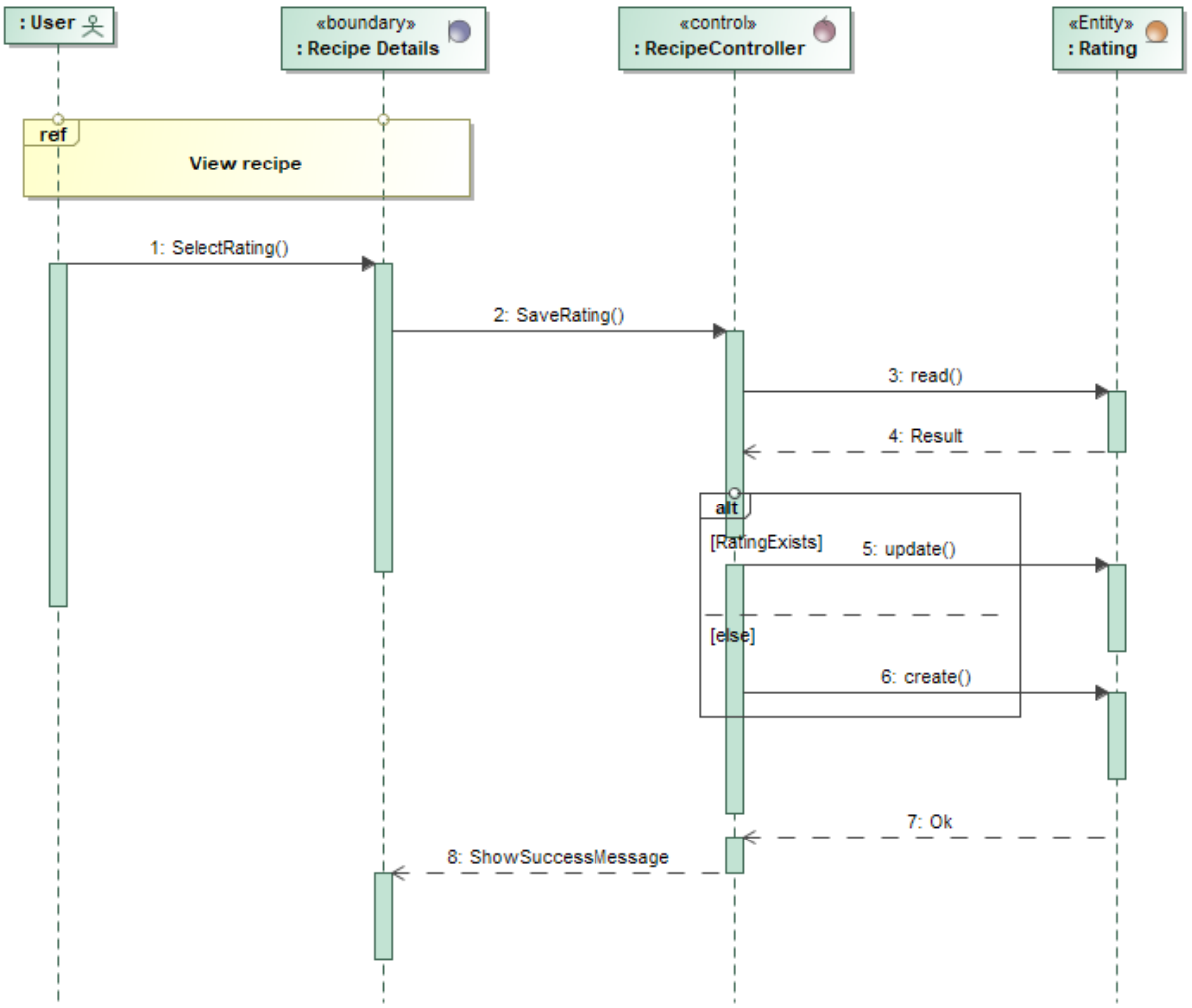


Diagram 15: Rate recipe

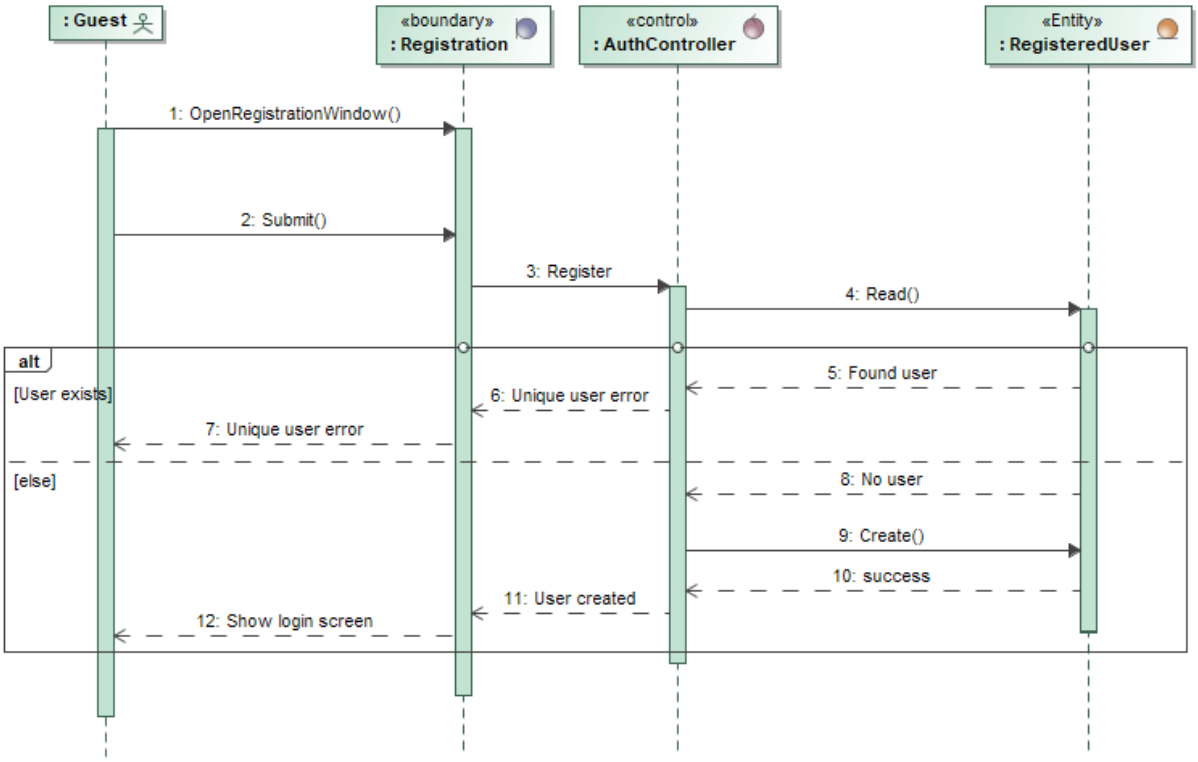


Diagram 16: Register

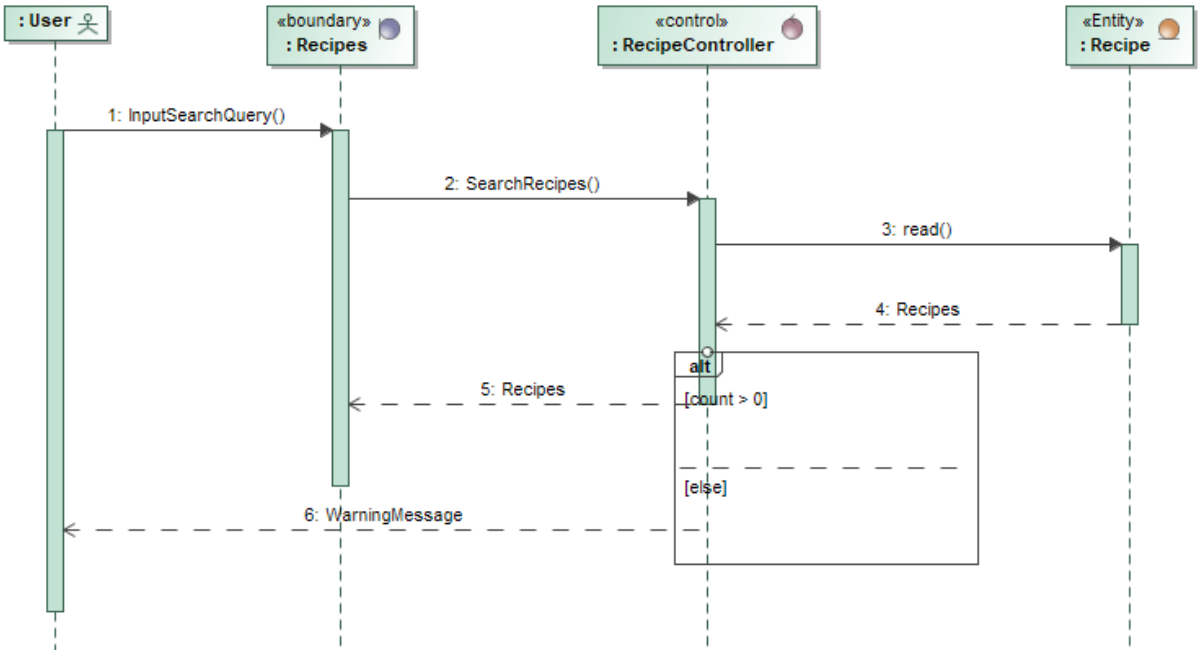


Diagram 17: Search recipes

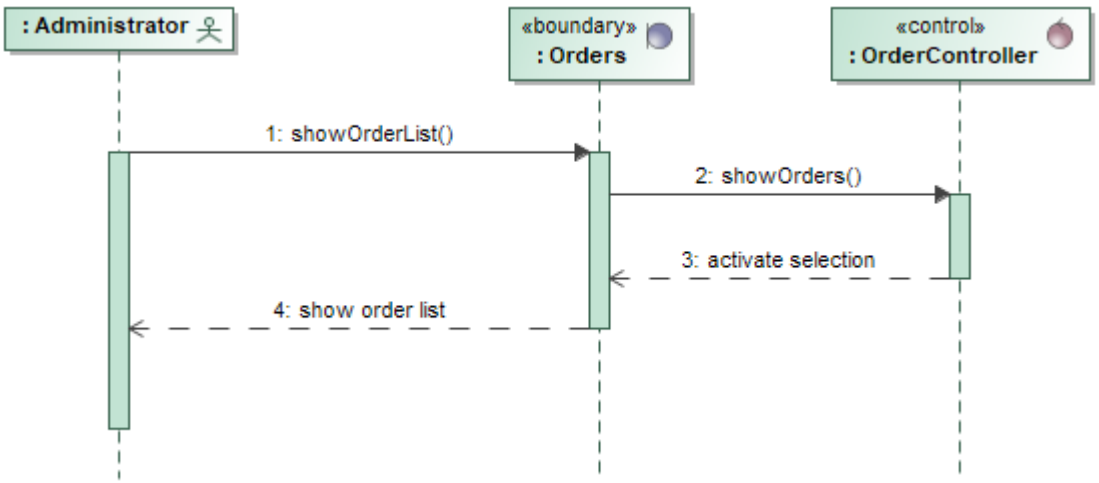


Diagram 18: View orders

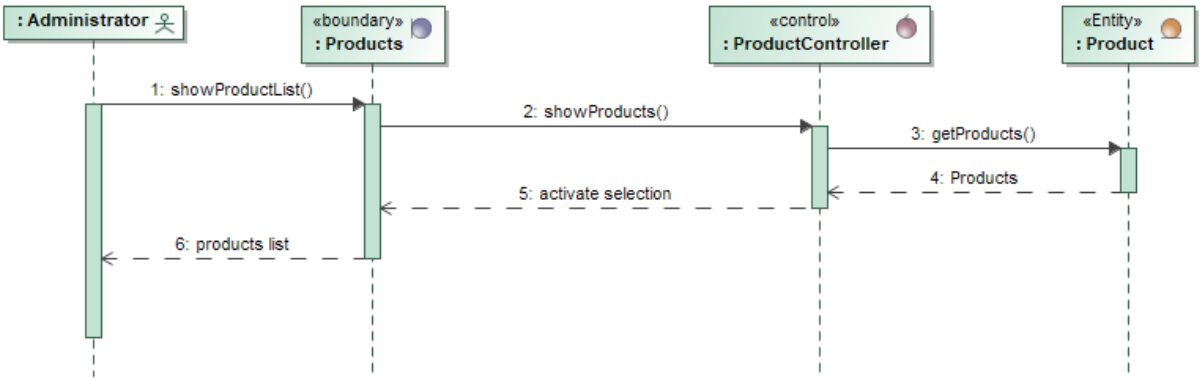


Diagram 19: View products

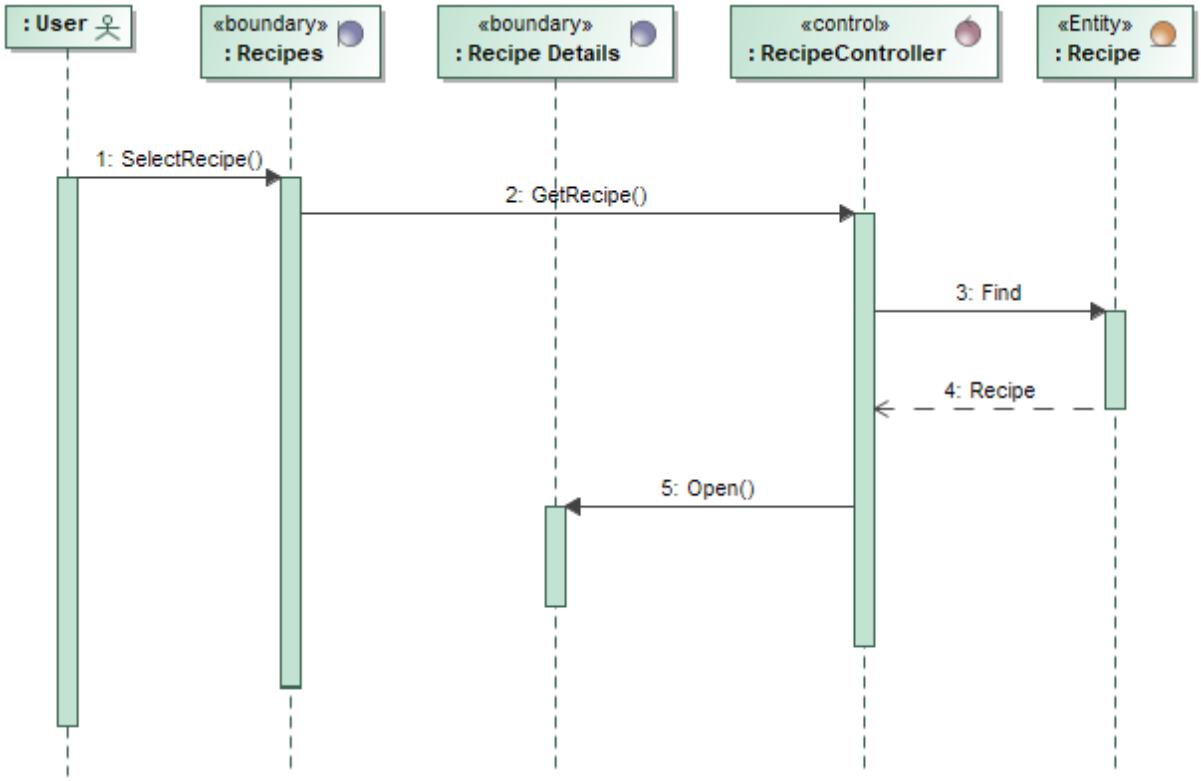
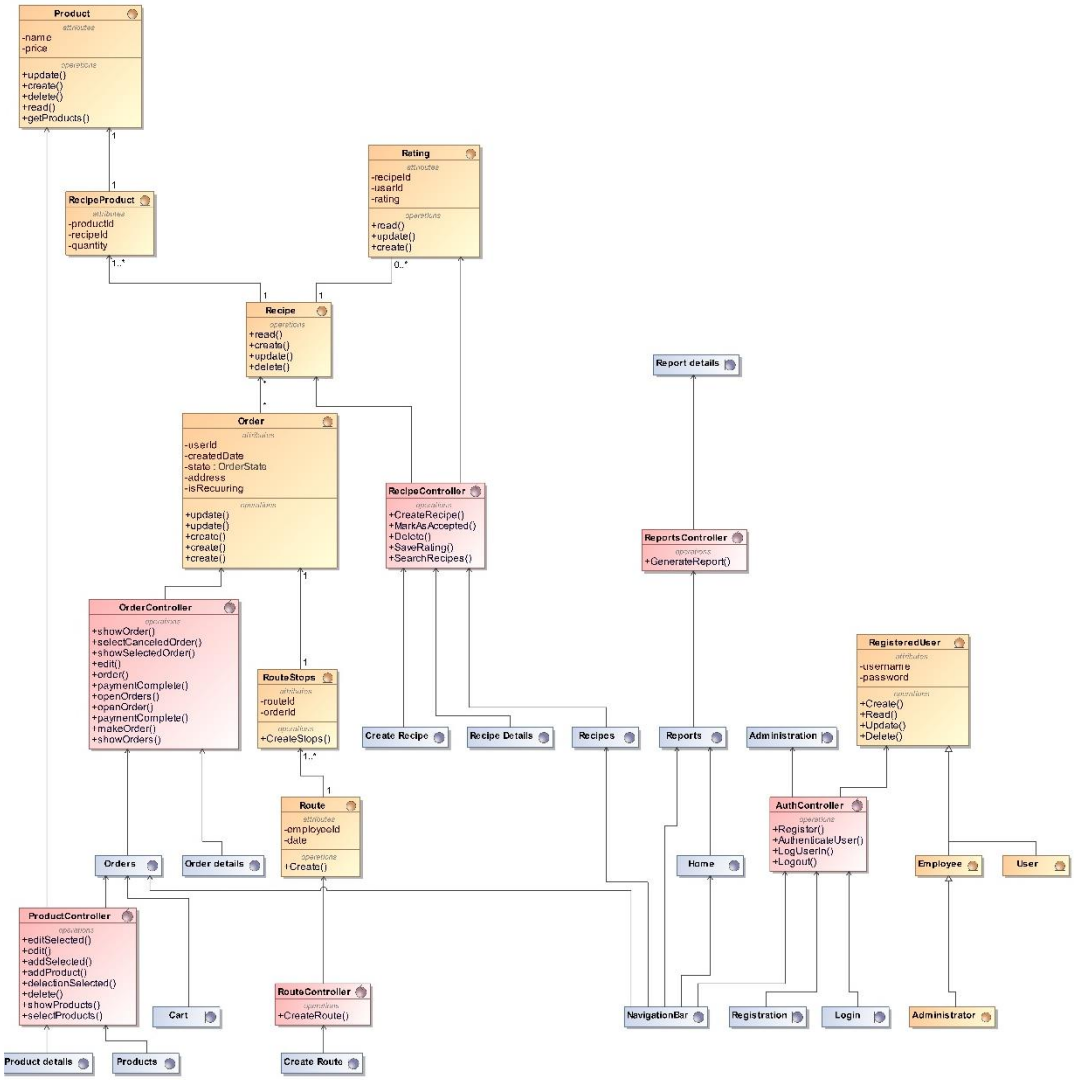


Diagram 20: View recipe

4.3. System class diagram

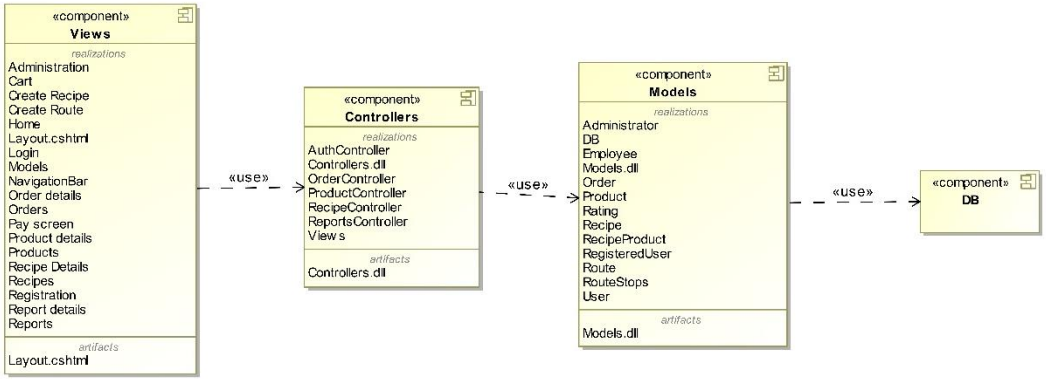
Our class diagram contains all of the classes, entities and boundries.



Here color coded entities are orange, controls pink and boundaries blue.

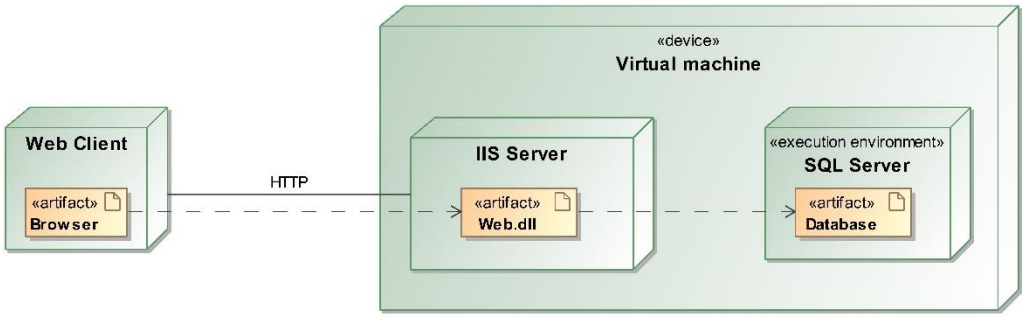
5. System realization

5.1. Component diagram



Our component diagram nearly matches package diagram and here we can actually see what artifacts are going to be generated to run our application.

5.2. Deployment diagram



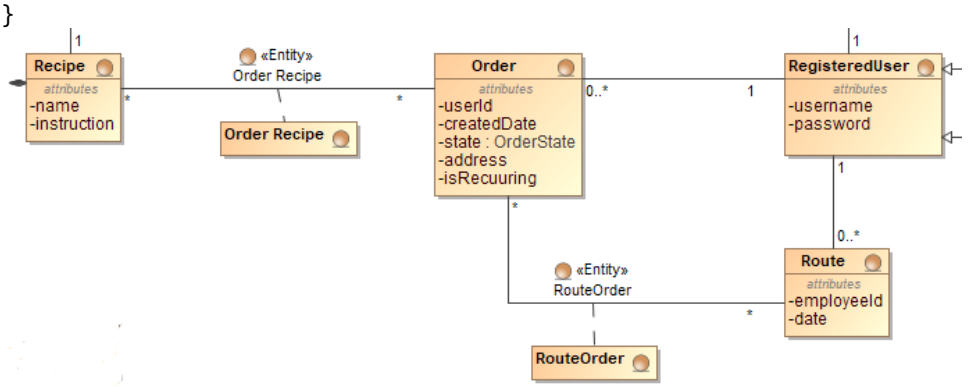
In the deployment diagram we can see a diagram how our system is going to operate after being deployed.

6. System realization

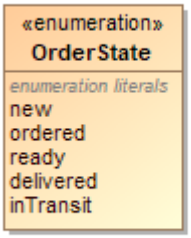
6.1. Aggregation, Composition, Dependency

```
public class Order
{
    public int ID { get; set; }
    public DateTime CreateDate { get; set; }
    public OrderState State { get; set; }
    public string Address { get; set; }
```

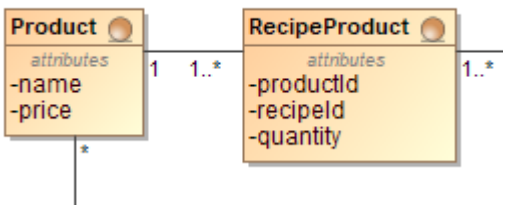
```
        public IList<Recipe> Recipes { get; set; } = new List<Recipe>();
        public Route Route { get; set; }
        public RegisteredUser User { get; set; }
    }
```



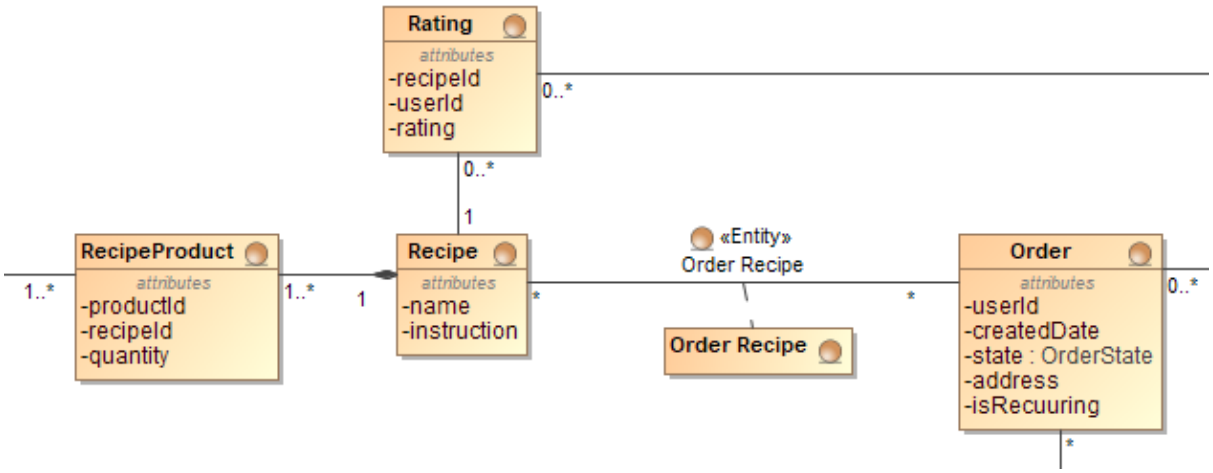

```
    public enum OrderState
    {
        New,
        Ordered,
        Ready,
        Delivered,
        InTransit
    }
```



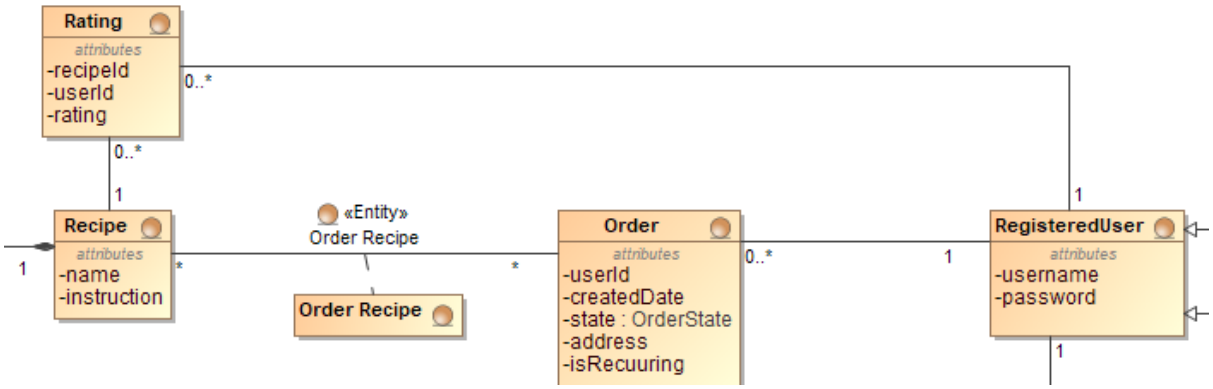

```
    public class Product
    {
        public int ID { get; set; }
        public string Name { get; set; }
        public decimal Price { get; set; }

        public IList<RecipeProduct> RecipeProducts { get; set; } = new
List<RecipeProduct>();
    }
```



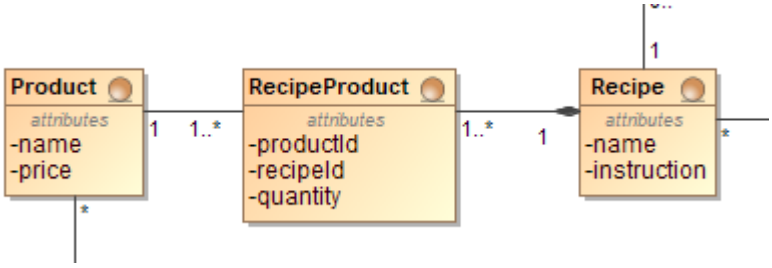
```
public class Recipe
    {
        public int ID { get; set; }
        public string Name { get; set; }
        public string Instruction { get; set; }
        public bool IsAccepted { get; set; }

        public IList<Order> Orders { get; set; } = new List<Order>();
        public IList<RecipeProduct> RecipeProducts { get; set; } = new
List<RecipeProduct>();
        public IList<Rating> Ratings { get; set; } = new List<Rating>();
    }
```



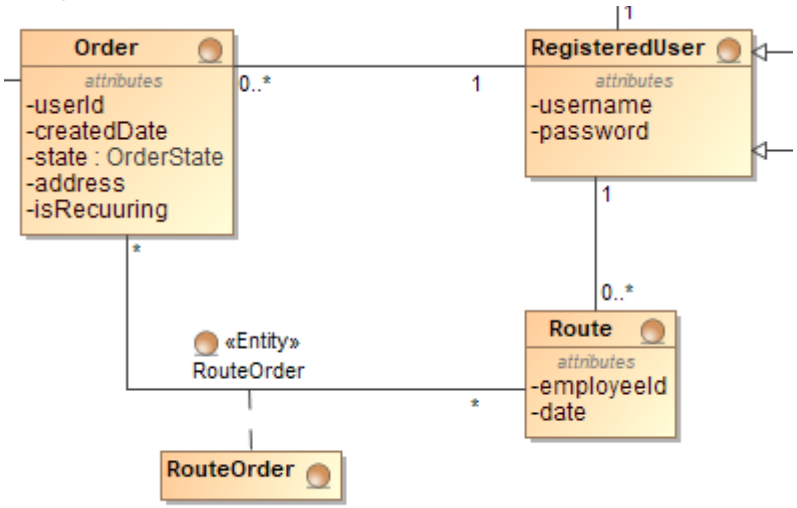

```
    public class Rating
    {
        public int ID { get; set; }
        public RegisteredUser User { get; set; }
        public Recipe Recipe { get; set; }
        [Column("Rating")]
        public int Amount { get; set; }
    }
```



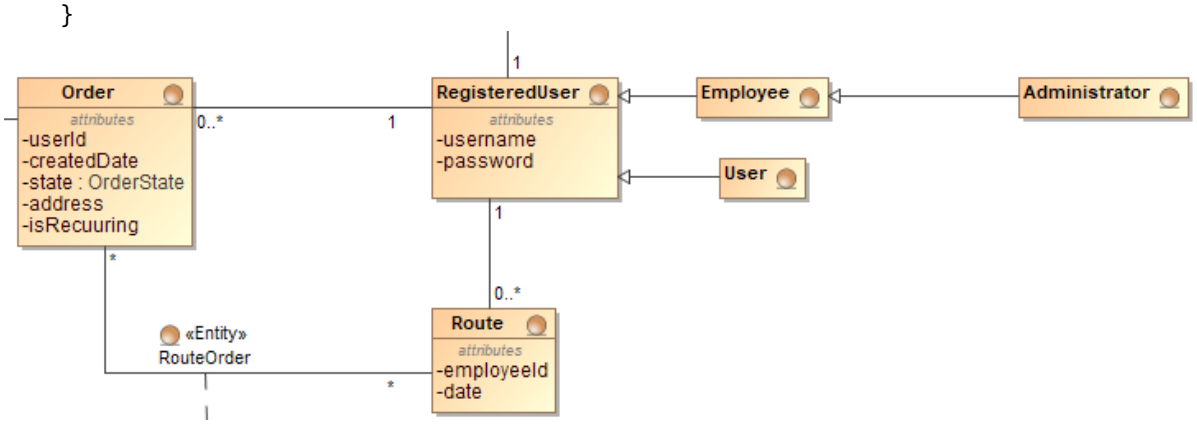

```
public class RecipeProduct
    {
        public int ID { get; set; }
        public Recipe Recipe { get; set; }
        public Product Product { get; set; }
        public double Quantity { get; set; }
    }
```




```
public class Route
{
    public int ID { get; set; }
    public DateTime Date { get; set; }
    public RegisteredUser Employee { get; set; }

    public IList<Order> Orders { get; set; } = new List<Order>();
}
```

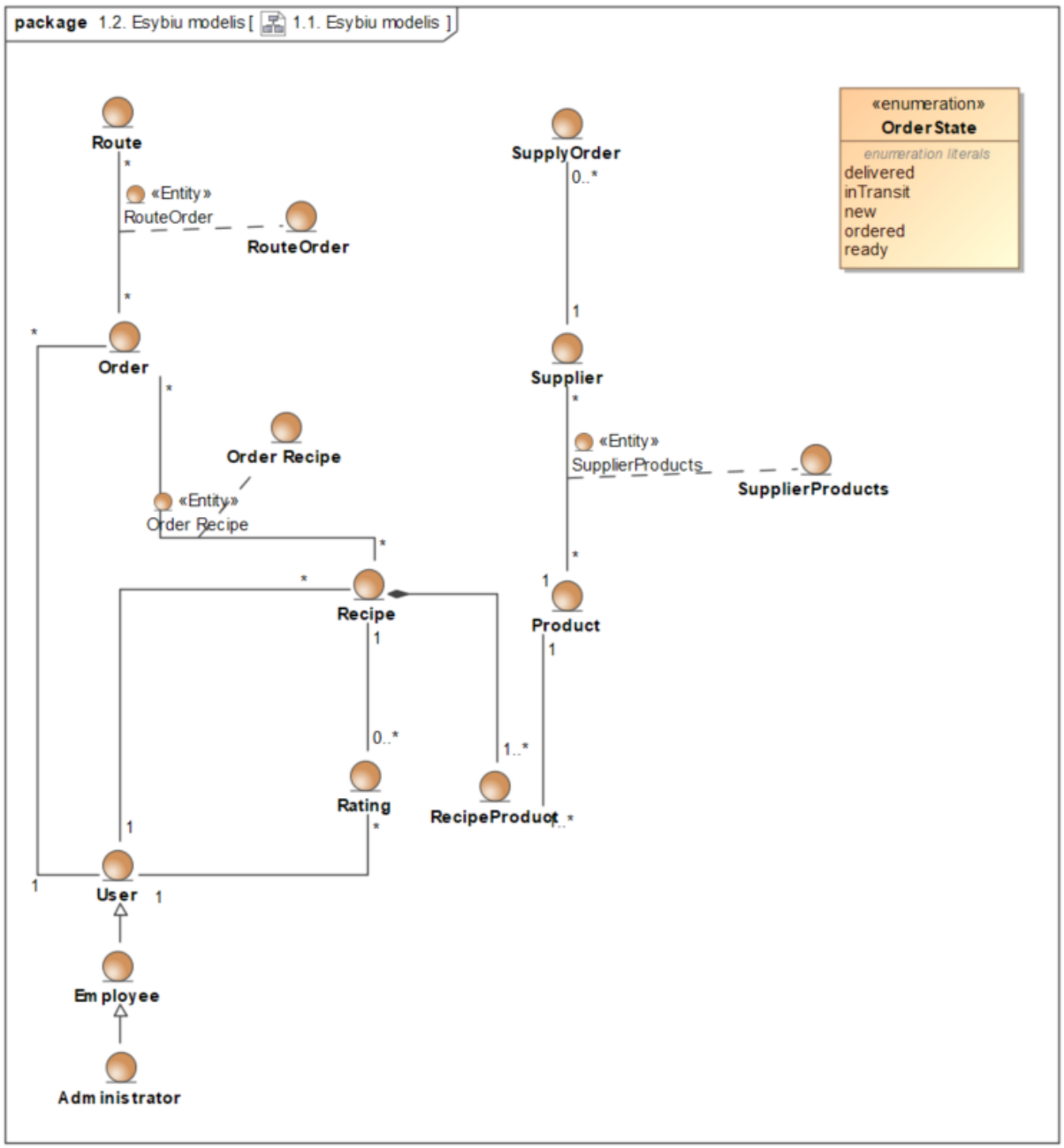



```
public class RegisteredUser
{
    public int ID { get; set; }
    public string Username { get; set; }
    public string Password { get; set; }
    public bool IsAdmin { get; set; }
    public bool IsEmployee { get; set; }

    public IList<Order> Orders { get; set; } = new List<Order>();
    public IList<Route> Routes { get; set; } = new List<Route>();
}
```

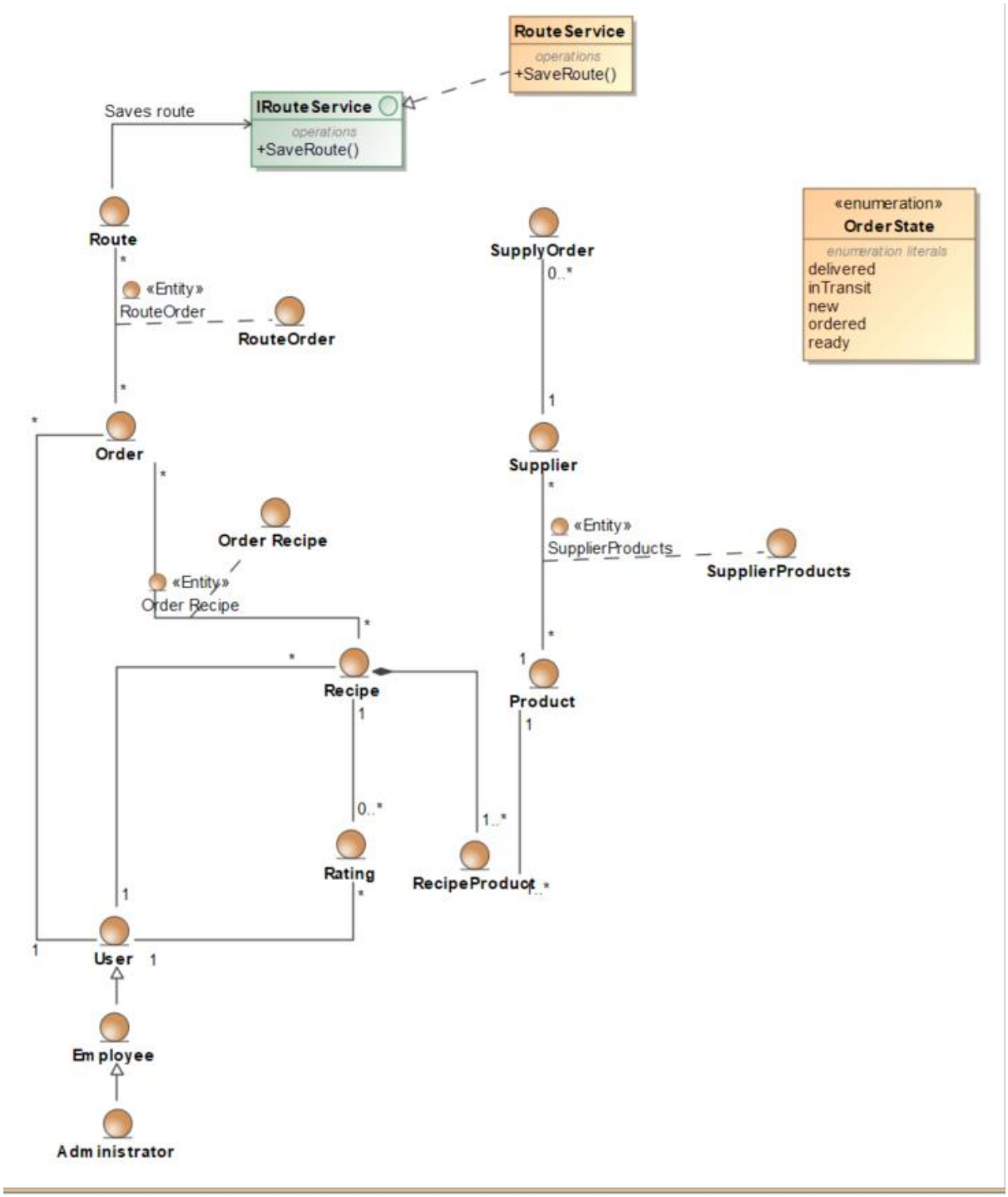


6.2. Dependency inversion

Before dependency inversion:



After dependency inversion:



6.3. Code generation

Forward engineering looks like this:

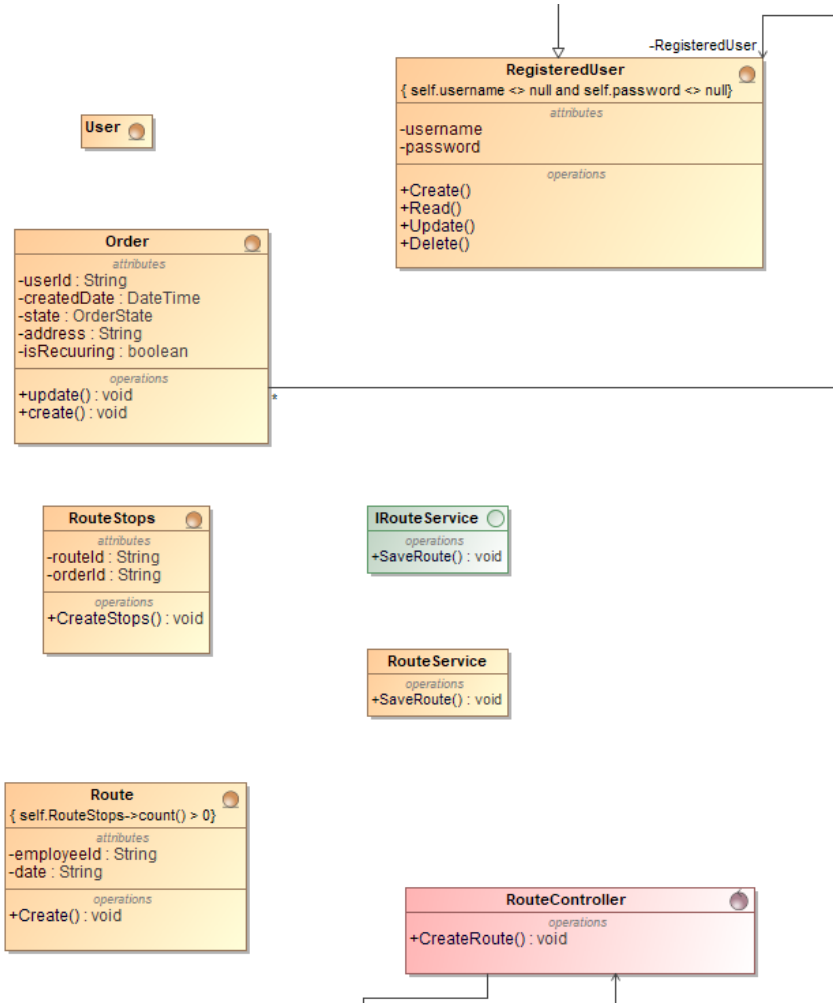


Figure 1 Forward engineering class model

```
/**
 * @(#) IRouteService.cs
 */

namespace ProjektoModelis.ReikalavimuModelis.EsybiuModelis
{
    public interface IRouteService
    {
        void SaveRoute( );
    }
}
```

Figure 2 Route interface - forward engineering

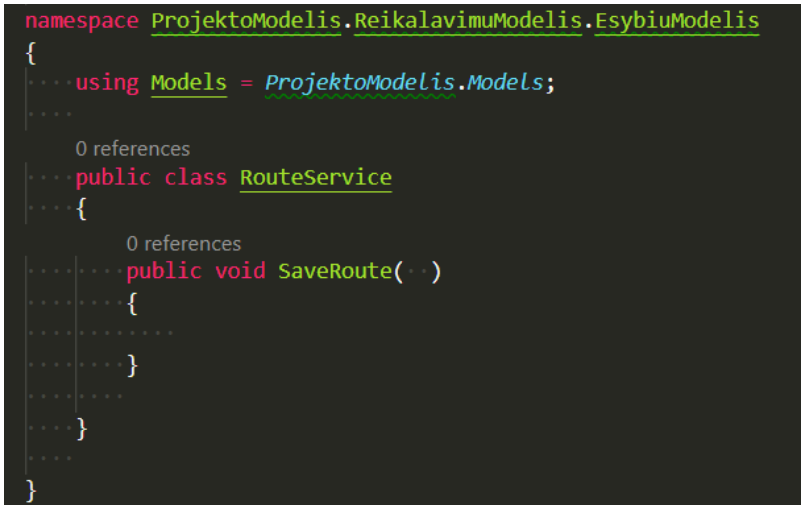


Figure 3 Route service - forward engineering

Figure 4 User entity - forward engineering

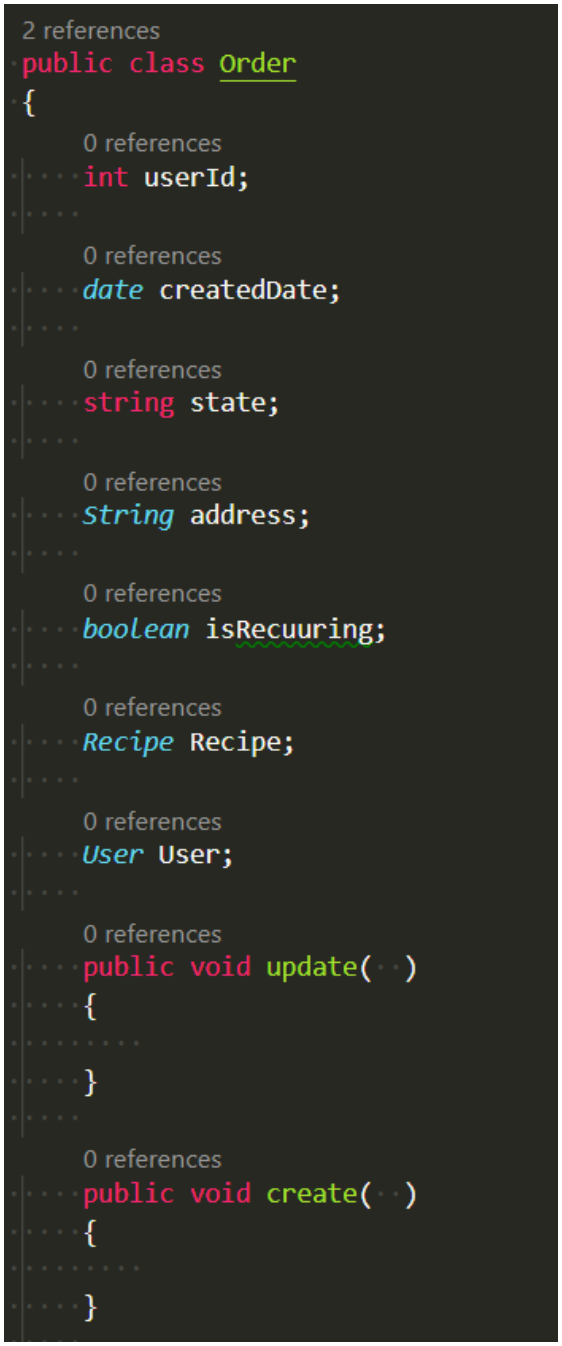


Figure 5 Order entity - forward engineering

After reverse engineering

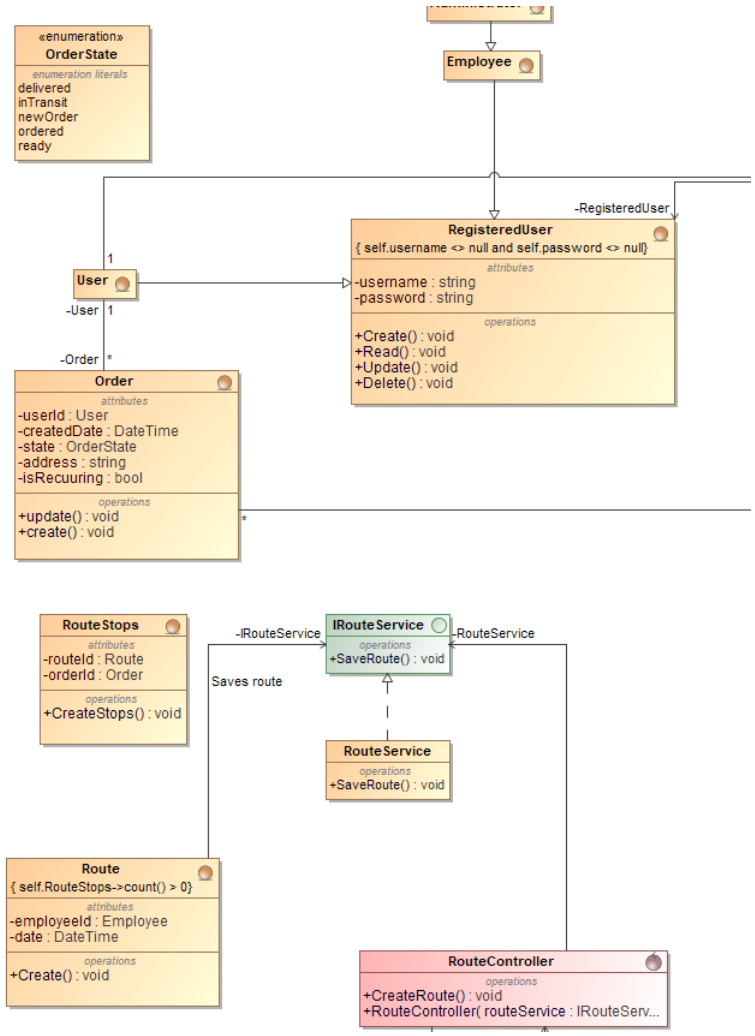


Figure 6 Class diagram - reverse engineering

```
namespace ProjektoModelis.ReikalavimuModelis.EsybiuModelis
{
    using Models = ProjektoModelis.Models;

    0 references
    public class RouteService : IRouteService
    {
        0 references
        public void SaveRoute( )
        {

        }

    }
}
```

Figure 7 Route service - reverse engineering

```
2 references
public class Order
{
    0 references
    User userId;

    0 references
    DateTime createdDate;

    0 references
    ProjektoModelis.ReikalavimuModelis.EsybiuModelis.OrderState state;

    0 references
    string address;

    0 references
    bool isRecuuring;

    0 references
    Recipe Recipe;

    0 references
    User User;

    0 references
    List<RouteStops> RouteStops;

    0 references
    public void update( )
    {

    }

    0 references
    public void create( )
    {

    }
```

Figure 8 Order entity - reverse engineering

```
namespace ProjektoModelis.Models
{
    2 references
    public class User : RegisteredUser
    {
        0 references
        ProjektoModelis.ReikalavimuModelis.EsybiuModelis.Recipe Recipe;

        0 references
        Order Order;

        0 references
        Rating Rating;

    }
}
```

Figure 9 User entity - reverse engineering

6.4. OCL

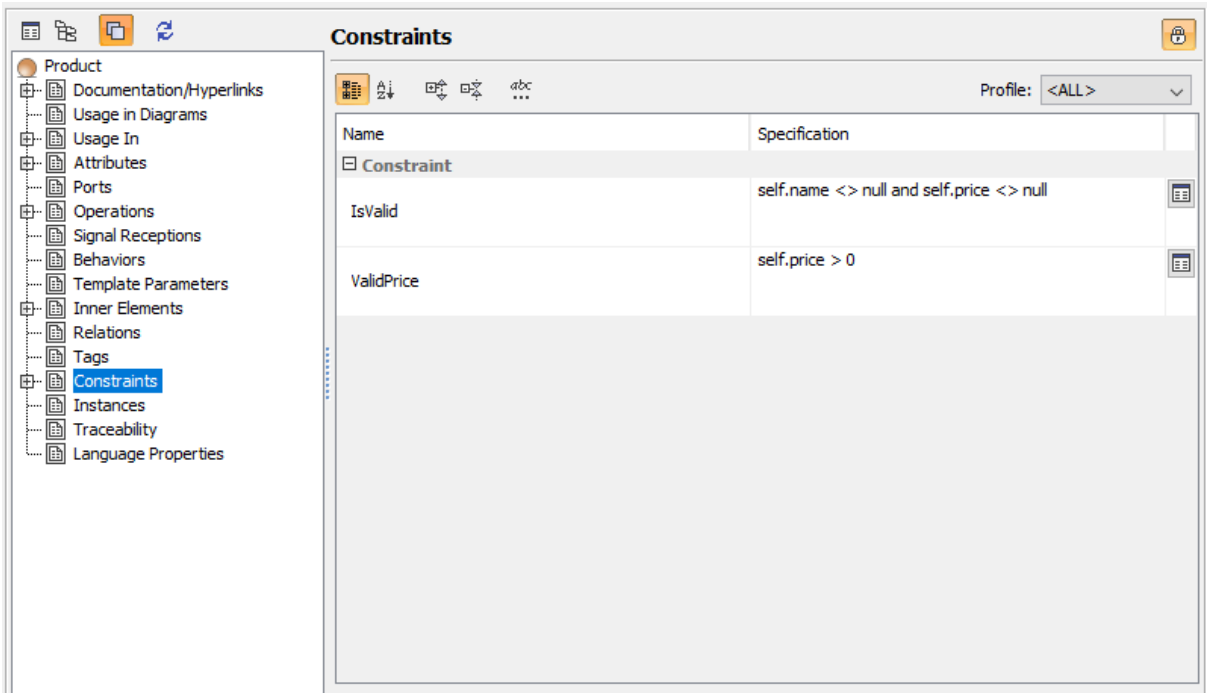


Figure 10 Product constraints

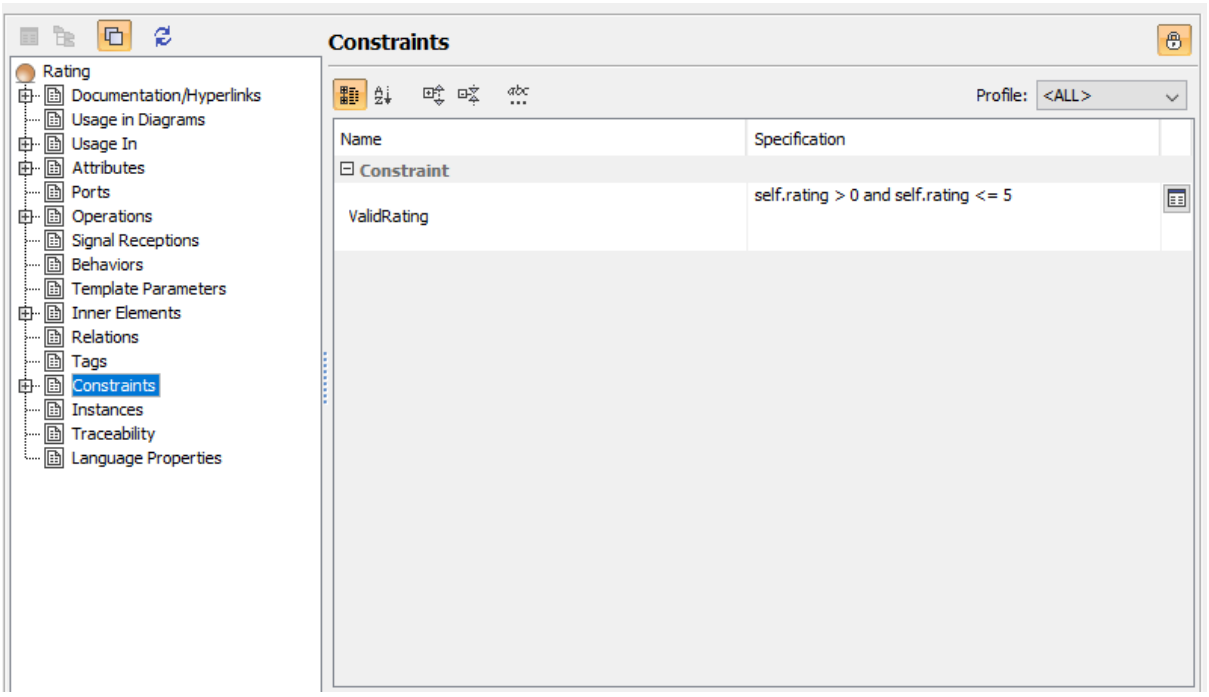


Figure 11 Rating constraints

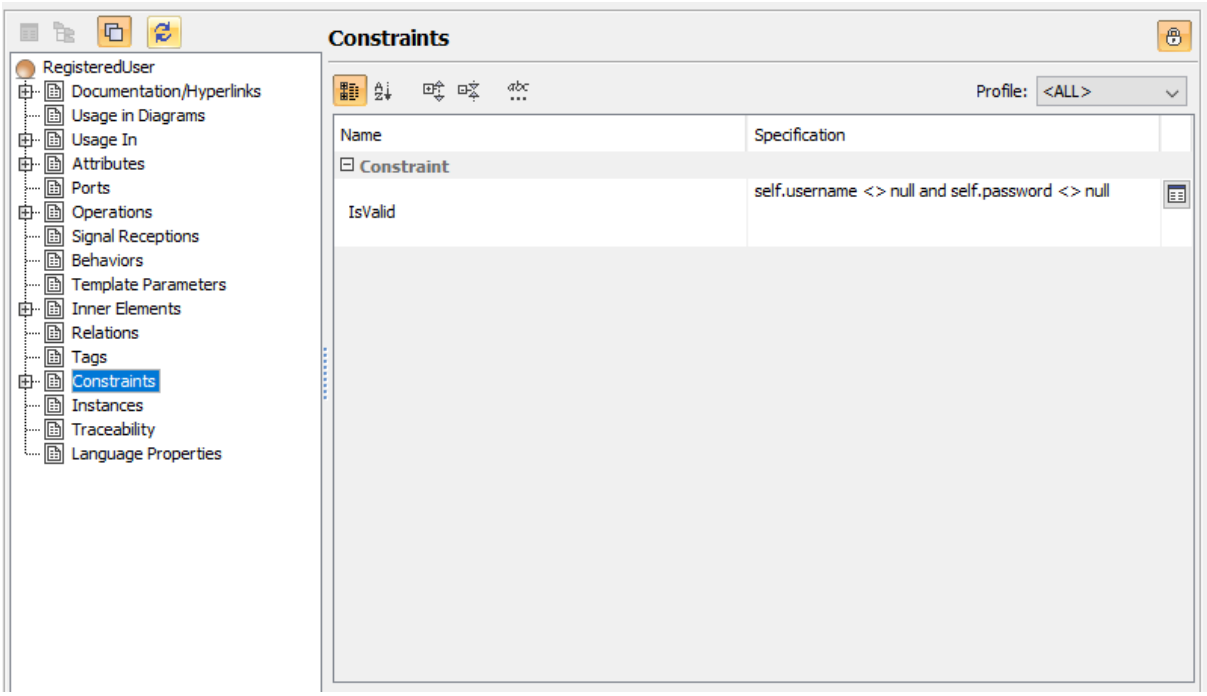


Figure 12 User constraints

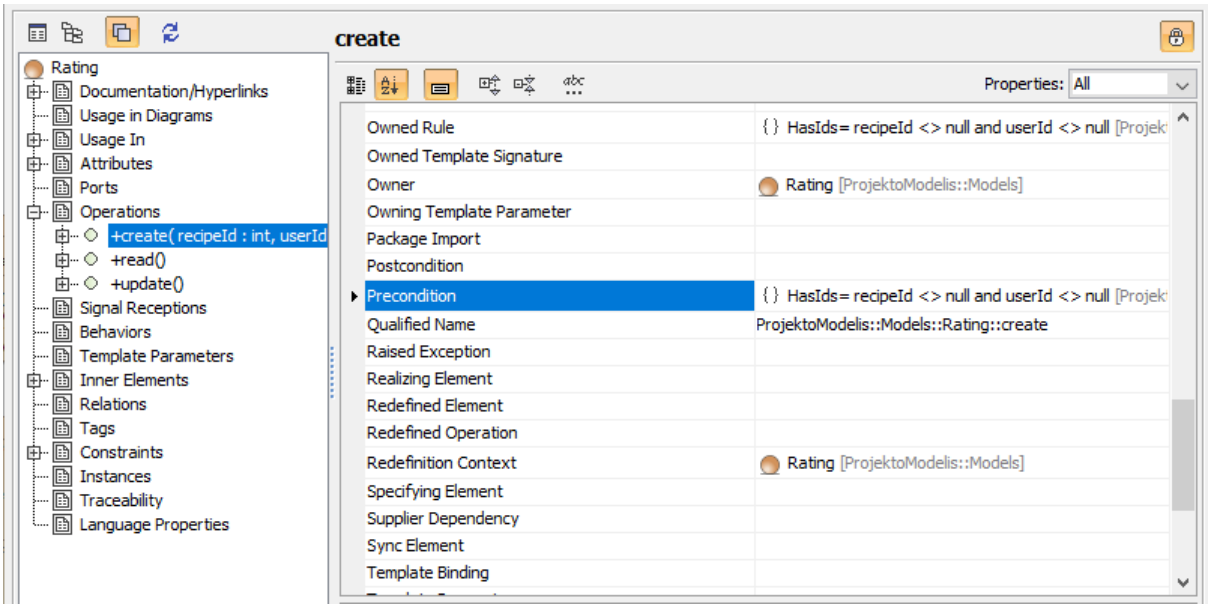


Figure 13 Rating createion pre-condition

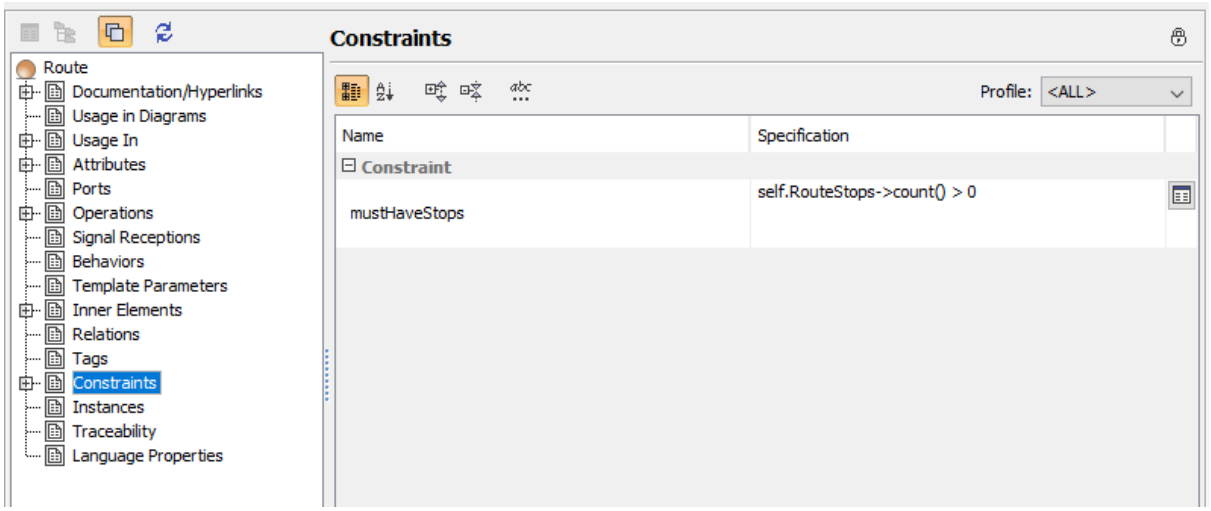


Figure 14 Route constraints

6.5. Use case diagram code realization

We are going to demonstrate, recipes use case code „Create recipe“, „View recipe“, „Rate recipe“, „Add recipe to cart“ realizations.

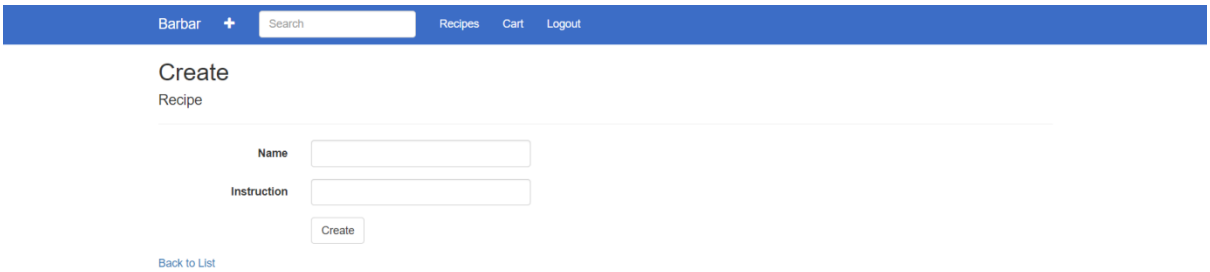


Figure 15. Create ne recipe window.

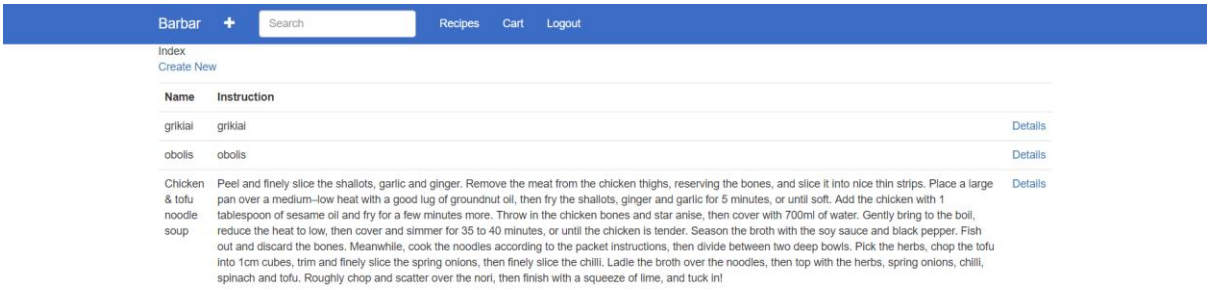


Figure 16. View recipe list window.

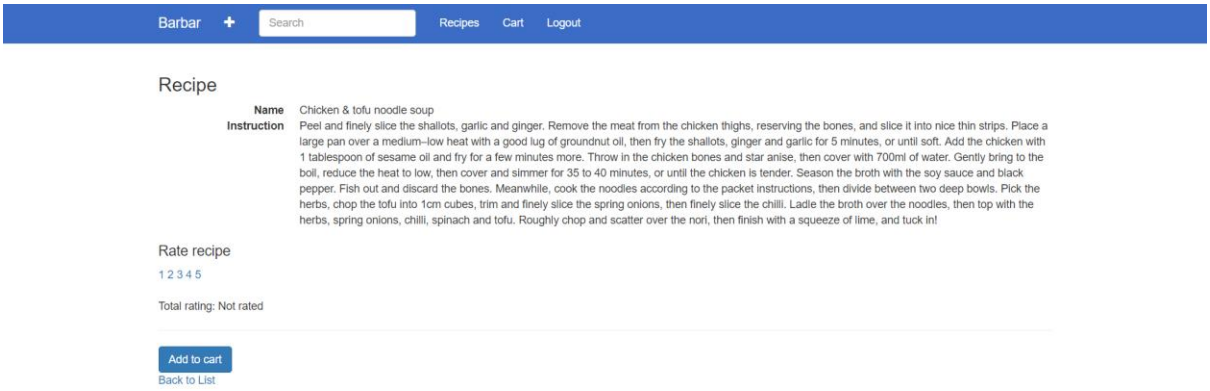


Figure 17. View recipe window.

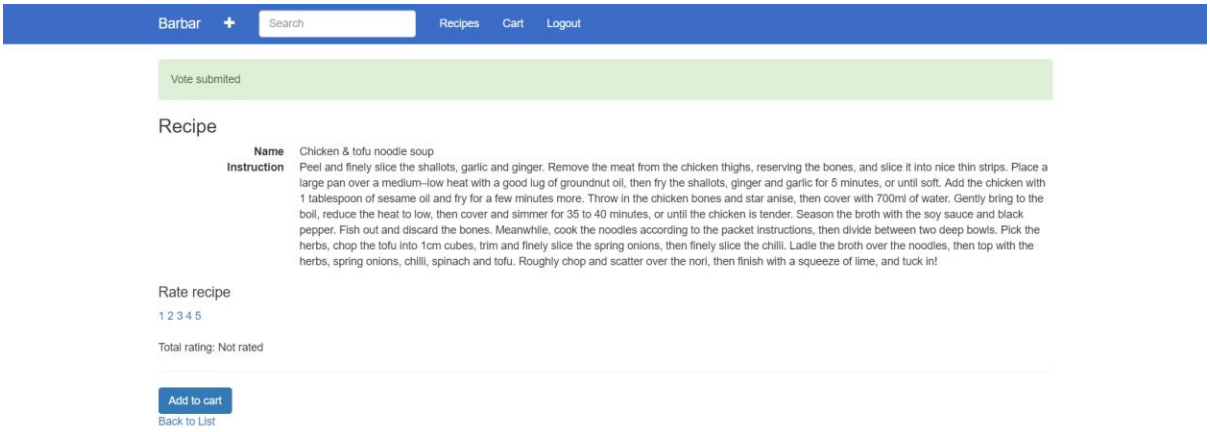


Figure 18. Rate recipe vote submited.

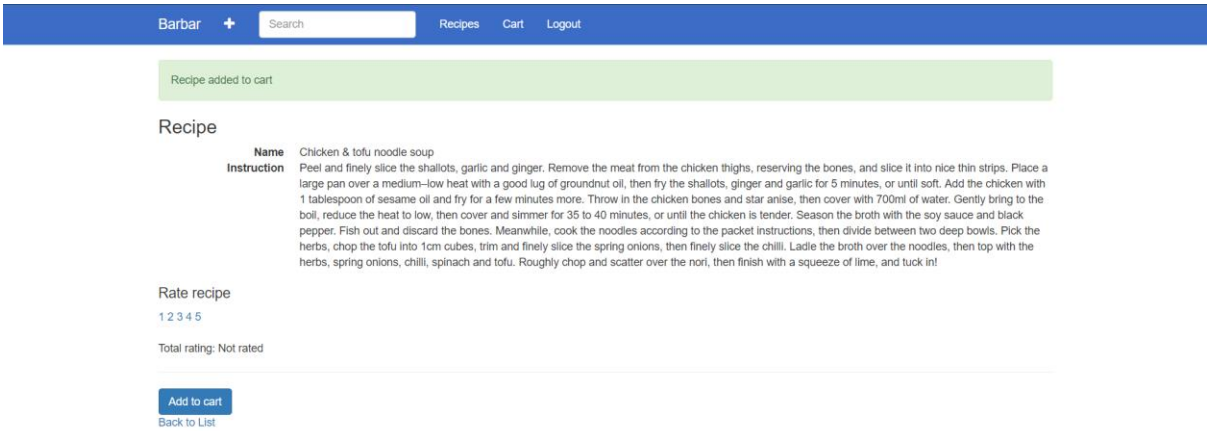


Figure 19. Recipe added to cart notification.

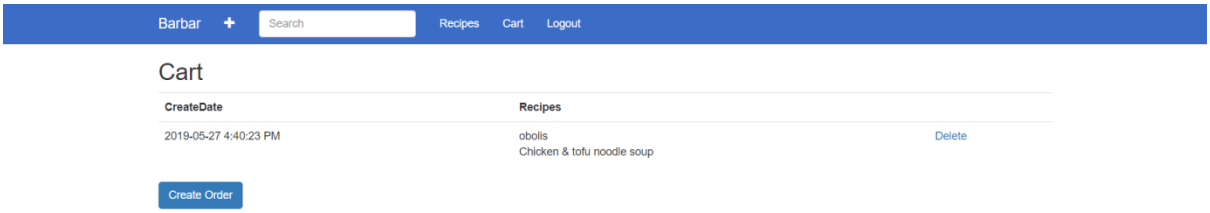


Figure 20. View recipe cart window.