



**National University of Computer and Emerging Sciences**

## **Software Design and Architecture**

---

### **Project- Online Food Delivery System**

<b>STUDENTS NAME:</b>	<b>Zain Javed, Muhammad Waqas Akhtar, Kazim Asif</b>
<b>REGISTRATION NUMBERS:</b>	<b>20I-0522, 20I-0525, 20I-2310</b>
<b>DEGREE PROGRAM:</b>	<b>BSSE</b>
<b>Section:</b>	<b>Q</b>
<b>SUBJECT NAME:</b>	<b>Software Design and Architecture</b>
<b>DATE OF SUBMISSION:</b>	<b>June 26, 2022.</b>
<b>SUBMITTED TO:</b>	<b>Dr. Uzair Khan</b>

**STUDENT SIGNATURE:**

Zain Javed, Muhammad Waqas Akhtar, Kazim Asif

**MARKS:**

---

**REMARKS:**

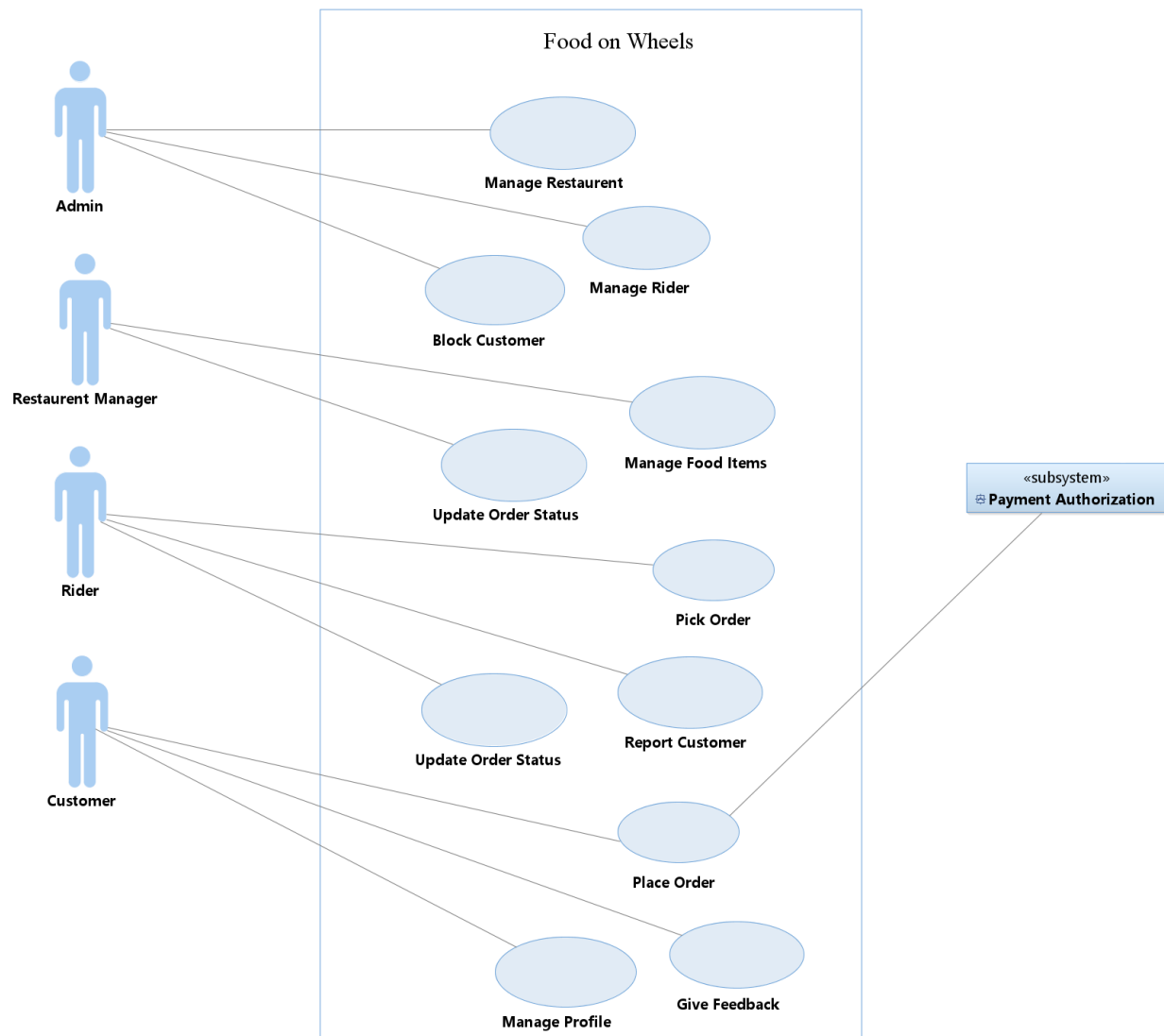
## Contents

Software Design and Architecture .....	1
Use Case Diagram.....	5
High Level Use Cases .....	5
UC01: Use case: Add Restaurant.....	5
UC02: Use case: Delete Restaurant .....	5
UC03: Use case: Add Food.....	6
UC04 Use case: Delete Food .....	6
UC05: Use case: Update Food .....	6
UC06: Use case: Block Customer.....	6
UC07: Use case: Manage Riders .....	6
UC08: Use case: Place Order.....	6
UC09: Use case: Give Feedback.....	7
UC10 .....	7
UC11: Use case: Report Customer .....	7
UC12: Use case: Update Status .....	7
UC13: Use case: Update Status .....	7
UC14: Use case: Manage Profile .....	7
Expanded Use Cases .....	8
UC01: Use case: Add Restaurant.....	8
UC02: Use case: delete Restaurant .....	8
UC03: Use case: Add Food.....	9
UC04 Use case: Delete Food .....	9
UC05: Use case: Update Food .....	10
UC06: Use case: Block Customer.....	11
UC07: Use case: Manage Riders .....	12
UC08: Use case: Place Order.....	12
UC09: Use case: Give Feedback.....	13
UC10: Use case: Update status .....	13
UC11: Use case: Pick Order .....	14

UC12: Use case: Update status .....	15
UC13: Use case: Report Customer .....	16
UC14: Use case Manage Profile .....	16
Domain Model .....	17
System Sequence Diagram .....	19
1. Manage Food .....	19
2. Manage Restaurants .....	20
3. Manage Riders .....	21
4. Block Customer .....	21
5. Place Order.....	22
6. Pick Order .....	24
7. Give Feedback .....	25
7.1 Give Feedback (Customer) .....	25
7.2 Show Feedback (Rider).....	25
8. Report Customer .....	26
8.1 Report Customer (Rider).....	26
8.2 Block Reported Customer (Admin) .....	27
9. Manage Profile (Customer).....	27
Sequence Diagram .....	27
1. Manage Riders .....	28
1.1. Add Rider .....	28
1.2. Delete Rider .....	28
2. Manage Restaurant.....	30
2.1. Add Restaurant.....	30
2.2. Display Restaurant .....	31
2.3. Delete Restaurant .....	32
3. Manage food .....	33
3.1 Add food .....	33
3.2 Remove food.....	33
3.3 Update food.....	34
3.4 Show Menu .....	34
4. Block Customer .....	35

4.1.	Display Violators .....	35
4.2.	Block Customer.....	36
5.	Pick Order .....	37
5.1.	Select Order .....	37
5.2.	Display Order.....	37
5.3	Display Picked Orders .....	38
5.4	Deliver Order .....	39
6.	Place Order.....	39
6.1	Show Restaurant .....	39
6.2	Select Restaurant.....	40
6.3	Show Food.....	40
6.4	Place Order.....	42
6.5	Display Bill .....	42
6.6	Get Payment Method .....	43
6.7	Pay by Card .....	43
6.8	Pay by cash .....	44
7.	Give Feedback .....	44
5.1	Show Riders .....	44
5.2	Give feedback .....	46
5.3	show Feedback.....	47
6.	Report Customer .....	47
6.1	Show customers list of Rider .....	47
6.2	Report Customer .....	48
6.3	Reported Customers List.....	48
6.4	Show Reasons List.....	49
6.5	Block reported Customers.....	50
7.	Manage Profile (Customer).....	50

## Use Case Diagram



## High Level Use Cases

### UC01: Use case: Add Restaurant

**Actor:** Admin

**Overview:** Firstly, the restaurant Manager opens the online food delivery System and gives the name, price, category and description of the food item. At last, he added the food.

### UC02: Use case: Delete Restaurant

**Actor:** Admin

**Overview:** Firstly, an admin opens the food delivery system and gives the instruction to system to display all the restaurants. Secondly, he selects the restaurant from the list and deletes it.

### **UC03: Use case: Add Food**

**Actor:** Restaurant Manager

**Overview:** Firstly, the restaurant Manager opens the online food delivery System and gives the name, price, category and description of the food item. At last, he added the food item.

### **UC04 Use case: Delete Food**

**Actor:** Restaurant Manager

**Overview:** Firstly, the restaurant Manager opens the online food delivery System and gives the instruction to the system to display all the foods of his restaurant. Secondly, he selects the food from the list of foods and deletes it.

### **UC05: Use case: Update Food**

**Actor:** Restaurant Manager

**Overview:** Firstly, the restaurant Manager opens the online food delivery System and gives the instruction to system to display all the food of his restaurant. Secondly, he selects the food from the list of foods. Thirdly, he updates the food details and save it.

### **UC06: Use case: Block Customer**

**Actor:** Admin

**Overview:** Firstly, the Admin opens the online food delivery System and gives the instruction to the System to display the list of customers who violates the conditions. Secondly, he selects the customers who violate the conditions. Thirdly, he blocks them from their restaurant.

### **UC07: Use case: Manage Riders**

**Actor:** Admin

**Overview:** Firstly, an admin opens the online food delivery System and adds the rider by giving details of rider or deletes the rider by selecting the rider from the list of riders. Secondly, he saves all the updates in the system.

### **UC08: Use case: Place Order**

**Actor:** Customer, Payment Authentication.

**Overview:** Firstly, a customer opens the online food delivery System and selects the restaurant. Secondly, he selects the food from the menu to eat. Thirdly, he selects the quantity of food and payment method. At the end, he places the order.

#### **UC09: Use case: Give Feedback**

**Actor:** Customer

**Overview:** The customer gives the feedback of the rider by selecting riders from the list who delivered them the orders. In addition to, he gives all the details in the feedback and uploads it on the system.

#### **UC10: Use case: Pick Order**

**Actor:** Rider

**Overview:** Firstly, the rider accesses the system to view the order placed by customers by rider's credentials. Then the rider selects the order, which he wants to pick and deliver. The order picked by the rider cannot see by other riders.

#### **UC11: Use case: Report Customer**

**Actor:** Rider

**Overview:** Firstly, the rider delivers the order to customer and after delivery, the rider can report the customer and give the reason of the report. After that, admin checks the reason and decide to block the customer or not.

#### **UC12: Use case: Update Status**

**Actor:** Rider

**Overview:** Firstly, the rider delivers the order to customer and after delivery, the rider collect the amount of that order if customer chooses cash on delivery method. Rider updates the status to deliver.

#### **UC13: Use case: Update Status**

**Actor:** Restaurant Manager

**Overview:** Firstly, after placing the order by customer, the manager updates the status of order "prepared", then "on the way".

#### **UC14: Use case: Manage Profile**

**Actor:** Customer

**Overview:** Firstly, the customer changes his/her details by going into manage profile section and then saves all the new details.

## **Expanded Use Cases**

### **UC01: Use case: Add Restaurant**

**Actor:** Admin

**Purpose:** Add restaurant to the list of restaurants.

**Overview:** Firstly, an admin opens the online food delivery System and adds the restaurant by giving the name, location and category of the restaurant. Lastly, he saves the restaurant.

**\*Use cases:** Login (Admin)

### **Typical Course of Events**

<b>Actor Action</b>	<b>System Response</b>
1. An admin opens the system and adds the new restaurant by giving the name, category and location of the restaurant.	
2. He added the restaurant in the restaurant list.	
	3. System saved the details of the restaurant.

### **UC02: Use case: delete Restaurant**

**Actor:** Admin

**Purpose:** Delete restaurant from the list of restaurants.

**Overview:** Firstly, an admin opens the food delivery system and gives the instruction to system to display all the restaurants. Secondly, he selects the restaurant from the list and deletes it.

**\*Use cases:** Login (Admin), Add Restaurant.

### **Typical Course of Events**



Actor Action	System Response
1. An admin opens the system and gives the instruction to system to display all the restaurants.	
	2. System displays the restaurant list.
3. He selects the restaurant from the list and deletes it.	
	4. System deleted the restaurant details.

### UC03: Use case: Add Food

**Actor:** Restaurant Manager

**Purpose:** Add new food to his restaurant.

**Overview:** Firstly, the restaurant Manager opens the online food delivery System and gives the name, price, category and description of the food item. At last, he added the food.

**\*Use cases:** Login (Restaurant Manager). Add Restaurant (Admin).

### Typical Course of Events

Actor Action	System Response
1. The Restaurant Manager opens the system and go for manage food item.	
2. Manager selects to add item and gives the name, price, category and description of the food item.	
	3. System saves the details of the new food to the selected category in the menu.

### Alternative Course:

**Line 2:**

### UC04 Use case: Delete Food

**Actor:** Restaurant Manager

**Purpose:** Delete food from the restaurant.

**Overview:** Firstly, the restaurant Manager opens the online food delivery System and gives the instruction to the system to display all the foods of his restaurant. Secondly, he selects the food from the list of foods and deletes it.

**\*Use cases:** Login (Restaurant Manager), Add Food

### Typical Course of Events

Actor Action	System Response
1. A Restaurant Manager opens the system and open the menu of his restaurant.	
	2. System displays all the foods of his restaurant.
3. Restaurant manager selects the food from the list of foods and deletes it.	
	4. System deletes the food from respected restaurant.

### UC05: Use case: Update Food

**Actor:** Restaurant Manager

**Purpose:** Update the food price in his restaurant.

**Overview:** Firstly, the restaurant Manager opens the online food delivery System and gives the instruction to system to display all the food of his restaurant. Secondly, he selects the food from the list of foods. Thirdly, he updates the food details and save it.

**\*Use cases:** Login (Restaurant Manager), Add Food.

### Typical Course of Events

Actor Action	System Response
1. A Restaurant Manager opens the system and open the menu of his restaurant.	
	2. System displays all the foods of his/her restaurant.
3. Manager selects the food from the list of foods.	
4. Manager selects to update the food.	
	5. System updates the food details in backend.
	6. System displays menu after updating.

### UC06: Use case: Block Customer

**Actor:** Admin

**Purpose:** Block customer due to violation of conditions.

**Overview:** Firstly, the rider delivers the order to customer and after delivery, the rider can report the customer and give the reason of the report. After that, admin checks the reason and decide to block the customer or not.

**\*Use cases:** Login (Admin).

### Typical Course of Events

Actor Action	System Response
1. The rider opens the online food delivery	
	2. System displays the list of that type of customers.
3. He selects all the customers who violate the conditions.	
4. He blocks them from their restaurant.	
	5. System updated the customer details.

### Alternative Course:

**Line 4:** System automatically blocks the Customers based on specific conditions.

## UC07: Use case: Manage Riders

**Actor:** Admin

**Purpose:** Add and delete the riders from the online food ordering system.

**Overview:** Firstly, an admin opens the online food delivery System and adds the rider by giving details of rider or deletes the rider by selecting the rider from the list of riders. Secondly, he saves all the updates in the system.

**\*Use cases:** Login (Admin).

### Typical Course of Events

Actor Action	System Response
1. The admin opens the system and adds the rider by giving details of rider or deletes the rider by selecting the rider from the list of riders.	
.	2. System updated the rider's details.

## UC08: Use case: Place Order

**Actor:** Customer, Payment Authentication Service.

**Purpose:** Place an order to the restaurant.

**Overview:** Firstly, a customer opens the online food delivery System and selects the restaurant. Secondly, he selects the food from the menu to eat. Thirdly, he selects the quantity of food and payment method. At the end, he places the order.

**\*Use cases:** Login (Customer).

### Typical Course of Events

Actor Action	System Response
1. The Customer opens the system and selects the restaurant.	

	2. System displays menu of that specific restaurant
3. He selects the food from the given list.	
	4. System asks for quantity.
5. He chooses the quantity of the food.	
	6. System asks for payment method.
7. He selects the payment method and places the order	
	8. System gives the order alert to the restaurant and rider.

### UC09: Use case: Give Feedback

**Actor:** Customer

**Purpose:** Give the feedback of rider on the app.

**Overview:** The customer gives the feedback of the rider by selecting riders from the list who delivered them the orders. In addition to, he gives all the details in the feedback and uploads it on the system.

**\*Use cases:** Login (Customer), Place Order, Pick Order.

### Typical Course of Events

Actor Action	System Response
1. After the order is received, the customer gives the feedback of the rider.	
2. He selects the rider from the rider list and gives detailed feedback.	
3. He uploads it on the system.	
	4. The System saves the feedback and added it to the rider's feedback.

### UC10: Use case: Update status

**Actor:** Restaurant manager

**Purpose:** Update the status of delivery/order

**Overview:** Firstly, after placing the order by customer, the manager updates the status of order “prepared”, then “on the way”.

**\*Use cases:** Place order

### Typical Course of Events

Actor Action	System Response
1. The customer places the order	
2. When food is prepared, the manager updates the status to “prepared”.	
	3. System updates the status and customer can see the updated status.
4. When rider picks the order from restaurant, the manager updates the order to “on the way”.	
	5. System again updates the status and customer can see the updated status.
6. If rider could not update the status to delivered after delivering the order. Restaurant manager can update the status to delivered.	
	7. System updates the status and customer can see the updated status.

### Alternative Course:

**Line 4:** The Restaurant can update status to deliver.

### UC11: Use case: Pick Order

**Actor:** Rider

**Purpose:** Confirm the order picking

**Overview:** Firstly, opens the system and goes into orders section. Then the rider selects the order, which he wants to pick and deliver. The order picked by the rider cannot be seen by other riders.

**\*Use cases:** Login (Rider), Place order

## Typical Course of Events

Actor Action	System Response
1. The rider opens the system and views the orders and locations of placement.	
2. Rider selects the order to pick.	
3. Rider confirms an order to pick.	
	4. System updates the status to pick and other riders for picking cannot see order.

## UC12: Use case: Update status

**Actor:** Rider

**Purpose:** Update the status of delivery/order

**Overview:** Firstly, the rider delivers the order to customer and after delivery, the rider collect the amount of that order if customer chooses cash on delivery method. Rider updates the status to delivered.

**\*Use cases:** Place order, pick order

## Typical Course of Events

Actor Action	System Response
1. Rider picks the order from restaurant.	
2. Rider delivers the order at given location.	
3. Rider collects amount of order from customer if customer selected the option “cash on delivery”.	
5. Rider updates the status of order delivery to delivered.	
	6. System updates the delivery status and customer can see the updated status.

**Alternative Course:**

**Line 4:**

1. The rider could not deliver the order at location.
2. Restaurant manager updates the status to 'delivered'.

### UC13: Use case: Report Customer

**Actor:** Rider

**Purpose:** report the customer to admin

**Overview:** Firstly, the rider delivers the order to customer and after delivery, the rider can report the customer and give the reason of the report. After that, admin checks the reason and decide to block the customer or not.

**\*Use cases:** Login (rider)

#### Typical Course of Events

Actor Action	System Response
1. The rider opens the online food delivery system and selects the option to report customer.	
	2. System displays all the customers he delivered food.
3. He selects the customer and enters the reason.	
4. He reports the customer	
	5. System sends the customer information to admin

### UC14: Use case: Manage Profile

**Actor:** Customer

**Purpose:** manage his own profile

**Overview:** Firstly, the customer changes his/her details by going into manage profile section and then saves all the new details.

**\*Use cases:** Login (Customer).



## Typical Course of Events

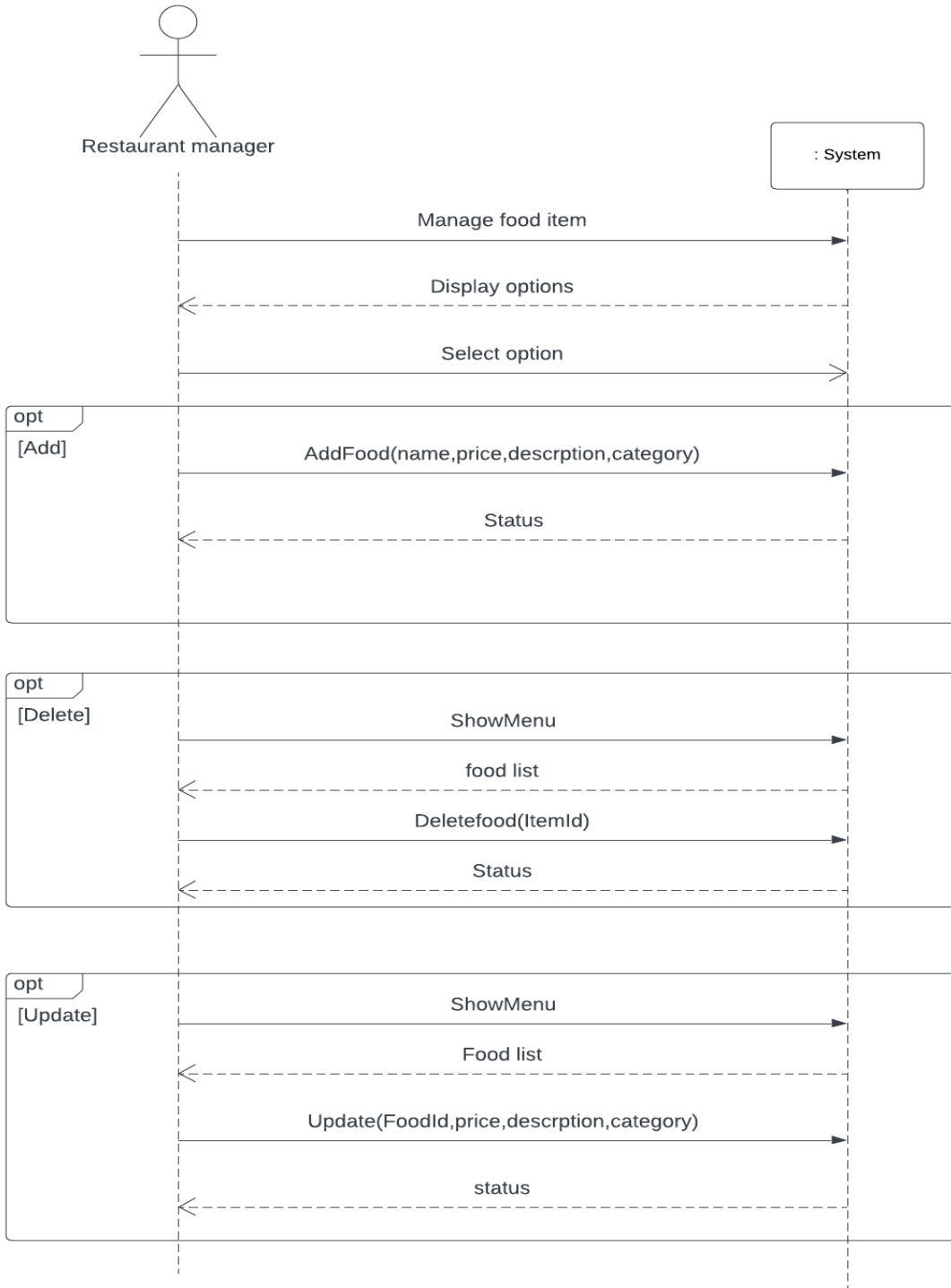
Actor Action	System Response
1. After the creation of his account, the customer goes to manage profile section.	2. The system asks for updated information.
3. He enters the updated name or phone number or any other detail in his account.	
4. He uploads it on the system.	
	5. The System saves the new details and updated his profile.

## Domain Model

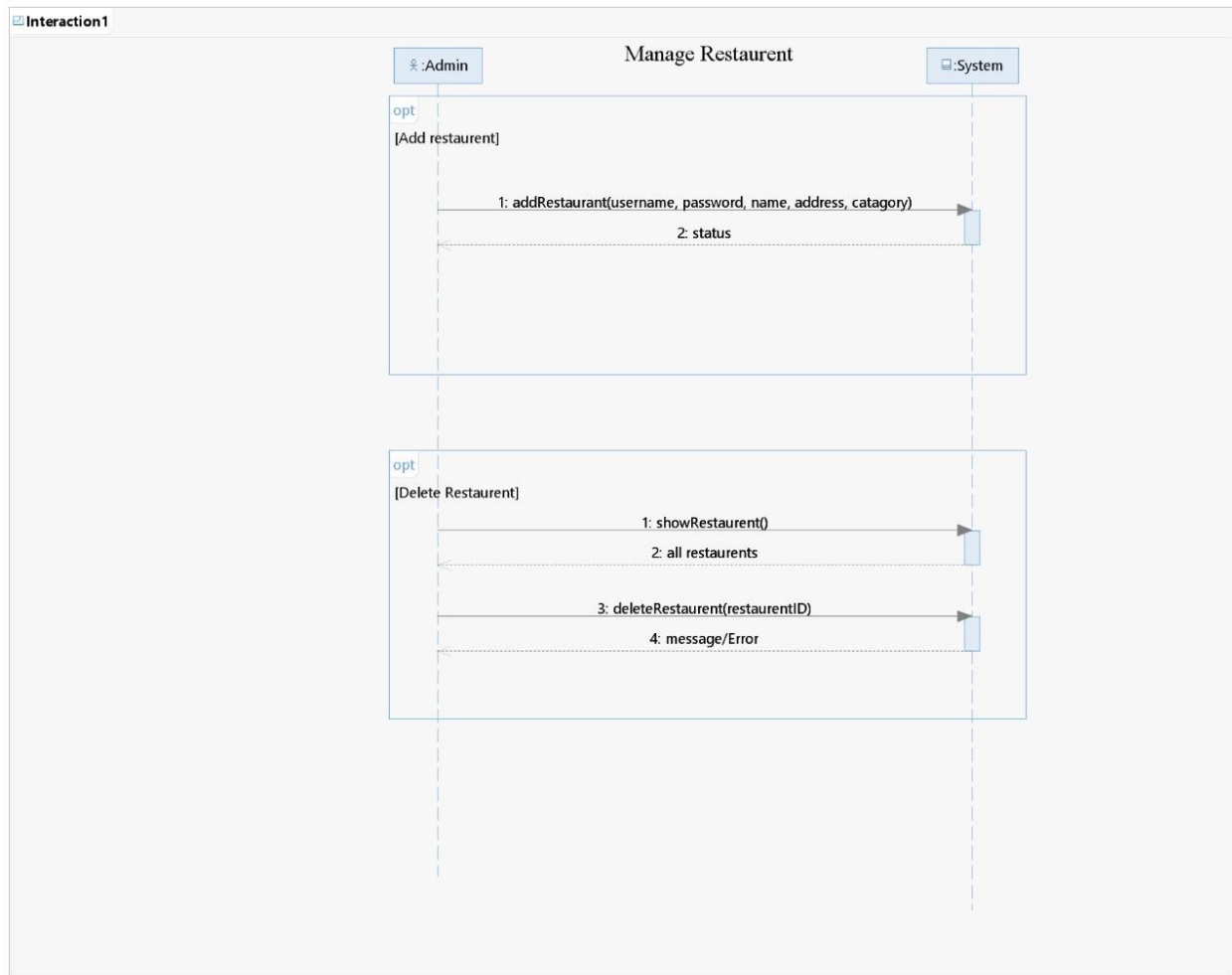


## System Sequence Diagram

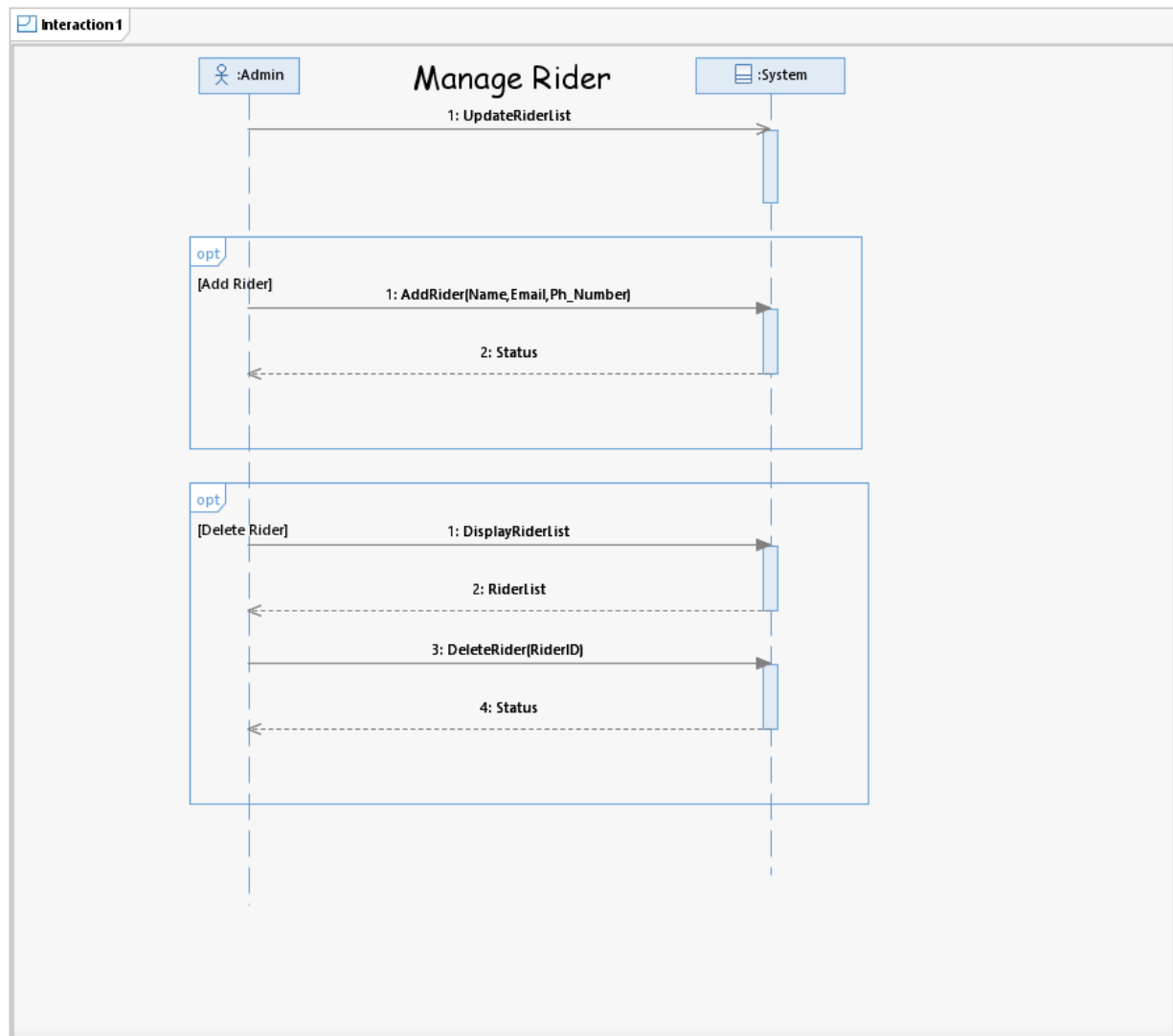
### 1. Manage Food



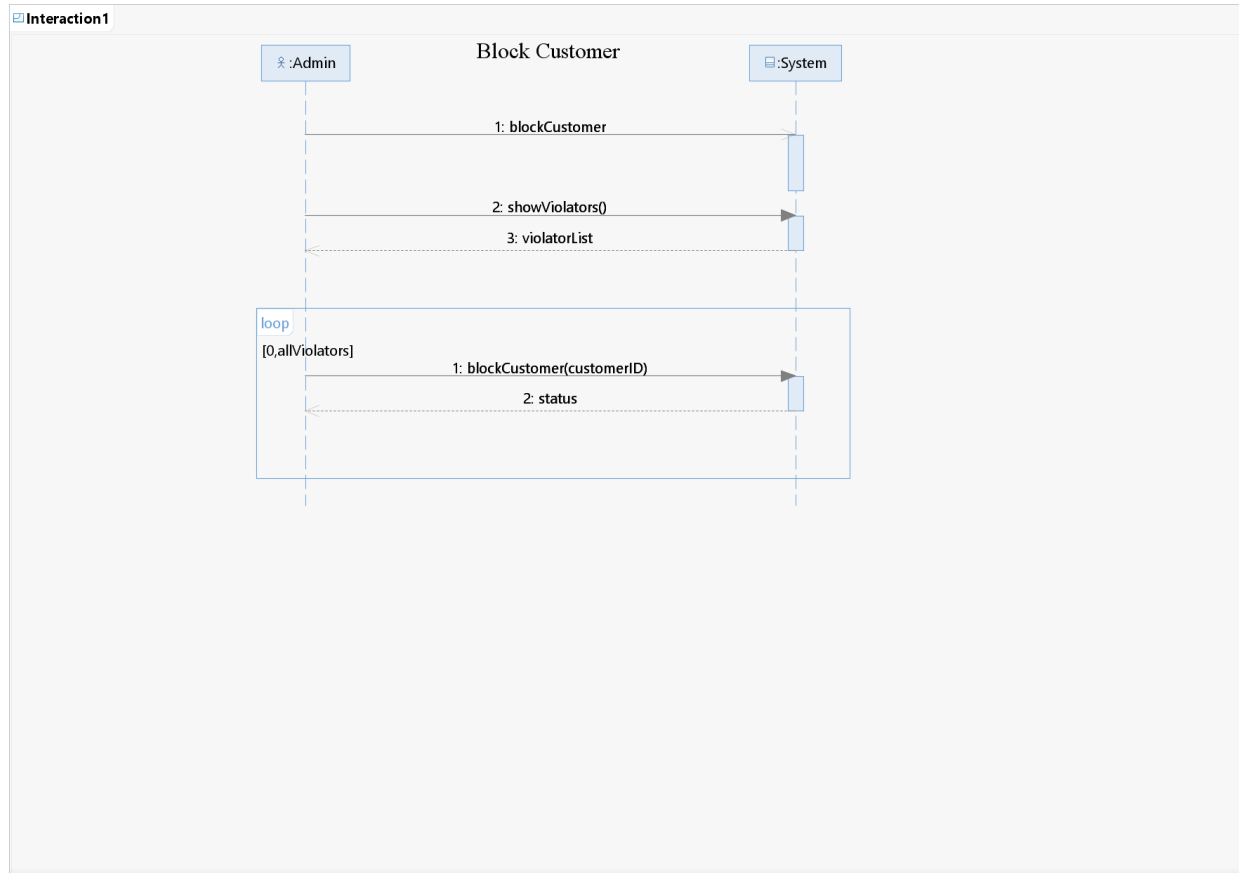
## 2. Manage Restaurants



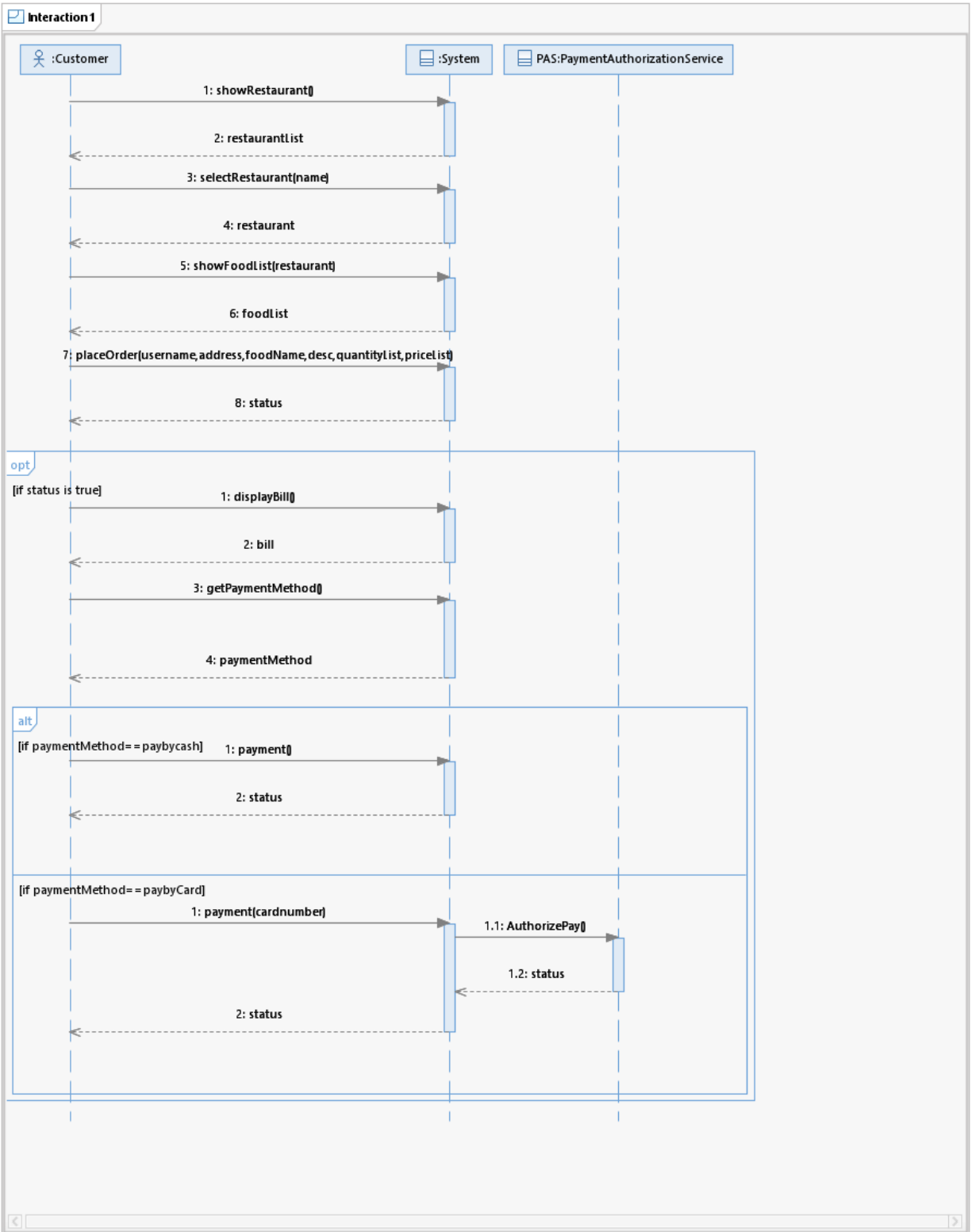
### 3. Manage Riders



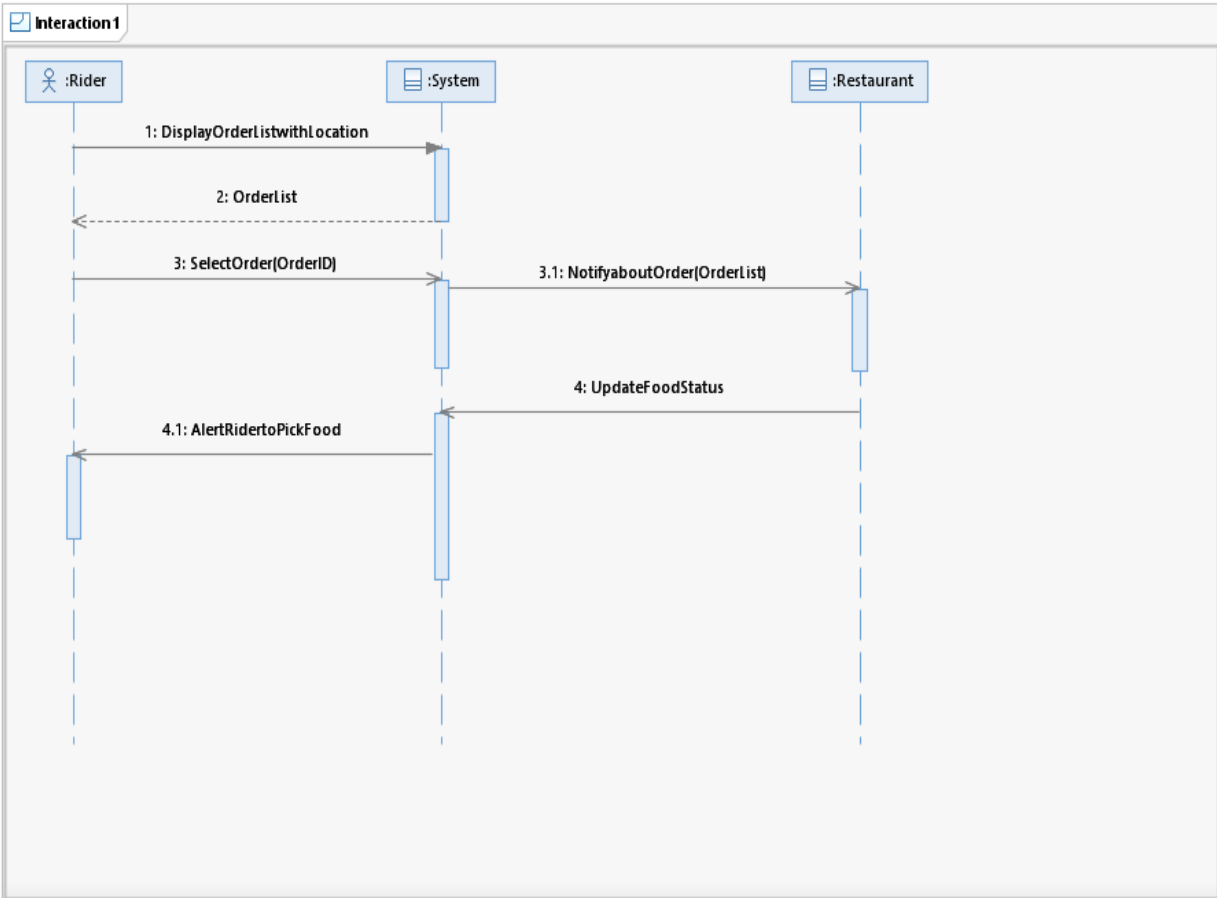
### 4. Block Customer



## 5. Place Order



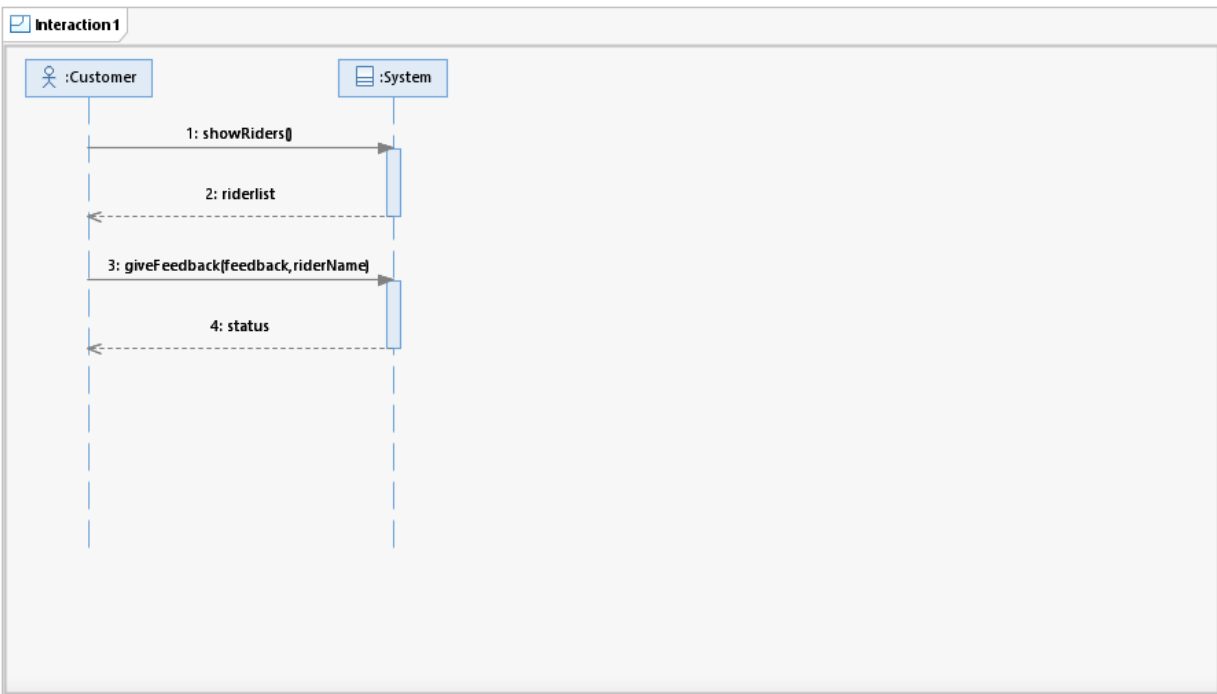
## 6. Pick Order



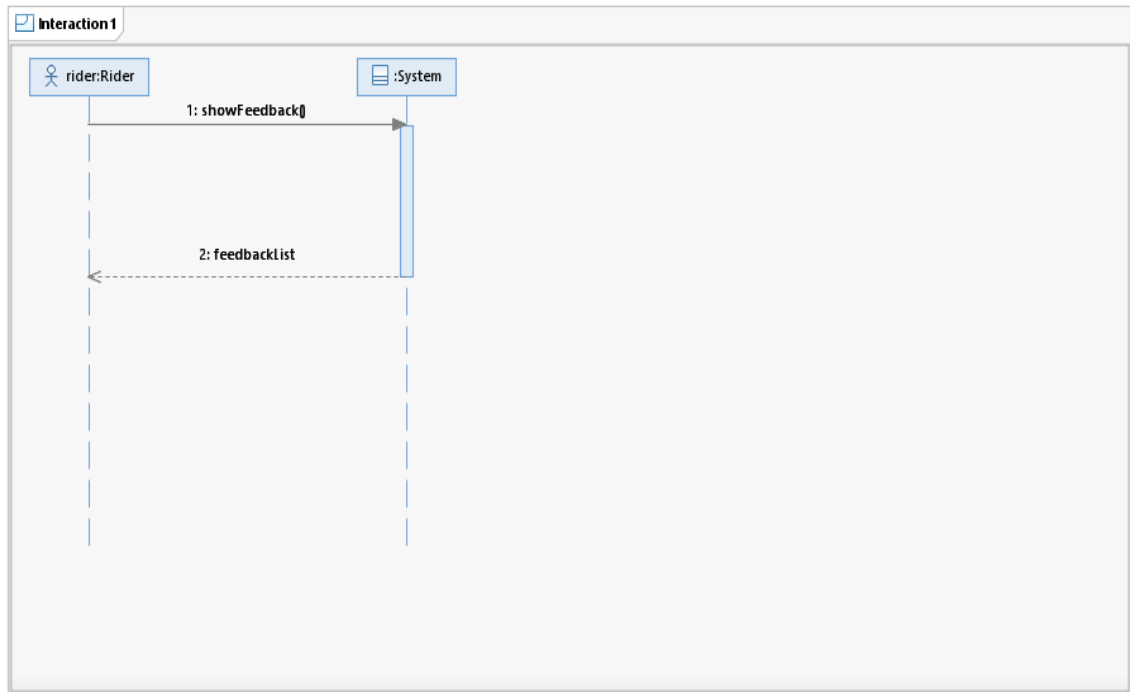


## 7. Give Feedback

### 7.1 Give Feedback (Customer)

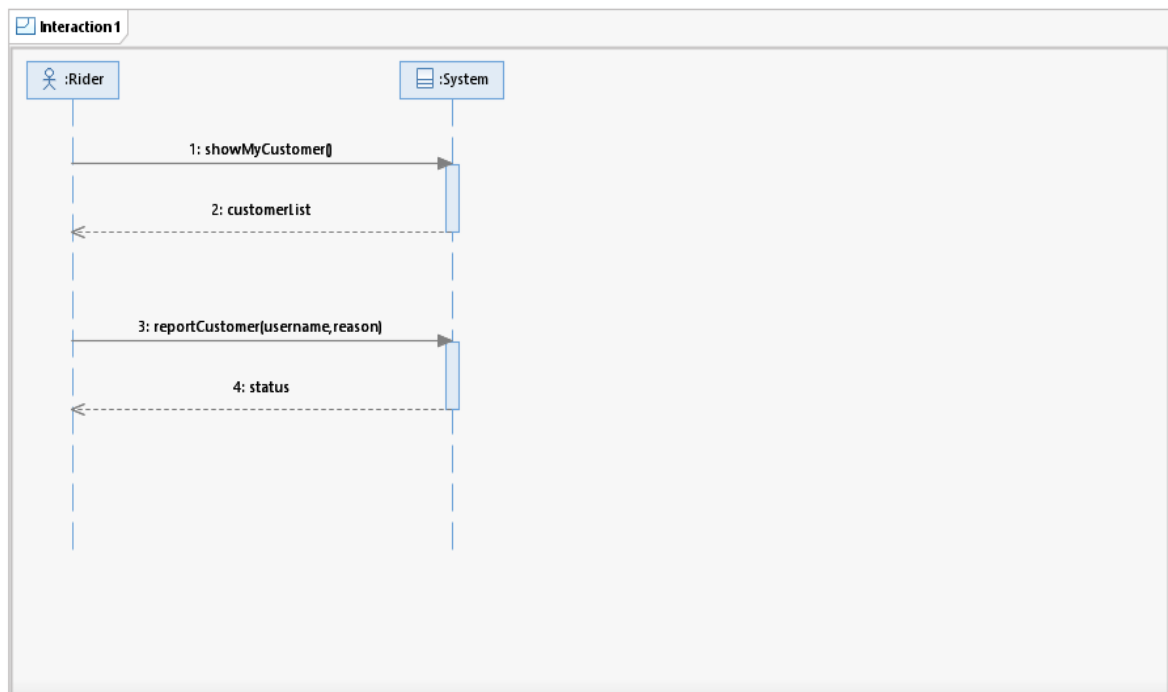


### 7.2 Show Feedback (Rider)

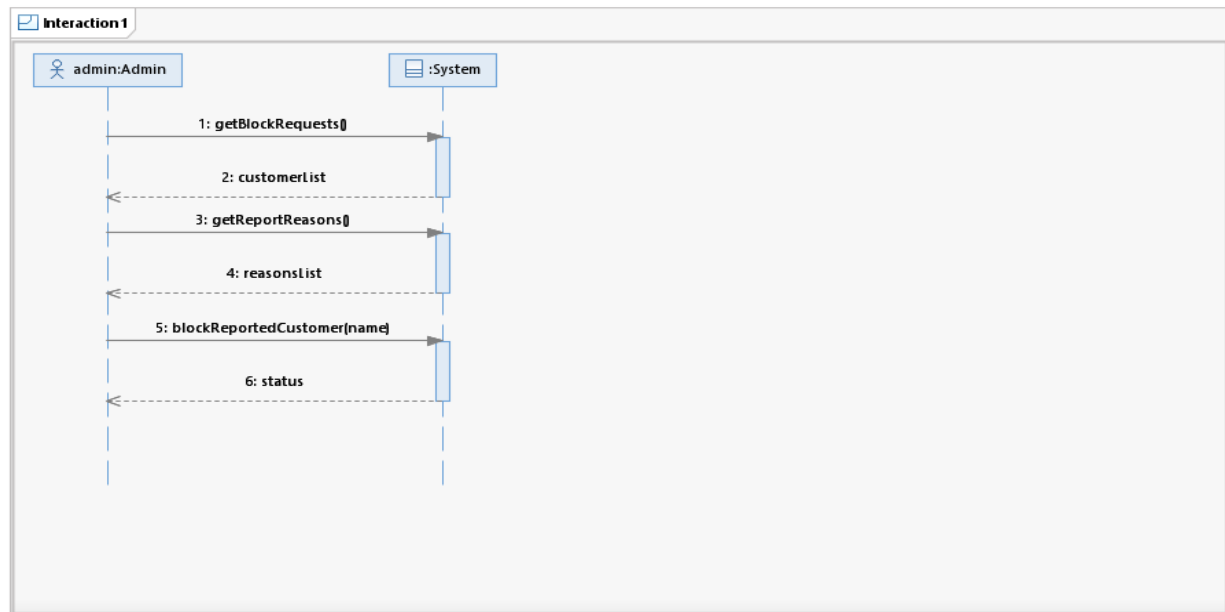


## 8. Report Customer

### 8.1 Report Customer (Rider)



## 8.2 Block Reported Customer (Admin)



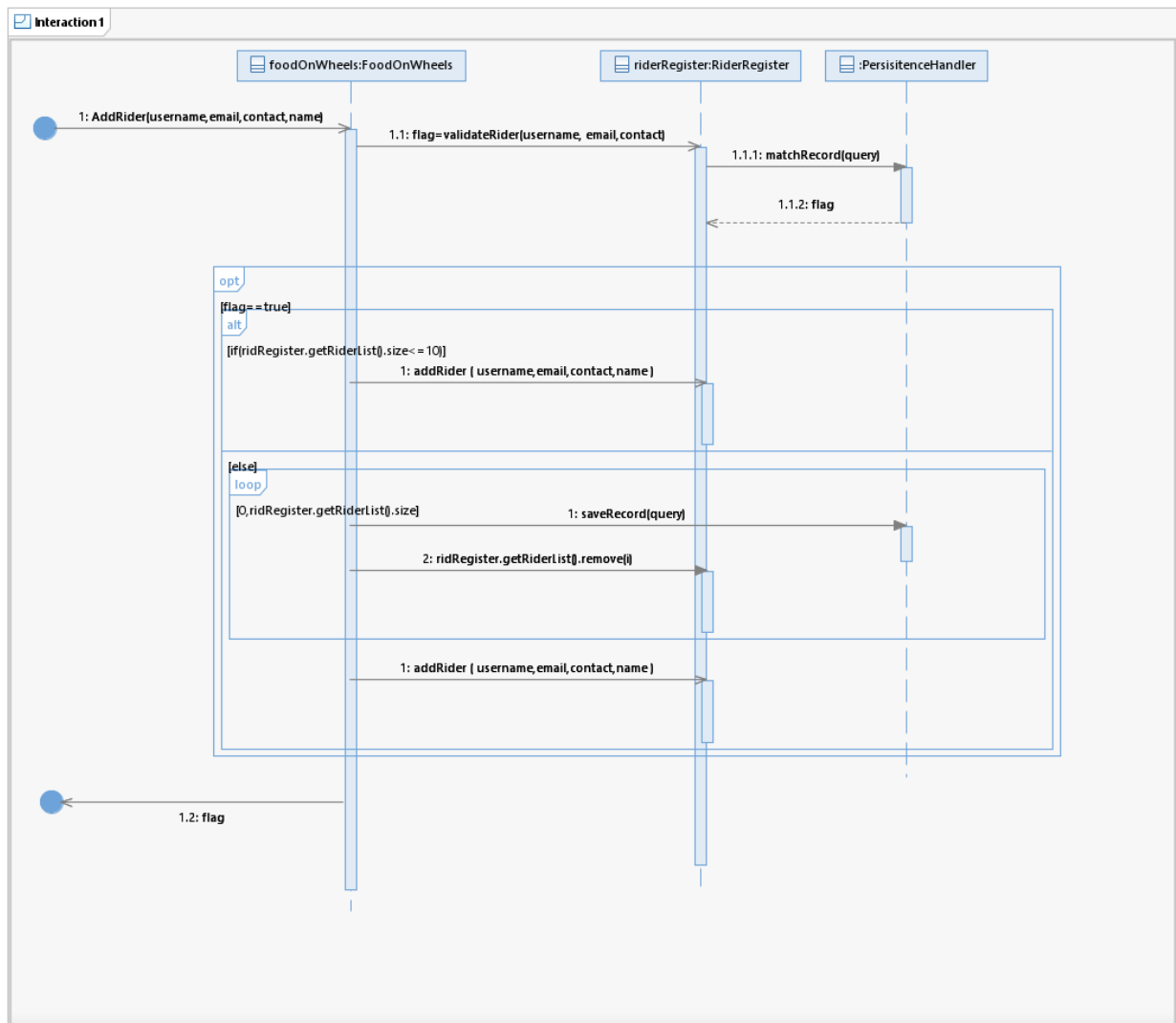
## 9. Manage Profile (Customer)



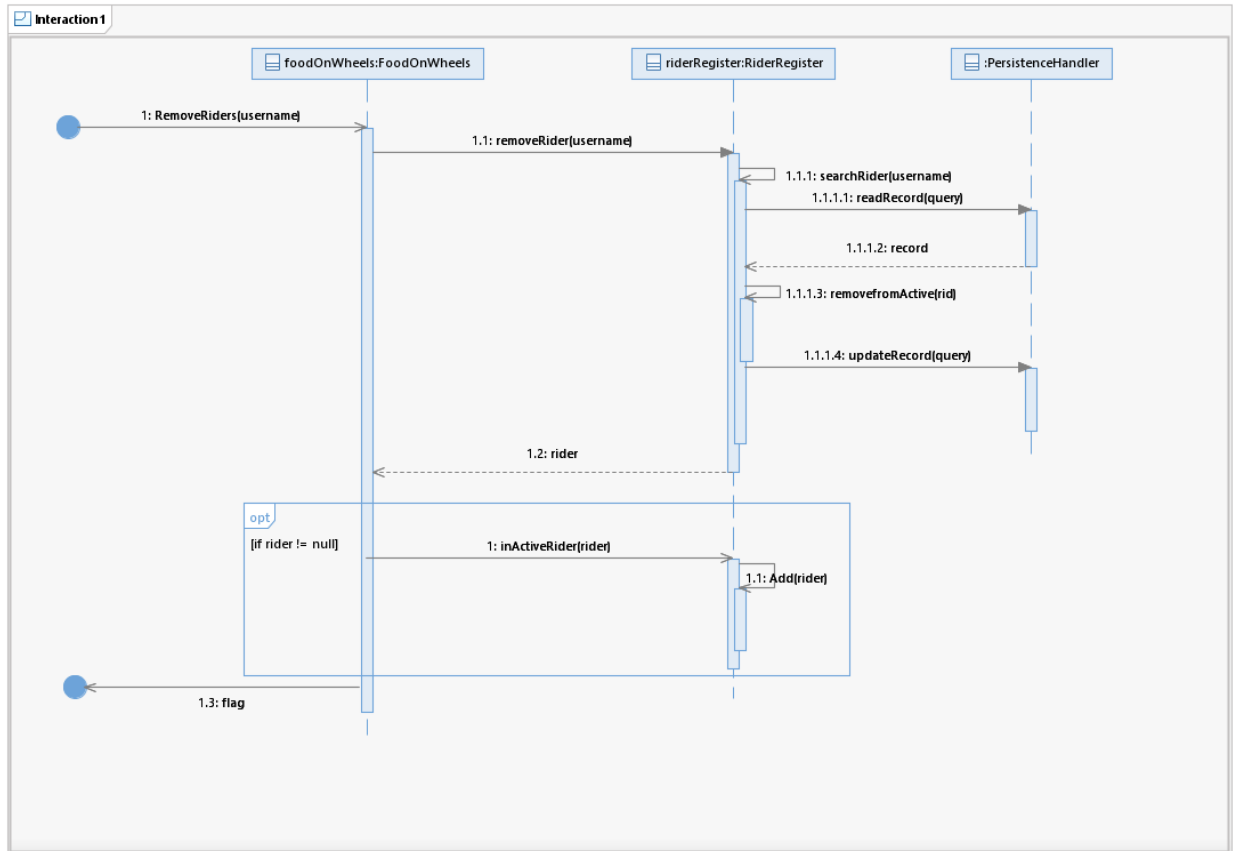
## Sequence Diagram

## 1. Manage Riders

### 1.1. Add Rider

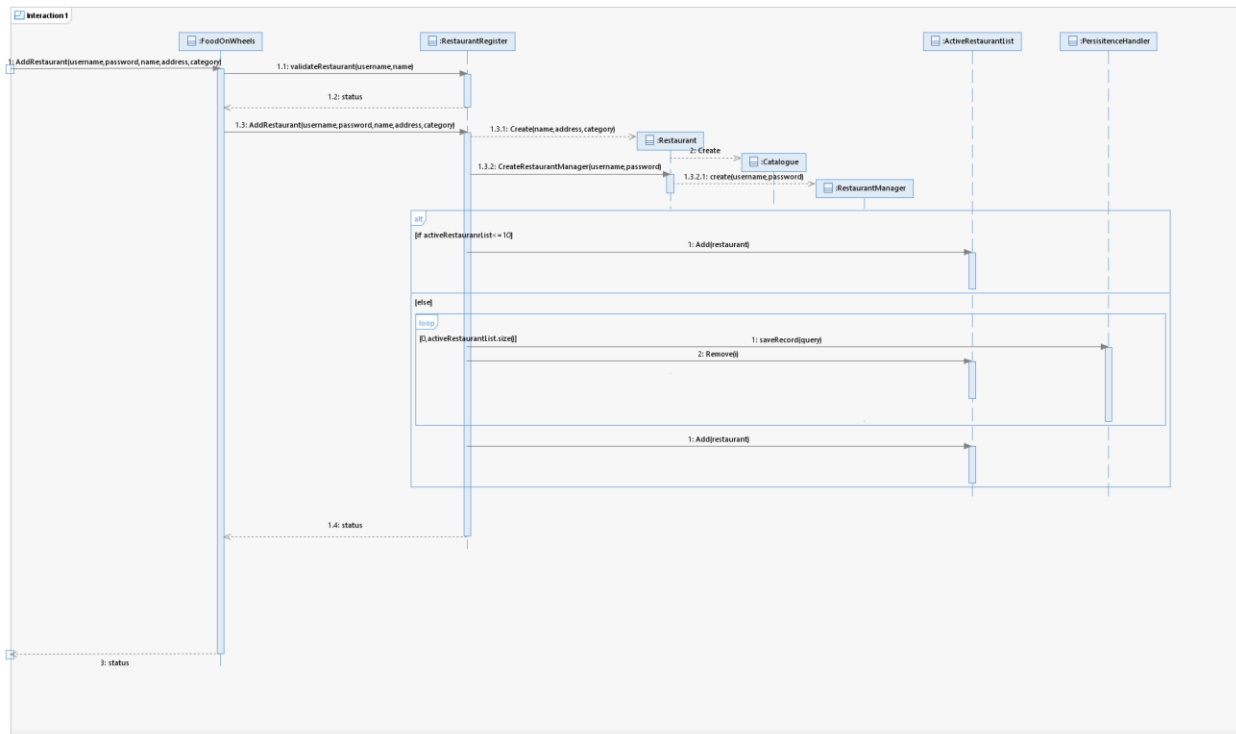


### 1.2. Delete Rider

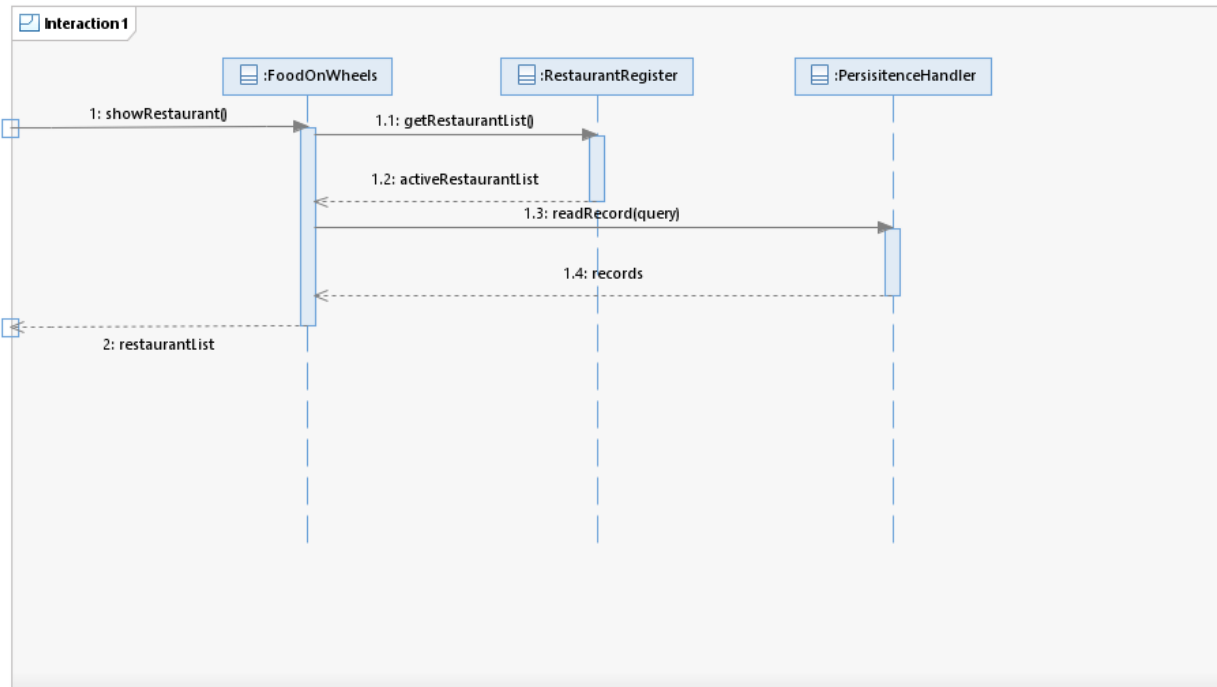


## 2. Manage Restaurant

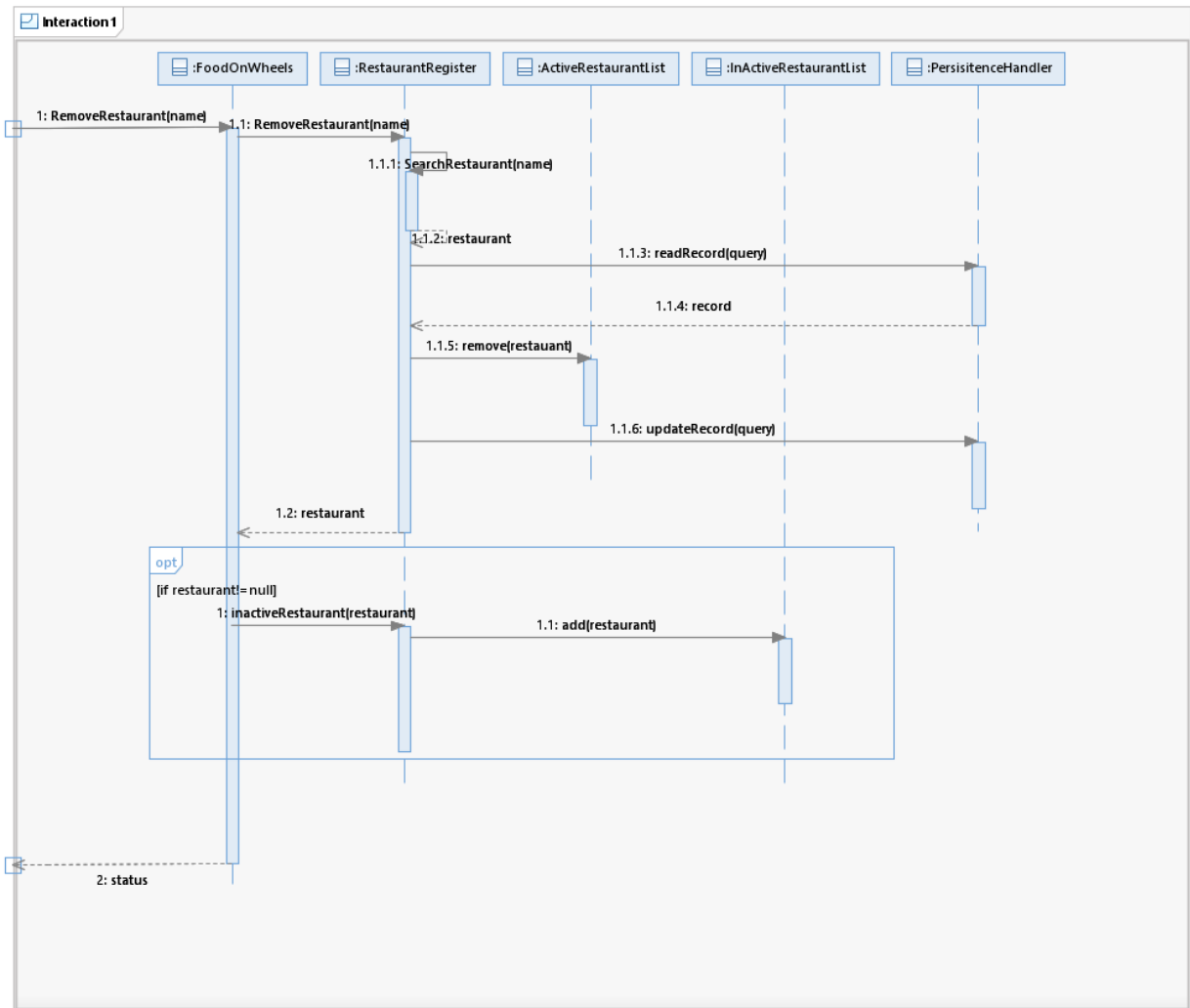
### 2.1. Add Restaurant



## 2.2. Display Restaurant



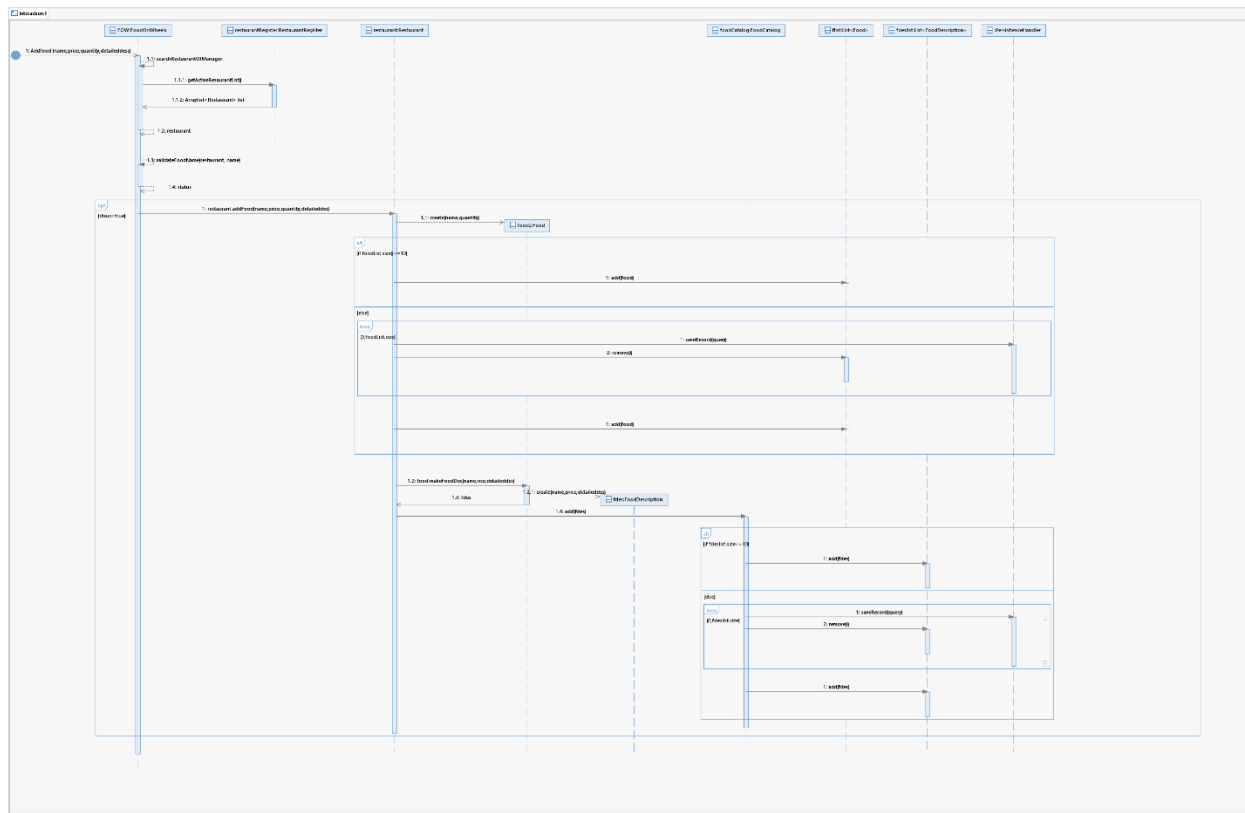
## 2.3. Delete Restaurant



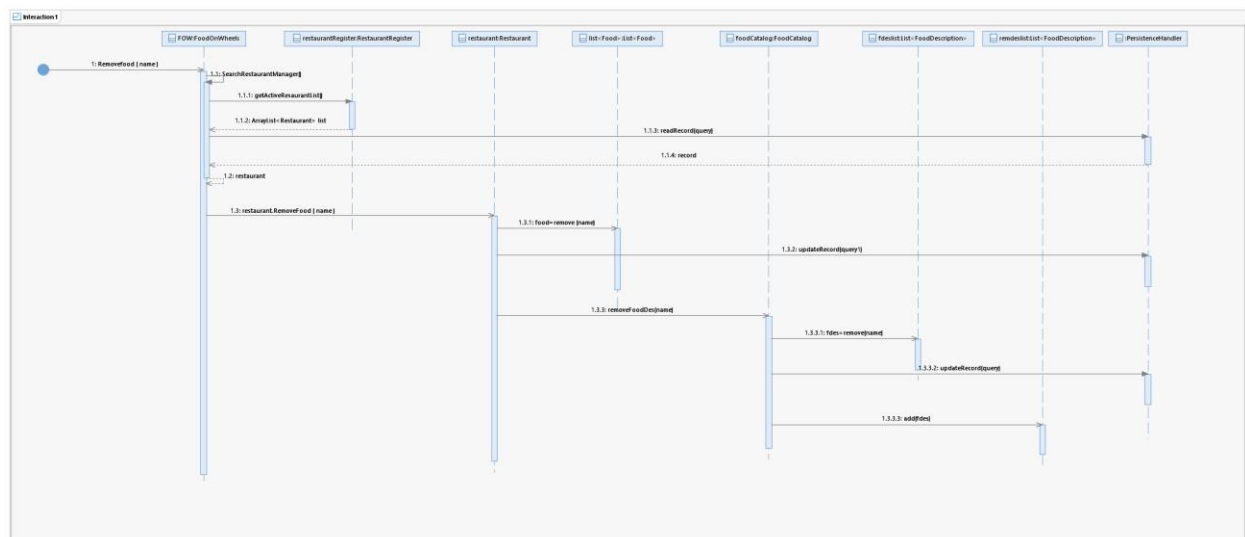


### 3. Manage food

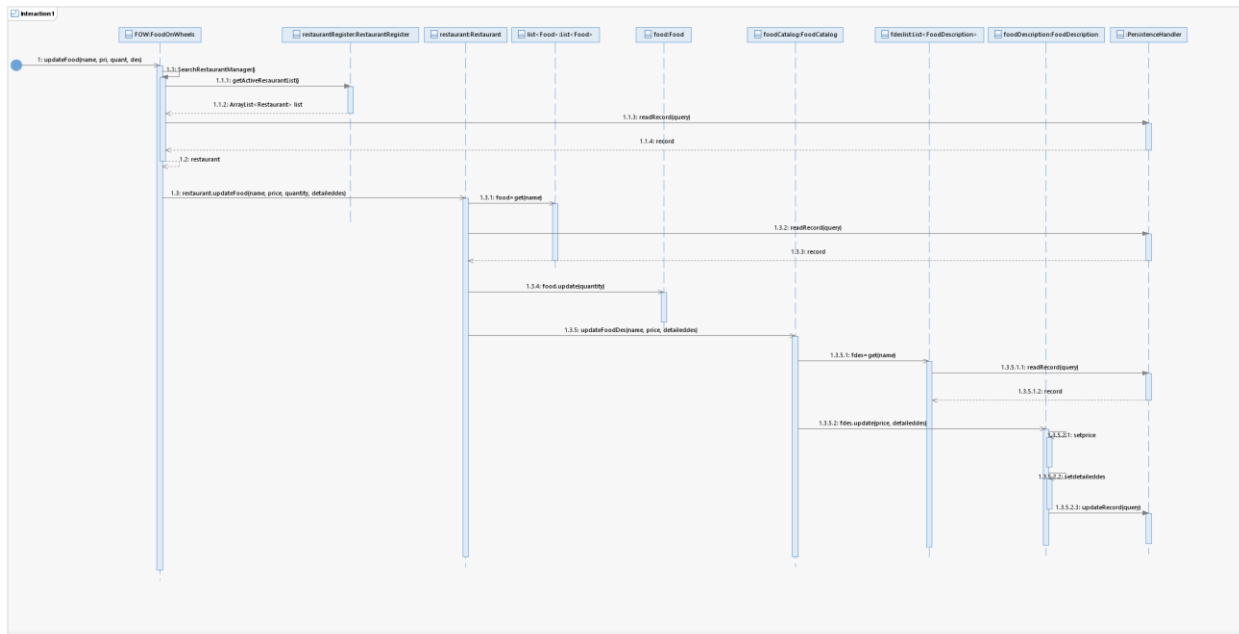
### 3.1 Add food



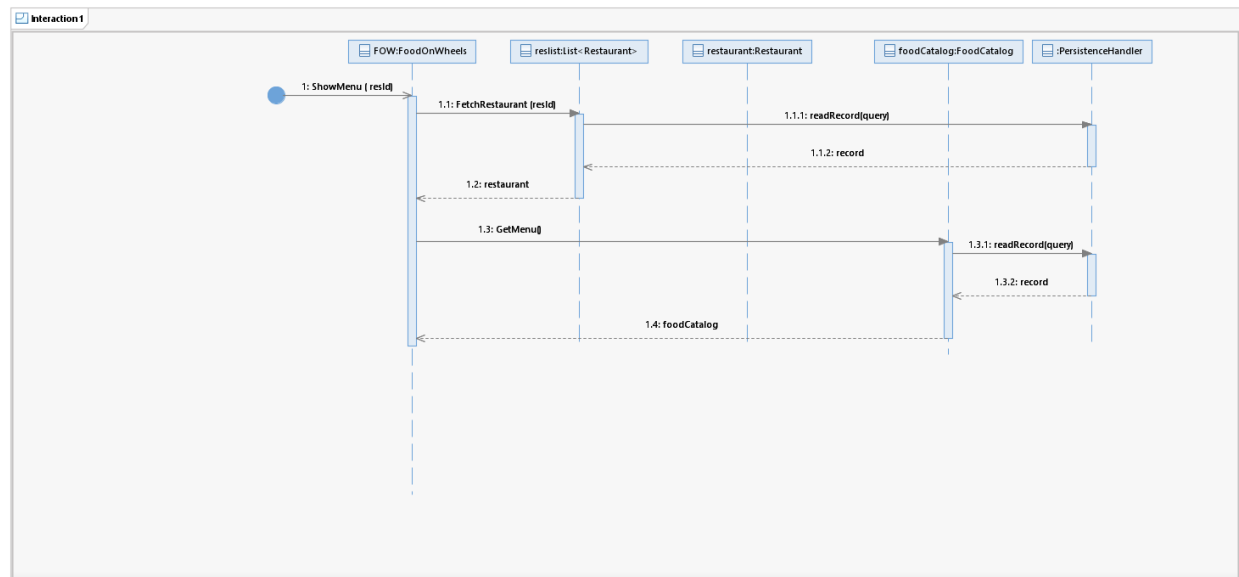
### 3.2 Remove food



### 3.3 Update food

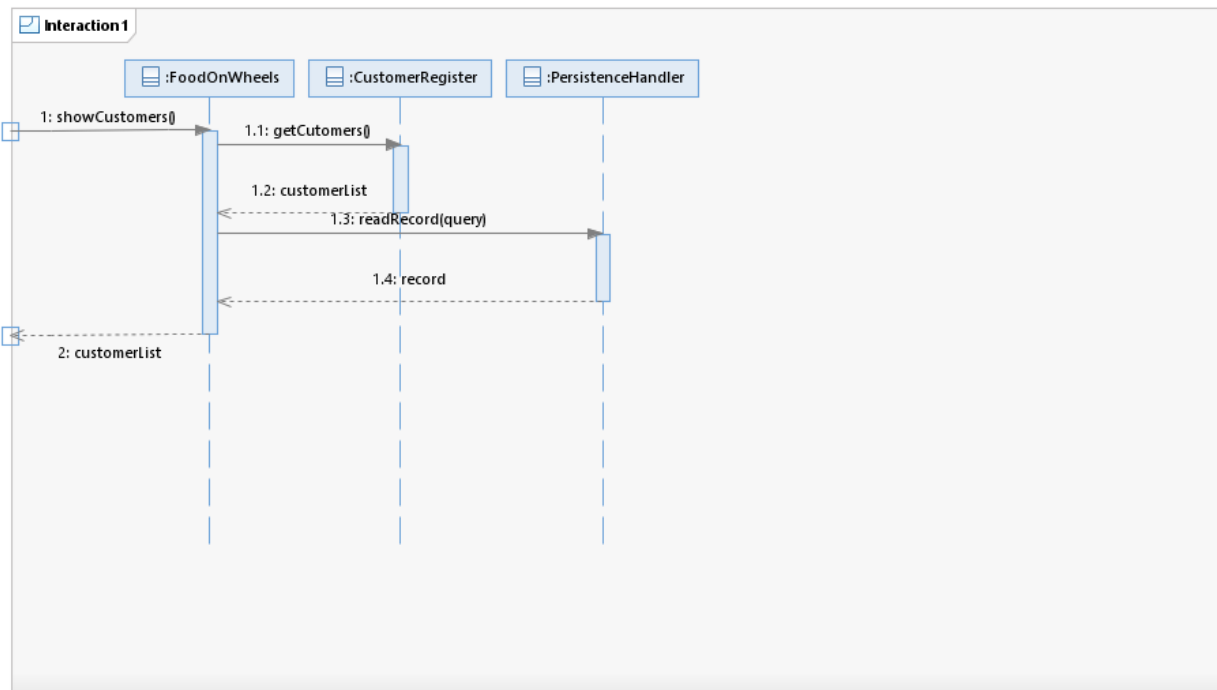


### 3.4 Show Menu

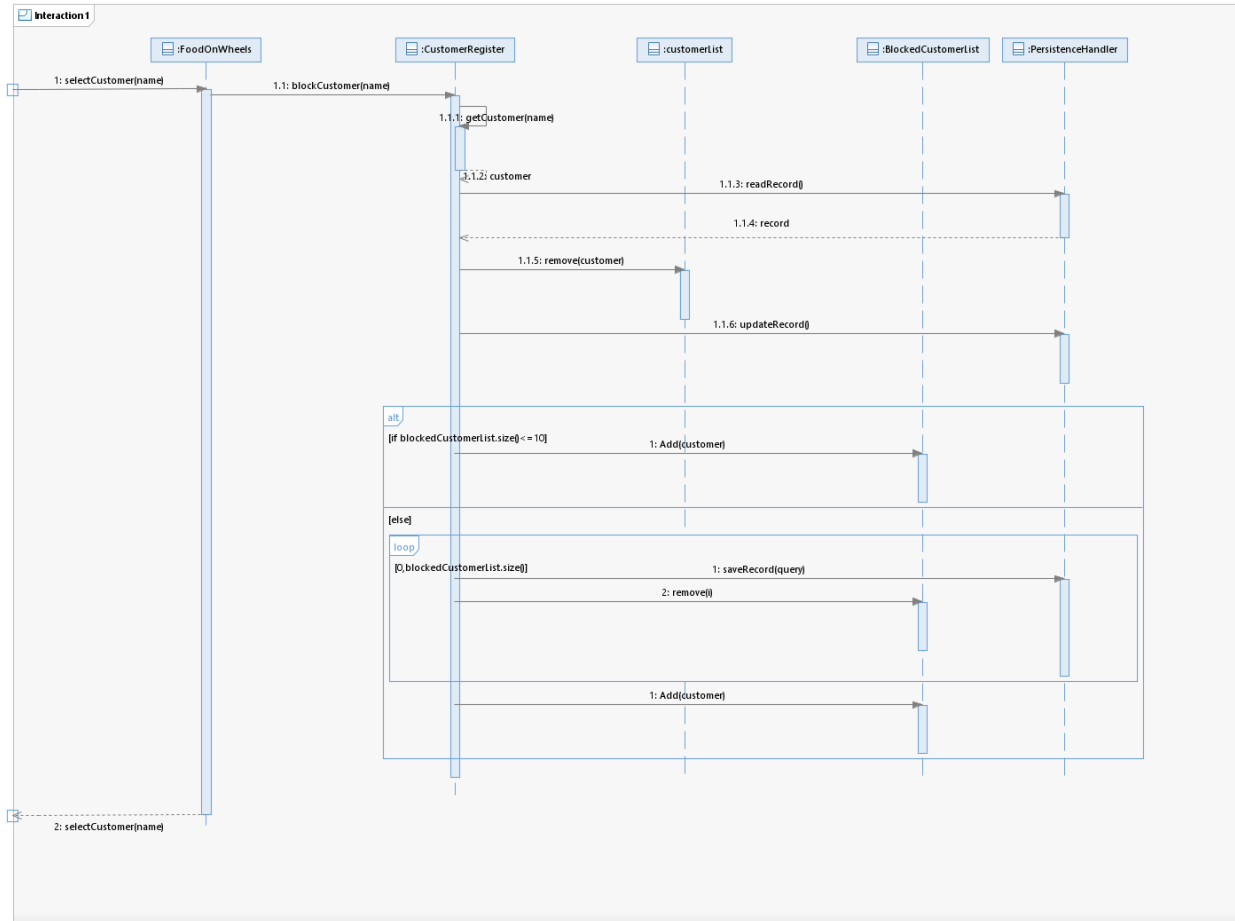


## 4. Block Customer

### 4.1. Display Violators

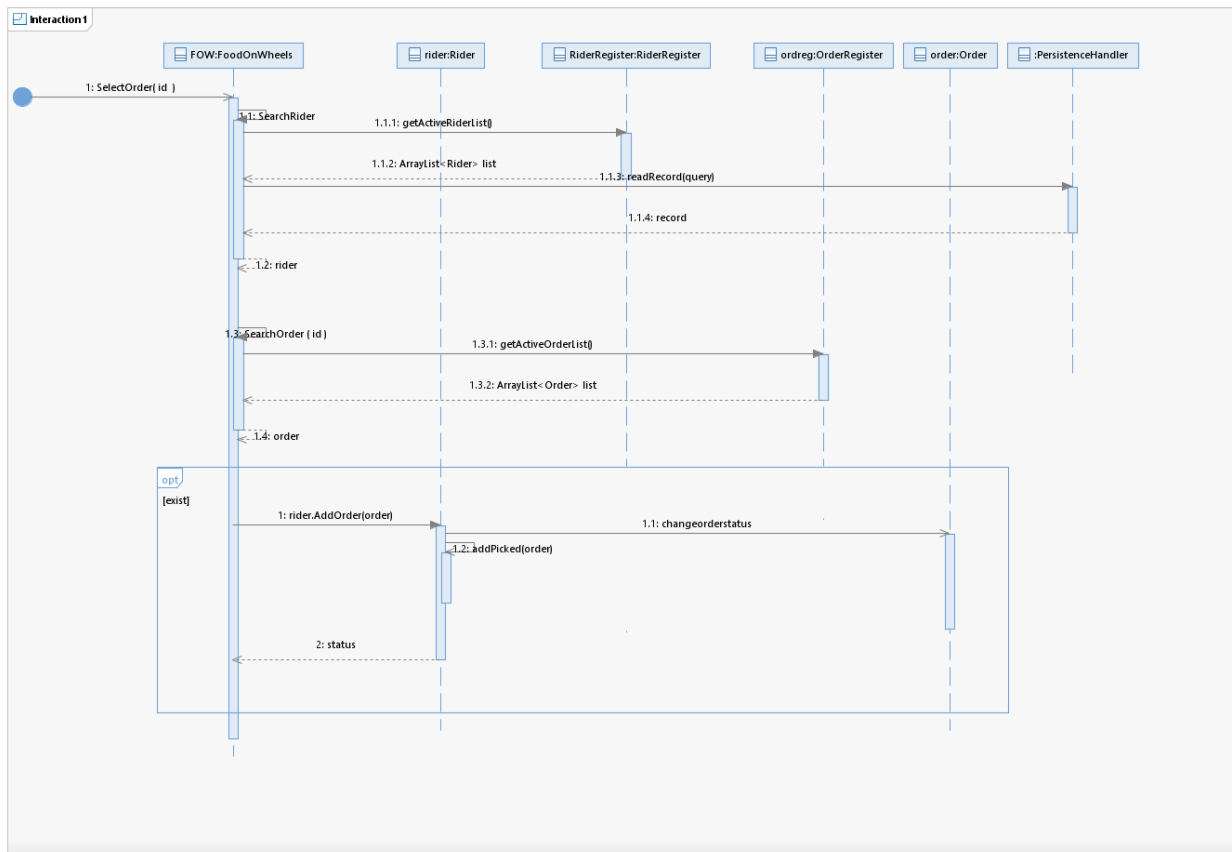


## 4.2. Block Customer

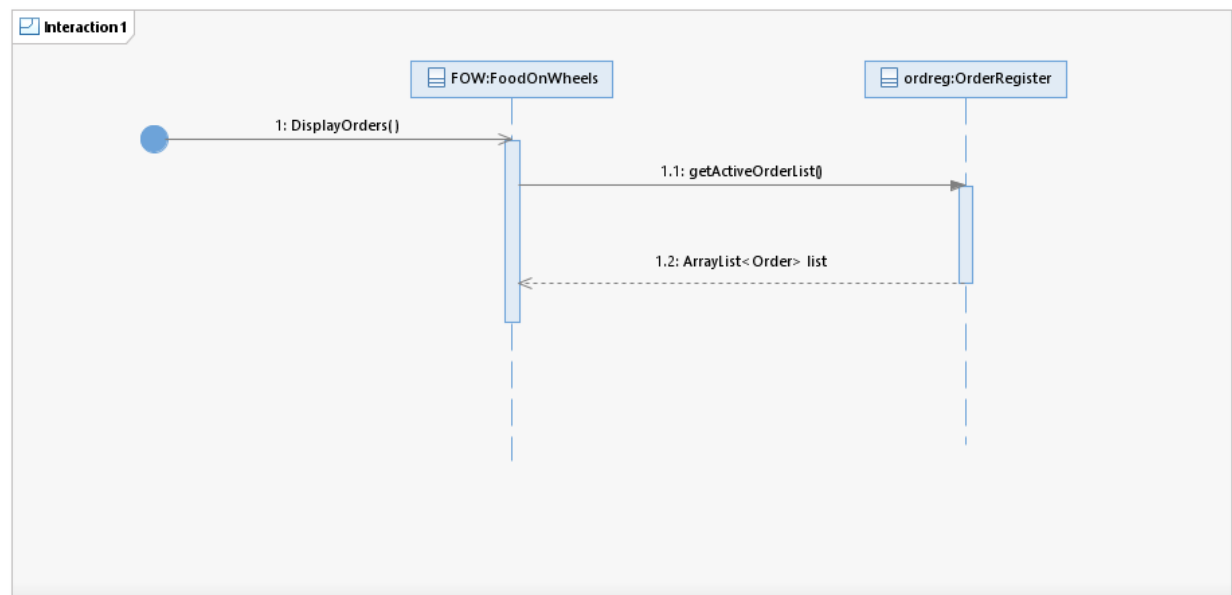


## 5. Pick Order

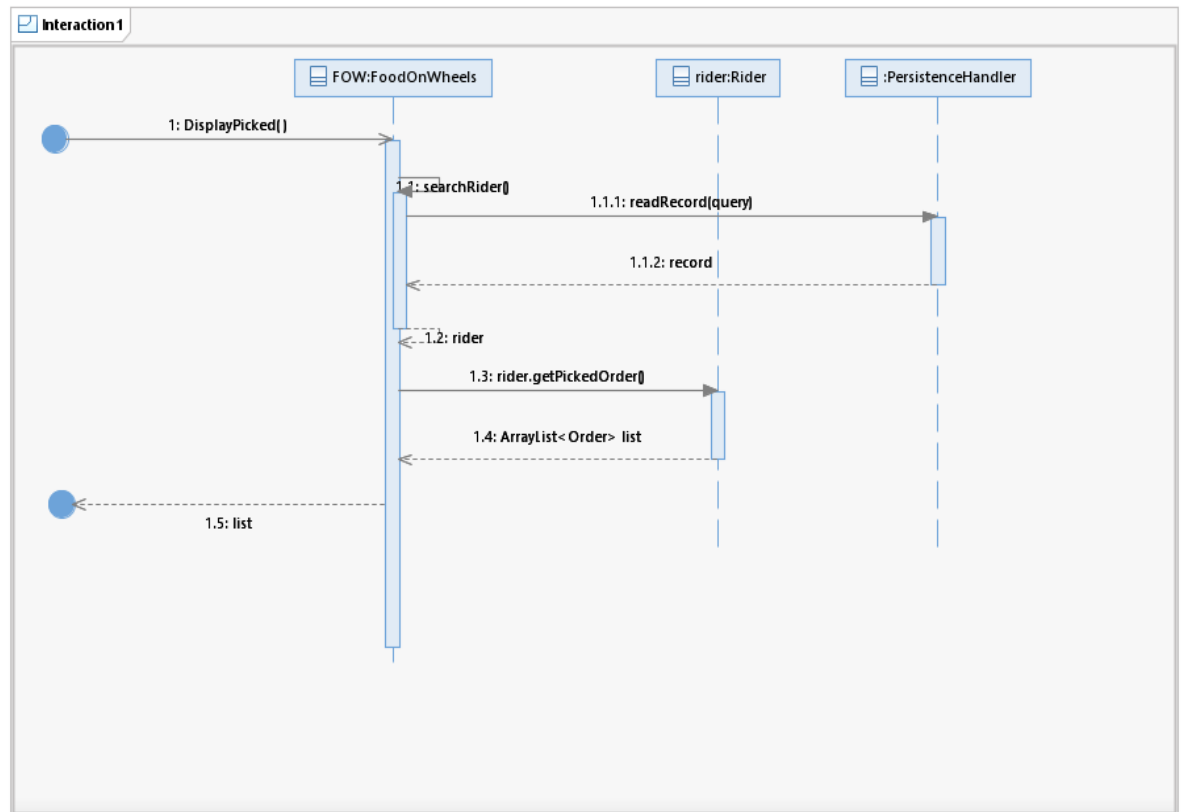
### 5.1. Select Order



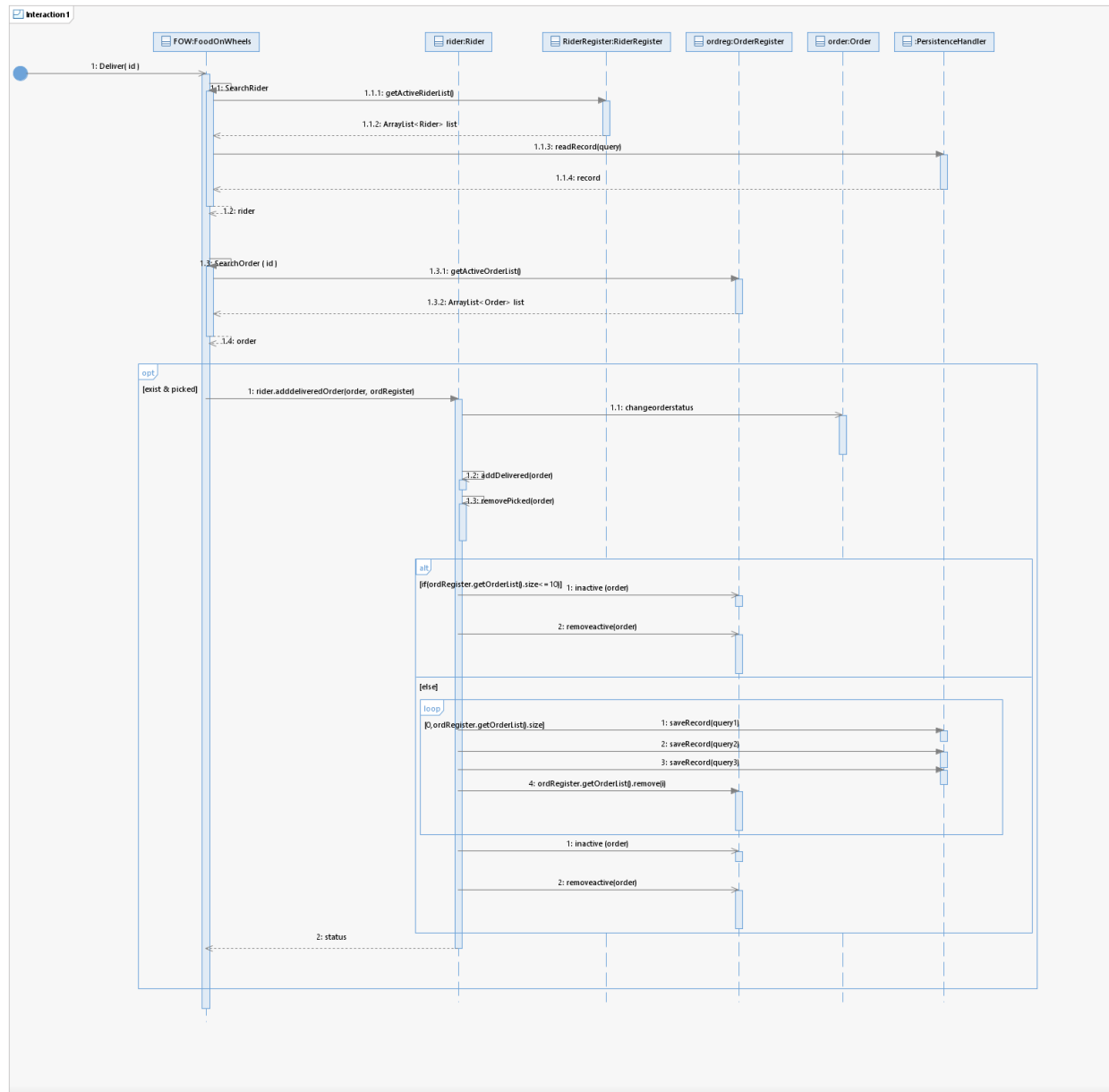
### 5.2. Display Order



### 5.3 Display Picked Orders

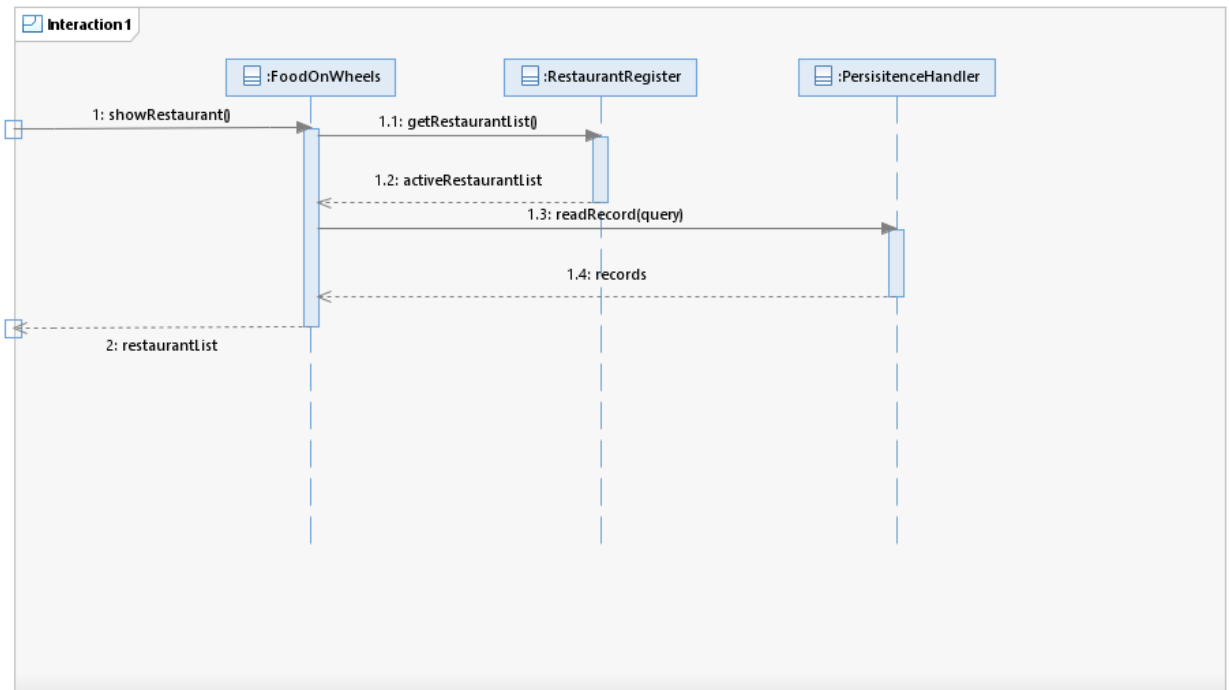


## 5.4 Deliver Order

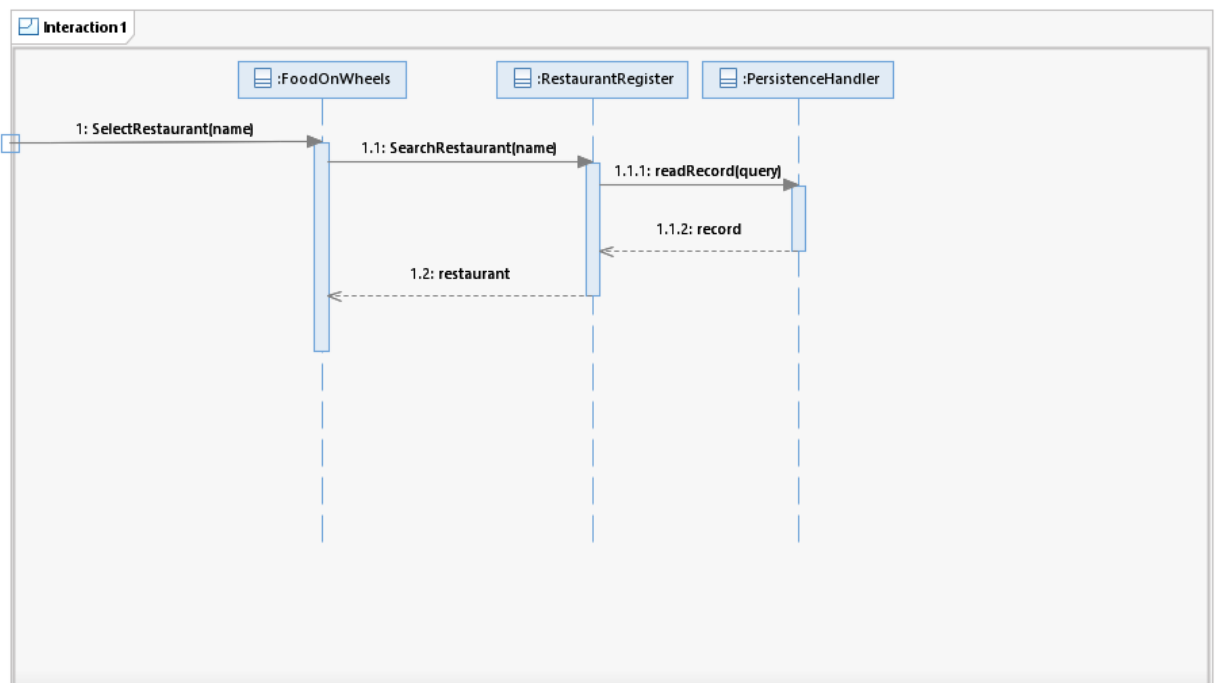


## 6. Place Order

### 6.1 Show Restaurant

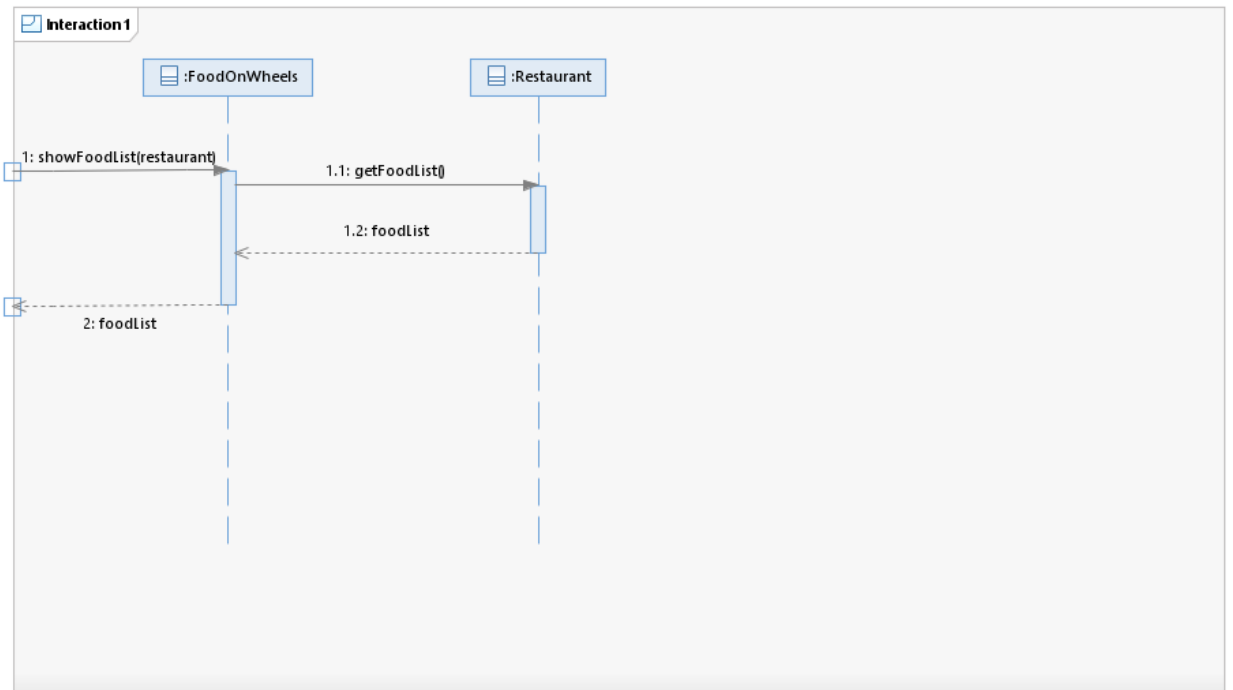


## 6.2 Select Restaurant

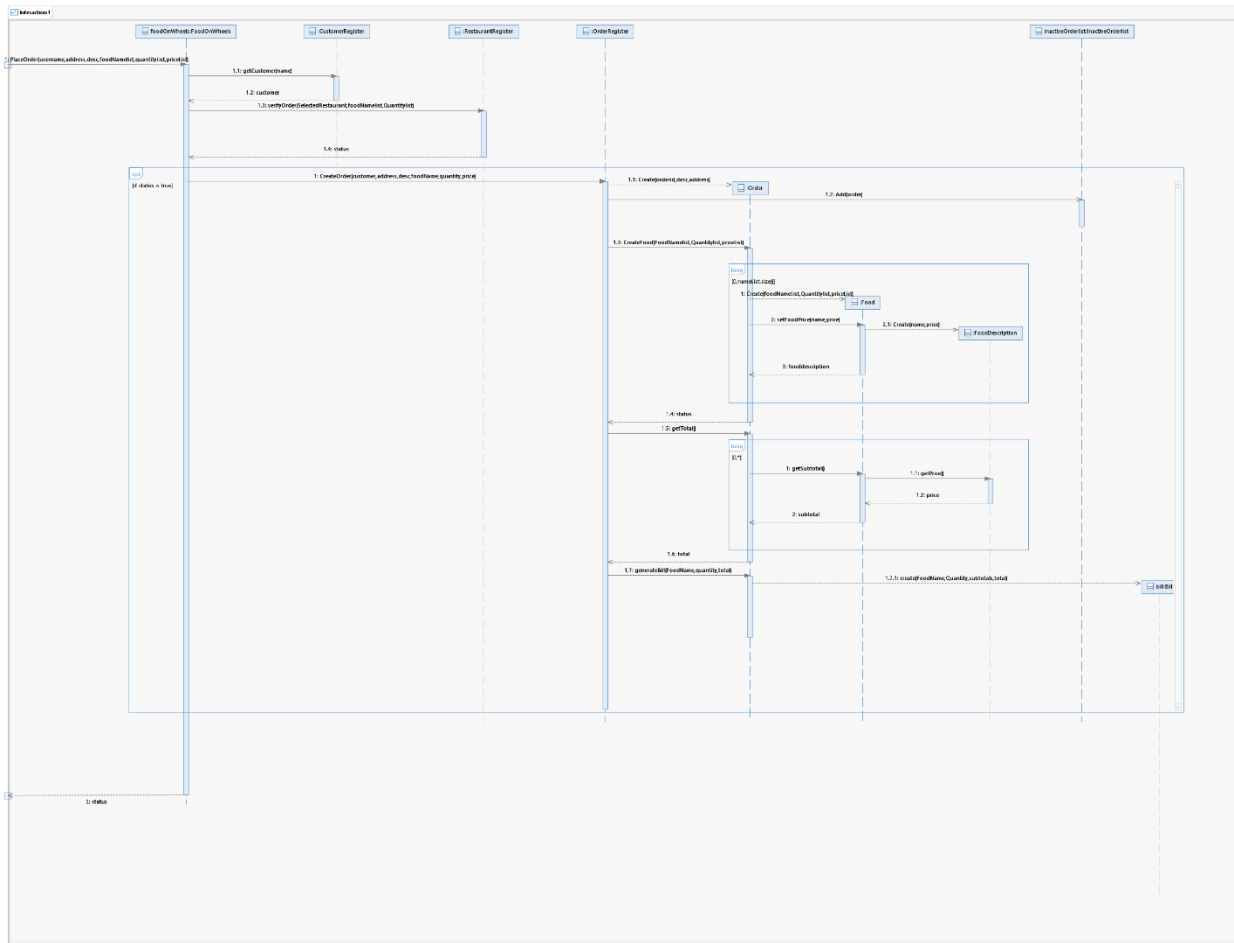


## 6.3 Show Food

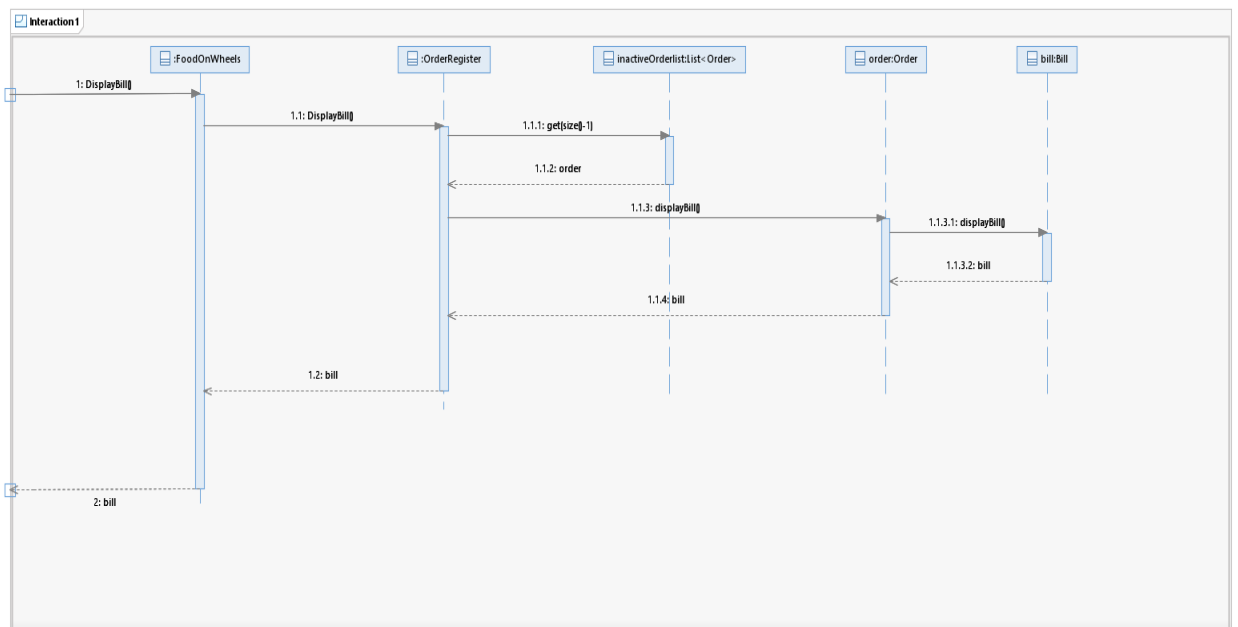




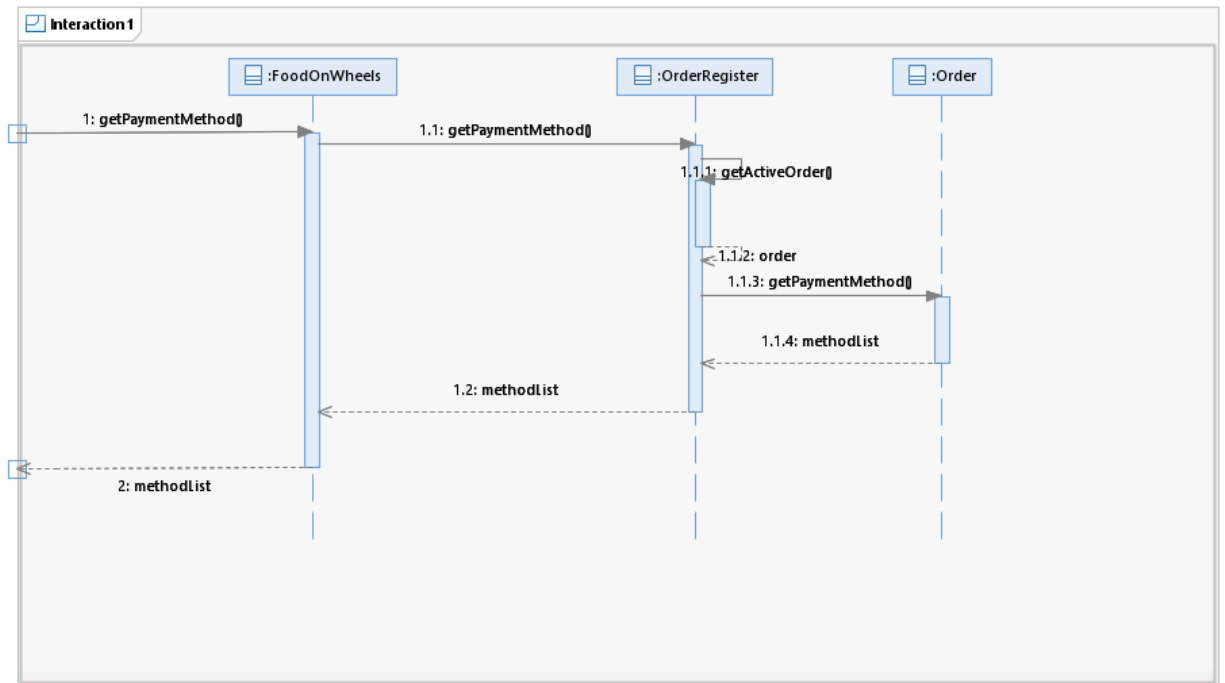
## 6.4 Place Order



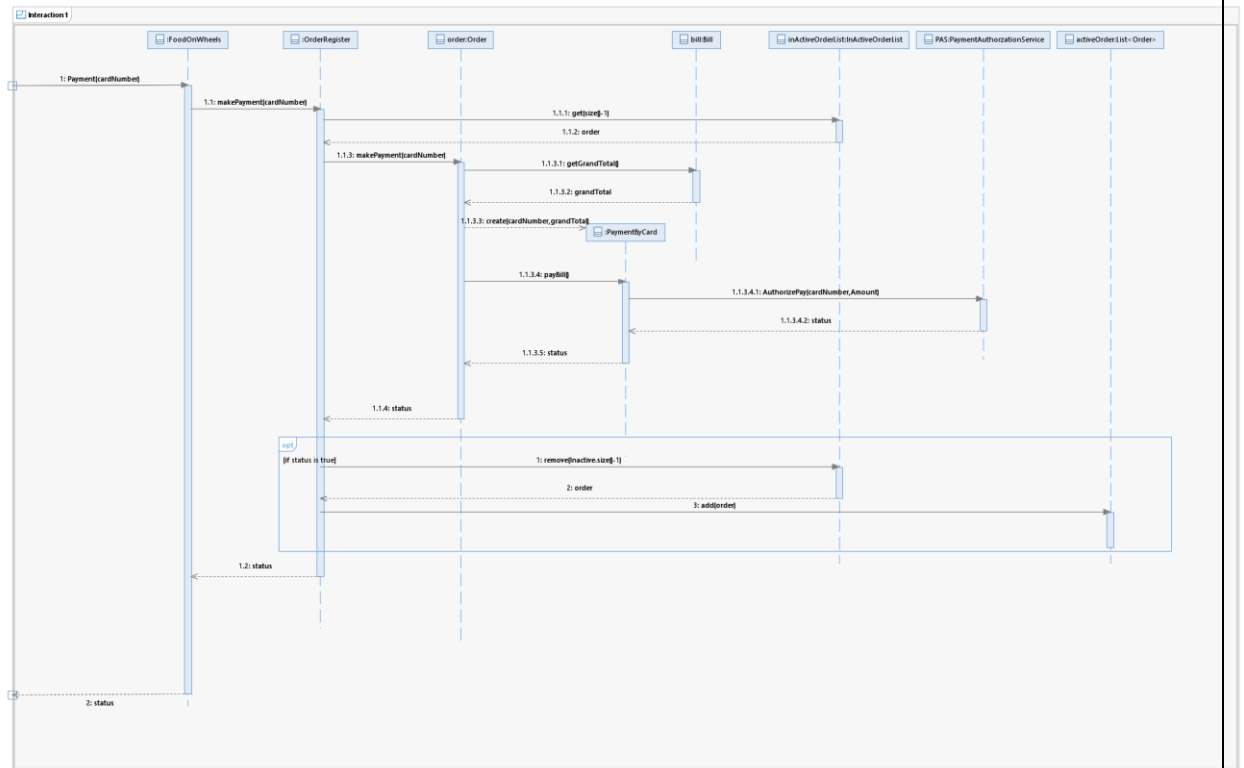
## 6.5 Display Bill



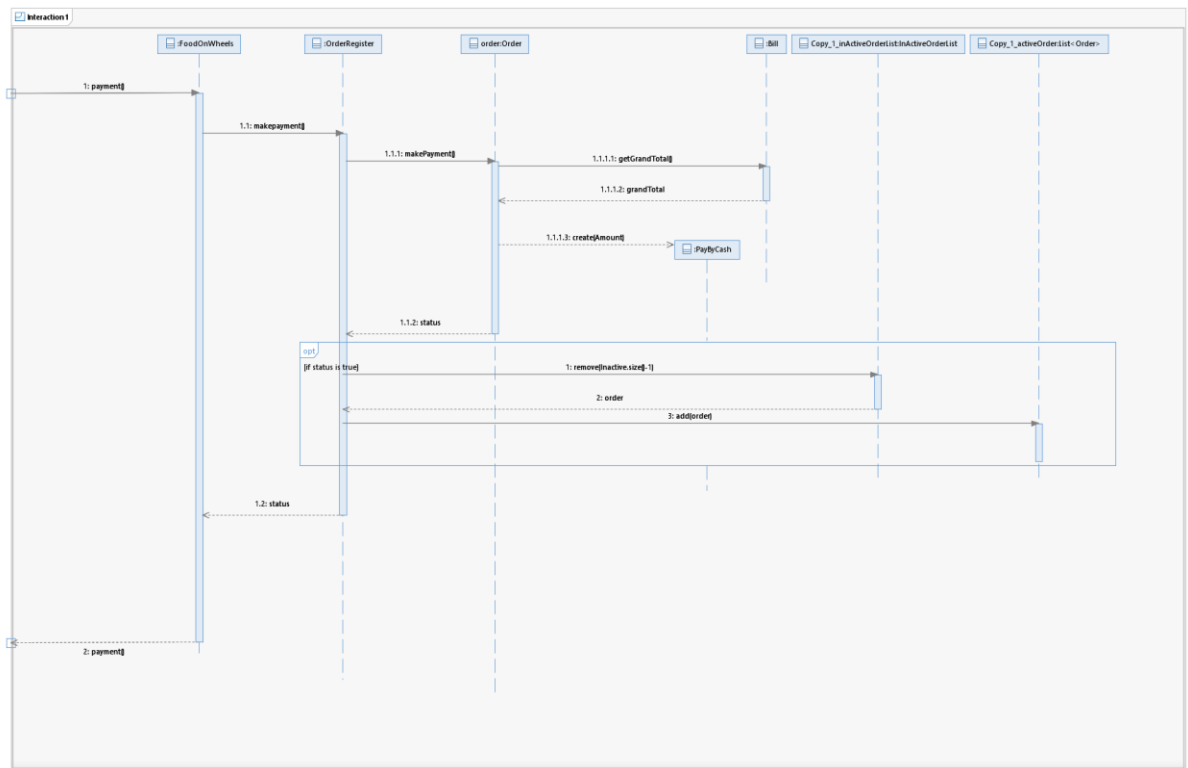
## 6.6 Get Payment Method



## 6.7 Pay by Card

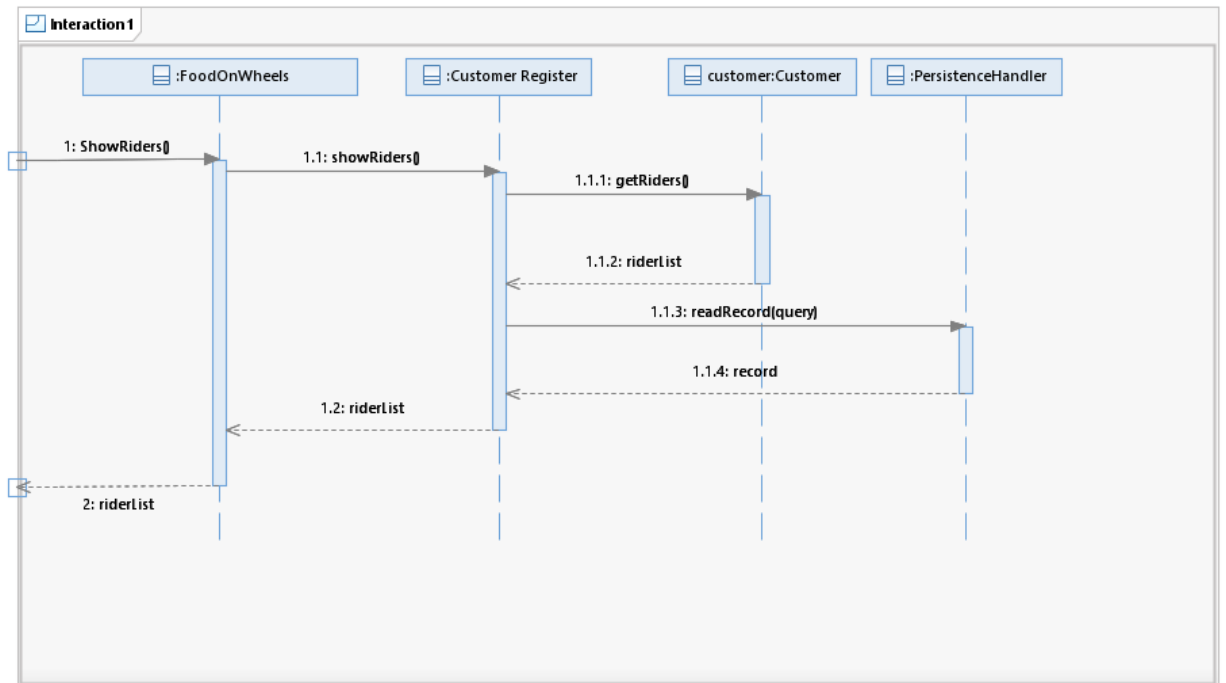


## 6.8 Pay by cash

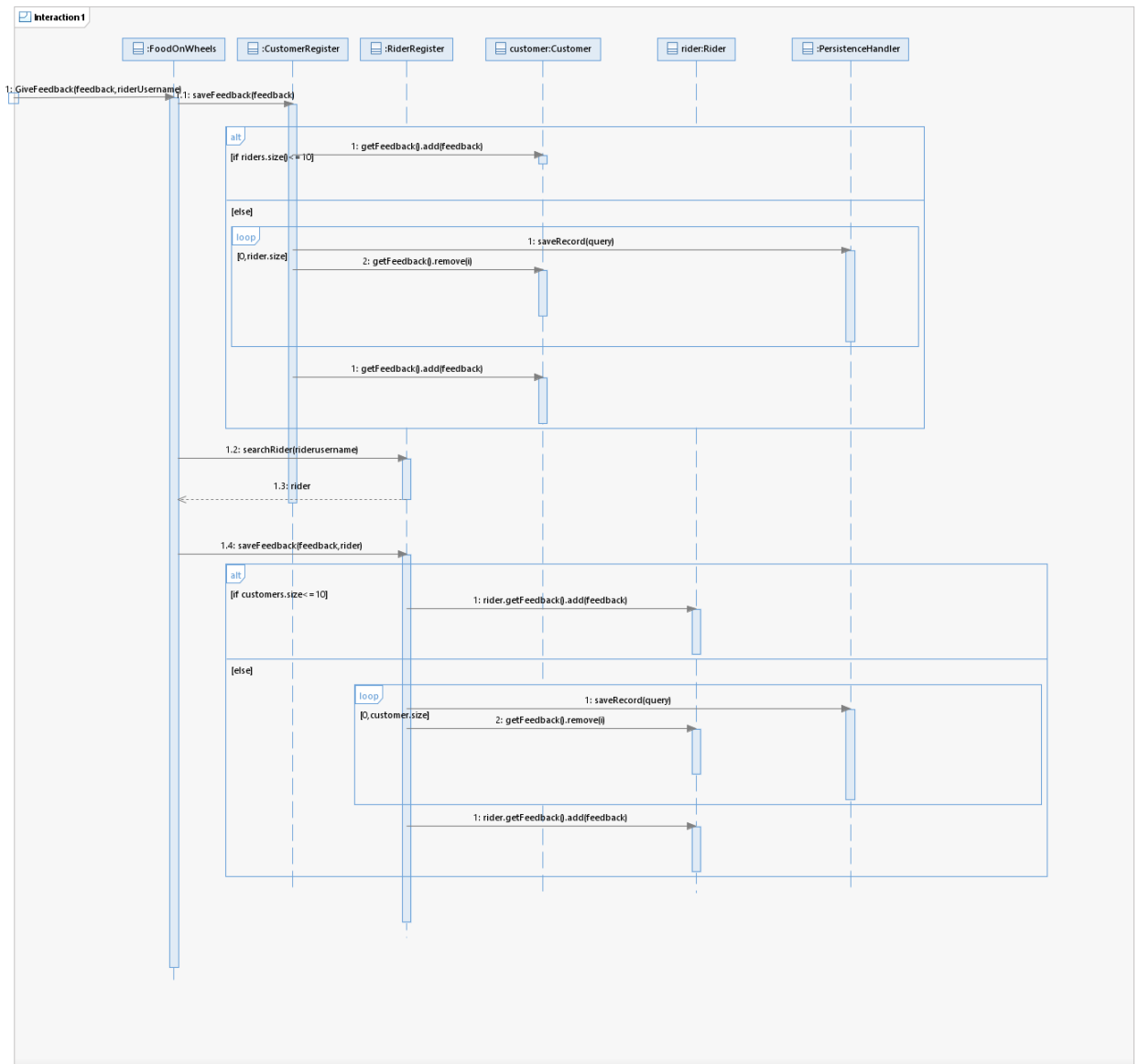


## 7. Give Feedback

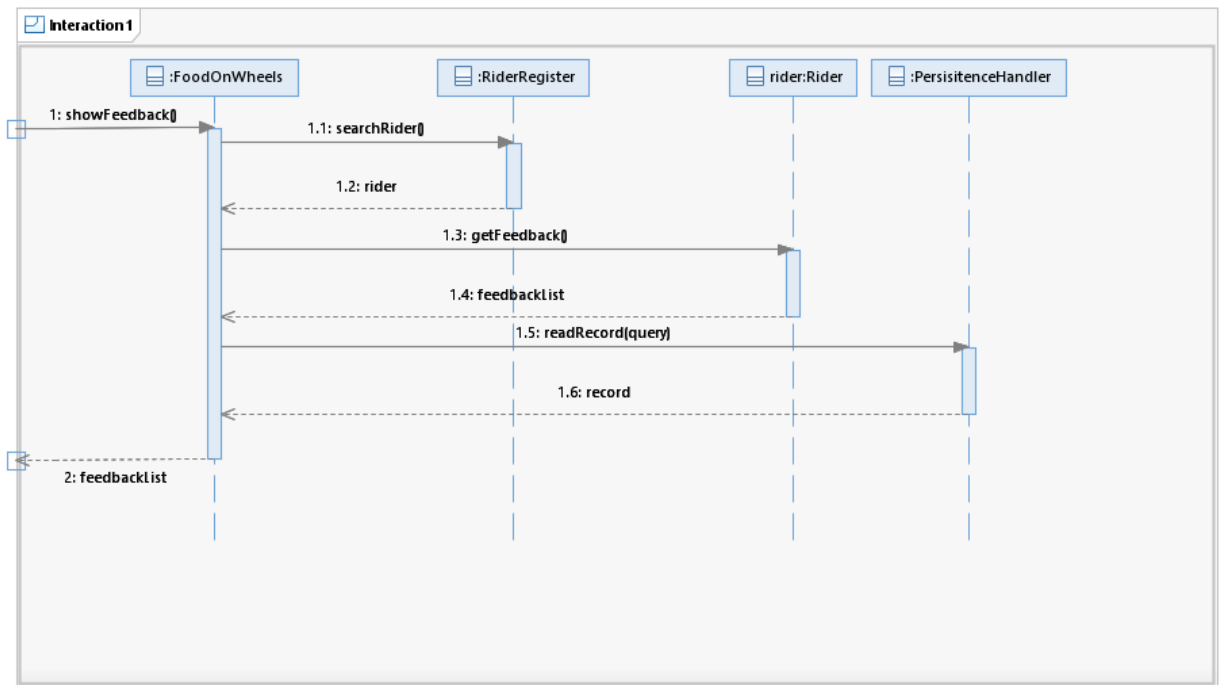
### 5.1 Show Riders



## 5.2 Give feedback

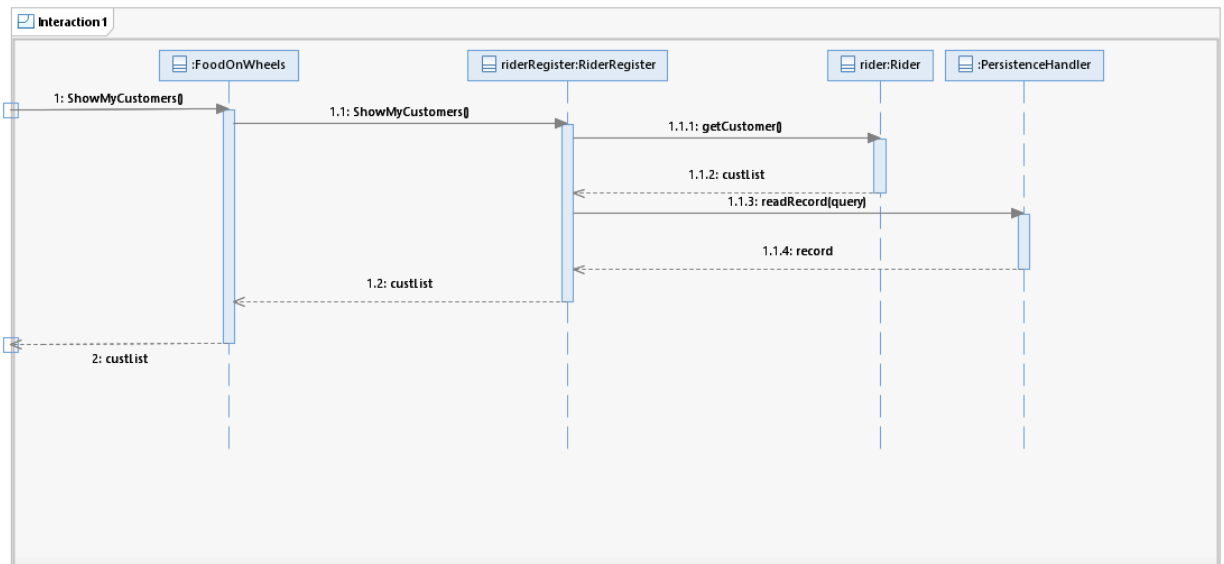


### 5.3 show Feedback

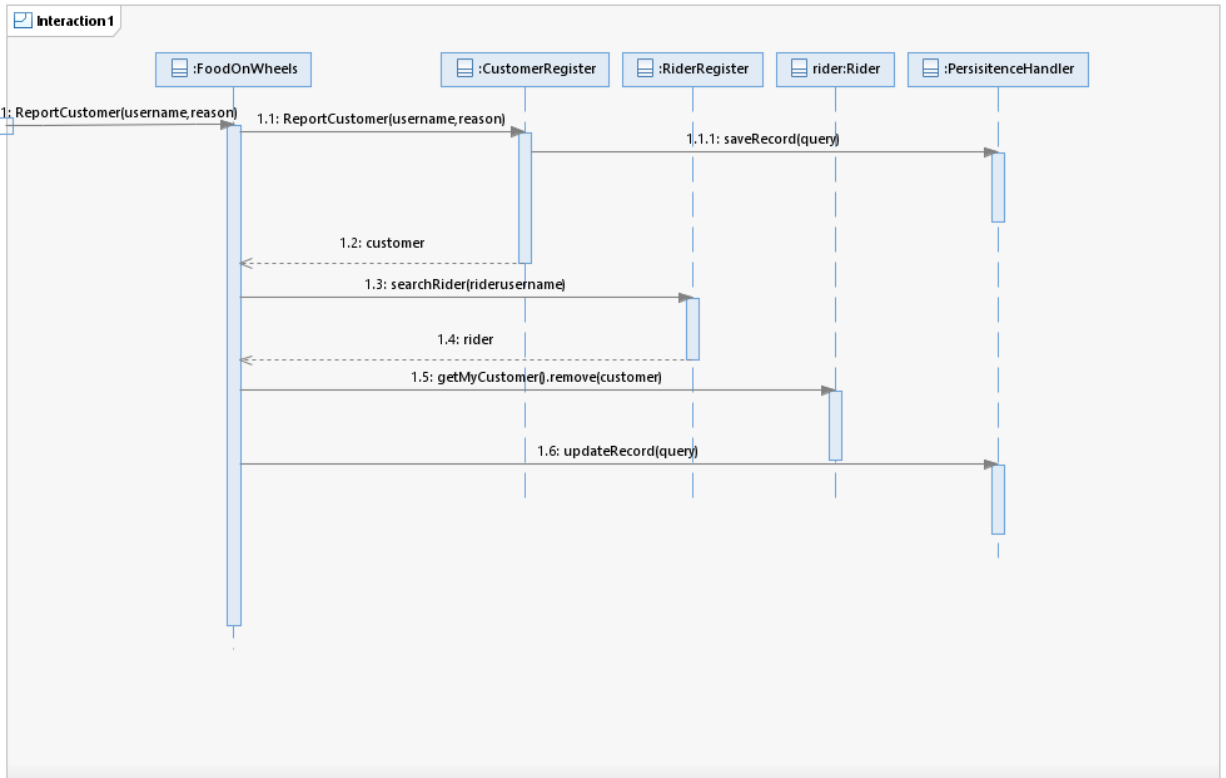


## 6. Report Customer

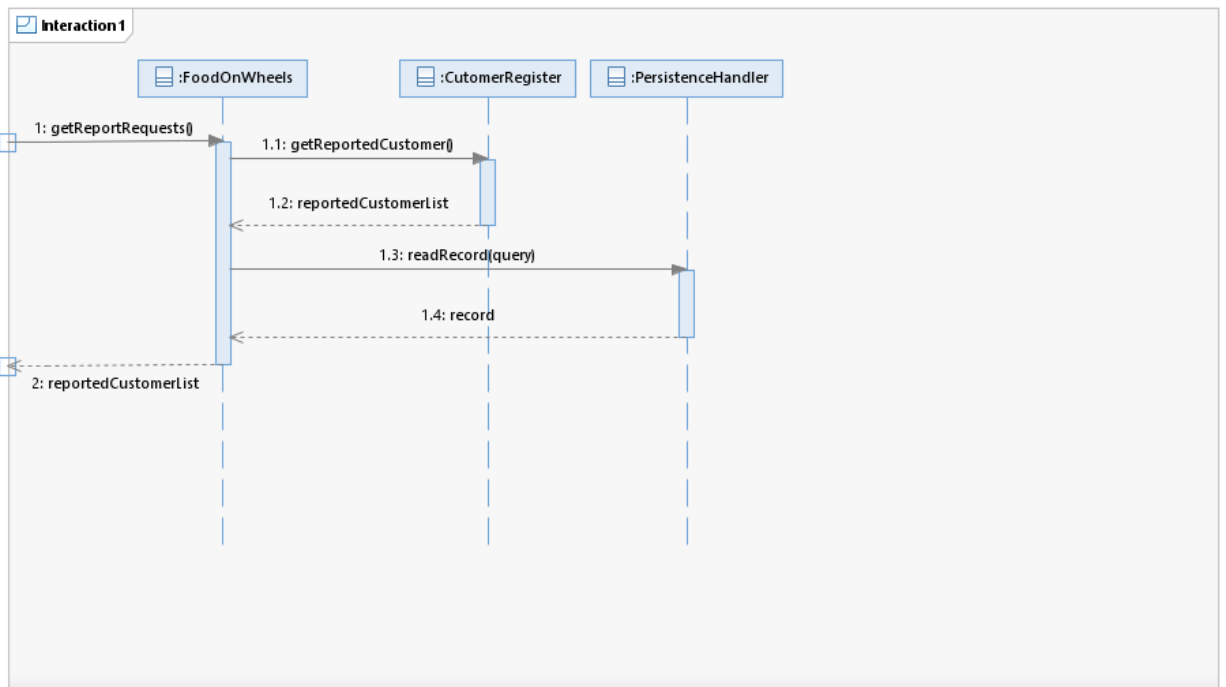
### 6.1 Show customers list of Rider



## 6.2 Report Customer

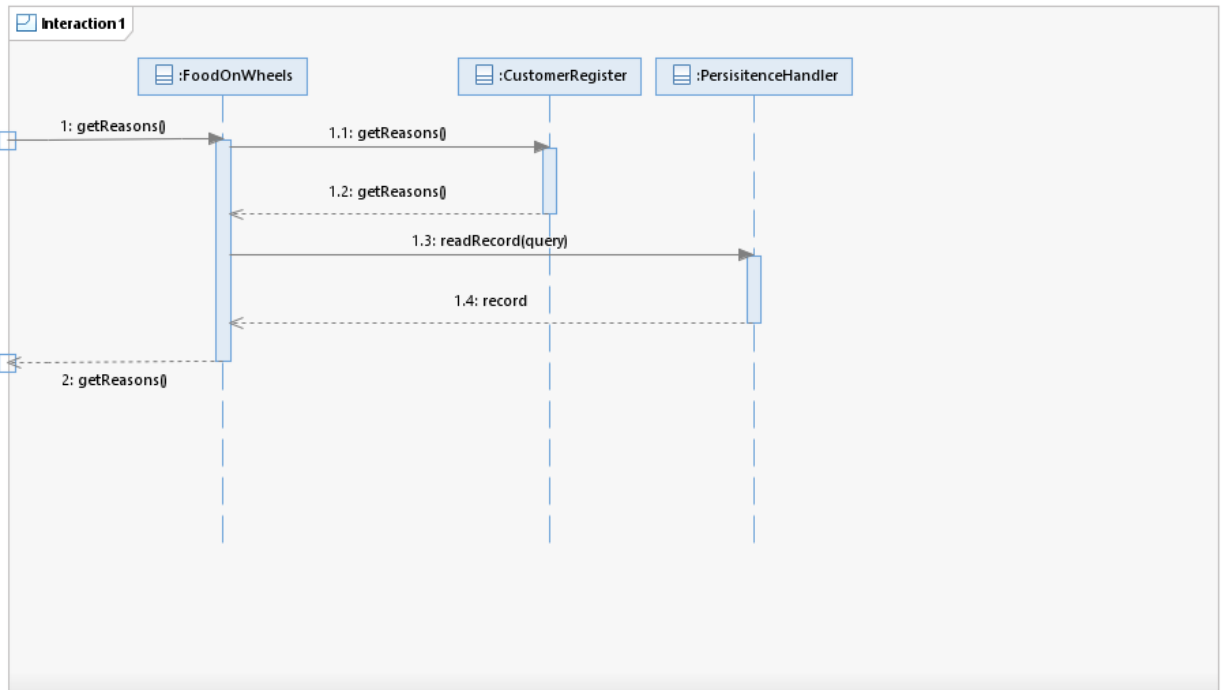


## 6.3 Reported Customers List

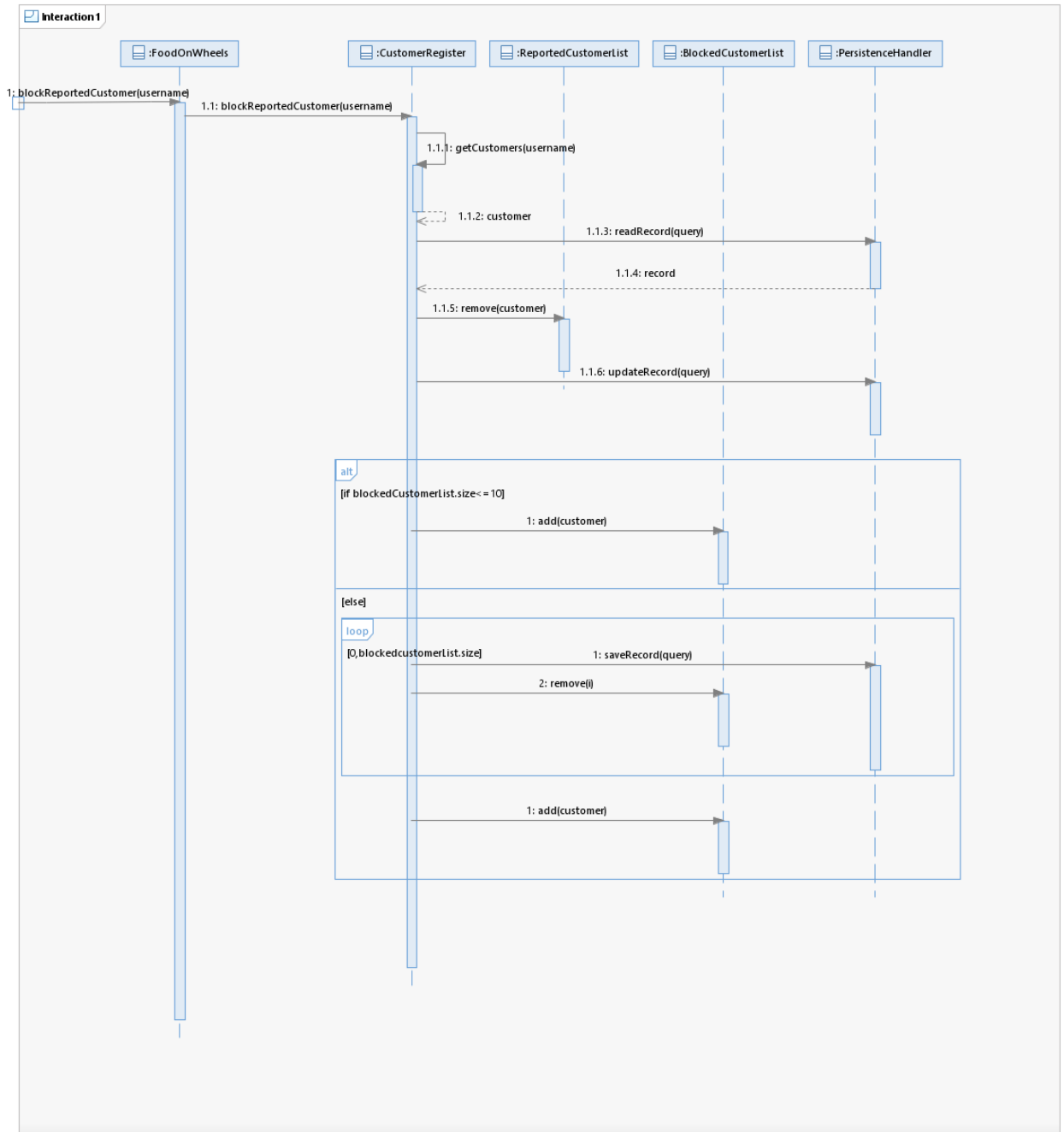




## 6.4 Show Reasons List



## 6.5 Block reported Customers



## 7. Manage Profile (Customer)

