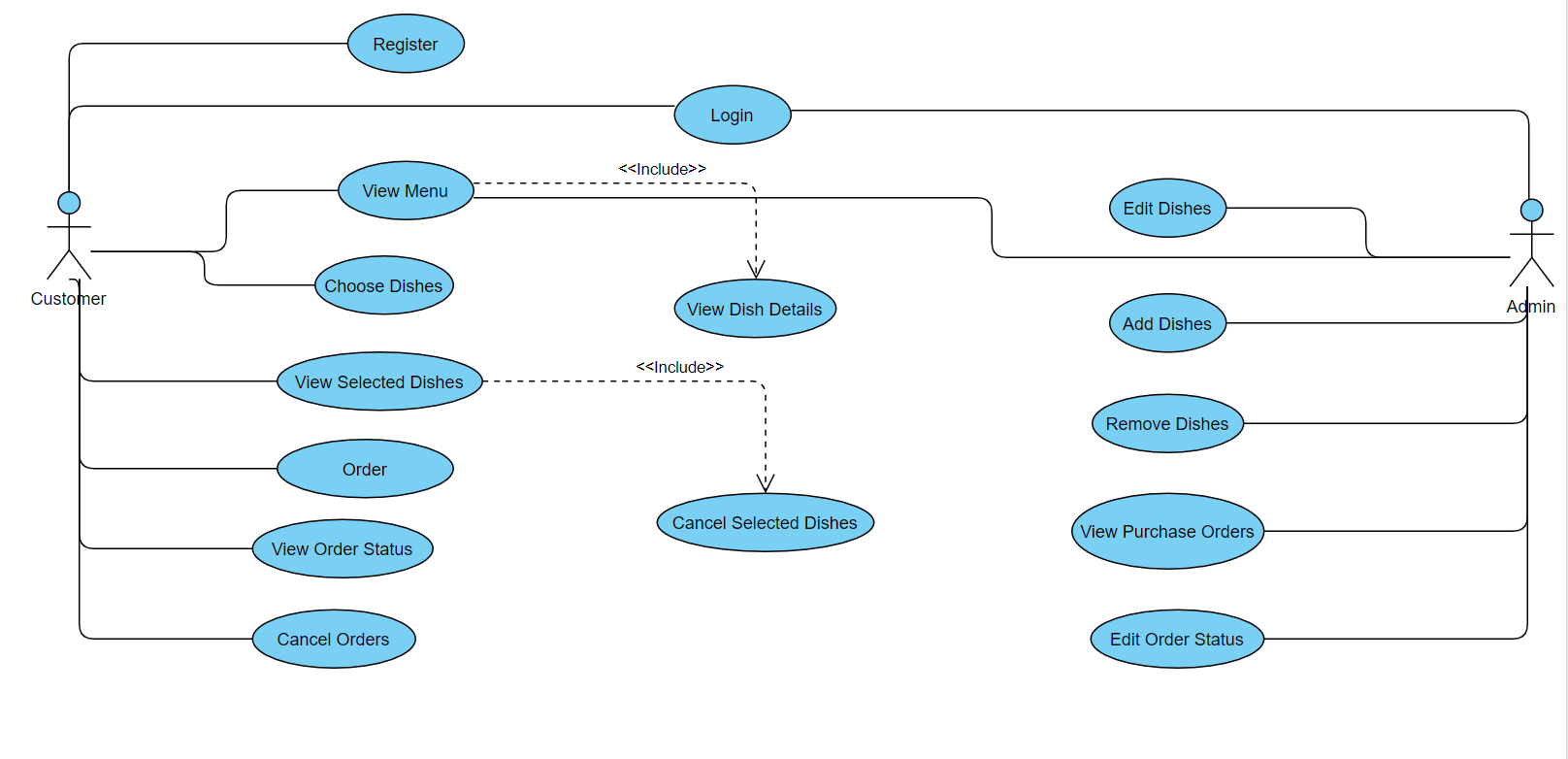
**Topic: A restaurant ‘s dish order management**

**Requirement:**

* User: needs an account to use functions in the app such as: View the list of dishes from there, View details of dishes, Select dishes, View selected dishes, Place orders, View menu status order ordered, Cancel order.
* - Admin: needs an account to perform functions such as: Update dishes (Add dishes, Modify dishes, Delete dishes), View ordered orders, Change order status

1. *General Usecase Diagram*

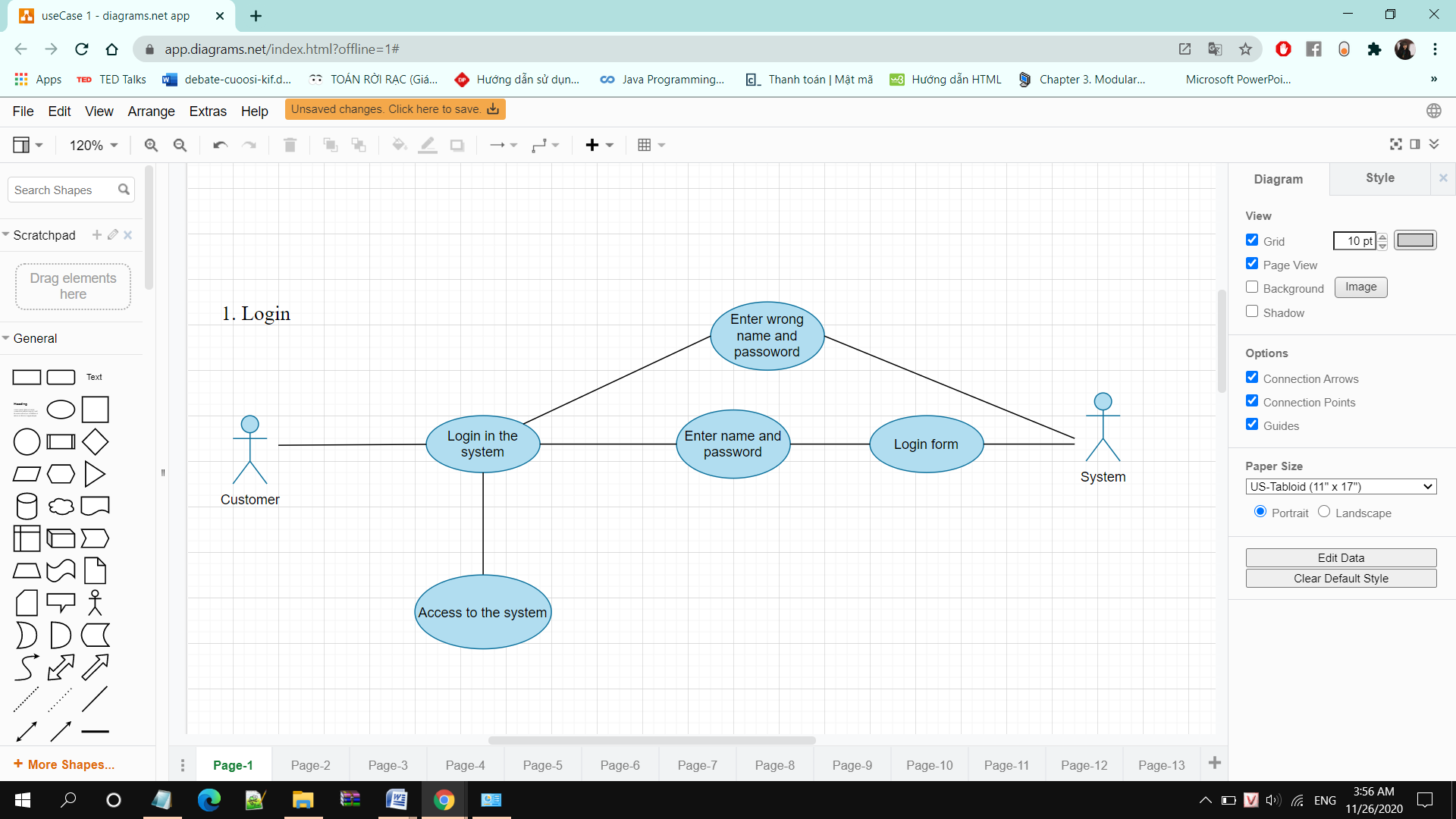


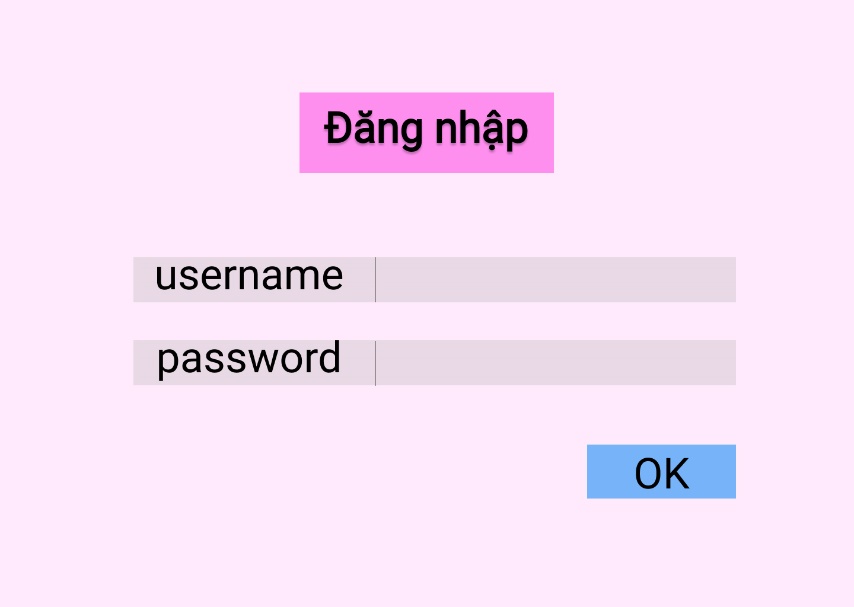
1. *Usecase specification (scenario)*
2. **Register**

|  |  |
| --- | --- |
| Use case | Register |
| Actors | Customer , system |
| Description | Customer register account in the system |
| Triggers | Customer clicks on “Sign up” button |
| Pre-conditions | Customer input information following to registration form request |
| Post-conditions | System register successfully and redirect to Home pages |
| Main Event Flow | 1. Customer access to the application 2. The system returns the registration form 3. Customer enters personal information to registration form 4. The system validates and confirm account |
| Exception Flow | * 1. Customer did not enter required data      1. The system notifies users to re-enter   2. Customer enters wrong input against request      1. The system notifies users to re-enter |

1. **Login**

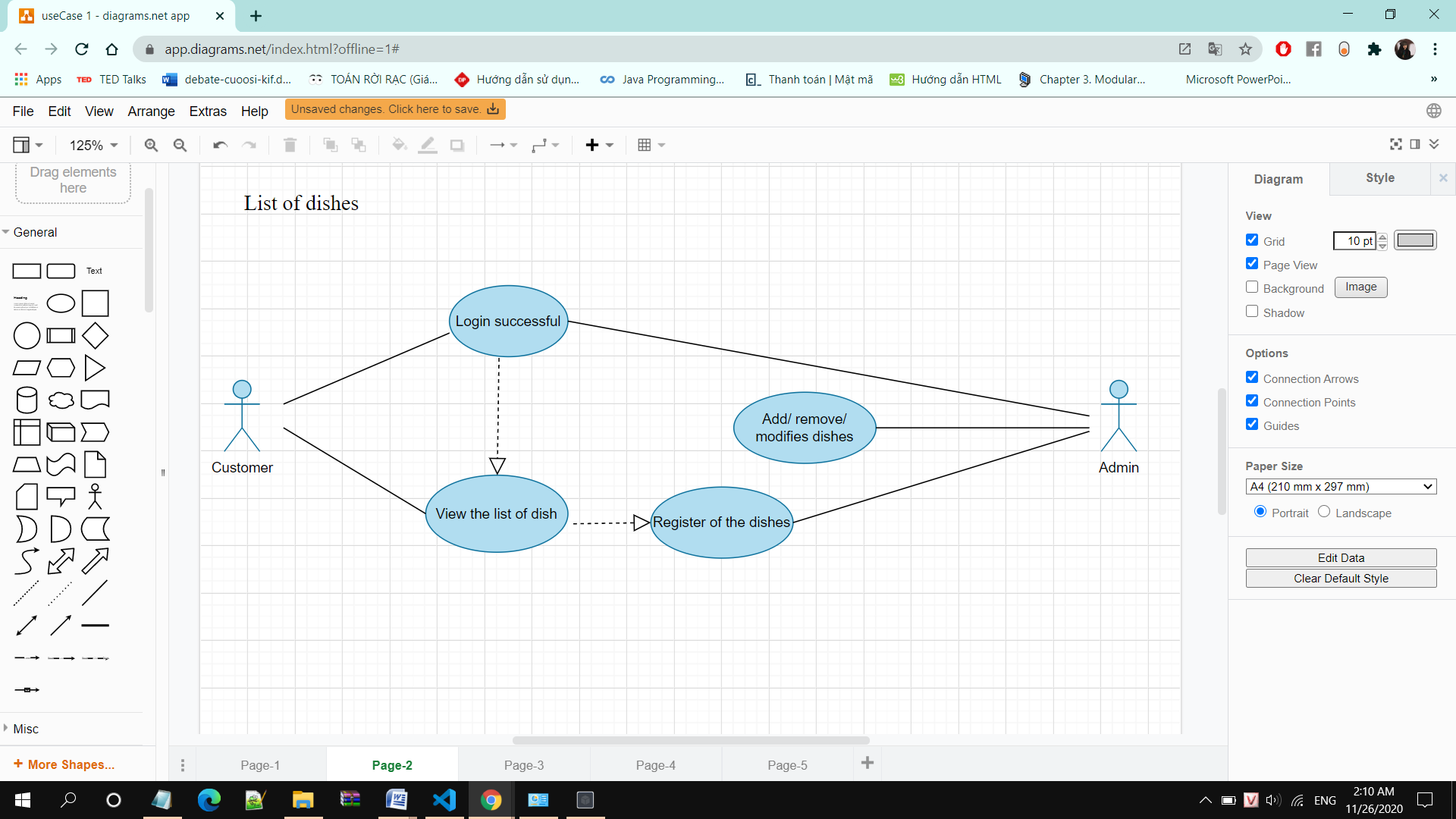
|  |  |
| --- | --- |
| Use case | Login |
| Actors | Customer , system |
| Description | Customer logins in the system |
| Triggers | Customer clicks on “login” button |
| Pre-conditions | Customer already has an account |
| Post-conditions | System return login form and login successfully |
| Main Event Flow | 1. Customer access to the application 2. The system returns the login form 3. Customer enters name and password and login 4. The system validates and returns the result. |
| Exception Flow | * 1. Customer did not enter name and password      1. The system notifies users to re-enter   2. Customer enters wrong name and password      1. The system notifies users to re-enter |

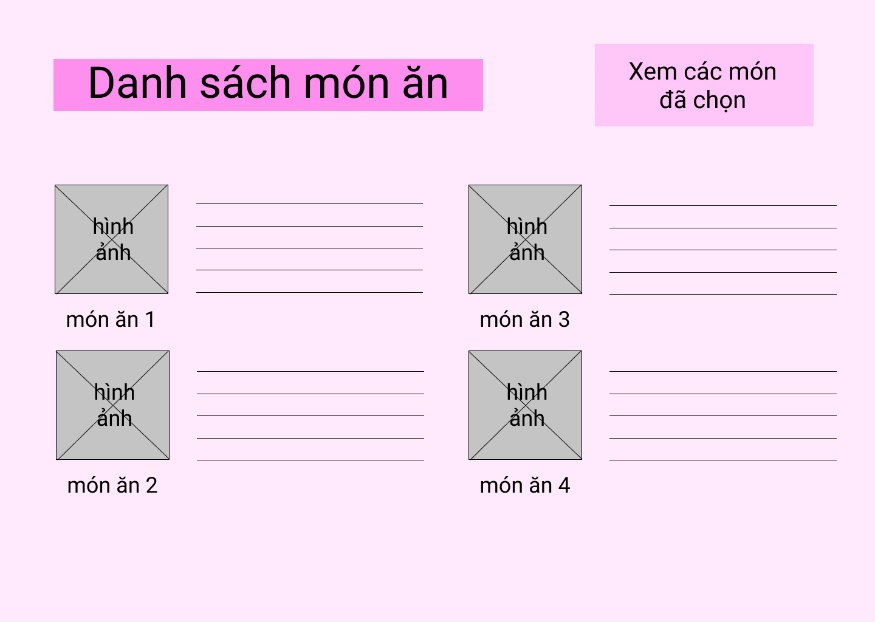




1. **View menu**

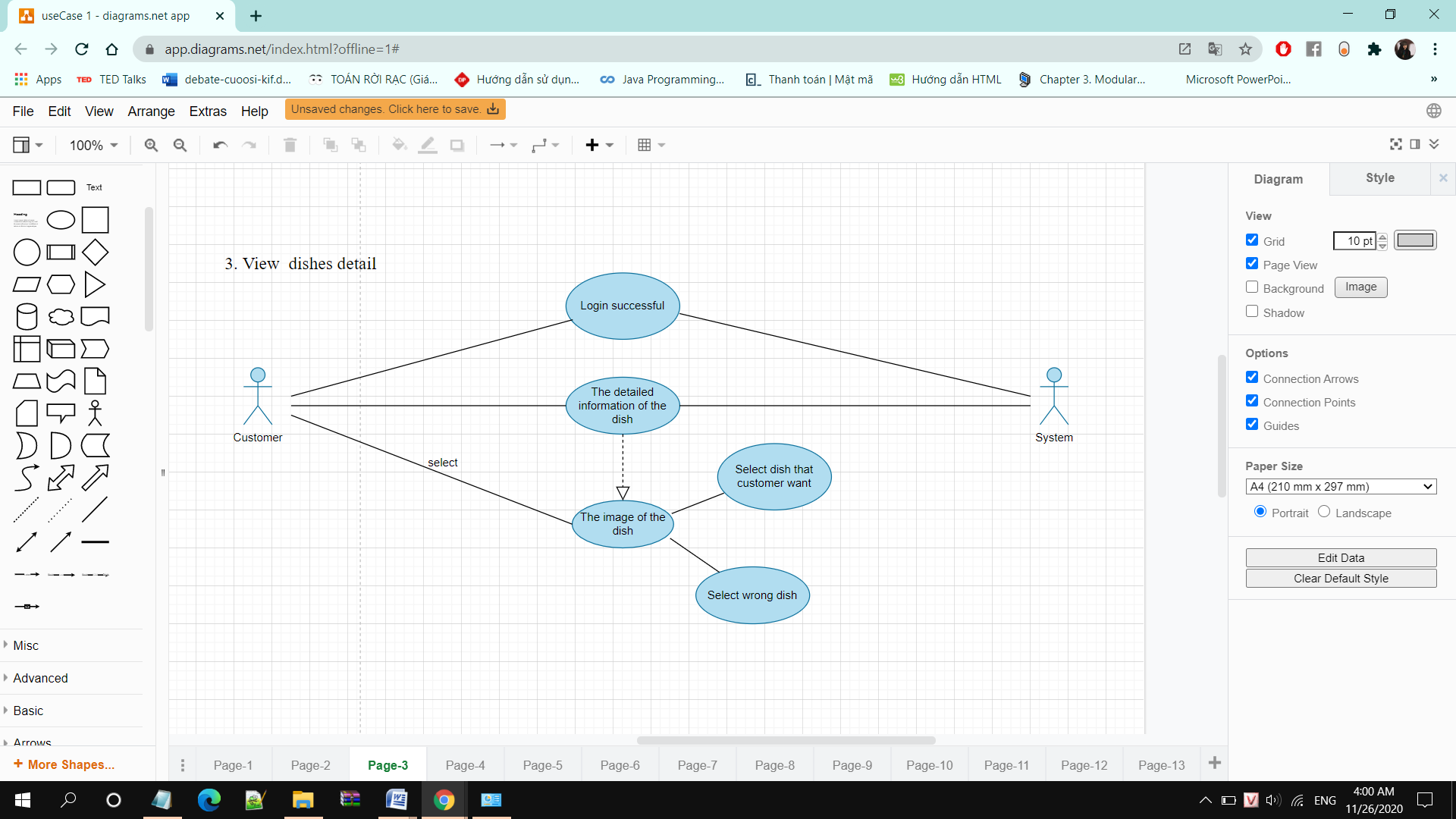
|  |  |
| --- | --- |
| Use Case | View menu |
| Actors | Customer, admin |
| Description | Customer views the list of dishes |
| Triggers | Customer logins successfully |
| Pre-conditions | Customer already has an account |
| Post-conditions | Customer views the entire the list of the dishes |
| Main Event Flow | 1. After user login successfully   The system displays the entire list of dishes in the main screen interface   1. Users view order dishes |
| Exception Flow | * 1. Customer did not enter name and password   The system notifies users to re-enter   * 1. Customer enters wrong name and password   The system notifies users to re-enter |

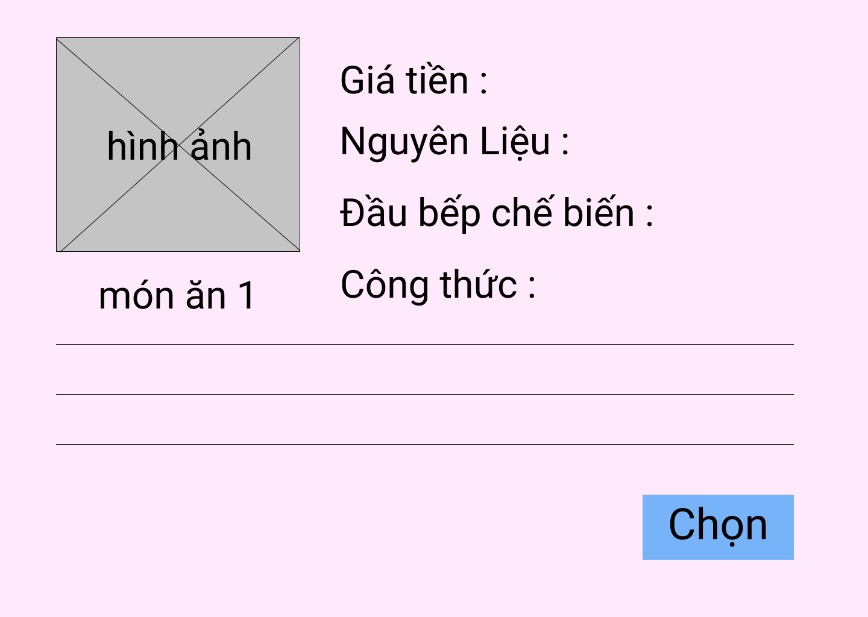
****

****

1. **View product details**

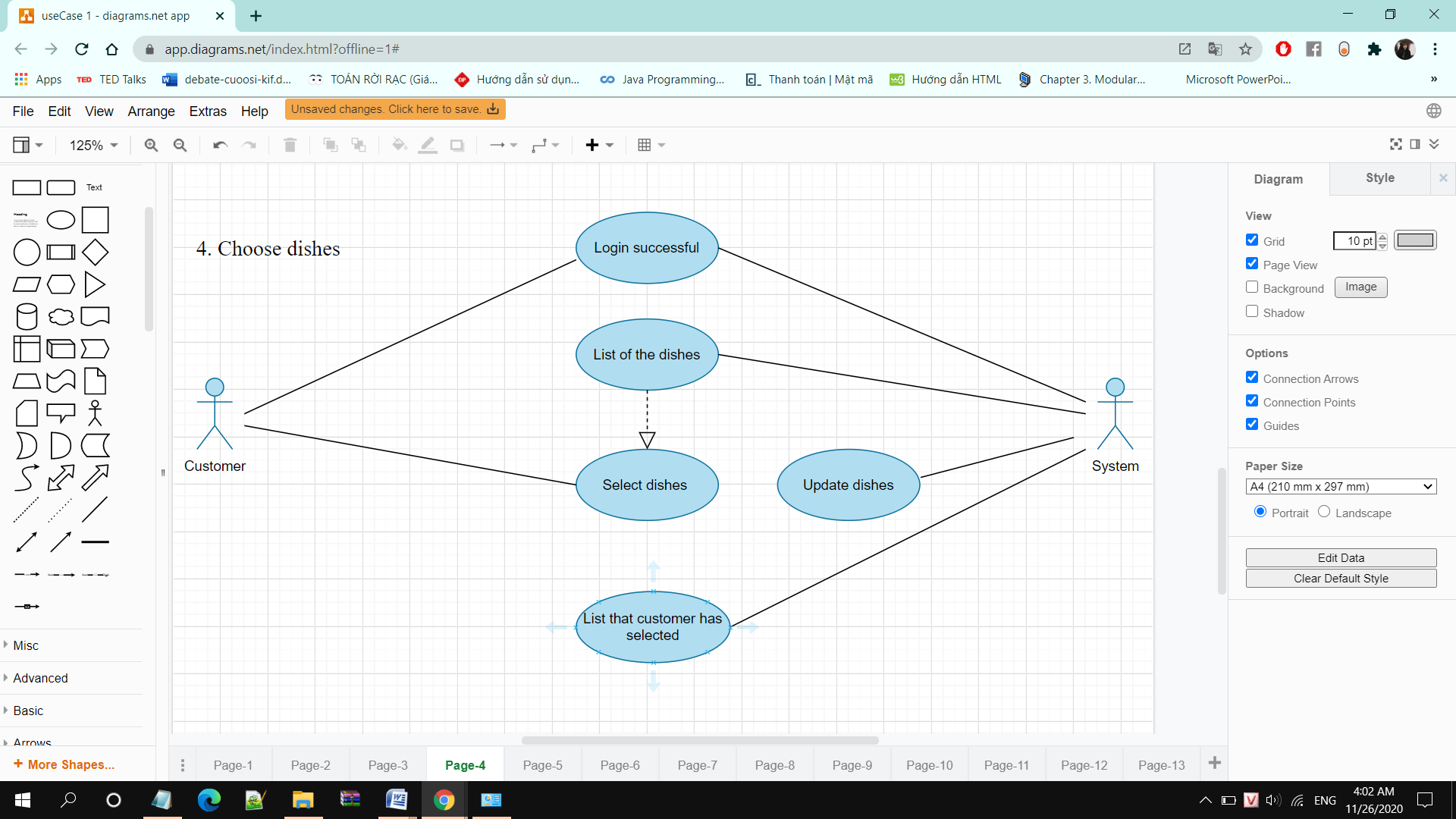
|  |  |
| --- | --- |
| Use Case | View dish details |
| Actors | Customer , system |
| Description | Customer views detailed information about the dish |
| Triggers | Customer clicks on the image of a dish in the list |
| Pre-conditions | Customer has account and login successful |
| Post-conditions | Customer views the completed about the dish |
| Main Event Flow | 1. Customer clicks on the images of the dish 2. The system retrieves data and displays detailed information about that dish |
| Exception Flow | Customer chooses the wrong dish  The system return the screen interface |

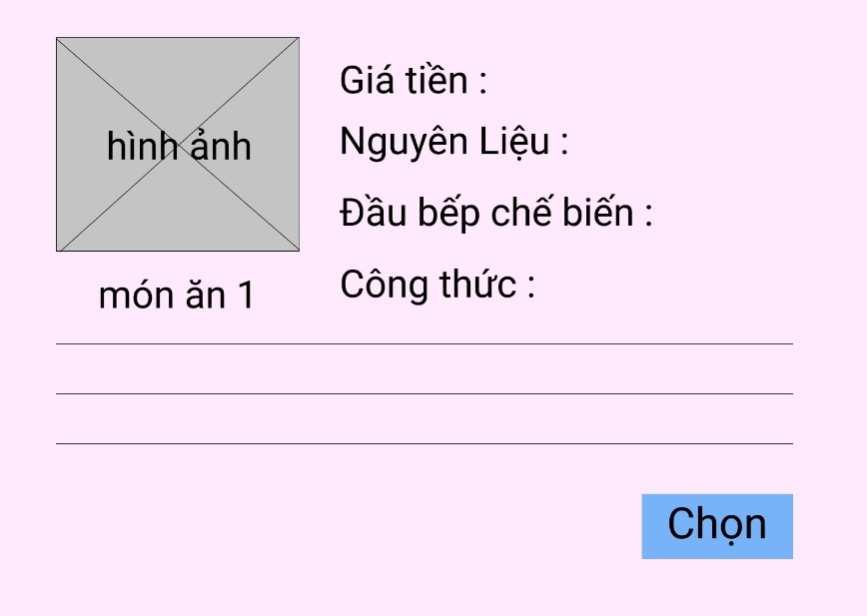
****

****

1. **Choose products**

|  |  |
| --- | --- |
| Use Case | Choose dishes |
| Actors | Customer , system |
| Description | Customer chooses one or more dishes |
| Triggers | Customer clicks on the “Select” on the dish details |
| Pre-conditions | Customer has account and login successful |
| Post-conditions | Customer return the main interfaceand chooses the desired dish |
| Main Event Flow | 1. Customer choose what dish he want 2. The system processes and display a notification that user has chosen and add to the list that the user has selectd |

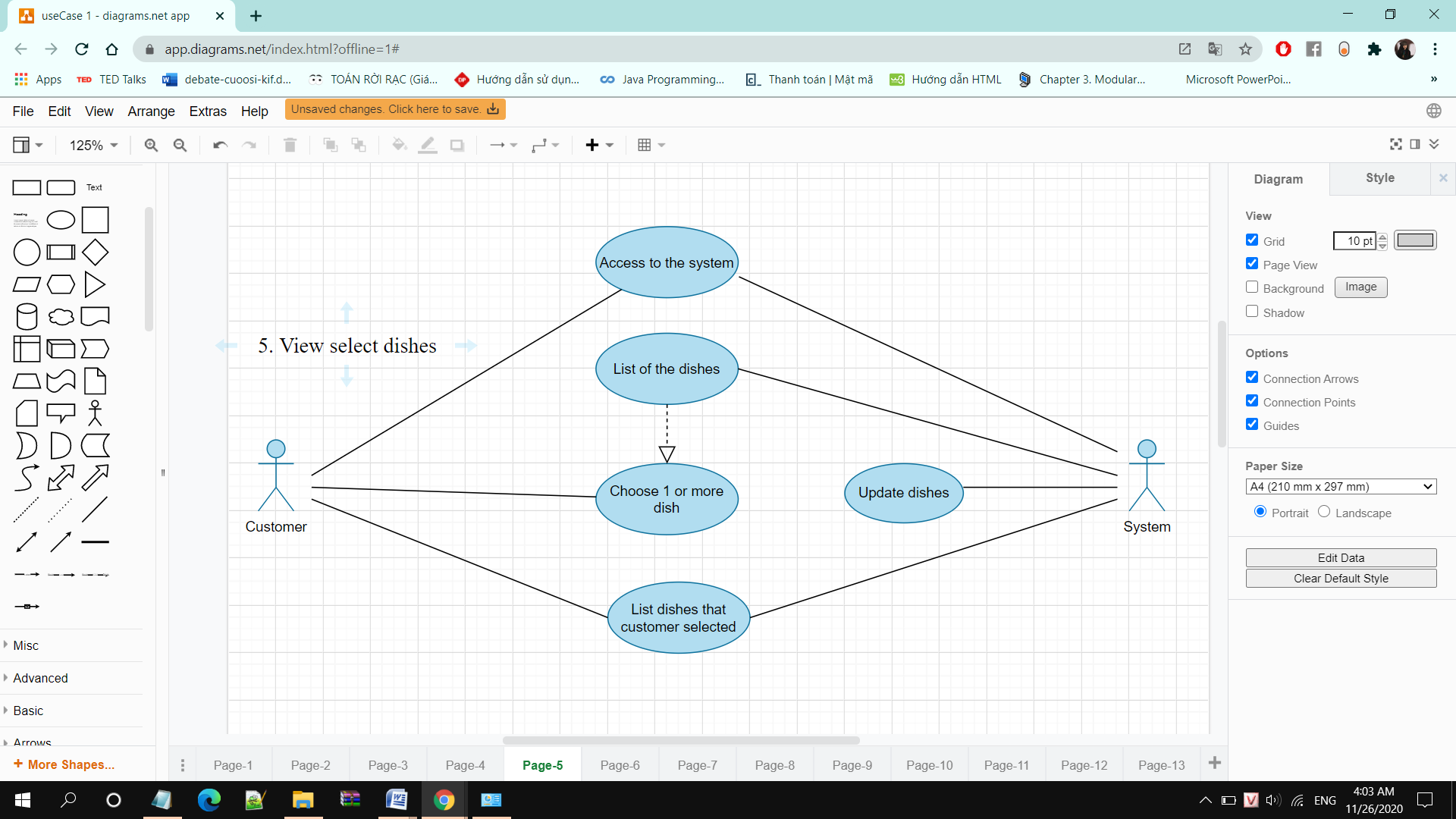
****

****

you selected this dish

1. **View selected dishes**

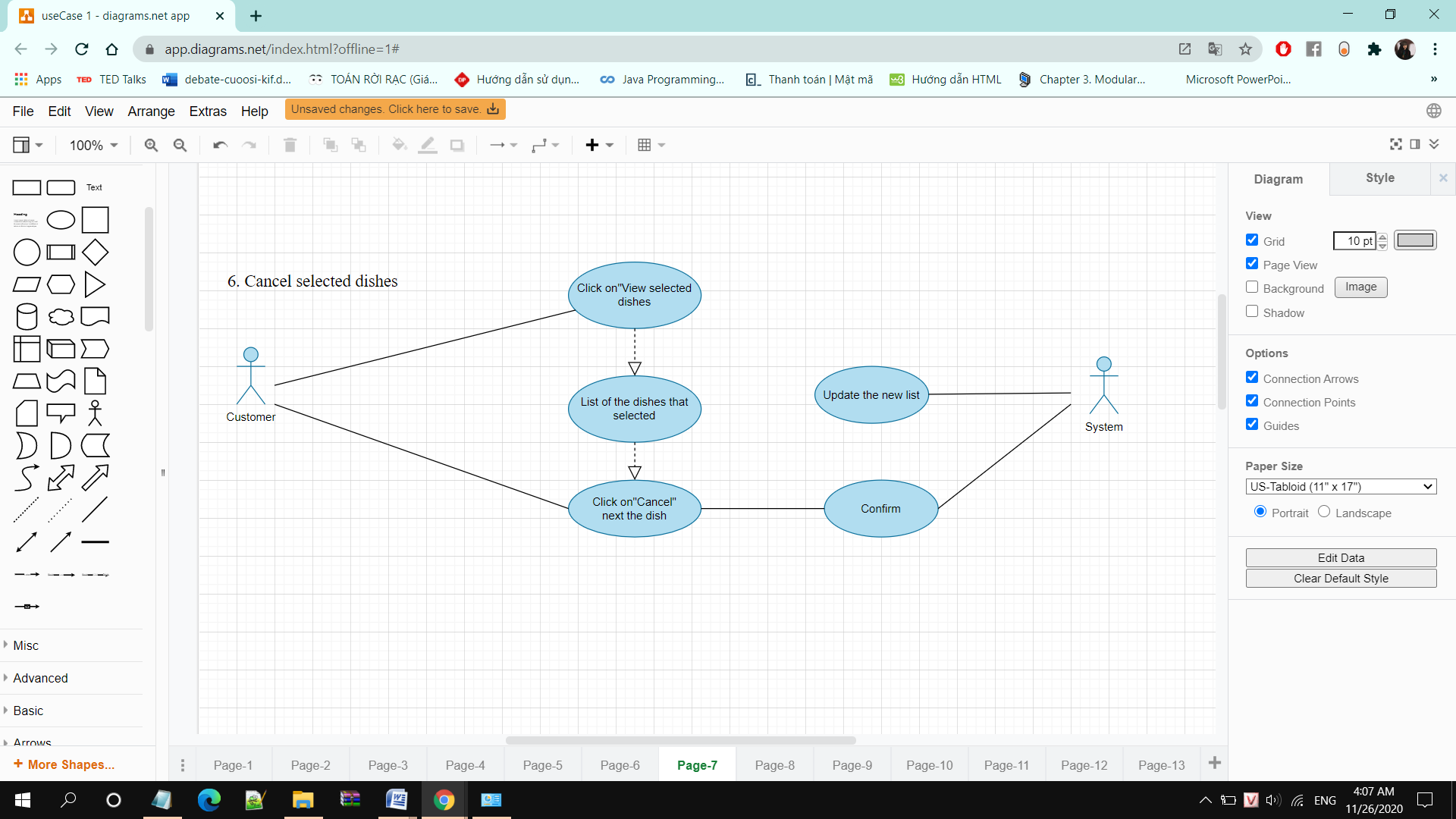
|  |  |
| --- | --- |
| Use Case | View selected dishes |
| Actors | Customer , admin |
| Description | Customer sees a list of the dishes they have selected |
| Triggers | Customer clicks on the “ View selected dishes” on the main interface |
| Pre-conditions | Customer selected 1 or more dishes |
| Post-conditions | Customer see the list of the dishes that they have selected |
| Main Event Flow | 1. If the dish chosen has been deleted, the system will display the deleted dish   Customer click on “Cancel” that dish |

****

****

1. **Cancel selected dishes**

|  |  |
| --- | --- |
| Use case | Cancel selected dishes |
| Actors | Customers, system |
| Description | Customers cancel 1 or more of the dishes they have chosen |
| Triggers | Customers click "cancel" next to the dish |
| Pre-conditions | Customers have selected 1 or more dishes and they are in view of the selected dishes |
| Post-condition | Customers will cancel the dishes |
| Main Event Flow | 1. Customers click on "View selected dishes" on the main interface  2. The system processes and displays the list of items that the user has selected  3. Customers click on the word "Cancel" next to the dish  4. The processing system cancel that item in the List of selected dishes  5. The system displays the list of selected dishes again |

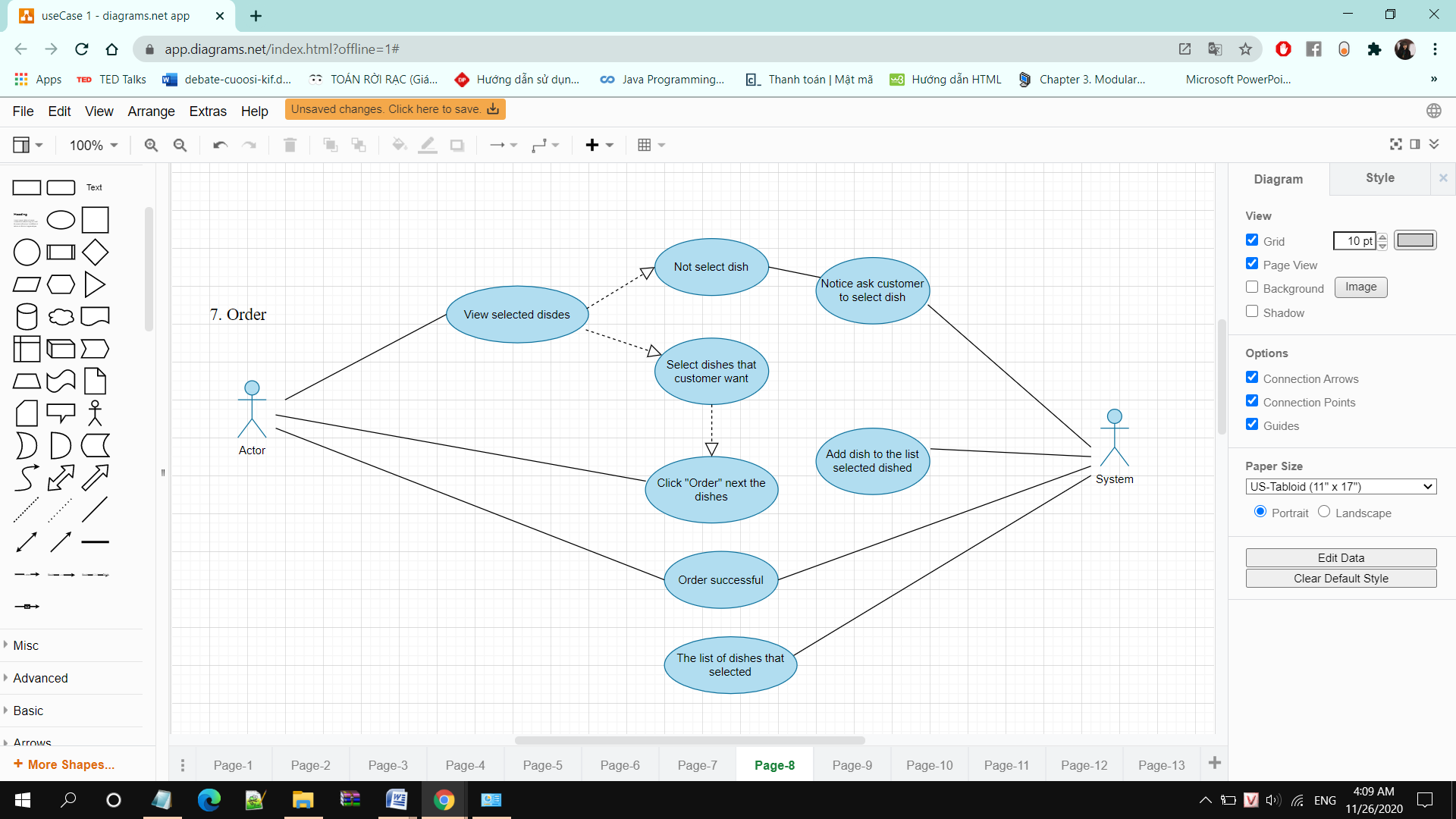
****

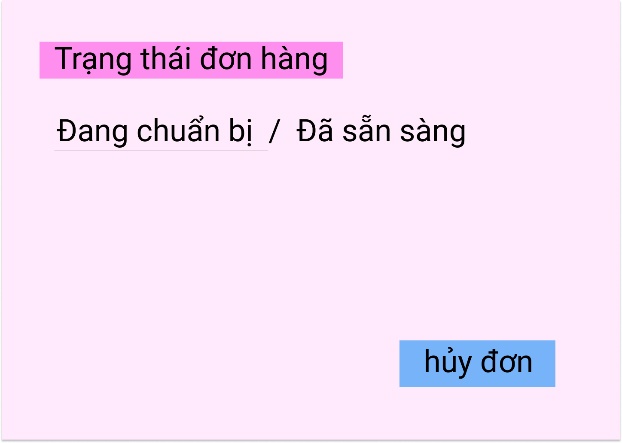
Cancel successfully

** **

1. **Order**

|  |  |
| --- | --- |
| Use case | Order |
| Actors | Customers, admin |
| Description | Customers order the dishes of their choice |
| Triggers | Customers click “order” on interface view selected dishes |
| Pre-conditions | Customers have selected 1 or more dishes and they are in view of the selected dishes |
| Post-conditions | Customers ordered successfully |
| Main Event Flow | 1. The customers click on "Order" on the interface of View selected dishes  2. The system processes and displays the notice of a successful order and switches to the View Order Status interface |
| Exception Flow | * 1. Customers have not chosen the item but click on the order   1.1.2. The system displays a notice asking the customers to choose an item before ordering |

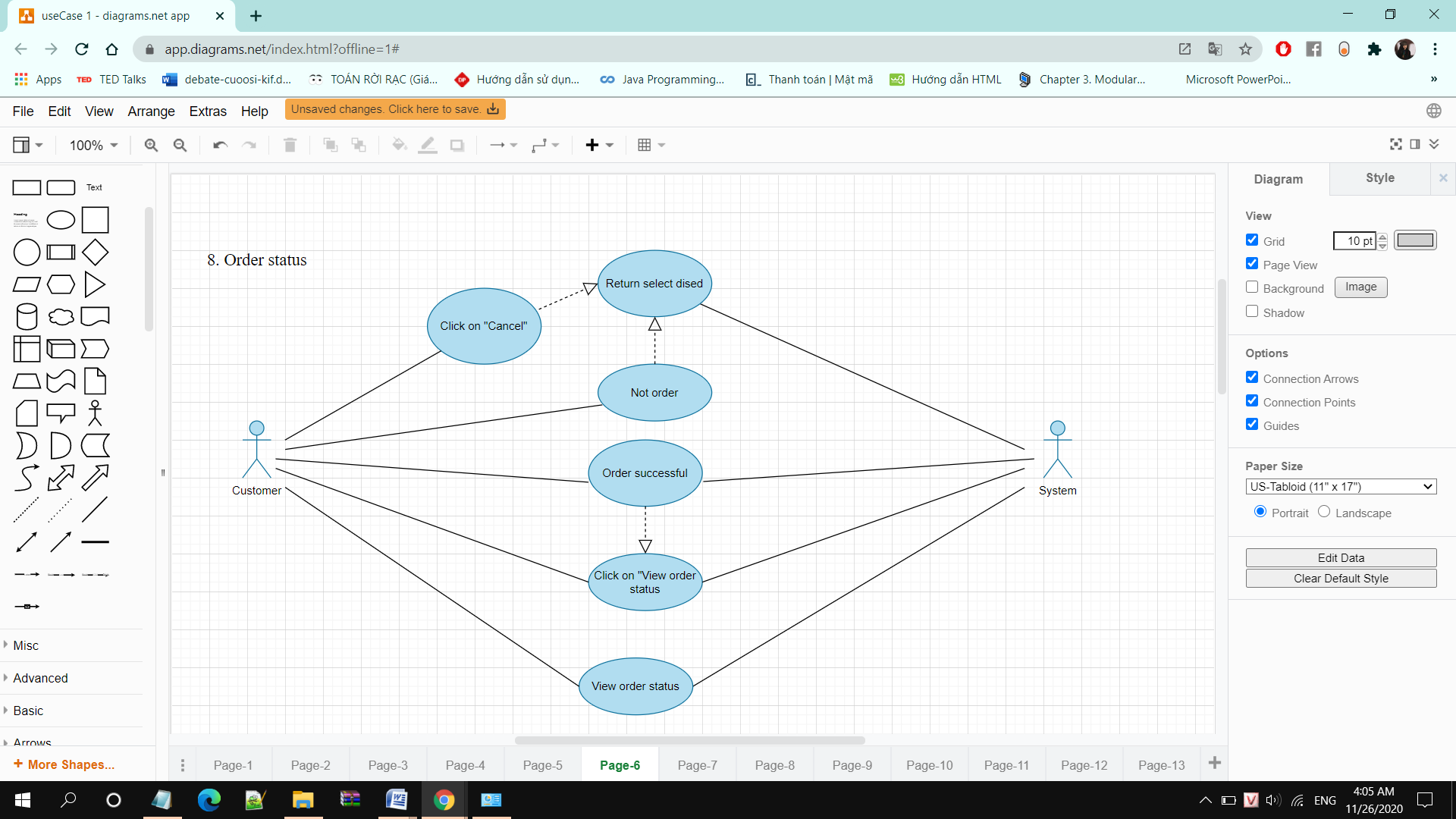
****

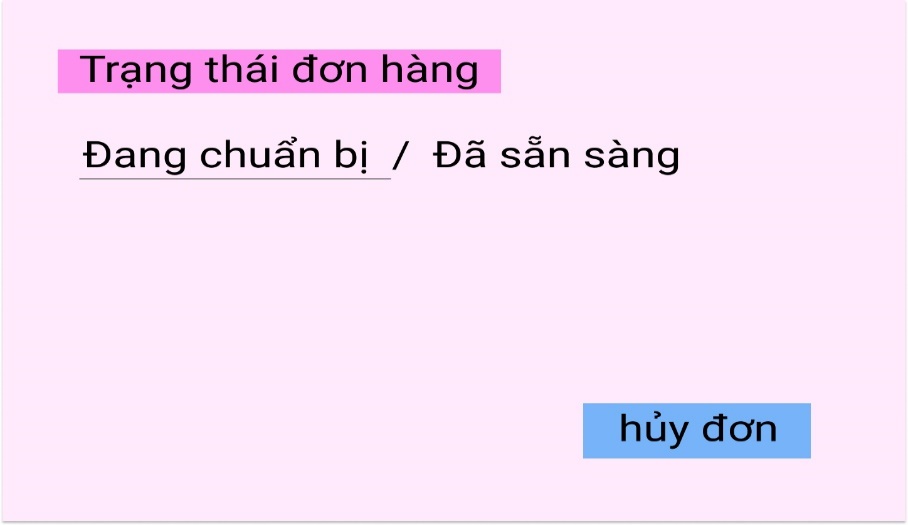
****

Order successfully

1. **View order status**

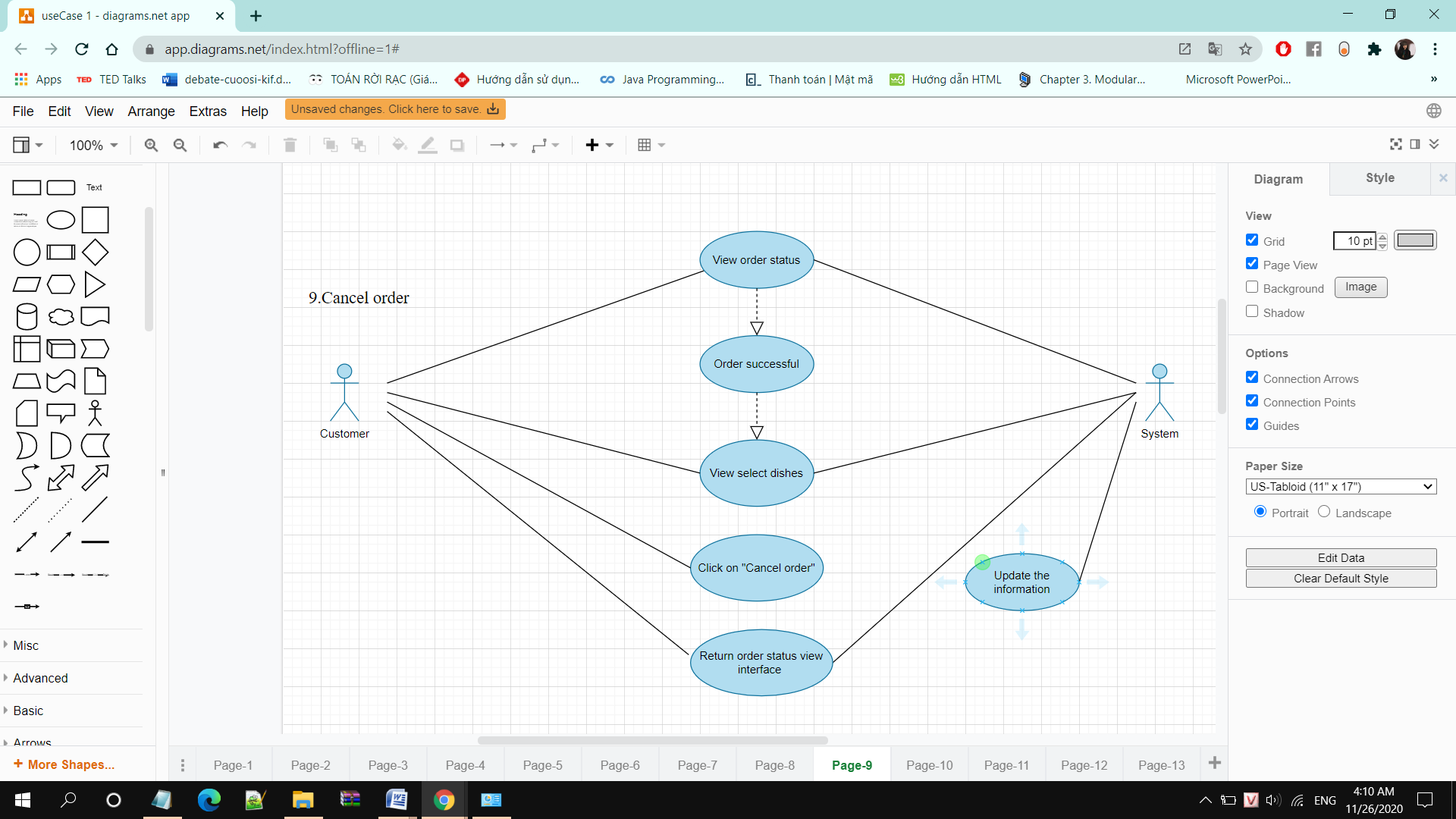
|  |  |
| --- | --- |
| Use case | Order status |
| Actors | Customers, admin |
| Description | Customers view order status |
| Triggers | Customers click “View order status” on inteface view selected dishes |
| Pre-conditions | Customers have ordered successfully |
| Post-conditions | Customers viewed order status |
| Main Event Flow | 1.Customers click on "View Order Status"  2. The system displays the order status |
| Exception Flow | 1.1. Customers have not placed an order, click "View order status".  1.1.1. The display system invites users to place an order  1.2. Customers have canceled the order and clicks "View order status"  1.2.1. The display system invites users to place an order |

****

****

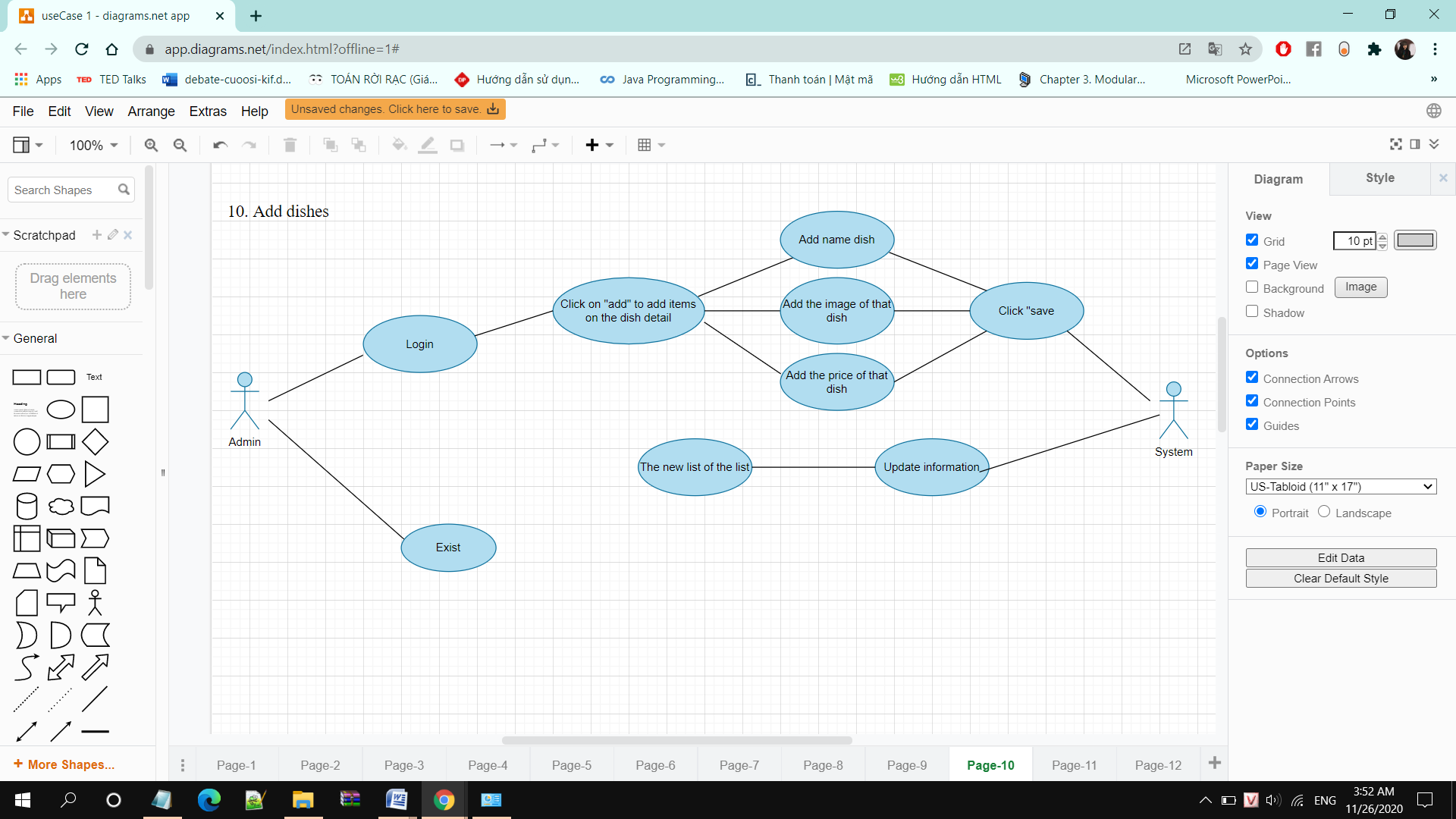
1. **Cancel orders**

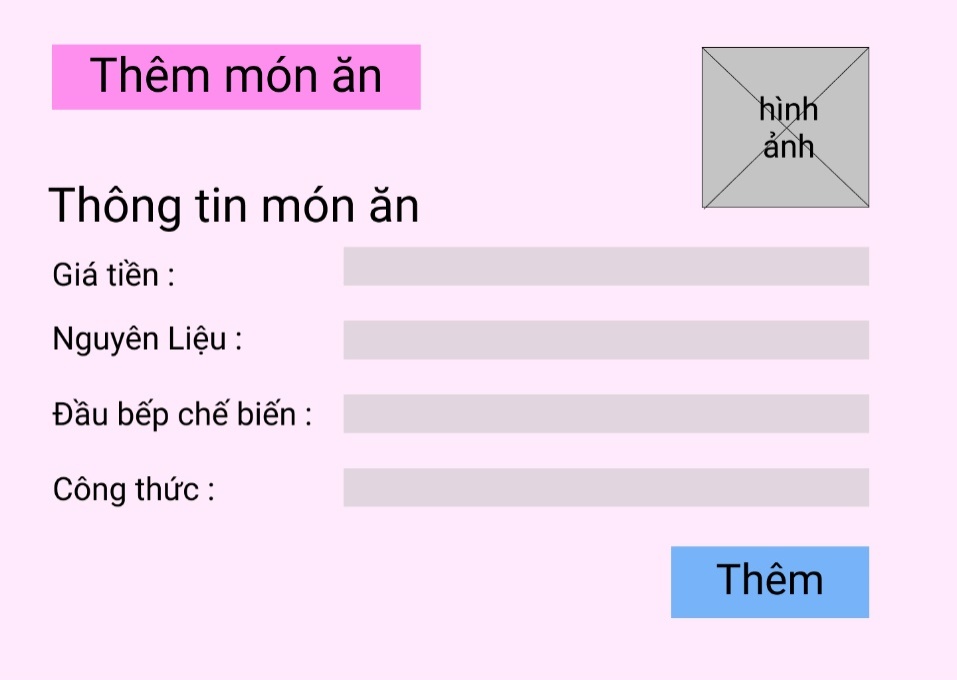
|  |  |
| --- | --- |
| Use case | Cancel orders |
| Actors | Customers, admin |
| Description | Customers cancel order that they have order |
| Trigger | Customers click “Cancel” on interface view selected dishes |
| Pre-conditions | Customers have ordered successfully and viewed order status |
| Post-conditions | Customers canceled order |
| Main Event Flow | 1. User clicks on “cancel order” on View order status interface  2. The system notifies the application cancellation confirmation  3. The user click”ok”  4. The system displays the user who has successfully canceled an order  5. The system returns to the order status view interface |

****

1. **Add product**

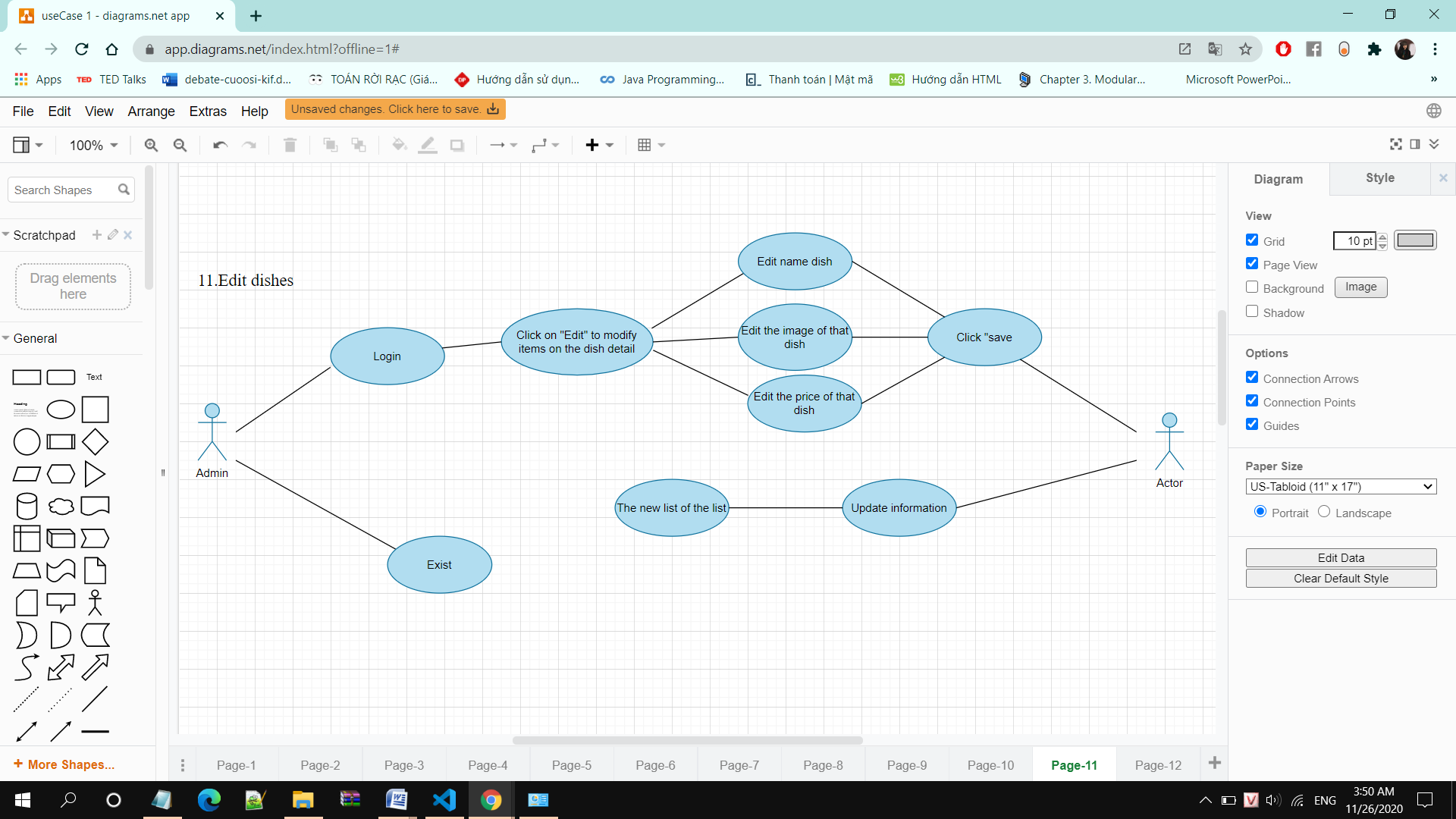
|  |  |
| --- | --- |
| Use Case | Add dishes |
| Actors | Admin |
| Description | Admin adds new dishes |
| Triggers | Admin click to add items on dish details |
| Pre-conditions | Admin has successfully logged in |
| Post-conditions | Admin returns to the main screen  The new dish is successfully added |
| Main Event Flow | 1. Admin click on "add" on the main interface 2. The system displays a form for the admin to add a new item 3. Admin follow the form and press save 4. The system checks and saves information in the database 5. The system displays “add successfully” |
| Exception Flow | * 1. . The admin adds an existing item in the List of dishes   3.1.1 The system check and notify that dishes already exist  3.1.2 admin add other dishes or exit |

****

****

1. **Edit product**

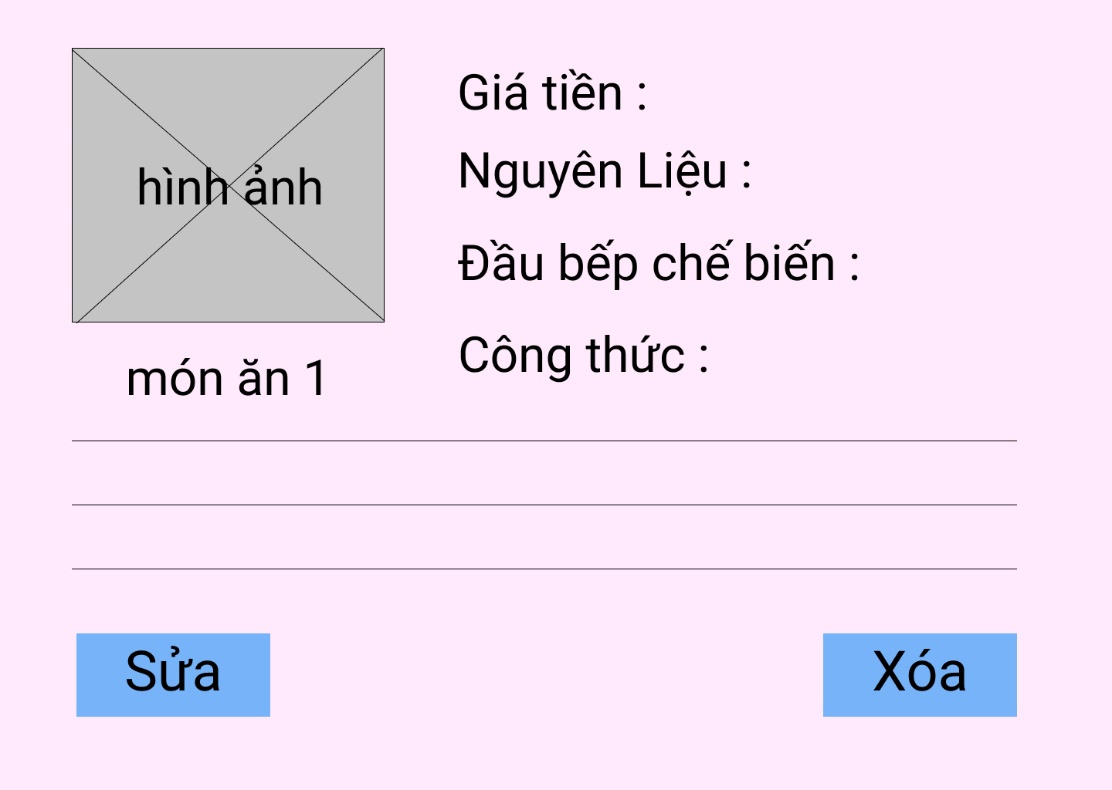
|  |  |
| --- | --- |
| Use Case | Edit selected dishes |
| Actors | Admin |
| Description | Admin modify information about dishes |
| Triggers | Admin click on edit dishes on dish detail interface |
| Pre-conditions | Admin login successfully |
| Post-conditions | Admin return main interface and edit information successfully |
| Main Event Flow | 1. Admin click on “Edit” on main interface 2. System display form to modify 3. Admin edit information in the form and press 4. The processing system saves the updated dish information 5. The system displays “add successfully” |

****

1. **Remove product**

|  |  |
| --- | --- |
| Use Case | Remove dishes |
| Actors | Admin |
| Description | Admin delete dishes |
| Triggers | Admin click on Delete dishes on dish detail |
| Pre-conditions | Admin login successfully |
| Post-conditions | Admin return main interface screen and dishes removed successfully |
| Main Event Flow | 1. Adminclick on “Delete” on own dish list interface 2. The system displays a message confirming deletion 3. Admin press delete 4. The system processes and displays deletion successfully |

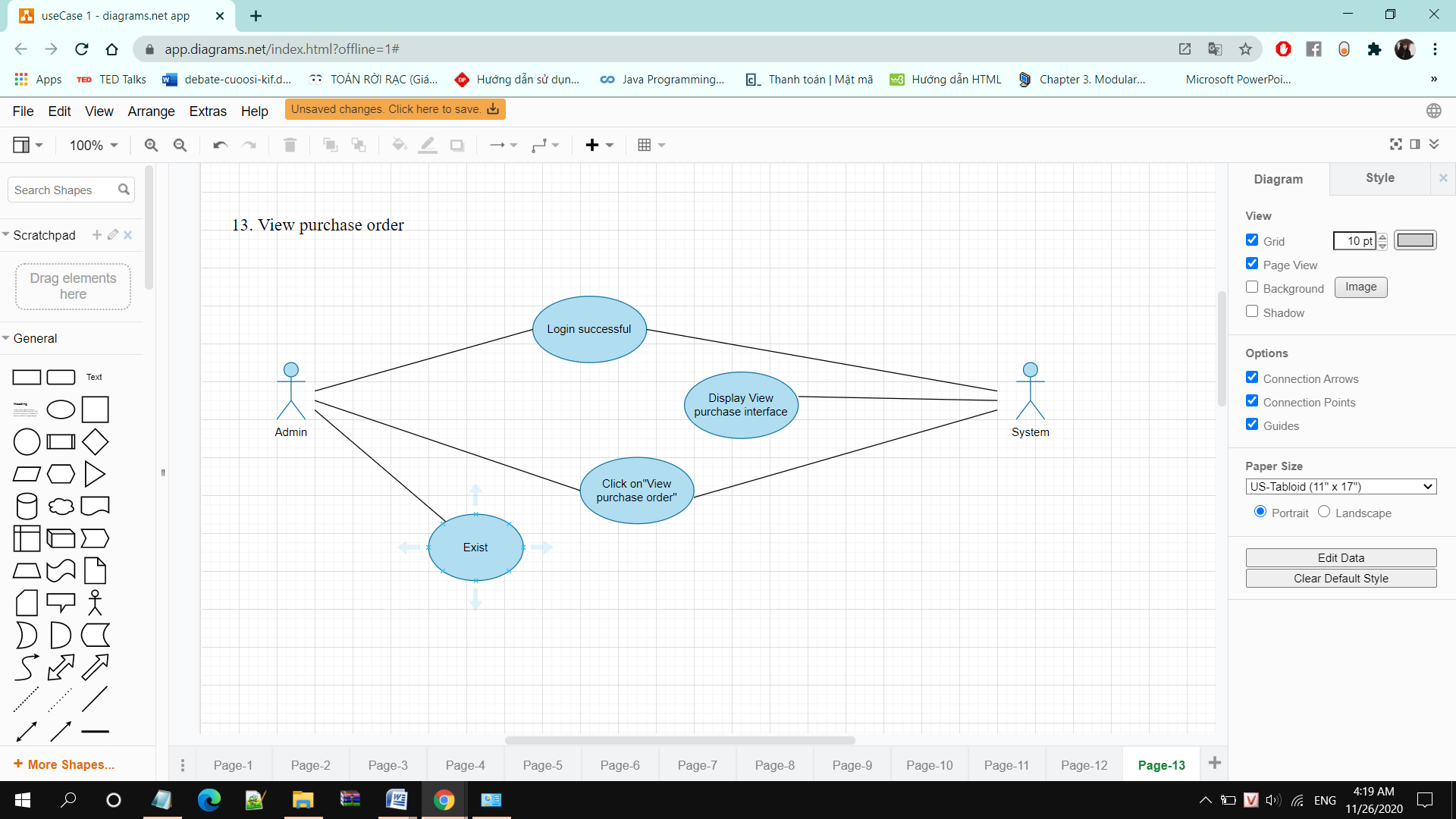
****

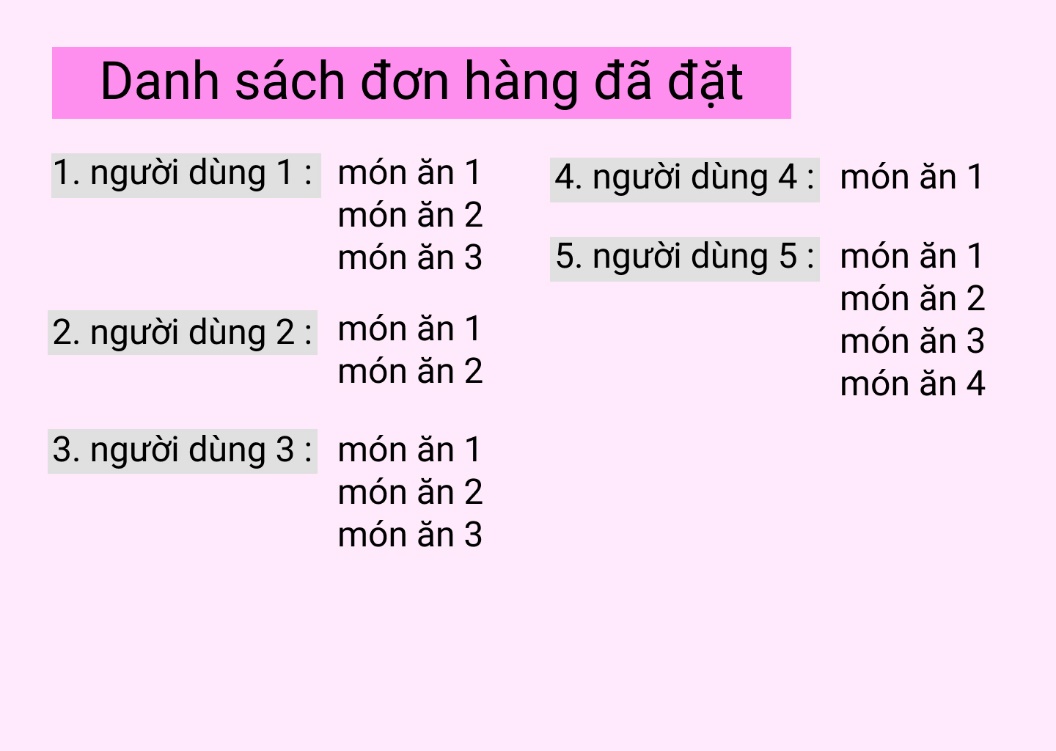
****

Interface Edit / Delete

1. **View purchase orders**

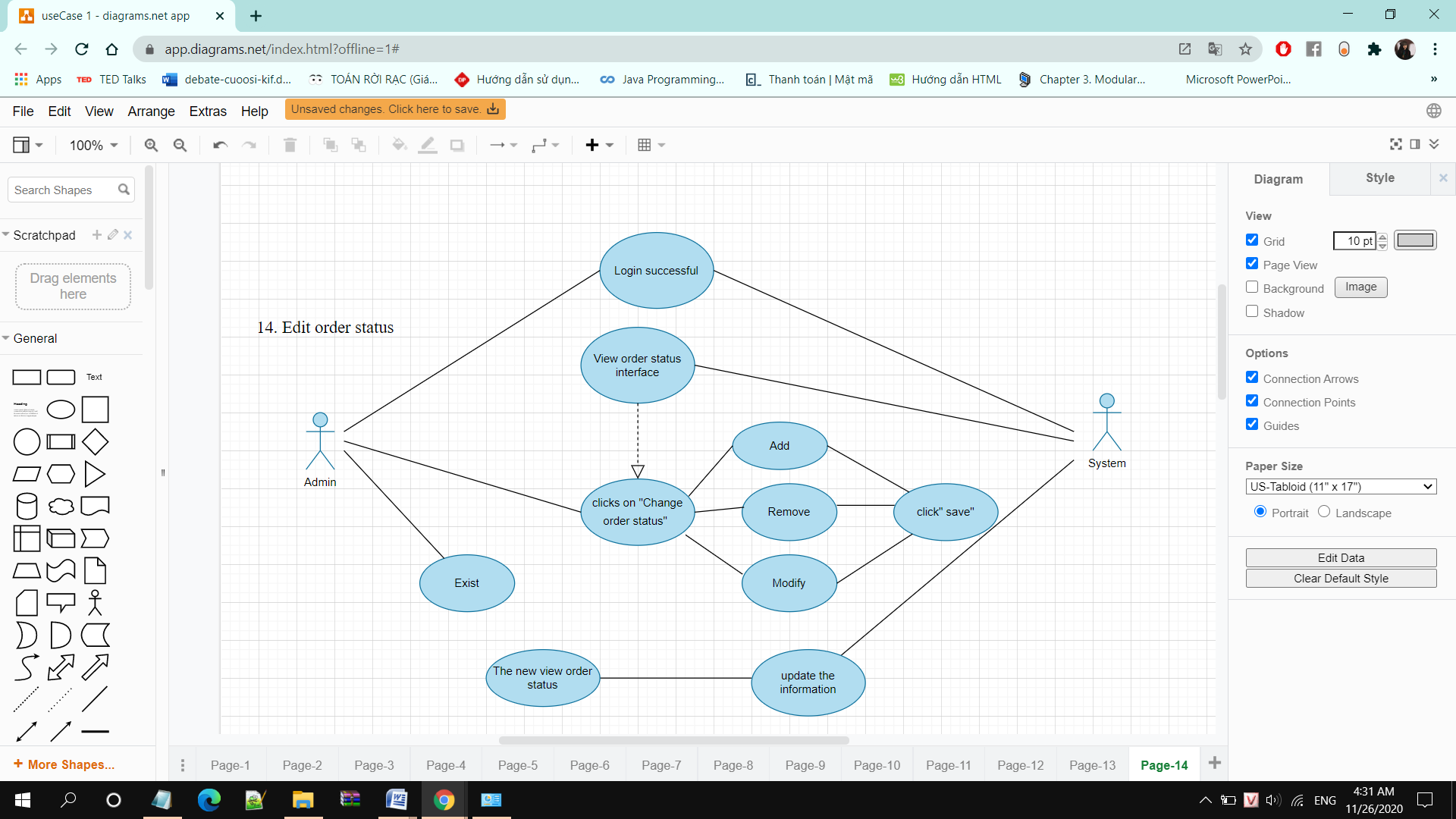
|  |  |
| --- | --- |
| Use Case | View purchase orders |
| Actors | Admin, system |
| Description | Admin view the orders that have been ordered |
| Triggers | Admin click on "View purchase orders" on the interface of the dish list |
| Pre-conditions | Admin has successfully logged in |
| Post-conditions | Admin returns to the main interface  Admin can see the orders |
| Main Event Flow | 1. Admin click on "View purchase orders" on the main interface 2. The system processes and displays orders that have been ordered according to the list of users |

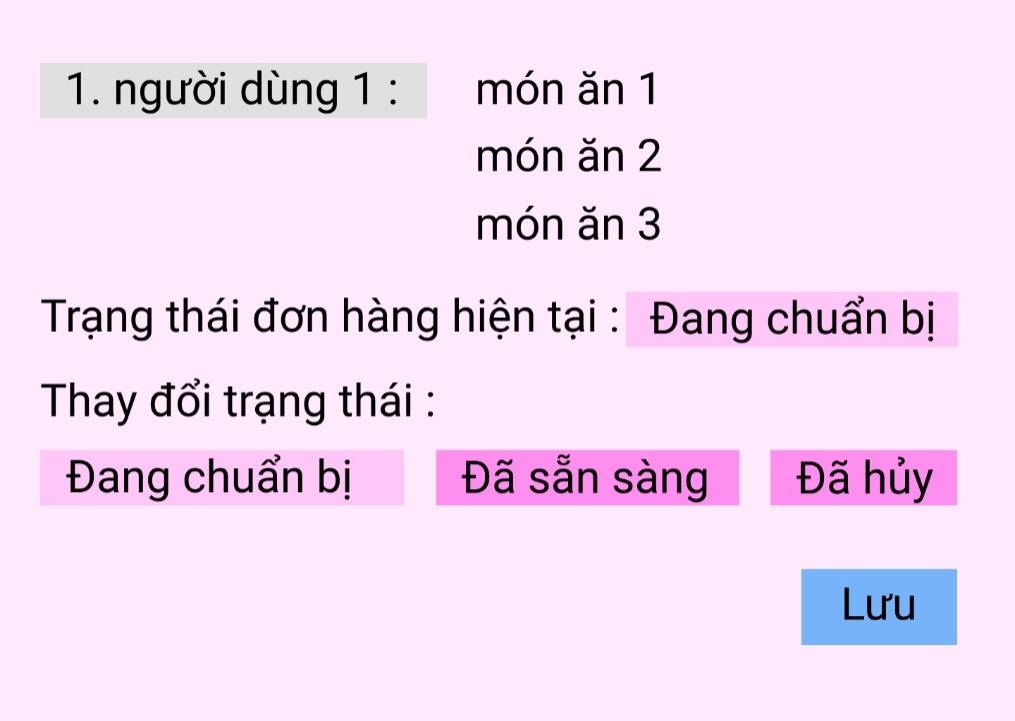
****

****

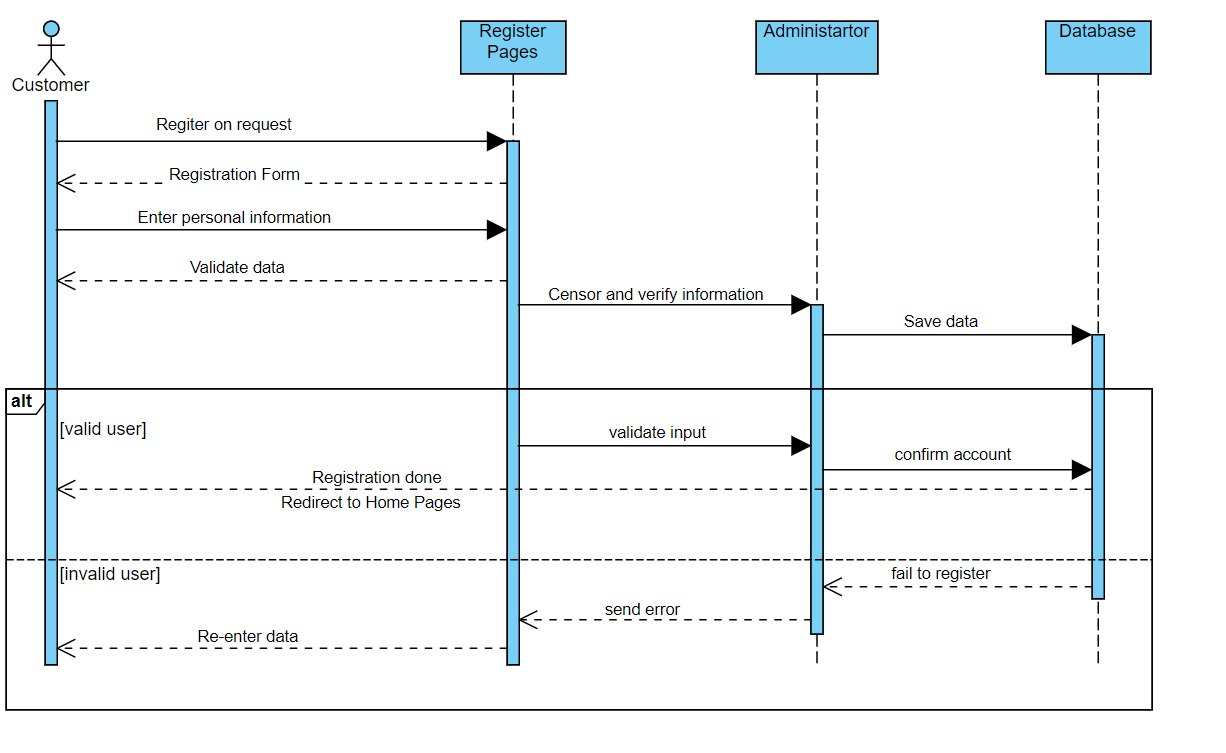
1. **Edit order status**

|  |  |
| --- | --- |
| Use Case | Edit order status |
| Actors | Admin |
| Description | Admin changes the status of orders that have been ordered |
| Triggers | Admin clicks on "Change order status" for an order |
| Pre-conditions | Admin has successfully logged in and is in the Ordered View Interface |
| Post-conditions | Admin returns to the view of ordered interface  Admin can change the status of the order (cooking, ready) |
| Main Event Flow | 1. Admin clicks on an order in View purchase orders 2. The system processes and displays the details of the person who ordered and the dishes 3. Admin select "Change order status" and change (from cooking to ready) 4. The system processes and displays “Edit successfully” |
| Exception Flow | * 1. When a user cancels an order, the system notifies that the order has been canceled by the user to the admin   2. The admin changes the status of the order to canceled |

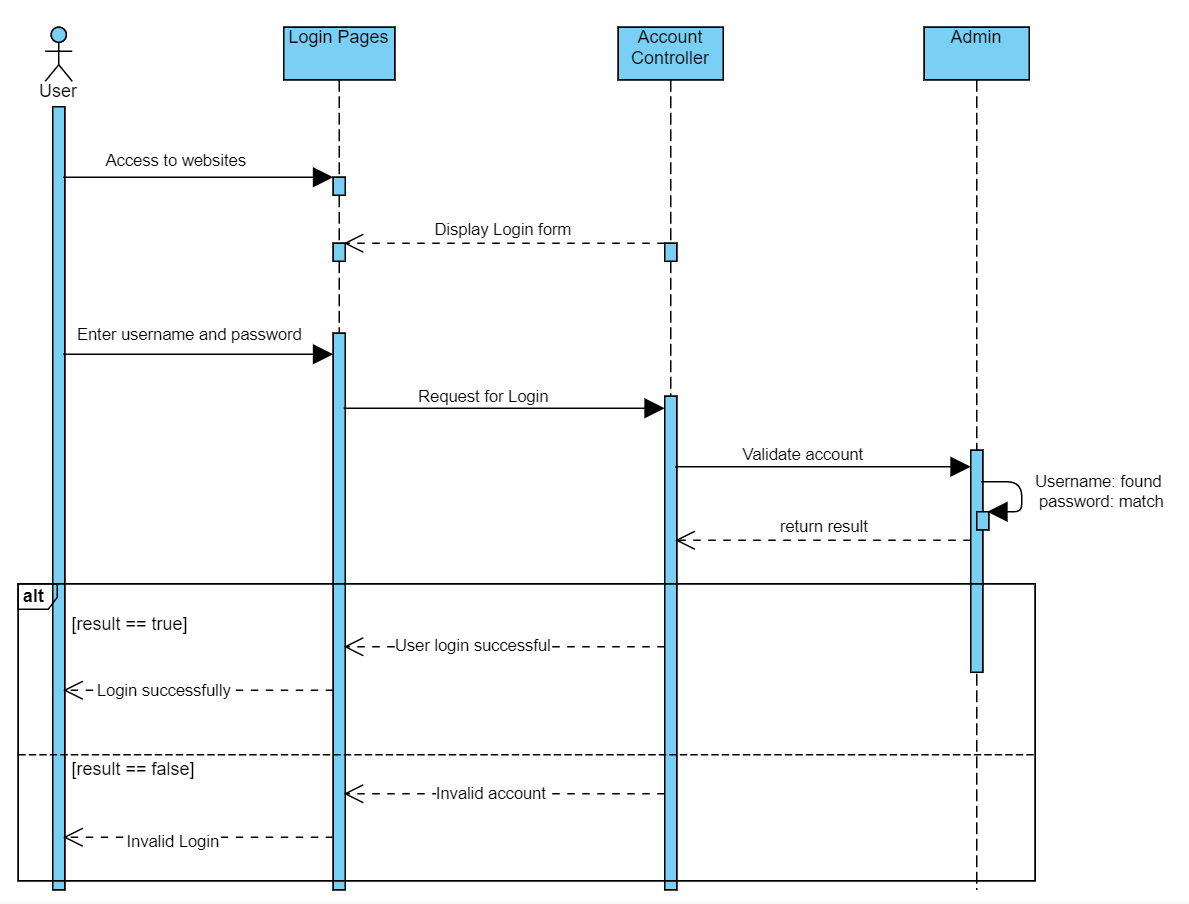
**

**

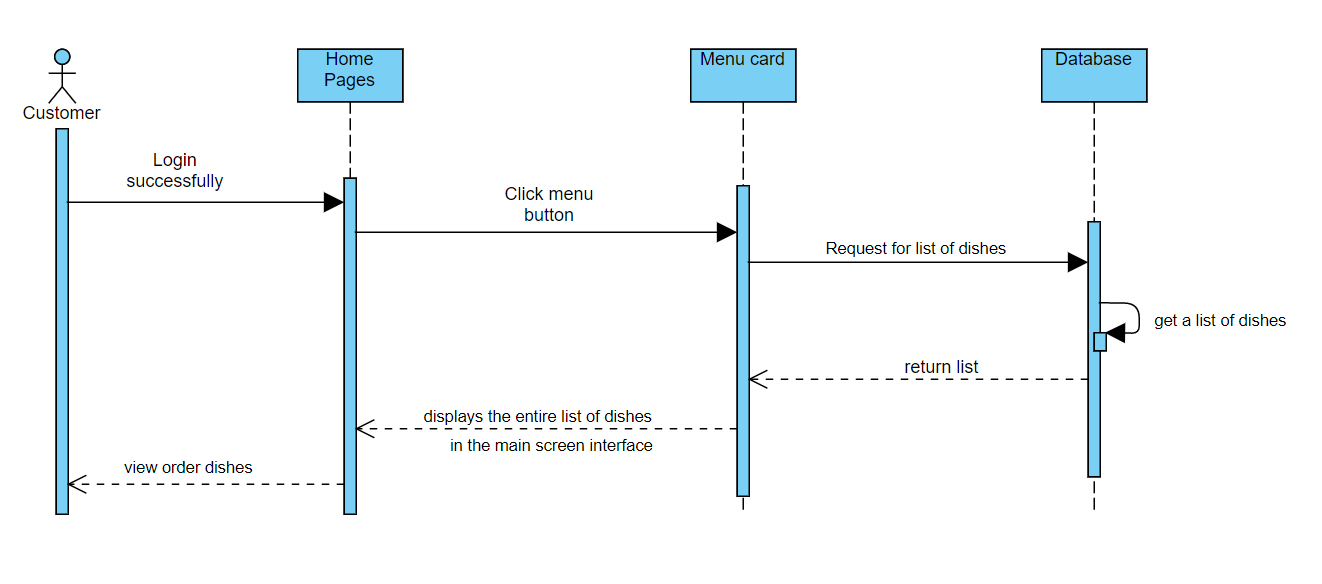
1. *Sequence diagram*
2. **Registration**



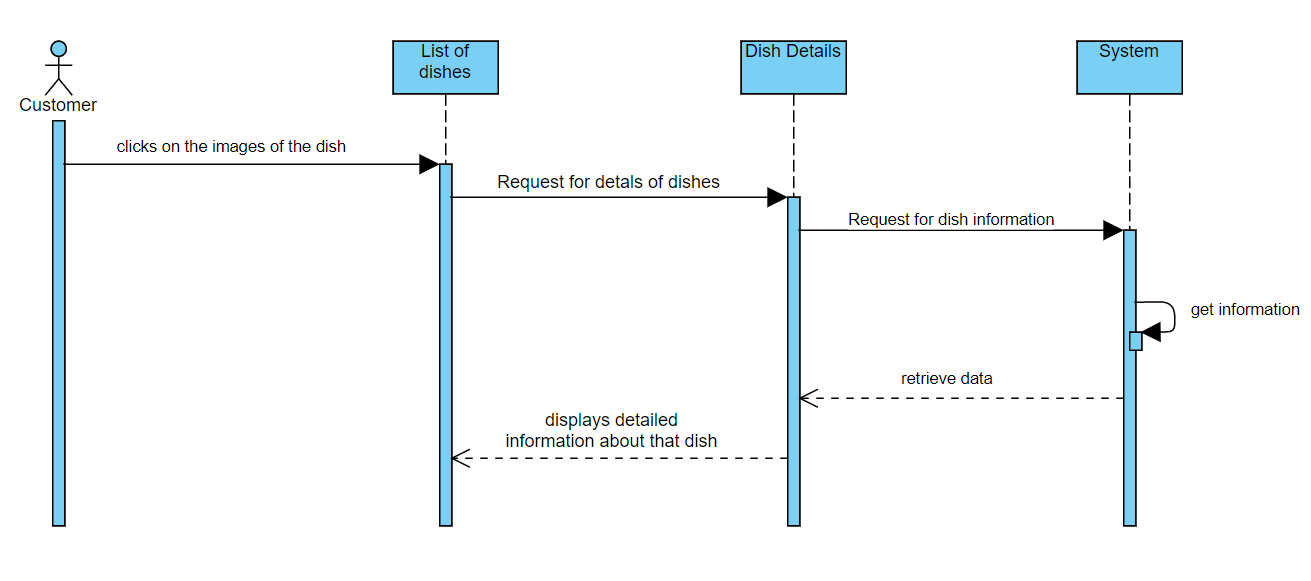
1. **Login**

****

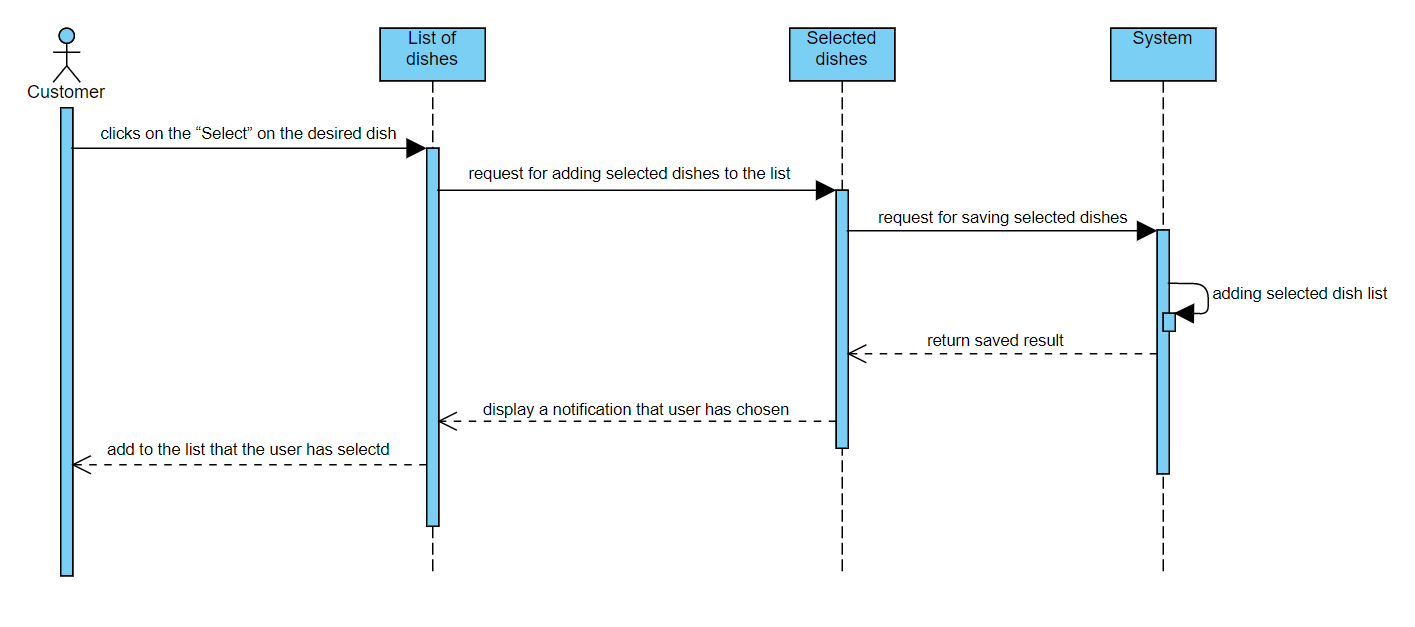
1. **View Menu**

****

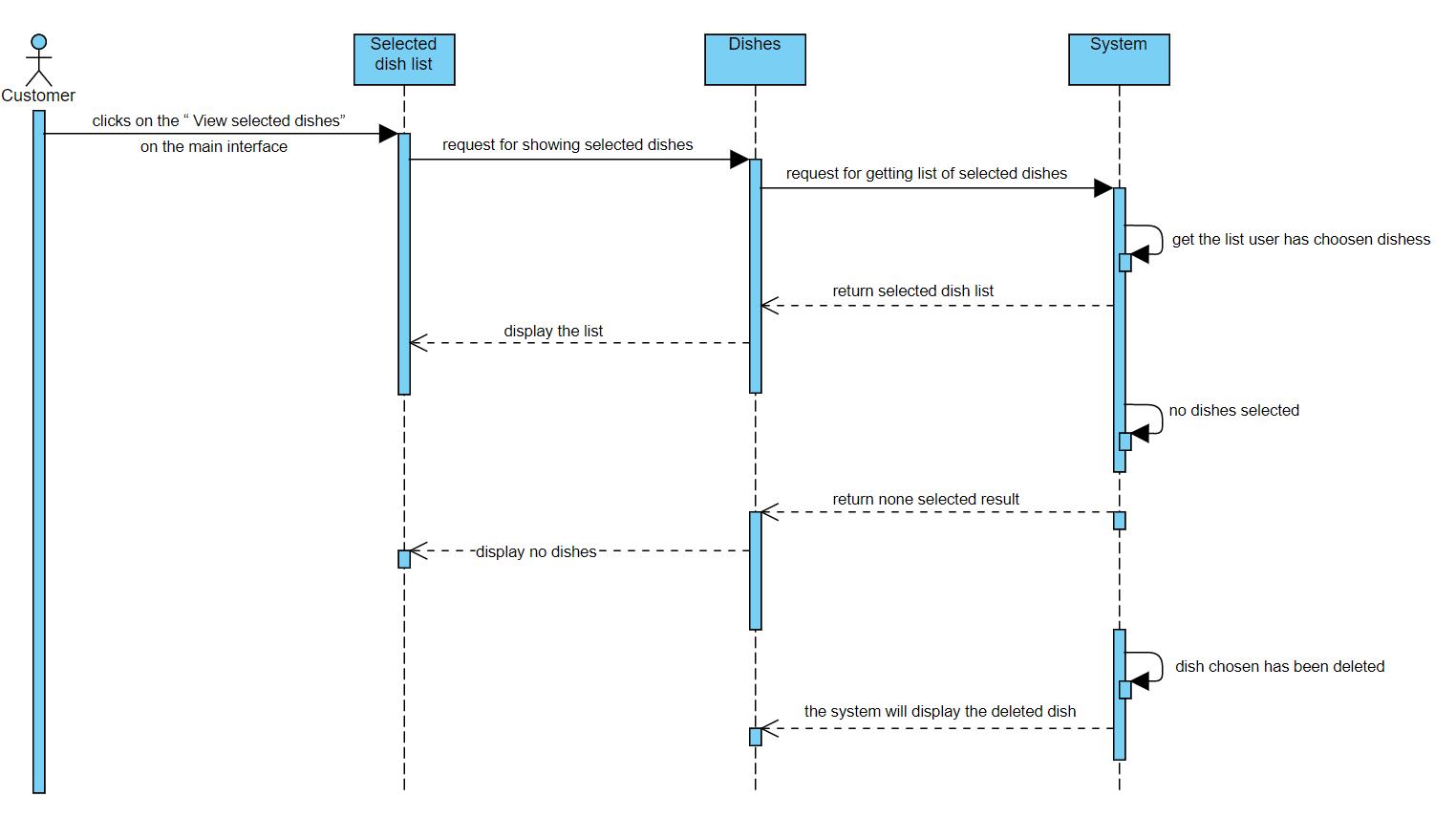
1. **View Dish Details**

****

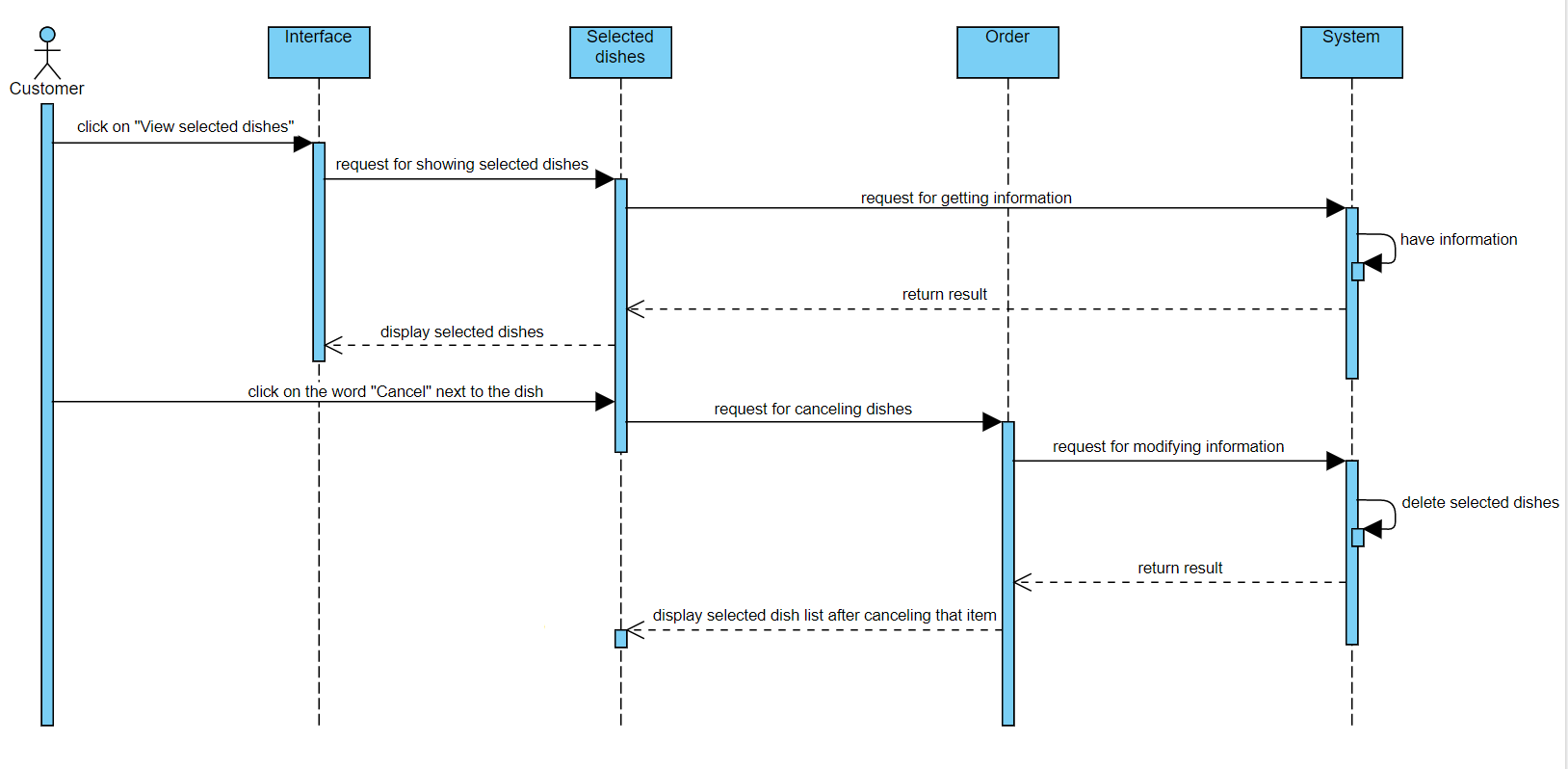
1. **Choose Dishes**

****

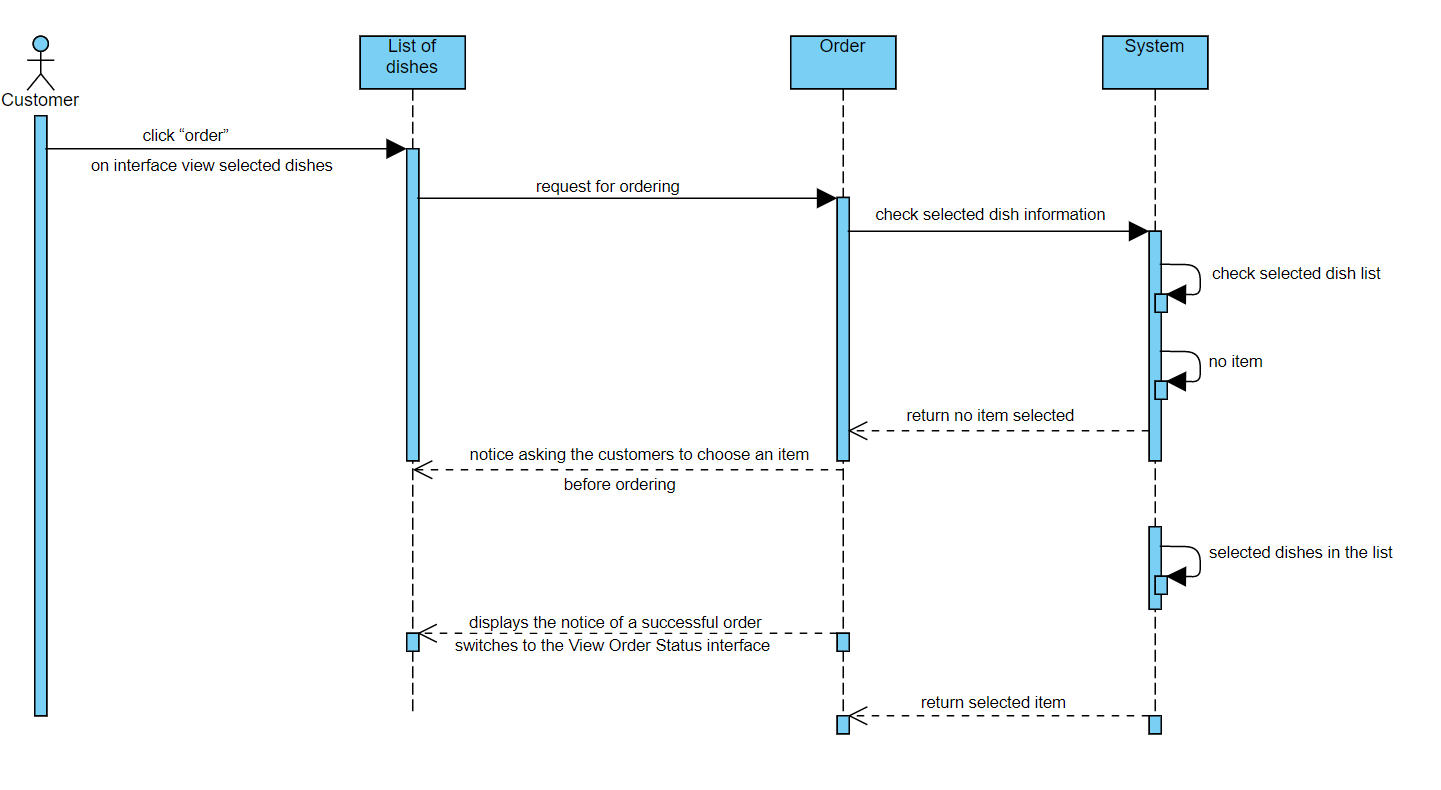
1. **View Selected Dishes**

****

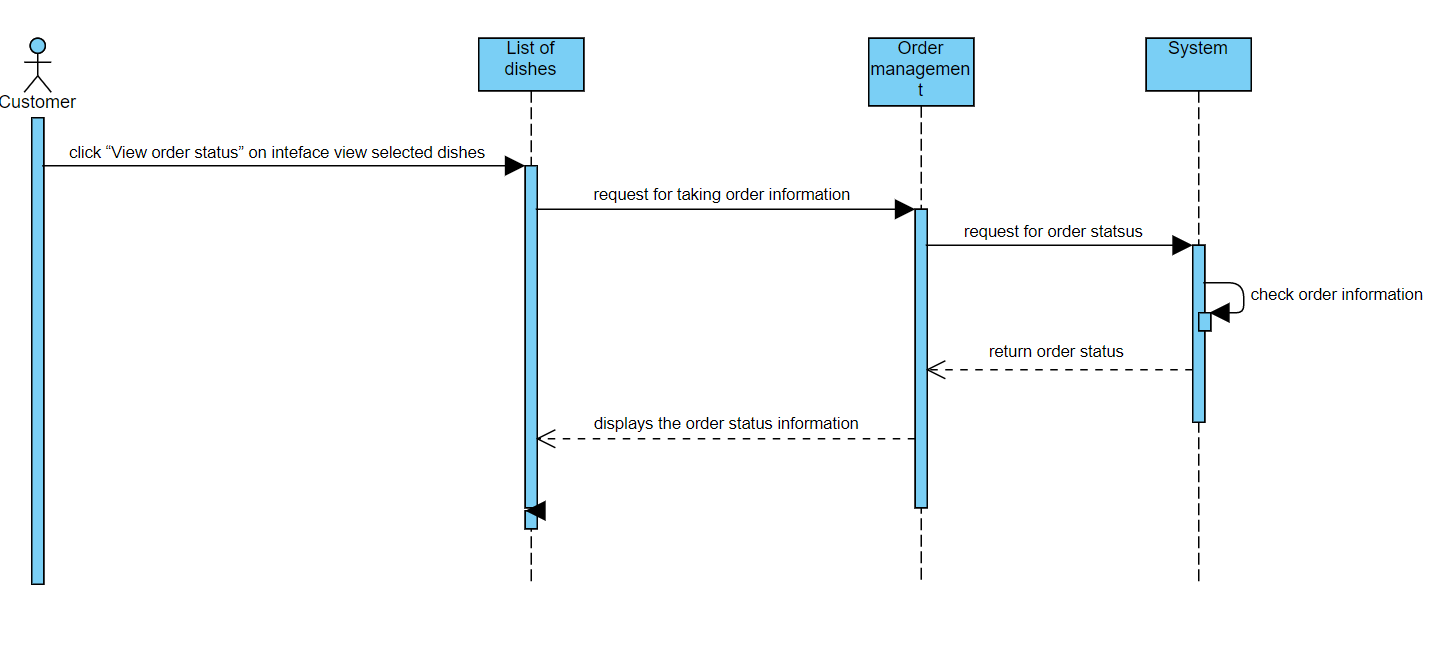
1. **Cancel selected dishes**



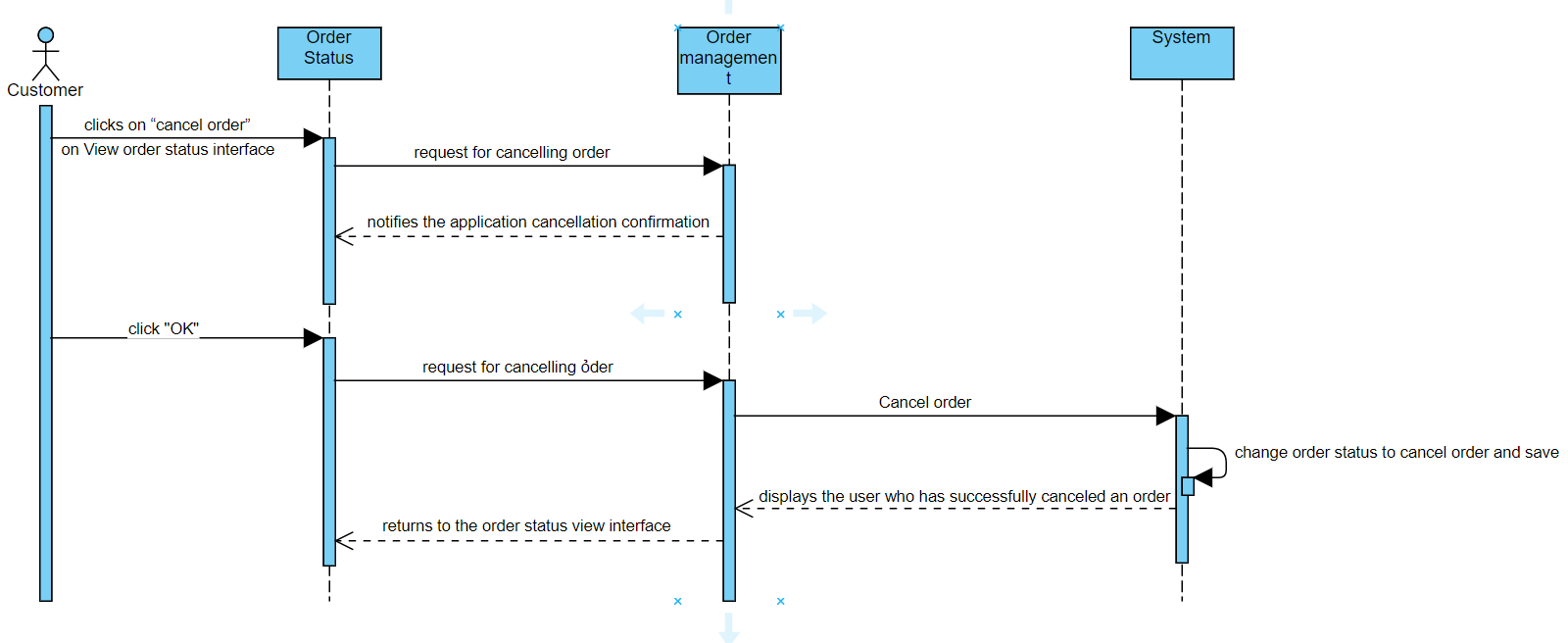
1. **Order**

****

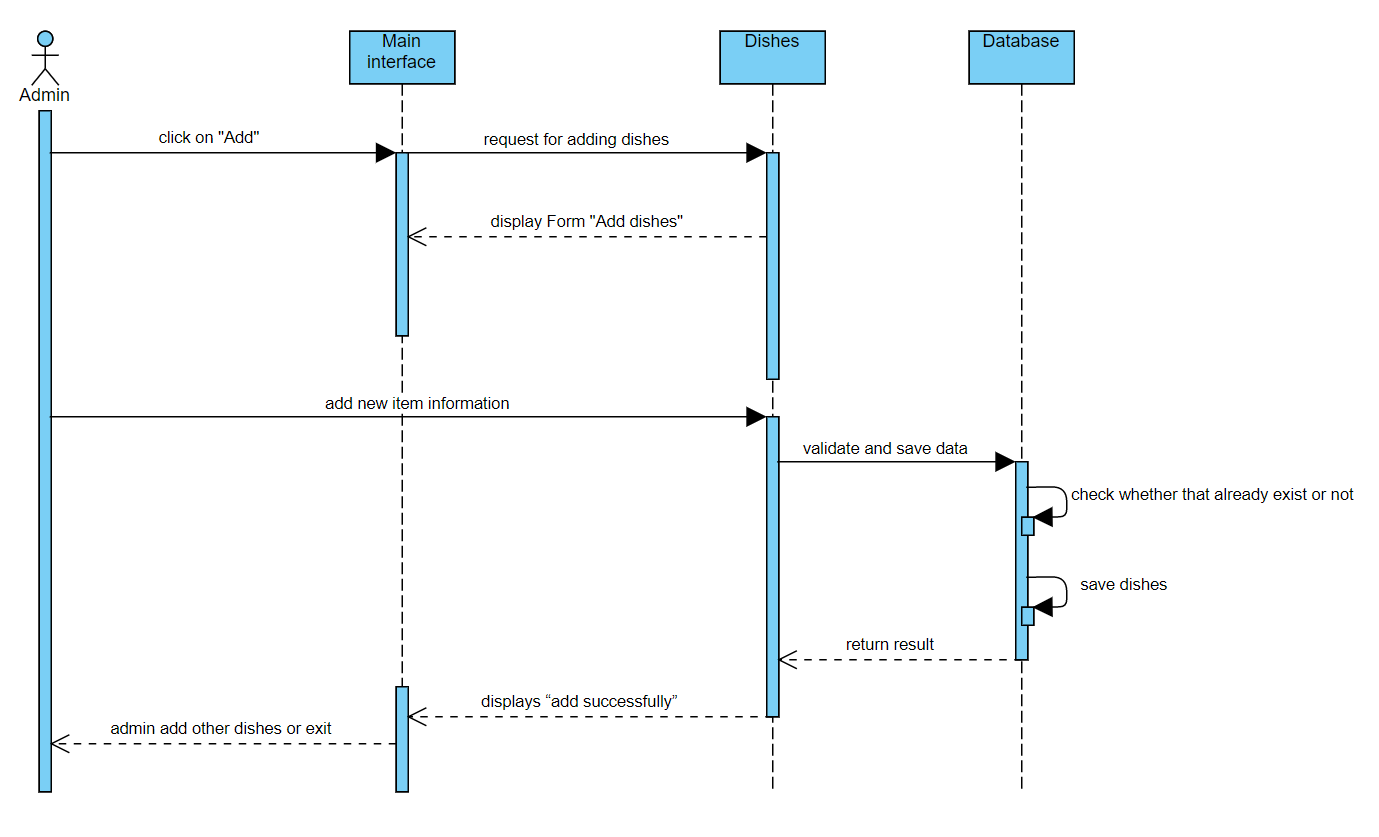
1. **View Order Status**



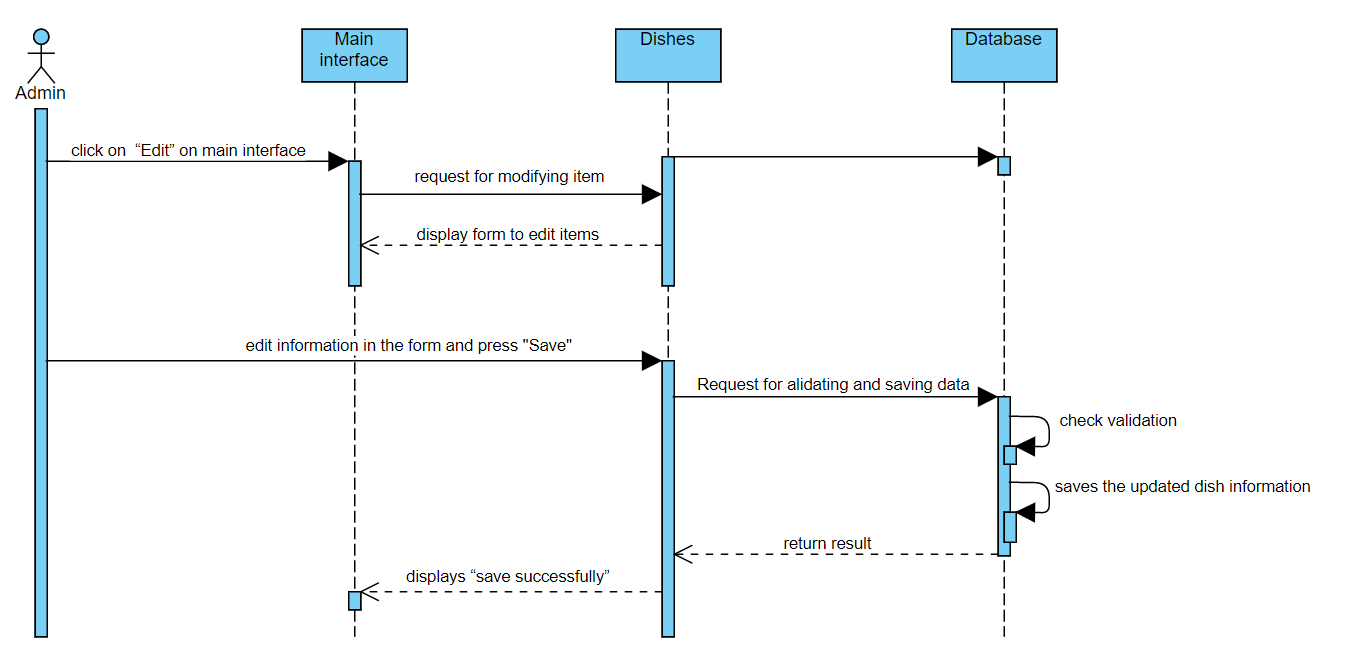
1. **Cancel Orders**

****

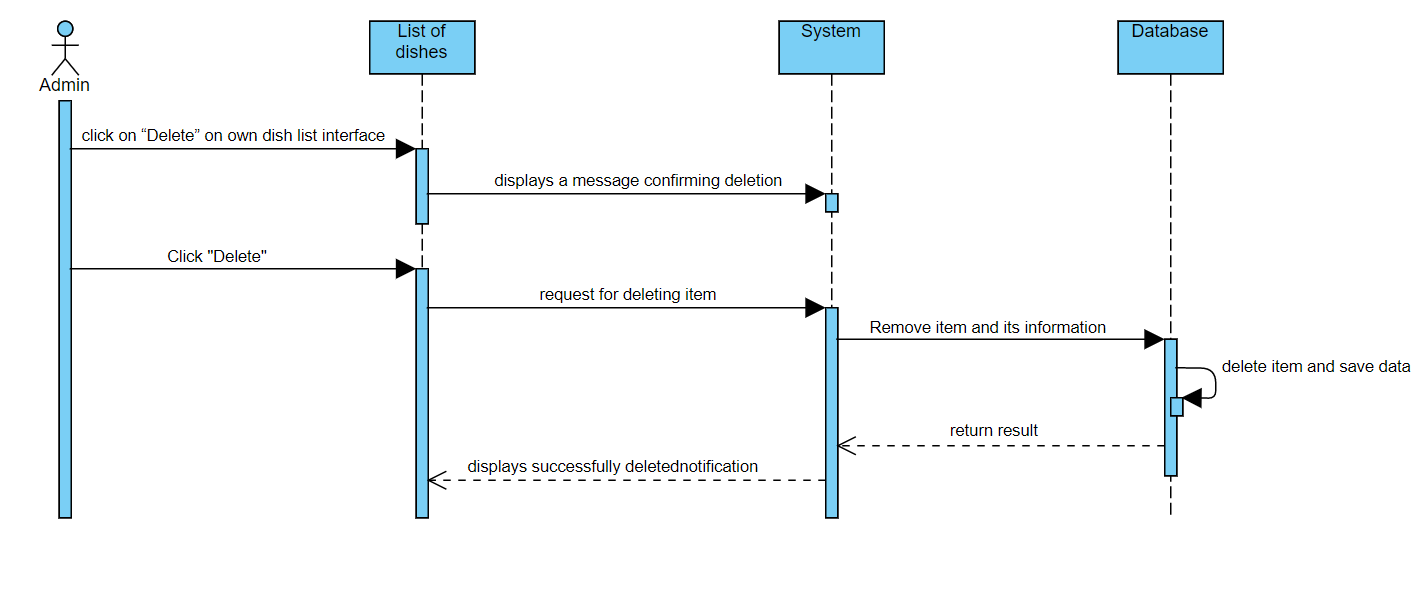
1. **Add dishes**

****

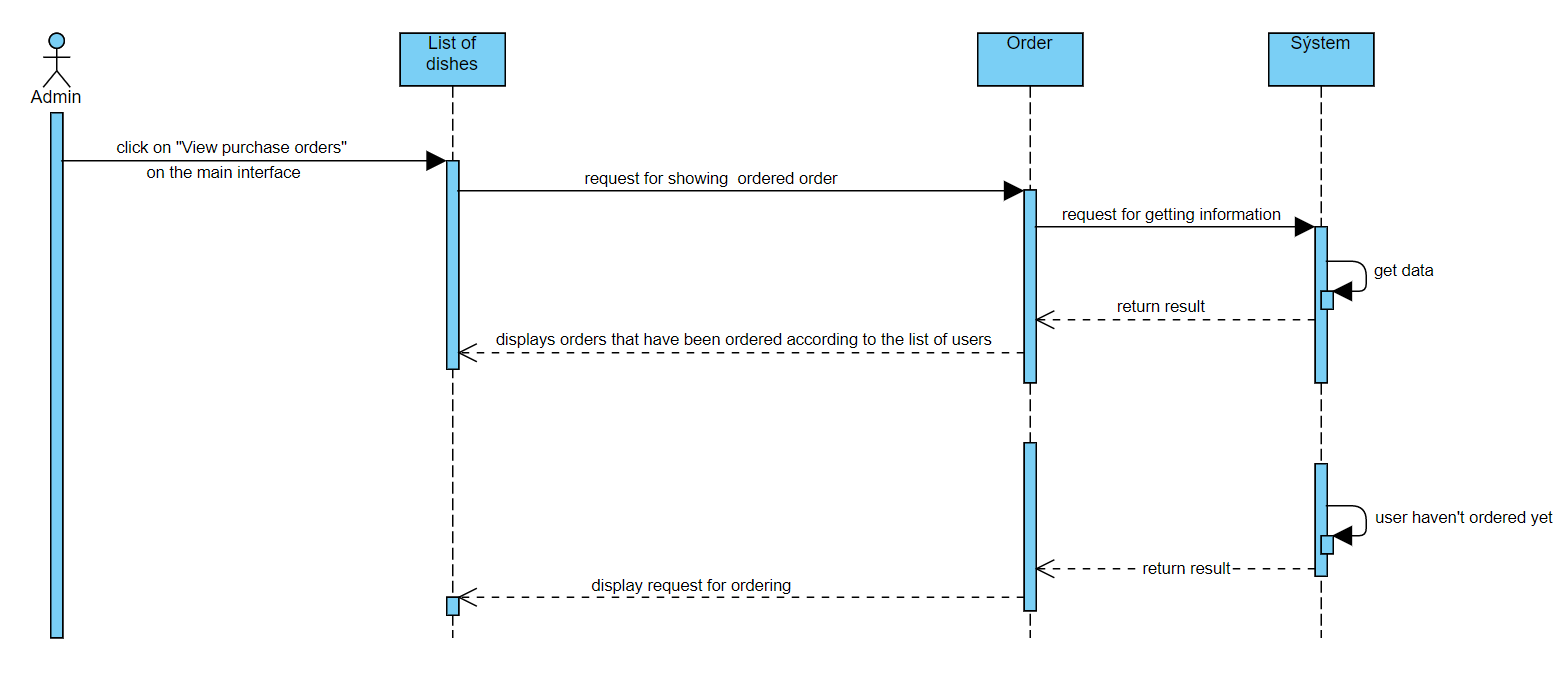
1. **Edit Dishes**

****

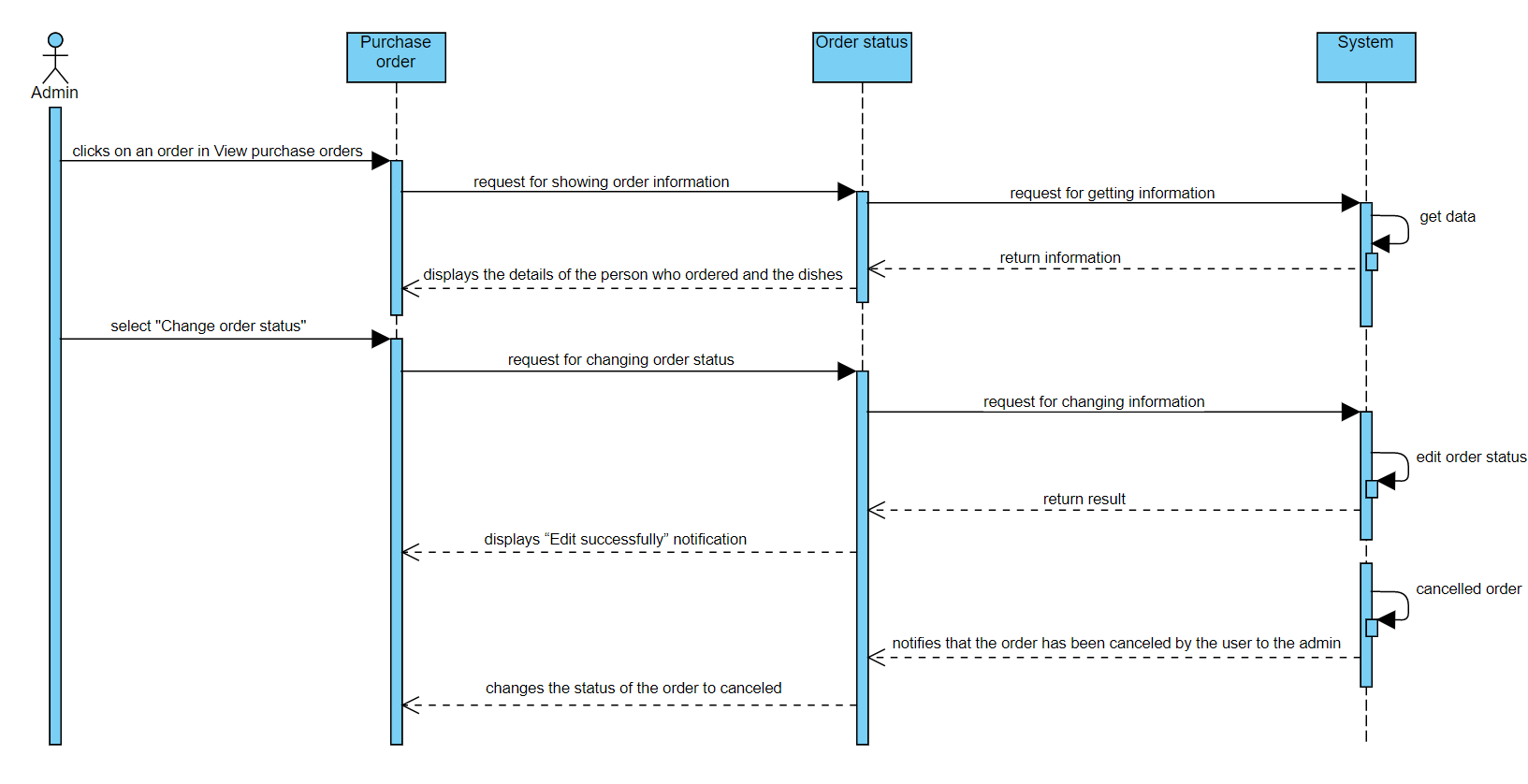
1. **Remove Dishes**

****

1. **View Purchase Orders**

****

1. **Edit Order Status**

****

1. *Biểu đồ lớp ( class diagram )*