

**[Class design]**

**DETAIL DESIGN DOCUMENT**

**Class design**

**Project Code:**

**Document Code: Class Design - v1.0**

**Hồ Chí Minh City, March 30th 2018**

**RECORD OF CHANGE**

\*A - Added M - Modified D – Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Effective Date** | **Changed Items** | **A\* M, D** | **Change Description** | **New Version** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**SIGNATURE PAGE**

**ORIGINATOR:** Nguyễn Tuyết Nga 31/3/2018

Đại học Bách Khoa TP HCM

**REVIEWERS:** Nguyễn Trường Giang 31/3/2018

Đại học Bách Khoa TP HCM

Thẩm Quốc Dũng 31/3/2018

Đại học Bách Khoa TP HCM

Phạm Khánh Huy Hoàng 31/3/2018

Đại học Bách Khoa TP HCM

**APPROVAL:**  TA – Hoàng Hùng

Đại học Bách Khoa TP HCM

**TABLE OF CONTENTS**

1 Introduction 3

1.1 Purpose 3

1.2 Definitions, Acronyms and Abbreviations 3

1.3 Standards and Conventions 3

1.4 References 3

1.5 Overview 3

2 COMMON package and mechanism 3

2.1 Common package 3

2.1.1 Class diagram 3

2.1.2 XXX class 3

2.2 Error, exception handling 3

2.2.1 Class Diagram 3

2.2.2 Usage mechanism 3

2.3 Log, trace and debug 3

2.4 Performance optimizing mechanism 3

2.5 Multilingual processing 3

3 Diagrams 3

4 Packages 3

4.1 XXX Package 3

4.1.1 Class diagram 3

4.1.2 External interface 3

4.1.3 XXX class 3

4.2 Implementation 3

5 Other considerations 3

6 Appendix 3

# 

# Introduction

## Purpose

Class Design can make reader easy to understand the system general and know about how can the procedure run

## Definitions, Acronyms and Abbreviations

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Standards and Conventions

## References

## Overview

* We have salesman class to save the data of identify of them.
* Change class to save the dateChange and actionType when salesman check the product
* Account class to create and check the account
* Approve class to save the dateApprove when salesman check the orders
* Customer class to save the data of identify of customer fill in the orders
* Orders class to save the data of the orders when customer order the product
* Product class to save the date of the product
* Cart class to save the id of cart when the customer put the product to the cart
* Manufacture class to save the date of manufacture, where the store buy product
* Has(Cart -> Orders) to save the transportation method, which client want to delivery
* Has(Product -> Cart) to count the product in cart

# COMMON package and mechanism

## Common package

### Class diagram

|  |  |  |
| --- | --- | --- |
| **No** | **Class Name** | **Description** |
| 01 | Salesman | This class is use for salesman, who sale the product to save the data of identify |
| 02 | Customer | This class is use for customer, who buy the product, to save the data of identify |
| 03 | Account | This class is use for create and check the account |
| 04 | Approve | This class is use for save the dateApprove when salesman check the orders |
| 05 | Orders | This class is use for save the data of the orders when customer order the product |
| 06 | Change | This class is use for save the dateChange and actionType when salesman check the product |
| 07 | Product | This class is use for save the date of the product |
| 08 | Cart | This class is use for save the id of cart when the customer put the product to the cart |
| 09 | Manufacture | This class is use for save the date of manufacture, where the store buy product |
| 10 | Has(Cart -> Orders) | This class is use for save the transportation method, which client want to delivery |
| 11 | Has(Product -> Cart) | This class is use for |

### Salesman class

Salesman class to save the data of identify of them.

* **Attributes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Attribute** | **Type** | **Default** | **Note** |
| 01 | id | int | None | Private |
| 02 | province | string | None | Public/ Static |
| 03 | street | string | None | Public/ Static |
| 04 | district | string | None | Public/ Static |
| 05 | phoneNum | string | None | Public/ Static |
| 06 | salesNum | int | None | Public/ Static |
| 07 | firstName | string | None | Public/ Static |
| 08 | lastName | string | None | Public/ Static |
| 09 | birthDate | datetime | None | Public/ Static |

* **Methods**

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | findProduct(key :string, product : Produtc) | This method is used for salesman search product in the system |
| 02 | viewInfo(key: string, product : Product) : void | This method is used for salesman see info product in the system. |
| 03 | updateInfo(key: string,product : Product) : void | Used for salesman update info of product in the system. |
| 04 | addNewProduct(product : Product) : void | Used for salesman add a new product in the system. |
| 05 | deleteProducts(product : Product) : boolean | Used for salesman remove a product in the system. |
| 06 | viewReport() : void | This method is used to view report. |
| 07 | login(account : Account) | Used for salesman login. |
| 08 | logout(account : Account) | Used for salesman logout. |
| 09 | manageOrders(orders) : int | This method used to manage the Orders. |

### Account class

Account class to create and check the account

* **Attributes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Attribute** | **Type** | **Default** | **Note** |
| 01 | id | int |  | Public/ Static |
| 02 | userName | string |  | Public/ Static |
| 03 | passWord | string |  | Public/ Static |
| 04 | active | boolean |  | Public/ Static |
| 05 | type | string |  | Public/ Static |

* **Methods**

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 02 | createAccount(username : string,password: string) | This method used to create account. |
| 03 | checkAccount(sales: Salesmam, id int) : boolean | This method used to check account |

### Cart class

Cart class to save the id of cart when the customer put the product to the cart

* **Attributes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Attribute** | **Type** | **Default** | **Note** |
| 01 | id | int |  | Public/ Static |
| 02 | totalPrice | int |  | Public/ Static |

* **Methods**

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | totalPrice(price : int,product :int) : int | Used to calculate the total price of the products. |
| 02 |  |  |

### Approve class

Approve class to save the dateApprove when salesman check the orders

* **Attributes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Attribute** | **Type** | **Default** | **Note** |
| 01 | dateApprove | datetime |  | Public/ Static |

* **Methods**

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 |  | <brief description of method. can be one sentence tell what the method does> |

### Change class

Change class to save the dateChange and actionType when salesman check the product

* **Attributes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Attribute** | **Type** | **Default** | **Note** |
| 01 | actionType | char |  | Public/ Static |
| 02 | dateChange | datetime |  | Public/ Static |

* **Methods**

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 |  | <brief description of method. can be one sentence tell what the method does> |

### Product

Product class to save the date of the product

* **Attributes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Attribute** | **Type** | **Default** | **Note** |
| 01 | id | int |  | Public/ Static |
| 02 | name | char |  | Public/ Static |
| 03 | description | char |  | Public/ Static |
| 04 | price | char |  | Public/ Static |
| 05 | image | int |  | Public/ Static |
| 06 | status | char |  | Public/ Static |
| 07 | quantityInStock | int |  | Public/ Static |

* **Methods**

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | GetProductInfo(userType:int, keyWord:char) | This method is used to get product infomation |

### Manufacture

Manufacture class to save the date of manufacture, where the store buy product

* **Attributes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Attribute** | **Type** | **Default** | **Note** |
| 01 | id | int |  | Public/ Static |
| 02 | manufactureName | char |  | Public/ Static |

* **Methods**

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 |  | <brief description of method. can be one sentence tell what the method does> |

### Customer

Customer class to save the data of identify of customer fill in the orders

* **Attributes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Attribute** | **Type** | **Default** | **Note** |
| 01 | id | int |  | Public/ Static |
| 02 | lastName | char |  | Public/ Static |
| 03 | firstName | char |  | Public/ Static |
| 04 | birthdate | date |  | Public/ Static |
| 05 | phoneNumber | int |  | Public/ Static |
| 06 | sex | char |  | Public/ Static |
| 07 | address | string |  | Public/ Static |
| 08 | street | string |  | Public/ Static |
| 09 | district | char |  | Public/ Static |
| 10 | city | char |  | Public/ Static |

* **Methods**

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | AddProductIntoCart(productId: char, cartId:char) | This method is used for customer to add new product into cart |
| 02 | Order(cartId:char) | This method is used for customer to order |

### Order class

Orders class to save the data of the orders when customer order the product

* **Attributes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Attribute** | **Type** | **Default** | **Note** |
| 01 | id | int |  | Public/ Static |
| 02 | ordersNumber | int |  | Public/ Static |
| 03 | ordersDate | date |  | Public/ Static |
| 04 | status | char |  | Public/ Static |

* **Methods**

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| 01 | GetOrders(orderId:int) | This method is used for getting the orders informations |

### Has (Product – Cart)

Has(Product -> Cart) to count the product in cart

* **Attributes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Attribute** | **Type** | **Default** | **Note** |
| 01 | quantityProduct | int |  | Public/ Static |

### Has (Cart – Orders)

* Has(Cart -> Orders) to save the transportation method, which client want to delivery
* **Attributes**

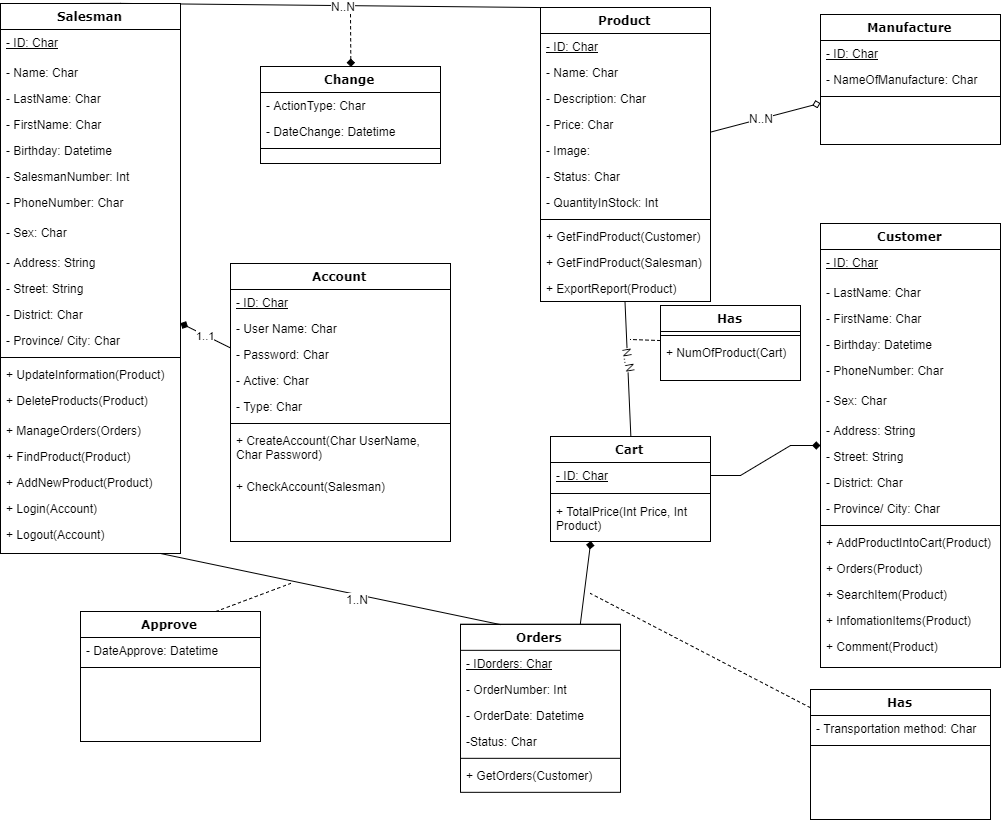
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Attribute** | **Type** | **Default** | **Note** |
| 01 | transportationMethod | char |  | Public/ Static |

* **Implementation**

*<How to implement the method, it can be in pseudo code or activity diagram or just words>*

## Error, exception handling

### Class Diagram



### Usage mechanism

## Log, trace and debug

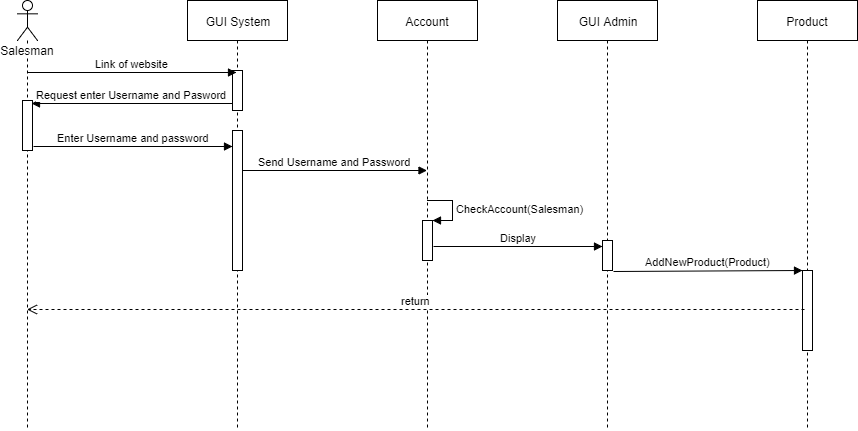
## Performance optimizing mechanism

## Multilingual processing

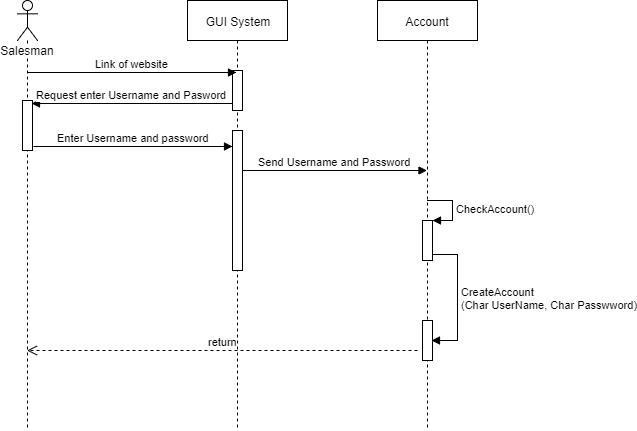
# Diagrams

## Admin

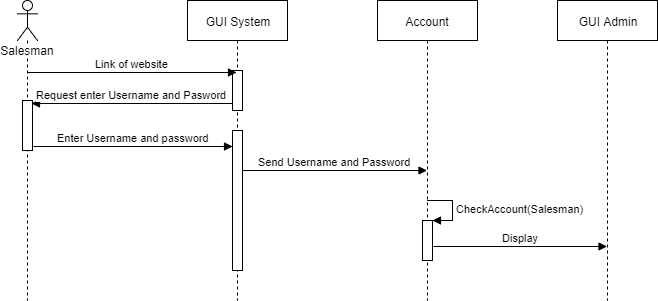
* Add new product



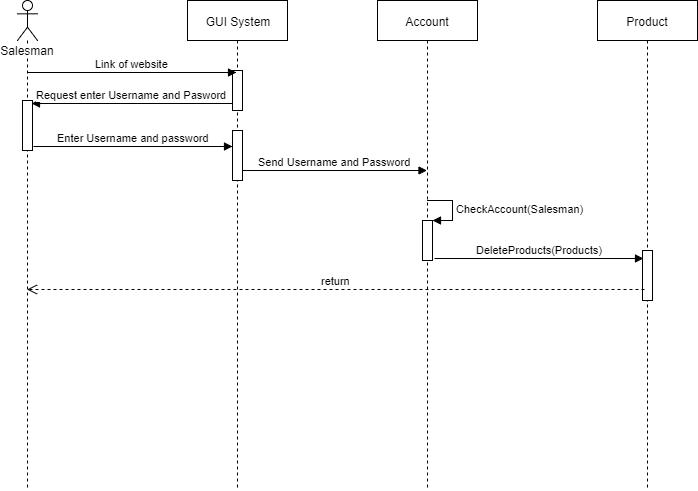
* Create account



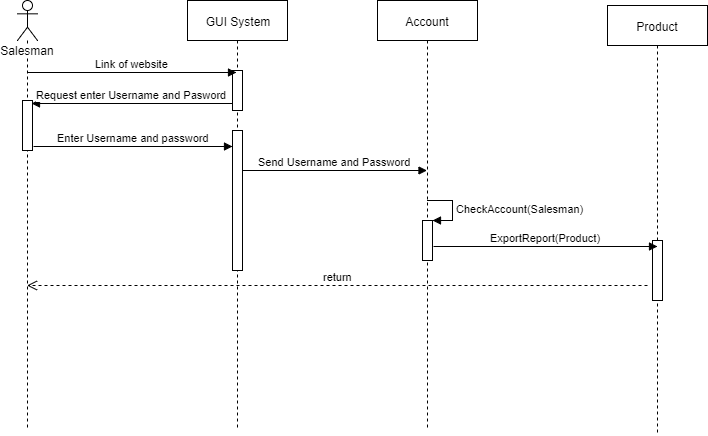
* Login



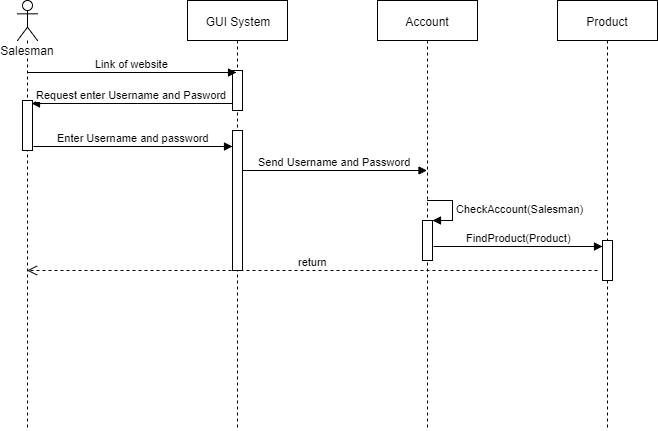
* Delete product



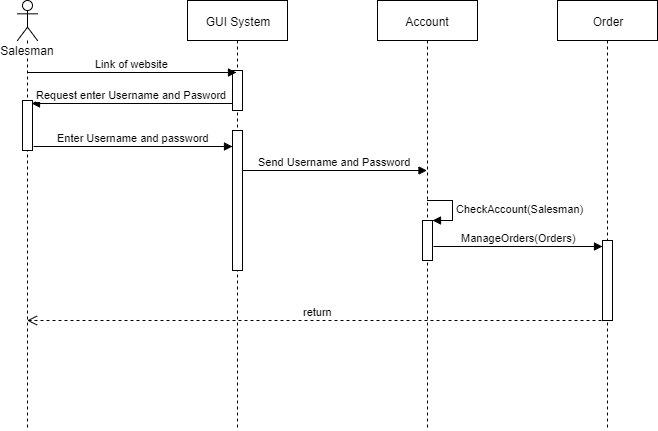
* Export Report



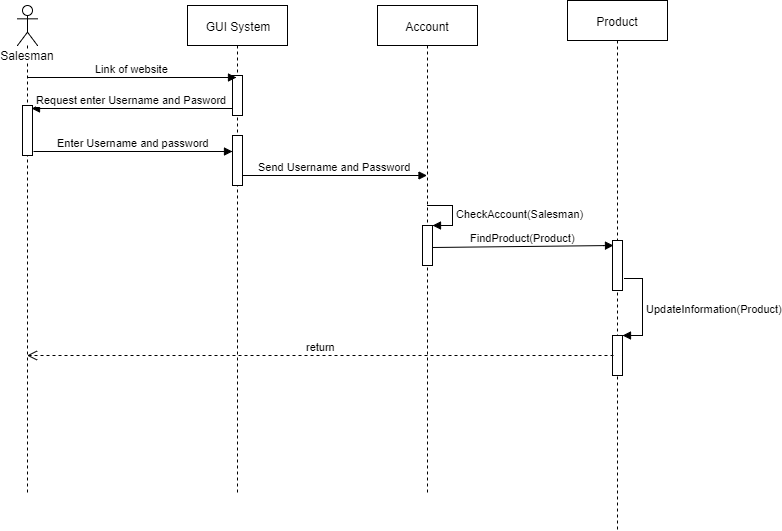
* Find Product



* Manage Orders

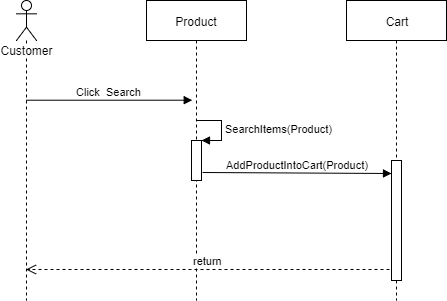


* Update Info

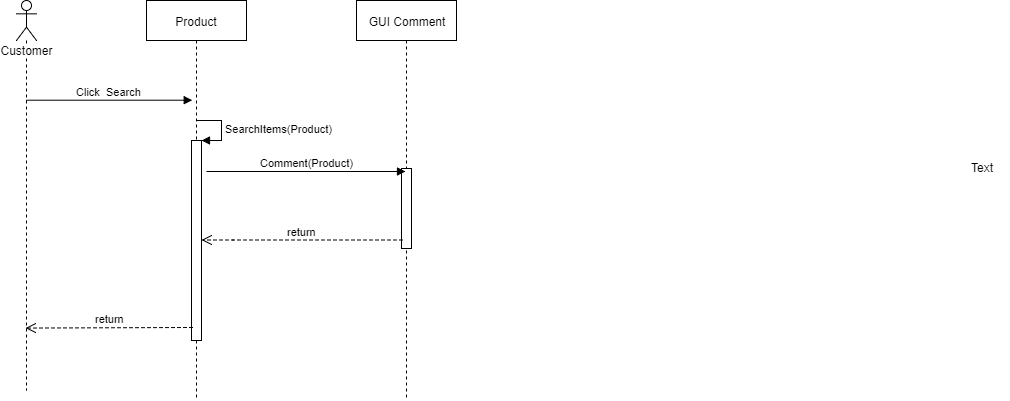


## Customer

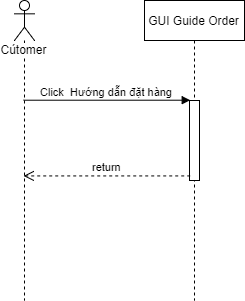
* Add cart



* Comment



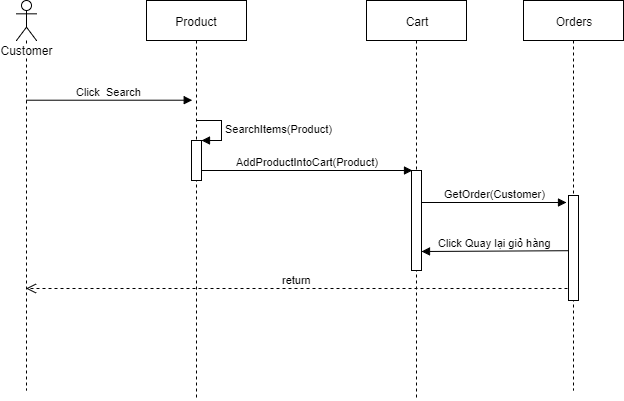
* Guide Order



* Info



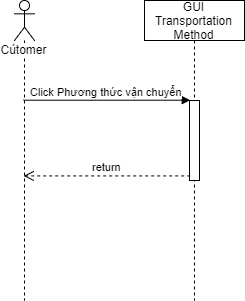
* Order



* Search Items



* Transportation method



# Packages

## XXX Package

### Class diagram

### External interface

### XXX class

## Implementation

# Other considerations

# Appendix