BEHAVIORAL
SCHEMA
An order is composed of **order lines** which represent a quantity of a product. The information about order lines is:

- **Name**: A copy of the product's name. *Derived Attribute*.
- **Model**: The current status of the order. *Derived Attribute*.
- **Total**: The total price of the order, taking into account the shipping costs and the taxes.

Likewise, order lines can have **order line attributes**, which represents the product attributes chosen for an order line product.

Order line attributes which correspond to downloadable product attributes have a specific attribute called *downloadCount* that keeps track of how many times the product has been downloaded by the customer.

## Example

The Vienna Mozart Orchestra plays concerts in Vienna in the most famous concert halls, including the Golden Hall in the Musikverein, a well-known building where the most popular New Year's Eve concert takes place.

Imagine that we would like to implement an **online** ticket shop based on the *osCommerce* solution. The following is a possible instantiation of an order:
9 BEHAVIORAL SCHEMA

9.1 INTRODUCTION

In the following sections we develop the behavioral schema of the osCommerce system.

The main purpose of the osCommerce behavioral schema is to specify the valid changes in the domain state, as well as the actions that the osCommerce system can perform.

Firstly, we show an overview use cases diagram which gives a general view of the most important functionalities of the system. Include relationships are not shown for purposes of clarity.

Afterwards, we specify each use case textually, as an interaction of activities between the primary actors and the system.

Use cases specification contains the mapping of use cases with the most important events of the system using textual references. Events of the system are presented alphabetically, in order to improve search. Each event is represented by an UML diagram, and its effect and constraints are specified using OCL operations.

9.2 USE CASE DIAGRAM
9.3 USE CASE SPECIFICATION

Use case

Change store data

**Primary Actor:** System administrator

**Precondition:** None.

**Trigger:** The system administrator wants to change the initial values of the store data.

**Main Success Scenario:**

1. The system displays the current values of the store data.
2. The system administrator provides a new value for one of the store attributes:
   
   - NameChange
   - OwnerChange
   - EMailAddressChange
   - EMailFromChange
   - ExpectedSortOrderChange
   - ExpectedSortFieldChange
   - SendExtraOrderChange
   - DisplayCartAfterAddingProductChange
   - AllowGuestToTellAFriendChange
   - DefaultSearchOperatorChange
   - StoreAddressAndPhoneChange
   - TaxDecimalPlacesChange
   - DisplayPricesWithTaxChange
   - SwitchToDefaultLanguageCurrencyChange
   - CountryChange
   - ZoneChange

3. The system validates that the value is correct.
4. The system saves the new value.
5. The system displays the new values of the store data.
   
The system administrator repeats steps 2-5 until he is done.
Assign minimum values

**Primary Actor:** System administrator

**Precondition:** None.

**Trigger:** The system administrator wants to change the minimum values of some attributes.

**Main Success Scenario:**

1. The system displays the current minimum values.
2. The system administrator provides a new value for one of the minimum values:
   - [→ FirstNameMinimumChange]
   - [→ LastNameMinimumChange]
   - [→ DateOfBirthMinimumChange]
   - [→ EmailAddressMinimumChange]
   - [→ StreetAddressMinimumChange]
   - [→ CompanyNameMinimumChange]
   - [→ PostCodeMinimumChange]
   - [→ CityMinimumChange]
   - [→ StateMinimumChange]
   - [→ TelephoneMinimumChange]
   - [→ PasswordMinimumChange]
   - [→ CreditCardOwnerNameMinimumChange]
   - [→ CreditCardNumberMinimumChange]
   - [→ ReviewTextMinimumChange]
3. The system validates that the value is correct.
4. The system saves the new value.
5. The system displays the new current minimum values.
   - The system administrator repeats steps 2-5 until he is done.

Assign maximum values

**Primary Actor:** System administrator

**Precondition:** None.

**Trigger:** The system administrator wants to change the maximum number of address book entries permitted for each customer.

**Main Success Scenario:**
1. The system displays the current maximum number of address book entries for each customer.
2. The system administrator provides the new maximum value:
   \[ \rightarrow \text{AddressBookEntriesMaximumChange} \]
3. The system validates that the value is correct.
4. The system saves the new value.
5. The system displays the new current maximum value.

**Use case**

**Change shown customer details**

**Primary Actor:** System administrator

**Precondition:** None.

**Trigger:** The system administrator wants to change whether some customer attributes are shown.

**Main Success Scenario:**

1. The system displays the current values of customer details configuration (shown or not shown).
2. The system administrator provides the new value for one of the customer details:
   \[ \rightarrow \text{GenderCustomerDetailChange} \]
   \[ \rightarrow \text{DateOfBirthCustomerDetailChange} \]
   \[ \rightarrow \text{CompanyCustomerDetailChange} \]
   \[ \rightarrow \text{SuburbCustomerDetailChange} \]
   \[ \rightarrow \text{StateCustomerDetailChange} \]
3. The system validates that the value is correct.
4. The system saves the new value.
5. The system displays the new current values of customer details configuration.
   The system administrator repeats steps 2-5 until he is done.

**Use case**

**Assign shipping and packaging configuration values**

**Primary Actor:** System administrator

**Precondition:** None.

**Trigger:** The system administrator wants to change the shipping and packaging configuration values.

**Main Success Scenario:**
1. The system displays the current shipping and packaging configuration values.

2. The system administrator provides the new value for one of the shipping and packaging configurable options:
   - \( \Rightarrow \text{PostCodeShippingConfigurationChange} \)
   - \( \Rightarrow \text{MaximumPackageWeightShippingConfigurationChange} \)
   - \( \Rightarrow \text{TypicalPackageTareWeightShippingConfigurationChange} \)
   - \( \Rightarrow \text{PercentageIncreaseForLargerPackagesShippingConfigurationChange} \)
   - \( \Rightarrow \text{CountryShippingConfigurationChange} \)

3. The system validates that the value is correct.

4. The system saves the new value.

5. The system displays the new current shipping and packaging configuration values.
   - The system administrator repeats steps 2-5 until he is done.

---

**Use case**

**Change download configuration values**

**Primary Actor:** System administrator

**Precondition:** None.

**Trigger:** The system administrator wants to change the download configuration values.

**Main Success Scenario:**

1. The system displays the current download configuration values.

2. The system administrator provides the new value for one of the download configuration options:
   - \( \Rightarrow \text{EnableDownloadConfigurationChange} \)
   - \( \Rightarrow \text{DaysExpireDelayDownloadConfigurationChange} \)
   - \( \Rightarrow \text{MaximumNumberDownloadConfigurationChange} \)

3. The system validates that the value is correct.

4. The system saves the new value.

5. The system displays the new current download configuration values.
   - The system administrator repeats steps 2-5 until he is done.

---

**Use case**

**Change stock configuration values**

**Primary Actor:** System administrator

**Precondition:** None.

**Trigger:** The system administrator wants to change the stock configuration values.
Main Success Scenario:

1. The system displays the current stock configuration values.
2. The system administrator provides the new value for one of the stock configuration options:
   - CheckStockConfigurationChange
   - SubtractStockConfigurationChange
   - AllowCheckoutStockConfigurationChange
   - ReorderLevelStockConfigurationChange
3. The system validates that the value is correct.
4. The system saves the new value.
5. The system displays the new current stock configuration values.
   - The system administrator repeats steps 2-5 until he is done.

Use case

Install a payment method

Primary Actor: Store administrator
Precondition: The payment method is not installed yet.
Trigger: The store administrator wants to install a payment method.

Main Success Scenario:

1. The system shows all the available payment methods and which of they are installed.
2. The store administrator selects a non installed payment method.
3. The store administrator provides the data of the payment method:
   - InstallAuthorizeNetPaymentMethod
   - InstallCreditCardPaymentMethod
   - InstallCashOnDeliveryPaymentMethod
   - InstallPaymentPaymentMethod
   - InstallCheckMoneyPaymentMethod
   - InstallNocheXPaymentMethod
   - InstallPayPalPaymentMethod
   - InstallTwoCheckOutPaymentMethod
   - InstallPSiGatePaymentMethod
   - InstallSECPaymentMethod
4. The system validates that the data is correct.
5. The system uninstalls the new payment method.
Use case

Uninstall a payment method

**Primary Actor:** Store administrator

**Precondition:** The payment method is installed.

**Trigger:** The store administrator wants to uninstall a payment method.

**Main Success Scenario:**

1. The system shows all the payment methods and which of they are installed.
2. The store administrator selects an installed payment method.
   
   ```
   -> UninstallAuthorizeNetPaymentMethod
   -> UninstallCreditCardPaymentMethod
   -> UninstallCashOnDeliveryPaymentMethod
   -> UninstallPaymentMethod
   -> UninstallCheckMoneyPaymentMethod
   -> UninstallNocexoPaymentMethod
   -> UninstallPayPalPaymentMethod
   -> UninstallTwoCheckOutPaymentMethod
   -> UninstallPSiGatePaymentMethod
   -> UninstallSECPaymentMethod
   ```
3. The system uninstalls the selected payment method.

**Extensions:**

2a. The payment method is used in an existing order:
   
   2a1. The system warns the store administrator that the payment method is used in the information of existing orders and that is only possible to disable the payment method.
   2a2. The system changes the status of the payment method to disabled:

   ```
   -> StatusPaymentMethodChange
   ```
   2a3. The use case ends.

Use case

Change payment method values

**Primary Actor:** System administrator

**Precondition:** The payment method is installed.

**Trigger:** The system administrator wants to change the configuration values of an installed payment method.
Main Success Scenario:

1. The system displays the installed payment methods.
2. The customer selects an installed payment method.
3. The system displays the current values of the payment method.
4. The system administrator provides the new values for the configurable attributes of the payment method:
   
   - \[\textit{EditAuthorizeNetPaymentMethod}\]
   - \[\textit{EditCreditCardPaymentMethod}\]
   - \[\textit{EditCashOnDeliveryPaymentMethod}\]
   - \[\textit{EditPaymentPointPaymentMethod}\]
   - \[\textit{EditCheckMoneyPaymentMethod}\]
   - \[\textit{EditNocheckPaymentMethod}\]
   - \[\textit{EditPayPalPaymentMethod}\]
   - \[\textit{EditTwoCheckOutPaymentMethod}\]
   - \[\textit{EditPSIShippingMethod}\]
   - \[\textit{EditSECPaymentMethod}\]

5. The system validates that the new values are correct.
6. The system saves the new values.
7. The system displays the new values of the payment method.

Use case

**Install a shipping method**

**Primary Actor:** Store administrator

**Precondition:** The shipping method is not installed yet.

**Trigger:** The store administrator wants to install a shipping method.

Main Success Scenario:

1. The system shows all the available shipping methods and which of they are installed.
2. The store administrator selects a non installed shipping method.
3. The store administrator provides the data of the shipping method.
   
   - \[\textit{InstallZoneRatesShippingMethod}\]
   - \[\textit{InstallFlatRateShippingMethod}\]
   - \[\textit{InstallPerItemShippingMethod}\]
   - \[\textit{InstallTableRateShippingMethod}\]
   - \[\textit{InstallUSPostalServiceShippingMethod}\]

4. The system validates that the data is correct.
5. The system creates an instance of the new shipping method.
Use case
Uninstall a shipping method

Primary Actor: Store administrator
Precondition: The shipping method is installed.
Trigger: The store administrator wants to uninstall a shipping method.

Main Success Scenario:

1. The system shows all the available shipping methods and which of they are installed.
2. The store administrator selects an installed shipping method.
   - \( \rightarrow UninstallZoneRatesShippingMethod \)
   - \( \rightarrow UninstallFlatRateShippingMethod \)
   - \( \rightarrow UninstallPerItemShippingMethod \)
   - \( \rightarrow UninstallTableRateShippingMethod \)
   - \( \rightarrow UninstallUSPostalServiceShippingMethod \)
3. The system deletes the instance of the selected shipping method.

Extensions:

2a. The shipping method is the shipping method used in an existing order:
   2a1. The system warns the store administrator that the shipping method is used in the information of existing orders and that is only possible to disable the shipping method.
   2a2. The system changes the \textit{enabled} attribute of the shipping method to false:
       - \( \rightarrow StatusShippingMethodChange \)
   2a3. The use case ends.

Use case
Change shipping method values

Primary Actor: System administrator
Precondition: The shipping method is installed.
Trigger: The system administrator wants to change the configuration values of an installed shipping method.

Main Success Scenario:

1. The system displays the installed shipping methods.
2. The customer selects an installed shipping method.
3. The system displays the current values of the selected shipping method.
4. The system administrator provides the new values for the configurable attributes of the shipping method:

   → EditZoneRatesShippingMethod
   → EditFlatRateShippingMethod
   → EditPerItemShippingMethod
   → EditTableRateShippingMethod
   → EditUSPostalServiceShippingMethod

5. The system validates that the new values are correct.
6. The system saves the new values.
7. The system displays the new values of the shipping method.

---

**Use case**

**Add a language**

**Primary Actor:** Store administrator

**Precondition:** None.

**Trigger:** The store administrator wants to add a new language.

**Main Success Scenario:**

1. The store administrator provides the details of the new language:

   → NewLanguage

2. The system validates that the data is correct.
3. The system saves the new language.

---

**Use case**

**Edit a language**

**Primary Actor:** Store administrator

**Precondition:** None.

**Trigger:** The store administrator wants to edit a language.

**Main Success Scenario:**

1. The store administrator selects the language to be edited.
2. The store administrator provides the new details of the selected language:

   → EditLanguage

3. The system validates that the data is correct.
4. The system saves the changes.
Use case
Delete a language

Primary Actor: Store administrator
Precondition: There are at least two languages.
Trigger: The store administrator wants to delete a language.

Main Success Scenario:

1. The store administrator selects the language to be deleted.
2. The store administrator confirms that he wants to delete the language:
   \[ \rightarrow \text{DeleteLanguage} \]
3. The system deletes the language.

Extensions:

2a. The deleted language is the default language of the store.
   \[ \rightarrow \text{SetDefaultLanguage} \]
2b. The deleted language is the current language of any active session.
   \[ \rightarrow \text{SetCurrentLanguage} \]

Use case
Set the default language

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to change the default language.

Main Success Scenario:

1. The store administrator selects the language which will become the default language.
2. The system updates the default language:
   \[ \rightarrow \text{SetDefaultLanguage} \]
Use case
Add a currency

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to add a new currency.

Main Success Scenario:
1. The store administrator provides the details of the new currency:
   \[ \rightarrow \text{NewCurrency} \]
2. The system validates that the data is correct.
3. The system saves the new currency.

Use case
Edit a currency

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to edit a currency.

Main Success Scenario:
1. The store administrator selects the currency to be edited.
2. The store administrator provides the new details of the selected currency:
   \[ \rightarrow \text{EditCurrency} \]
3. The system validates that the data is correct.
4. The system saves the changes.

Use case
Delete a currency

Primary Actor: Store administrator
Precondition: There are at least two currencies.
Trigger: The store administrator wants to delete a currency.

Main Success Scenario:
1. The store administrator selects the currency to be deleted.
2. The store administrator confirms that he wants to delete the currency:
   \[\rightarrow \text{DeleteCurrency}\]
3. The system deletes the currency.

**Extensions:**

2a. The deleted currency was the default currency.
   2a1. The system sets any of the available currencies as the default currency:
       \[\rightarrow \text{SetDefaultCurrency}\]
2b. The deleted currency is the current currency of an active session.
   2b1. The system sets any of the available currencies as the current currency:
       \[\rightarrow \text{SetCurrentCurrency}\]
2c. The currency is the currency of an order:
   2c1. The system changes the status of the currency to disable.
       \[\rightarrow \text{CurrencyStatusChange}\]
   2c2. The use case ends.

**Use case**

**Update currencies**

**Primary Actor:** Store administrator

**Precondition:** None.

**Trigger:** The store administrator wants to update automatically via Internet the change values for currencies.

**Main Success Scenario:**

1. The system connects to the change information server.
2. The value change is automatically updated for all the currencies:
   \[\rightarrow \text{UpdateCurrencyValueChange}\]

**Use case**

**Set the default currency**

**Primary Actor:** Store administrator

**Precondition:** None.

**Trigger:** The store administrator wants to change the default currency.

**Main Success Scenario:**
1. The store administrator selects the currency which will become the default currency.
2. The system updates the default currency:
   \[\rightarrow \text{SetDefaultCurrency}\]

**Use case**

**Add a country**

**Primary Actor:** Store administrator

**Precondition:** None.

**Trigger:** The store administrator wants to add a country.

**Main Success Scenario:**

1. The store administrator provides the details of the new country:
   \[\rightarrow \text{NewCountry}\]
2. The system validates that the data is correct.
3. The system saves the new country.

**Use case**

**Edit a country**

**Primary Actor:** Store administrator

**Precondition:** None.

**Trigger:** The store administrator wants to edit a country.

**Main Success Scenario:**

1. The store administrator selects the country to be edited.
2. The store administrator provides the new details of the selected country:
   \[\rightarrow \text{EditCountry}\]
3. The system validates that the data is correct.
4. The system saves the changes.

**Use case**

**Delete a country**

**Primary Actor:** Store administrator

**Precondition:** The country is not the location of any address.
Trigger: The store administrator wants to delete a country.

Main Success Scenario:

1. The store administrator selects the country to be deleted.
2. The system warns the store administrator of the number of zones which are part of the country to be deleted.
3. The store administrator confirms that he wants to delete the country and their zones:
   [→DeleteCountry]
4. The system deletes the country and their zones.

Use case
Add a zone

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to add a zone.

Main Success Scenario:

1. The store administrator provides the details of the new zone:
   [→NewZone]
2. The system validates that the data is correct.
3. The system saves the new zone.

Use case
Edit a zone

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to edit a zone.

Main Success Scenario:

1. The store administrator selects the zone to be edited.
2. The store administrator provides the new details of the selected zone:
   [→EditZone]
3. The system validates that the data is correct.
4. The system saves the changes.
Use case
Delete a zone

Primary Actor: Store administrator
Precondition: The zone is not the location of any address.
Trigger: The store administrator wants to delete a zone.

Main Success Scenario:

1. The store administrator selects the zone to be deleted.
2. The store administrator confirms that he wants to delete the zone:
   \[\rightarrow DeleteZone\]
3. The system deletes the zone.

Use case
Add a tax zone

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to add a tax zone.

Main Success Scenario:

1. The store administrator provides the details of the new tax zone:
   \[\rightarrow NewTaxZone\]
2. The system validates that the data is correct.
3. The system saves the new tax zone.

Use case
Edit a tax zone

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to edit a tax zone.

Main Success Scenario:

1. The store administrator selects the tax zone to be edited.
2. The store administrator provides the new details of the selected tax zone:
3. The system validates that the data is correct.
4. The system saves the changes.

**Use case**

**Delete a tax zone**

**Primary Actor:** Store administrator  
**Precondition:** None.  
**Trigger:** The store administrator wants to delete a tax zone.

**Main Success Scenario:**
1. The store administrator selects the tax zone to be deleted.
2. The store administrator confirms that he wants to delete the tax zone:  
   [→DeleteTaxZone]
3. The system deletes the tax zone and all the associated tax rates.

**Use case**

**Add a tax class**

**Primary Actor:** Store administrator  
**Precondition:** None.  
**Trigger:** The store administrator wants to add a tax class.

**Main Success Scenario:**
1. The store administrator provides the details of the new tax class:  
   [→NewTaxClass]
2. The system validates that the data is correct.
3. The system saves the new tax class.

**Use case**

**Edit a tax class**

**Primary Actor:** Store administrator  
**Precondition:** None.  
**Trigger:** The store administrator wants to edit a tax class.
Main Success Scenario:

1. The store administrator selects the tax class to be edited.
2. The store administrator provides the new details of the selected tax class:
   
   \[ \rightarrow \text{EditTaxClass} \]

3. The system validates that the data is correct.
4. The system saves the changes.

Use case
Delete a tax class

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to delete a tax class.

Main Success Scenario:

1. The store administrator selects the tax class to be deleted.
2. The system informs the store administrator about how many products are associated to the deleted tax class.
3. The store administrator confirms that he wants to delete the tax class:
   
   \[ \rightarrow \text{DeleteTaxClass} \]

4. The system deletes the tax class and all the associated tax rates.

Extensions:

2a. The store administrator doesn’t want to delete the tax class.
   
   2a1. The use case ends.

Use case
Add a tax rate

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to add a tax rate.

Main Success Scenario:

1. The store administrator provides the details of the new tax rate:
   
   \[ \rightarrow \text{NewTaxRate} \]
2. The system validates that the data is correct.
3. The system saves the new tax rate.

**Use case**

**Edit a tax rate**

**Primary Actor:** Store administrator  
**Precondition:** None.  
**Trigger:** The store administrator wants to edit a tax rate.

**Main Success Scenario:**

1. The store administrator selects the tax rate to be edited.
2. The store administrator provides the new details of the selected tax rate:
   
   $\rightarrow \text{EditTaxRate}$

3. The system validates that the data is correct.
4. The system saves the changes.

**Use case**

**Delete a tax rate**

**Primary Actor:** Store administrator  
**Precondition:** None.  
**Trigger:** The store administrator wants to delete a tax rate.

**Main Success Scenario:**

1. The store administrator selects the tax rate to be deleted.
2. The store administrator confirms that he wants to delete the tax rate:
   
   $\rightarrow \text{DeleteTaxRate}$

3. The system deletes the tax rate.

**Use case**

**Add a product**

**Primary Actor:** Store administrator  
**Precondition:** None.  
**Trigger:** The store administrator wants to add a product to the store catalog.
Main Success Scenario:

1. The store administrator selects the product category.
2. The store administrator provides the product data:
   
   \[\rightarrow \text{NewProduct}\]
3. The system validates that the data is correct.
4. The system saves the new product.
5. The store administrator provides a product attribute:
   
   \[\rightarrow \text{NewProductAttribute}\]
6. The system validates that the product attribute is correct.
7. The system saves the new product attribute.
   The store administrator repeats steps 5-7 until he is done.

Extensions:

5a. The product does not have product attributes:
   
   5a1. The use case ends.
5b. The product option is new:
   
   5b1. Add a product option
5c. The product option value is new:
   
   5c1. Add a product option value.

**Use case**

**Add a product option**

**Primary Actor:** Store administrator

**Precondition:** None.

**Trigger:** The store administrator wants to add a product option to the store catalog.

Main Success Scenario:

1. The store administrator provides the product option data:

   \[\rightarrow \text{NewProductOption}\]
2. The system validates that the data is correct.
3. The system saves the new product option.
Use case

Edit a product option

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to edit a product option.

Main Success Scenario:

1. The store administrator selects the product option to be edited.
2. The store administrator provides the new details of the selected product option:
   
   $\text{[} \rightarrow \text{EditProductOption}]$

3. The system validates that the data is correct.
4. The system saves the changes.

Use case

Delete a product option

Primary Actor: Store administrator
Precondition: The product option has not products or values linked to it.
Trigger: The store administrator wants to delete a product option.

Main Success Scenario:

1. The store administrator selects the product option to be deleted.
2. The system asks for the confirmation of the store administrator.
3. The store administrator confirms that he wants to delete the product option:
   
   $\text{[} \rightarrow \text{DeleteProductOption}]$

4. The system deletes the product option.

Use case

Add a product option value

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to add a value to a product option.

Main Success Scenario:
1. The store administrator selects the product option.
2. The store administrator provides the product option value data:
   \[ \rightarrow \text{NewProductOptionValue} \]
3. The system validates that the data is correct.
4. The system saves the new product option value.

**Use case**

**Edit a product option value**

**Primary Actor:** Store administrator  
**Precondition:** None.  
**Trigger:** The store administrator wants to edit a product option value.

**Main Success Scenario:**

1. The store administrator selects the product option value to be edited.
2. The store administrator provides the new details of the selected product option value:
   \[ \rightarrow \text{EditProductOptionValue} \]
3. The system validates that the data is correct.
4. The system saves the changes.

**Use case**

**Delete a product option value**

**Primary Actor:** Store administrator  
**Precondition:** The product option value has not products linked to it.  
**Trigger:** The store administrator wants to delete a product option value.

**Main Success Scenario:**

1. The store administrator selects the product option value to delete.
2. The system asks for the confirmation of the store administrator.
3. The store administrator confirms that he wants to delete the product option value:
   \[ \rightarrow \text{DeleteProductOptionValue} \]
4. The system deletes the product option value.
**Use case**

### Add a product attribute

**Primary Actor:** Store administrator

**Precondition:** None.

**Trigger:** The store administrator wants to assign an attribute to a product.

**Main Success Scenario:**

1. The store administrator selects the product.
2. The store administrator provides the attribute and the product attribute data (increment and sign):
   - \([\rightarrow \text{NewProductAttribute}]\)
   - \([\rightarrow \text{NewDownloadableProductAttribute}]\)
3. The system validates that the data is correct.
4. The system saves the new product attribute.

**Extensions:**

2a. The product option is new:
   - 2a1. *Add a product option.*
2b. The product option value is new:
   - 2b1. *Add a product option value.*

**Use case**

### Edit a product attribute

**Primary Actor:** Store administrator

**Precondition:** None.

**Trigger:** The store administrator wants to edit a product attribute.

**Main Success Scenario:**

1. The store administrator selects the product attribute to be edited.
2. The store administrator provides the new details for the product attribute:
   - \([\rightarrow \text{AttributeChange}]\)
   - \([\rightarrow \text{IncrementAndSignAttributeChange}]\)
   - \([\rightarrow \text{EditDownloadableAttribute}]\)
3. The system validates that the data is correct.
4. The system saves the changes.
   The system repeats steps 2-4 until he is done.
Use case
Delete a product attribute

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to delete a product attribute.

Main Success Scenario:

1. The store administrator selects the product attribute to be deleted.
2. The system asks for the confirmation of the store administrator.
3. The store administrator confirms that he wants to delete the product attribute:
   \[ \rightarrow \text{DeleteProductAttribute} \]
4. The system deletes the product attribute.

Extensions:

3a. The product attribute is part of an existing order line:
   3a1. The system changes the status of the product attribute to disable.
       \[ \rightarrow \text{ProductAttributeStatusChange} \]
   3a2. The use case ends

Use case
Edit a product

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to edit a product.

Main Success Scenario:

1. The store administrator selects the product to be edited.
2. The store administrator provides the new values for the attributes of the product:
   \[ \rightarrow \text{EditProduct} \]
3. The system validates that the data is correct.
4. The system saves the changes.
Use case
Delete a product

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to delete a product.

Main Success Scenario:

1. The store administrator selects the product to be deleted.
2. The system asks for the confirmation of the store administrator.
3. The store administrator confirms that he wants to delete the product:
   [→DeleteProduct]
4. The system deletes the product and their product attributes.

Extensions:

3a. The product is part of an order:
   3a1. The system changes the status of the product to disable.
       [→ProductStatusChange]
   3a2. The use case ends.

Use case
Move a product

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to change the category of a product.

Main Success Scenario:

1. The store administrator selects the product to be moved.
2. The store administrator indicates the new category of the selected product, if any:
   [→MoveProduct]
3. The system validates that the data is correct.
4. The system saves the new placement.
Use case
Link a product

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to link a product to another category.

Main Success Scenario:

1. The store administrator selects the product to be linked.
2. The store administrator indicates the new category of the selected product, if any:
   \[ \rightarrow \text{LinkProduct} \]
3. The system links the product.

Use case
Add a product category

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to add a category.

Main Success Scenario:

1. The store administrator provides the details of the new product category, including its parent category, if any:
   \[ \rightarrow \text{NewCategory} \]
2. The system validates that the data is correct.
3. The system saves the new category.

Use case
Edit a product category

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to edit a category.

Main Success Scenario:

1. The store administrator selects the category to be edited.
2. The store administrator provides the new details of the selected category:
   \[\rightarrow\text{EditCategory}\]
3. The system validates that the data is correct.
4. The system saves the changes.

**Use case**

**Move a product category**

**Primary Actor:** Store administrator  
**Precondition:** None.  
**Trigger:** The store administrator wants to change the placement of a category in the category hierarchy.

**Main Success Scenario:**

1. The store administrator selects the category to be moved.  
2. The store administrator indicates the new parent category, if any:  
   \[\rightarrow\text{MoveCategory}\]  
3. The system validates that the data is correct.  
4. The system saves the new placement.

**Use case**

**Delete a product category**

**Primary Actor:** Store administrator  
**Precondition:** None.  
**Trigger:** The store administrator wants to delete a category.

**Main Success Scenario:**

1. The store administrator selects the category to be deleted.
2. The system warns the store administrator of the number of subcategories and products linked to the category to be deleted.  
3. The store administrator confirms that he wants to delete the category:  
   \[\rightarrow\text{DeleteCategory}\]  
4. The system deletes the selected category and its subcategories. The products linked to the deleted category or its subcategories are removed from the system if they do not participate in any orders. The system changes the status of the products which participate in orders to out of stock.
### Use case

#### Add a special

**Primary Actor:** Store administrator  
**Precondition:** None.  
**Trigger:** The store administrator wants to add a special.

**Main Success Scenario:**

1. The store administrator selects the product which will be offered in a special price.
2. The store administrator provides the details of the special:  
   
   \[
   \text{[→NewSpecial]} 
   \]
3. The system validates that the data is correct.
4. The system saves the new special.

#### Use case

#### Edit a special

**Primary Actor:** Store administrator  
**Precondition:** None.  
**Trigger:** The store administrator wants to edit a special.

**Main Success Scenario:**

1. The store administrator selects the special to be edited.
2. The store administrator provides the new details of the selected special:  
   
   \[
   \text{[→EditSpecial]} 
   \]
3. The system validates that the data is correct.
4. The system saves the changes.

### Use case

#### Delete a special

**Primary Actor:** Store administrator  
**Precondition:** None.  
**Trigger:** The store administrator wants to delete a special.

**Main Success Scenario:**
1. The store administrator selects the special to be deleted.
2. The system asks for the confirmation of the store administrator.
3. The store administrator confirms that he wants to delete the special:
   \[ \rightarrow \text{DeleteSpecial} \]
4. The system deletes the special.

**Add a manufacturer**

**Primary Actor:** Store administrator

**Precondition:** None.

**Trigger:** The store administrator wants to add a manufacturer.

**Main Success Scenario:**

1. The store administrator provides the details of the new manufacturer:
   \[ \rightarrow \text{NewManufacturer} \]
2. The system validates that the data is correct.
3. The system saves the new manufacturer.

**Edit a manufacturer**

**Primary Actor:** Store administrator

**Precondition:** None.

**Trigger:** The store administrator wants to edit a manufacturer.

**Main Success Scenario:**

1. The store administrator selects the manufacturer to be edited.
2. The store administrator provides the new details of the selected manufacturer:
   \[ \rightarrow \text{EditManufacturer} \]
3. The system validates that the data is correct.
4. The system saves the changes.
**Use case**

**Delete a manufacturer**

**Primary Actor:** Store administrator  
**Precondition:** None.  
**Trigger:** The store administrator wants to delete a manufacturer.

**Main Success Scenario:**

1. The store administrator selects the manufacturer to delete.  
2. The system warns the store administrator of the number of products linked to the manufacturer to be deleted.  
3. The store administrator confirms that he wants to delete the manufacturer:  
   
   $\Rightarrow$DeleteManufacturer  

4. The system deletes the manufacturer and, if requested, changes the status of the products manufactured by it to out of stock.

**Use case**

**Add a banner**

**Primary Actor:** Store administrator  
**Precondition:** None.  
**Trigger:** The store administrator wants to add a new banner.

**Main Success Scenario:**

1. The store administrator provides the details of the new banner:  
   
   $\Rightarrow$NewBanner  

2. The system validates that the data is correct.  
3. The system saves the new banner.

**Use case**

**Edit a banner**

**Primary Actor:** Store administrator  
**Precondition:** None.  
**Trigger:** The store administrator wants to edit a banner.

**Main Success Scenario:**
1. The store administrator selects the banner to be edited.
2. The store administrator provides the new details of the selected banner:
   \[\rightarrow \text{EditBanner}\]
3. The system validates that the data is correct.
4. The system saves the changes.

**Use case**

Delete a banner

**Primary Actor:** Store administrator

**Precondition:** None.

**Trigger:** The store administrator wants to delete a banner.

**Main Success Scenario:**

1. The store administrator selects the banner to be deleted.
2. The store administrator confirms that he wants to delete the banner:
   \[\rightarrow \text{DeleteBanner}\]
3. The system deletes the banner.

**Use case**

Add a banner group

**Primary Actor:** Store administrator

**Precondition:** None.

**Trigger:** The store administrator wants to add a new banner group.

**Main Success Scenario:**

1. The store administrator provides the details of the new banner group:
   \[\rightarrow \text{NewBannerGroup}\]
2. The system validates that the data is correct.
3. The system saves the new banner.
Use case

Edit a banner group

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to edit a banner group.

Main Success Scenario:

1. The store administrator selects the banner group to be edited.
2. The store administrator provides the new details of the selected banner group:
   
   \[ \text{[\rightarrow EditBannerGroup]} \]

3. The system validates that the data is correct.
4. The system saves the changes.

Use case

Delete a banner group

Primary Actor: Store administrator
Precondition: The banner group doesn’t contain any banners.
Trigger: The store administrator wants to delete a banner.

Main Success Scenario:

1. The store administrator selects the banner group to be deleted.
2. The store administrator confirms that he wants to delete the banner group:

   \[ \text{[\rightarrow DeleteBannerGroup]} \]

3. The system deletes the banner.

Use case

Send an email

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to send an email to customers.

Main Success Scenario:
1. The store administrator selects the addressee customer, or one of the predefined set of addressee customers (all the customers or all the newsletter subscribers).
2. The store administrator specifies the sender address.
3. The store administrator provides the subject and the message.
4. The store administrator confirms that he wants to send the email.
5. The system sends the email.

**Use case**

**Create a newsletter**

**Primary Actor:** Store administrator

**Precondition:** None.

**Trigger:** The store administrator wants to create a new newsletter.

**Main Success Scenario:**

1. The store administrator selects the type of the newsletter (newsletter or product notification).
2. The store administrator provides the title and the content of the newsletter:
   
   ```plaintext
   [→NewNewsletter]
   [→NewProductNotification]
   ```
3. The system validates that the data is correct.
4. The system saves the newsletter.

**Use case**

**Edit a newsletter**

**Primary Actor:** Store administrator

**Precondition:** The newsletter is unlocked.

**Trigger:** The store administrator wants to edit a newsletter.

**Main Success Scenario:**

1. The store administrator selects the newsletter to be edited.
2. The store administrator provides the new details of the selected newsletter:
   
   ```plaintext
   [→EditNewsletter]
   [→EditProductNotification]
   ```
3. The system validates that the data is correct.
4. The system saves the changes.
Use case

Delete a newsletter

Primary Actor: Store administrator
Precondition: The newsletter is unlocked.
Trigger: The store administrator wants to delete a newsletter.

Main Success Scenario:

1. The store administrator selects the newsletter to be deleted.
2. The store administrator confirms that he wants to delete the newsletter:
   \[\rightarrow DeleteNewsletter\]
3. The system deletes the newsletter.

Use case

Lock a newsletter

Primary Actor: Store administrator
Precondition: The newsletter is unlocked.
Trigger: The store administrator wants to indicate to the other administrators that a newsletter is pending to be delivered.

Main Success Scenario:

1. The store administrator selects the newsletter to be locked.
   \[\rightarrow LockNewsletter\]
2. The system saves the change.

Use case

Unlock a newsletter

Primary Actor: Store administrator
Precondition: The newsletter is locked.
Trigger: The store administrator wants to indicate to the other administrators that a newsletter ceases to be locked.

Main Success Scenario:

1. The store administrator selects the newsletter to be unlocked.
   \[\rightarrow UnlockNewsletter\]
2. The system saves the change.

Use case

Send a newsletter

Primary Actor: Store administrator
Precondition: The newsletter is locked.
Trigger: The store administrator wants to send a newsletter.

Main Success Scenario:

1. The store administrator selects the newsletter which will be sent.
2. The system sends the newsletter to all the newsletter subscribers.
3. The system saves that the newsletter has been sent.
   \[\rightarrow\text{SendNewsletter}\]

Extensions:

2a. The newsletter is a product notification.
   2a1. The store administrator selects which products are implied in the notification.
   2a2. The system sends the newsletter to customers who are subscribed to some of the notified products.
   2a3. The use case continues at step 3.

Use case

Create a customer

Primary Actor: Customer
Precondition: None.
Trigger: A customer wants to open an account in the store.

Main Success Scenario:

1. The customer provides the required customer data:
   \[\rightarrow\text{NewCustomer}\]
2. The system validates the customer data.
3. The system saves the new account.
### Use case
**Change password**

**Primary Actor:** Customer  
**Precondition:** The customer is logged in.  
**Trigger:** A customer wants to change his password.

**Main Success Scenario:**

1. The customer provides the old password.  
2. The customer provides the new password twice.  
   \[ \rightarrow \text{PasswordChange} \]  
3. The system validates that the data is correct.  
4. The system saves the changes.

### Use case
**Change customer details**

**Primary Actor:** Customer  
**Precondition:** The customer is logged in.  
**Trigger:** A customer wants to change its customer details.

**Main Success Scenario:**

1. The customer provides the new customer details.  
   \[ \rightarrow \text{EditCustomerDetails} \]  
2. The system validates that the data is correct.  
3. The system saves the changes.

### Use case
**Administrate address book**

**Primary Actor:** Customer  
**Precondition:** The customer is logged in and the number of addresses is less than the maximum number of address entries permitted.  
**Trigger:** A customer wants to view or change the address book.

**Main Success Scenario:**
1. The system displays the current address book entries of the customer.
2. The customer selects an address book entry to be edited:
   
   \[ \rightarrow \text{EditCustomerAddress} \]

3. The system validates that the data is correct.
4. The system saves the changes and displays the new address book.
   
   The customer repeats steps 1-4 until he is done.

**Extensions:**

2a. The customer doesn’t want to change the address book:
   
   2a1. The use case ends.

2b. The customer wants to add a new address book entry:
   
   2b1. The customer provides the required data:
   
   \[ \rightarrow \text{NewCustomerAddress} \]

   2b2. The use case continues at step 3.

2c. The customer wants to delete an address book entry:
   
   2c1. The customer selects the address book entry:
   
   \[ \rightarrow \text{DeleteCustomerAddress} \]

   2c2. The use case continues at step 3.

2d. The customer wants to change the default address book entry:
   
   2d1. The customer selects the new default address book entry:
   
   \[ \rightarrow \text{PrimaryCustomerAddressChange} \]

   2d2. The use case continues at step 3.

---

**Use case**

**Administrate subscriptions**

**Primary Actor:** Customer

**Precondition:** The customer is logged in.

**Trigger:** A customer wants to view or change their product notification subscriptions.

**Main Success Scenario:**

1. The system displays the details of the current product notification subscriptions of the customer.
2. The customer adds a new product subscription:
   
   \[ \rightarrow \text{NewProductNotificationSubscription} \]

3. The system validates that the data is correct.
4. The system saves the changes and displays the new product notification subscriptions.
   
   The customer repeats steps 1-4 until he is done.

**Extensions:**
2a. The customer doesn’t want to change their product notification subscriptions:
   2a1. The use case ends.
2b. The customer wants to be subscribed or unsubscribed to all product notifications:
   [→ EditGlobalNotifications]
2c. The customer wants to delete a product notification subscription:
   2c1. The customer selects the product:
      [→ DeleteProductNotificationSubscription]
   2c2. The use case continues at step 3.

---

**Use case**

**Edit a customer**

**Primary Actor:** Store administrator  
**Precondition:** None.  
**Trigger:** The store administrator wants to edit a customer.

**Main Success Scenario:**

1. The store administrator selects the customer to be edited.  
2. The store administrator provides the new details of the selected customer:  
   [→ EditCustomer]
3. The system validates that the data is correct.  
4. The system saves the changes.

---

**Delete a customer**

**Primary Actor:** Store administrator  
**Precondition:** None.  
**Trigger:** The store administrator wants to delete a customer.

**Main Success Scenario:**

1. The store administrator selects the customer to be deleted.  
2. The system asks for the confirmation of the store administrator.  
3. The store administrator confirms that he wants to delete the customer:  
   [→ DeleteCustomer]
4. The system deletes the customer and their addresses, reviews, notification subscriptions and shopping carts.
Extensions:

3a. The customer has orders:
   3a1. The system changes the status of the customer to disable.
       \( \rightarrow \text{CustomerStatusChange} \)
   3a2. The system deletes customer's addresses, reviews, notification subscriptions
       and shopping carts.
   3a3. The use case ends.

Use case

Open session

Primary Actor: Customer
Precondition: None.
Trigger: A customer starts using the system.

Main Success Scenario:

1. The system creates an anonymous session:
   \( \rightarrow \text{NewSession} \)

Use case

Finish session

Primary Actor: Customer
Precondition: None.
Trigger: A customer finishes using the system.

Main Success Scenario:

1. The system deletes the current session.
   \( \rightarrow \text{DeleteSession} \)

Extensions:

1a. The customer is logged in and the session has a non empty shopping cart.
   1a1. The shopping cart is saved.
**Use case**

**Log in**

**Primary Actor:** Customer  
**Precondition:** The customer is not logged in yet.  
**Trigger:** A customer logs in the system.

**Main Success Scenario:**

1. The customer introduces their identification data.  
2. The system validates the identification data.  
3. The customer becomes the owner of the current session.  
   
   → *Log in*

**Extensions:**

3a. The customer has a shopping cart from a previous session.  
   3a1. The previous shopping cart is restored.  
   
   → *Restore Previous Shopping Cart*

**Use case**

**LogOut**

**Primary Actor:** Customer  
**Precondition:** The customer is logged in.  
**Trigger:** A customer logs out from the system.

**Main Success Scenario:**

1. The current session becomes anonymous.  
   
   → *LogOut*

**Extensions:**

1a. The customer has a non empty shopping cart.  
   1a1. The shopping cart is saved.
### Use case
**Change the current language**

**Primary Actor:** Customer  
**Precondition:** None.  
**Trigger:** A customer wants to change the current language of the session.

**Main Success Scenario:**

1. The store administrator selects the language which will become the current language.  
2. The system updates the current language.  
   
   \[
   \rightarrow \text{SetCurrentLanguage}
   \]

### Use case
**Change the current currency**

**Primary Actor:** Customer  
**Precondition:** None.  
**Trigger:** A customer wants to change the current currency of the session.

**Main Success Scenario:**

1. The store administrator selects the currency which will become the current currency.  
2. The system updates the current currency.  
   
   \[
   \rightarrow \text{SetCurrentCurrency}
   \]

### Use case
**Add a review**

**Primary Actor:** Customer  
**Precondition:** None.  
**Trigger:** A customer wants to write a review of a product.

**Main Success Scenario:**

1. The customer selects a product.  
2. The customer provides the content and the rate of the review.  
   
   \[
   \rightarrow \text{NewReview}
   \]
3. The system validates that the data is correct.
4. The system saves the review.

**Extensions:**

2a. The customer is not logged in:
   
   2a1. The customer logs in:
       
       [→Login]

   2a2. The use case continues at step 2.

---

**Use case**

**Edit a review**

**Primary Actor:** Store administrator

**Precondition:** None.

**Trigger:** The store administrator wants to edit a review.

**Main Success Scenario:**

1. The store administrator selects the review to be edited.
2. The store administrator provides the modified text and the new rating of the selected review.
   
   [→EditReview]
3. The system validates that the data is correct.
4. The system saves the changes.

---

**Use case**

**Delete a review**

**Primary Actor:** Store administrator

**Precondition:** None.

**Trigger:** The store administrator wants to delete a review.

**Main Success Scenario:**

1. The store administrator selects the review to be deleted.
2. The system asks for the confirmation of the store administrator.
3. The store administrator confirms that he wants to delete the review:
   
   [→DeleteReview]
4. The system deletes the review.
Use case

Place and order

Primary Actor: Customer
Precondition: The customer is logged in.
Trigger: A customer wants to place and order.

Main Success Scenario:

1. At any time before step 10 the customer logs in:
   \[→ LogIn\]
2. The system adds the contents of the anonymous shopping cart to the customer shopping cart.
3. The system displays the contents of the shopping cart.
4. The customer browses the product catalog.
   \[→ ReadProductInfo\]
5. The customer selects a product to buy:
   \[→ AddProductToShoppingCart\]
6. The system displays the contents of the shopping cart.
7. The customer changes the contents of the shopping cart:
   \[→ UpdateShoppingCart\]
8. The system updates the shopping cart.
9. The system displays the contents of the updated shopping cart.

   The customer repeats steps 3,4 and 7 as necessary to build his order.
10. The customer checks out the order.
11. The system shows the shipping address and the available shipping methods.
12. The customer selects the preferred shipping method.
13. The system shows the billing address and the available payment methods.
14. The customer selects the preferred payment method.
15. The system displays a summary of the order.
16. The customer confirms the order:
   \[→ OrderConfirmation\]
17. The system saves the order.
18. The system sends an email to the customer and to the store extra order emails with the information about the order.

Extensions:

1a. The customer is new:
   1a1. Create customer.
6a. The configurable option Display cart after adding a product is disabled
The customer repeats steps 4 and 5 as necessary.

16a. The customer wants to change the contents of the shopping cart:
   16a1. The customer changes the contents of the shopping cart:
      \[\rightarrow\text{OrderConfirmation}\]
   16a2. The customer continues with the checkout procedure at step 11.

11a, 16a. The customer wants to change the shipping address:
   11a1. The system shows the know addresses of the customer.
   11a2. The customer selects a different shipping address.
   11a3. The customer continues with the checkout procedure at step 11.

13a, 16b. The customer wants to change the billing address:
   13a1. The system shows the know addresses of the customer.
   13a2. The customer selects a different billing address.
   13a3. The customer continues with the checkout procedure at step 13.

16c. The customer wants to change the shipping method:
   16c1. The customer selects the new shipping method.
   16c2. The customer continues with the checkout procedure at step 13.

16d. The customer wants to change the payment method:
   16d1. The customer selects the new payment method.
   16d2. The customer continues with the checkout procedure at step 15.

11a2a,16a2a. The customer wants to define a new shipping address:
   11a2a1. The customer gives the new address:
      \[\rightarrow\text{OrderConfirmation}\]
   11a2a2. The system saves the address.
   11a2a3. The customer continues with the checkout procedure at step 11.

13a2a,16b2a. The customer wants to define a new billing address:
   13a2a1. The customer gives the new address:
      \[\rightarrow\text{NewCustomerAddress}\]
   13a2a2. The system saves the address.
   13a2a3. The customer continues with the checkout procedure at step 13.

---

**Use case**

**Cancel an order**

**Primary Actor:** Store administrator

**Precondition:** None.

**Trigger:** The store administrator wants to cancel an order.
Main Success Scenario:

1. The store administrator selects the order to be cancelled.
2. The system asks for the confirmation of the store administrator.
3. The store administrator confirms that he wants to cancel the order:
   \[\rightarrow \text{CancelOrder}\]
4. The system sets the order status to cancelled.

Use case

Add an order status

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to add a new order status.

Main Success Scenario:

1. The store administrator provides the details of the new order status:
   \[\rightarrow \text{NewOrderStatus}\]
2. The system validates that the data is correct.
3. The system saves the new order status.

Use case

Edit an order status

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to edit an order status.

Main Success Scenario:

1. The store administrator selects the order status to be edited.
2. The store administrator provides the new details of the selected order status:
   \[\rightarrow \text{EditOrderStatus}\]
3. The system validates that the data is correct.
4. The system saves the changes.
Use case
Delete an order status

Primary Actor: Store administrator
Precondition: The deleted order status is not the current status of any order.
Trigger: The store administrator wants to delete an order status.

Main Success Scenario:
1. The store administrator selects the order status to be deleted.
2. The store administrator confirms that he wants to delete the order status:
   \[\rightarrow DeleteOrderStatus\]
3. The system deletes the order status.

Extensions:
2a. The order status has been an status of an order:
   2a1. The system changes the status of the order status to disable.
       \[\rightarrow ProductStatusChange\]
   2a2. The use case ends.

Use case
Change the status of an order

Primary Actor: Store administrator
Precondition: None.
Trigger: The store administrator wants to change the status of an order.

Main Success Scenario:
1. The system shows the orders and their status.
2. The store administrator selects the order which will be edited.
3. The system shows the applicable order status.
4. The store administrator selects the new status.
   \[\rightarrow UpdateOrderStatus\]
5. The system validates that the data is correct.
6. The system saves the changes.
Use case
Set cancelled order status

Primary Actor: Store administrator
Precondition: The order status is not yet the cancelled status.
Trigger: The store administrator wants to indicate to the system which order status is used to indicate that an order is cancelled.

Main Success Scenario:

1. The store administrator selects an order status.
2. The system register that the selected order status represents cancelled orders.
   [$\Rightarrow$ SetCancelledOrderStatus]

Use case
Set default order status

Primary Actor: Store administrator
Precondition: The order status is not yet the default status.
Trigger: The store administrator wants to indicate to the system which order status is assign when an order is created.

Main Success Scenario:

1. The store administrator selects an order status.
2. The system register that the selected order status is the default order status.
   [$\Rightarrow$ SetDefaultOrderStatus]

Use case
Show a banner

Primary Actor: System.
Precondition: None.
Trigger: The system shows a banner.

Main Success Scenario:

1. The system shows a banner.
   [$\Rightarrow$ ShowBanner]
Use case

Click a banner

Primary Actor: Customer
Precondition: None.
Trigger: The customer clicks on a banner.

Main Success Scenario:

1. The customer clicks on a banner.
   \[\rightarrow ClickBanner\]
2. The system redirects the online store to the banner's web page.

Use case

Read a review

Primary Actor: Customer
Precondition: None.
Trigger: A customer wants to read a review of a product.

Main Success Scenario:

1. The system shows a summary of the reviews of the product.
2. The customer selects a review.
   \[\rightarrow ReadReview\]
3. The system shows the selected review.

Use case

Download a product

Primary Actor: Customer
Precondition: The customer is logged in.

   The customer purchased the product.
   Download is enabled by the system.
   The download is not expired and the number of download has not been exceeded.

Trigger: A customer wants to download a purchased product.
Main Success Scenario:

1. The customer selects the purchased product to be downloaded.
   
   [→ProductDownload]

2. The system allows the customer downloading the product.

**Use case**

Show manufacturer's web

**Primary Actor:** Customer

**Precondition:** None.

**Trigger:** A customer wants to be redirected to the manufacturer's web page.

Main Success Scenario:

1. The customer selects a manufacturer.
   
   [→ClickManufacturer]

2. The customer is redirected to the manufacturer's web page.

**Use case**

Show products under stock

**Primary Actor:** Store administrator.

**Precondition:** None.

**Trigger:** The store administrator wants to obtain which products have to be reordered.

Main Success Scenario:

1. The system shows the set of products under stock:
   
   [→ShowUnderStockProducts]

**Use case**

Show expected products

**Primary Actor:** Store administrator, Customer.

**Precondition:** None.

**Trigger:** The store administrator or the customer wants to obtain which products will be in stock soon.
Main Success Scenario:

1. The system shows the set of expected products:
   \[\rightarrow \text{ShowExpectedProducts}\]

**Use case**

**Show orders of a customer**

**Primary Actor:** Store administrator.

**Precondition:** None.

**Trigger:** The store administrator wants to obtain the orders of a customer.

**Main Success Scenario:**

1. The system shows the list of customers.
2. The store administrator selects a customer.
3. The system shows the orders of the selected customer:
   \[\rightarrow \text{ShowOrdersOfCustomer}\]

**Use case**

**Show previous orders**

**Primary Actor:** Customer.

**Precondition:** The customer is logged in.

**Trigger:** A Customer wants to visualize their orders.

**Main Success Scenario:**

1. The system shows the previous orders made by the customer:
   \[\rightarrow \text{ShowOrdersOfCustomer}\]

**Use case**

**Show best viewed products**

**Primary Actor:** Store administrator, Customer.

**Precondition:** None.

**Trigger:** The store administrator or the customer wants to visualize the most viewed products.
Main Success Scenario:

1. The system shows the products in stock ordered by the number of times which has been visualized by the customers.
   
   [→ShowBestViewedProducts]

Use case

Show best products purchased

Primary Actor: Store administrator, Customer.
Precondition: None.
Trigger: The store administrator or the customer wants to visualize the most purchased products.

Main Success Scenario:

1. The system shows the products in stock ordered by the number of times which has been purchased by the customers.
   
   [→ShowBestPurchasedProducts]

Use case

Show customer’s orders total

Primary Actor: Store administrator.
Precondition: None.
Trigger: The store administrator wants to visualize the total amount of money spent by each customer in the online store.

Main Success Scenario:

1. The system shows the customers and the total price of their orders.
   
   [→ShowCustomersOrdersTotal]

Use case

Online customers

Primary Actor: Store administrator.
Precondition: None.
Trigger: The store administrator wants to visualize the customers who are online.
Main Success Scenario:

1. The system shows the online customers. 
   \[\rightarrow ShowOnlineCustomers\]

**Use case**

**Show specials**

**Primary Actor:** Store administrator, Customer.
**Precondition:** None.
**Trigger:** The store administrator or the customer wants to visualize products on offer.

Main Success Scenario:

1. The system shows the products on offer. 
   \[\rightarrow ShowSpecials\]

**Use case**

**Show products of a category**

**Primary Actor:** Store administrator, customer.
**Precondition:** None.
**Trigger:** The store administrator or the customer wants to visualize the products contained in a category.

Main Success Scenario:

1. The store administrator or the customer selects a category. 
2. The system shows the products of the selected category. 
   \[\rightarrow ShowProductsOfCategory\]

**Use case**

**Show products of a manufacturer**

**Primary Actor:** Store administrator, customer.
**Precondition:** None.
**Trigger:** The store administrator or the customer wants to visualize the products produced by a manufacturer.
Main Success Scenario:

1. The store administrator or the customer selects a manufacturer.
2. The system shows the products manufactured by the selected manufacturer.
   
   \[\rightarrow \text{ShowProductsOfManufacturer}\]

Use case

Show new products

Primary Actor: Customer.
Precondition: None.
Trigger: The customer wants to visualize the last launched products.

Main Success Scenario:

1. The system shows the last products on sale.
   
   \[\rightarrow \text{ShowNewProducts}\]

Use case

Show reviews of a product

Primary Actor: Store administrator, customer.
Precondition: None.
Trigger: The store administrator or the customer wants to visualize the reviews of a product.

Main Success Scenario:

1. The store administrator or the customer selects a product.
2. The system shows the reviews of the selected product
   
   \[\rightarrow \text{ShowReviewsOfProduct}\]

Use case

Tell to a friend

Primary Actor: Customer.
Precondition: None.
Trigger: A Customer wants to send the current web page to a friend with a comment by email.
Main Success Scenario:

1. The customer provides his name, his friend's name, his friend's email and the message about the web page.
2. The system sends the email with a link to the current web page.

Use case

**Generate an invoice**

**Primary Actor:** Store administrator.

**Precondition:** None.

**Trigger:** The store administrator wants to generate an invoice corresponding to an order.

Main Success Scenario:

1. The store administrator selects the order.
2. The system shows a printable invoice.

Use case

**Generate a packaging slip**

**Primary Actor:** Store administrator.

**Precondition:** None.

**Trigger:** The store administrator wants to generate a packaging slip of an order.

Main Success Scenario:

1. The store administrator selects the order.
2. The system shows a printable packaging slip.
9.4 EVENTS SPECIFICATION

**Event**

AddProductToShoppingCart

- **Event diagram**

- **Initial Integrity Constraints**

  context AddProductToShoppingCart::AttributesAreFromProduct(): Boolean
  body: self.product.attribute -> includesAll(self.attribute)

  context AddProductToShoppingCart::AttributesAreOfDifferentOptions(): Boolean
  body: self.attribute -> isUnique(option)

- **Effect**

  context AddProductToShoppingCart::effect()
  post ShoppingCartItemIsCreated:
  - sci.oclisNew and
  - sci.oclisTypeOf(ShoppingCartItem) and
  - sci.quantity = self.quantity and
  - sci.product = self.product and
  - sci.attribute = self.attribute and
  - if self.session.shoppingCart -> notEmpty() then
    - The session has a shopping cart
    - self.session.shoppingCart.shoppingCartItem -> includes(sci)
  - else
    - The session does not have a shopping cart
  - if self.session.customer -> isEmpty() then
    - The session is Anonymous
    - sc.oclisNew() and
    - sc.oclisTypeOf(AnonymousShoppingCart) and