* Use-Case: getting information.

1. **Actor:** guest-user \ registered-user.
2. **Precondition:**

There is at least one registered shop \ product.

Registered-user succeed Login use-case.

1. **Parameters:** the user accesses to the wanted shop \ product.
2. **Actions:**

1. The user approach the relevant store \ product for more information.

2. The system will present the relevant additional information of the product (depending on the type of product, from electrical product to food).

|  |  |  |
| --- | --- | --- |
| Use-Case | Parameter | Expected Output |
| Getting information. | The user has received additional information regarding the relevant product \ store. | Success |
|  | The user receive Irrelevant (not belong to the product) \ incomplete information. | Fail |

* Use-Case: search products.

1.**Actor:** guest-user / registered-user.

2. **Precondition:** none.

3.**Parameters:** Identifying information for the wanted product.

The user has the wanted shop \ product.

4.**Actions:**

1. The user gives the relevant information for the desired product.

2. The system returns the relevant products found.

|  |  |  |
| --- | --- | --- |
| Use-Case | Parameter | Expected Output |
| Search products. | The user has received products that match the preliminary information. | Success |
|  | * The user receives Irrelevant products to the given information. * Do not founds right product even its in the system. | Fail |

* Use-Case: saving products in a shopping Basket.

1.**Actor:** guest-user / registered-user.

2. **Precondition:** none.

3.**Parameters:**

The user has a product he wants to save.

4.**Actions:**

1. The user select the product.

2. The system check if the product available with the relevant shop, invoking Use-Case: check if the product available.

3. The system attaches the selected product to the user shopping Basket.

4.system updates stock of the product by invoking **Use-case** Change Item’s Detail. ???

|  |  |  |
| --- | --- | --- |
| Use-Case | Parameter | Expected Output |
| Saving products in a shopping Basket. | The user received the product into his shopping basket. | Success |
|  | * The user received the wrong product into his shopping basket. * The System attach the product to another user. * The system will not be able to attach the product to the shopping cart. | Fail |

* Use-Case: check if the product available.

1.**Actor:** system

2. **Precondition:** user activate Use-Case: saving products in a shopping Basket.

3.**Parameters:**

Quantity of product, product id, shop

4.**Actions:**

1. The system is approaches to the relevant store.

2. system check at the inventory that the quantity at the shop is ≥ quantity the user wants.

3. system notifies if quantity of the product is available.

|  |  |  |
| --- | --- | --- |
| Use-Case | Parameter | Expected Output |
| Check if the product available. | The system manages to get correct information for the required product. | Success |
|  | The system receives incorrect information for the required product. | Fail |

* Use-Case: editing the shopping cart

1.**Actor:** guest-user / registered-user.

2. **Precondition:** none.

3.**Parameters:**

The user has a product he wants to save.

4.**Actions:**

1. the user request the shopping basket.

2. He sends a request to the shopping basket to change the quantity of product.

3. the system operate the Use-Case: check if the product available of the new quantity.

4.shopping basket update the right quantity of the product according to the answer of the use-case.

|  |  |  |
| --- | --- | --- |
| Use-Case | Parameter | Expected Output |
| Editing the shopping cart. | * the product will not appear in the user cart. * appears with the new wanted quantity. * System do not edit the quantity due to shop disapproval. | Success |
|  | * the product will appear with quantity ‘0’. * wrong quantity / another product will be updated. | Fail |

* Use-Case: check-out (buying)

1.**Actor:** guest-user / registered-user / system.

2. **Precondition:** none.

3.**Parameters:**

The user has a product he wants to save.

4.**Actions:**

1.system check for each product in the shopping basket if its available by invoking **Use-Case**: check if the product available.

2. The system calculate the total amount of the products for the available products by invoking **Use-Case**: calculate the total amount of the products in the shopping basket.

3. The user transmits payment information.

4. The System divides the payments by stores (products for each shop).

5. The system activate the **Use-case** (payment service) for each shop.

6. according to the use case result the system sends the invoicing both the user and for the stores associated with the transaction.

7. The system cleans the user shopping cart.

|  |  |  |
| --- | --- | --- |
| Use-Case | Parameter | Expected Output |
| Check-out (buying) | The user manages to make a payment for the products kept in the shopping cart, each store receives an invoice for its part in the transaction. | Success |
|  | * The system is unable to make the payment * The stores do not receive notification of the transaction. * A payment is made that does not match the user's shopping cart. | Fail |

* Use-Case: calculate the total amount of the products in the shopping basket.

1.**Actor:** system.

2. **Precondition:** none.

3.**Parameters:**

The user has a product he wants to save.

4.**Actions: -** invoke this use-case for each product in the shopping basket.

1. system check available of the product by invoking Use-Case: check if the product available.

2. according to the answer of the use-case, calculate product discount by invoking Use-case: activate product discount.

3. – for each shop the system check shop discount policy by invoking Use-case: activate shop discount policy.

|  |  |  |
| --- | --- | --- |
| Use-Case | Parameter | Expected Output |
| Calculate the total amount of the products in the shopping basket. | The system calculates the correct price for the shopping cart under the existing constraints (checking quantities, discounts and more). | Success |
|  | * Incorrect scheme of product prices. | Fail |

* Use-case: activate product discount.

1.**Actor:** system.

2. **Precondition:** invoking at Use-Case: calculate the total amount of the products in the shopping basket.

3.**Parameters:** product to buy.

The user has a product he wants to buy.

4.**Actions:**

1. system is approaching the relevant shop.

2. system checking if there is a discount for the current product.

3. If there is a discount, the system returns the current price of the product.

|  |  |  |
| --- | --- | --- |
| Use-Case | Parameter | Expected Output |
| Activate product discount | The system updates the price of the product according to the existing discount. | Success |
|  | * The system does not update the price of the product in accordance with the existing discount * The system updates the price of the product not in accordance with the existing discount. | Fail |

* Use-case: activate shop discount policy.

1.**Actor:** system.

2. **Precondition:** invoking at Use-Case: calculate the total amount of the products in the shopping basket.

3.**Parameters:** product to buy.

The user has a product he wants to buy.

4.**Actions:**

1. system is approaching the relevant shop.

2. system checking the shop discount policy.

3. If there is a discount, the system returns the current price of the shop’s products.

|  |  |  |
| --- | --- | --- |
| Use-Case | Parameter | Expected Output |
| Activate shop discount policy. | The system updates the price of the product not in accordance with the store's discount policy. | Success |
|  | * The system does not update the price of the products in accordance with the existing discount * The system updates the price of the products not in accordance with the existing discount. | Fail |