**Use Cases**

**System use cases:**

**I.1:**

|  |  |
| --- | --- |
| Initialize system | **Title** |
| System Manager | **Actors** |
| None | **Preconditions** |
| None | **Postconditions** |
| External Services | **Parameters** |
| The system is initialized and loaded. | **Expected result** |
| * System manager initializes the system. | **Actions** |

**I.2:**

|  |  |
| --- | --- |
| Change/replace/insert a connection with external services. | **Title** |
| System manager | **Actors** |
| External services exists | **Preconditions** |
| None | **Postconditions** |
| The external service | **Parameters** |
| The changes admitted – the external service is connected to the system successfully . | **Expected result** |
| * The system manager connect change/replace/insert the external service. * The system approves the manager's action . * The system completes its activities as usual. | **Actions** |

**I.3:**

|  |  |
| --- | --- |
| Payment | **Title** |
| User | **Actors** |
| * Payment service exists. * The user is agree with the payment policy. | **Preconditions** |
| Backing to the previous state if the request denied. | **Postconditions** |
| The deals details | **Parameters** |
| The payment is Approved/denied | **Expected result** |
| * The user completes the payment process. * The system sends a payment request to the payment service * The payment service sends back the result with is an approval or deny. * The system sends to the user the result of the payment service . | **Actions** |

**I.4:**

|  |  |
| --- | --- |
| Supplying request | **Title** |
| 1. Store founder 2. Store owner 3. Store manager | **Actors** |
| 1. Supplier exists in the Supplying system. 2. Each actor must action under the permission that he has. | **Preconditions** |
| None | **Postconditions** |
| Costumer details and package details. | **Parameters** |
| The request approved/denied | **Expected result** |
| * the actor puts the details of the package in the system. * The system sends a supplying request to the external supplying system. * The Supplying service sends back the result of the request which is approval or deny. * The system sends to the actor the result back from the Supplying service. | **Actions** |

**I.5:**

|  |  |
| --- | --- |
| Real time notification | **Title** |
| Market | **Actors** |
| * The user exists in the system. * The user is member. * The user is active. | **Preconditions** |
| None | **Postconditions** |
| Message details and user details | **Parameters** |
| The user gets the message successfully at real time | **Expected result** |
| * An event occurs that leads to a member to get notification about it during his connection. * Checking if the user is connected. * Sends the message to the user. | **Actions** |

**I.6:**

|  |  |
| --- | --- |
| Pending notification | **Title** |
| Market | **Actors** |
| * The user exists in the system. * The user is member. * The user is offline | **Preconditions** |
| None | **Postconditions** |
| Message details and user details | **Parameters** |
| The user gets the message successfully at the next first time he logged in to the system. | **Expected result** |
| * An event occurs that leads to a member to get notification about it. * Checking that the user is offline. * Save the notification that the user should get until he connects. * The user connects and gets the message succefuly. | **Actions** |

**User use cases:**

**II.1.1:**

|  |  |
| --- | --- |
| Entering to the system | **Title** |
| User | **Actors** |
| None | **Preconditions** |
| The user gets an empty Cart | **Postconditions** |
| None | **Parameters** |
| * The user gets in to the system successfully. * The user get an empty cart that he could buy what ever he wants an available. | **Expected result** |
| * The user opens the system. * The system gives the user a cart. * The system shows the user a page to activate like a gust. | **Actions** |

**II.1.2:**

|  |  |
| --- | --- |
| Exiting from the system | **Title** |
| User | **Actors** |
| None | **Preconditions** |
| * If the user is gust: the system removes his cart * If the user is member: the system saves his cart. | **Postconditions** |
| None | **Parameters** |
| The user is no longer activated in the system and the "post conditions" occurred successfully. | **Expected result** |
| * The user quit the system. * If the user is guest: the system clear his cart and removes it. * If the user is member: the system saves his cart state. | **Actions** |

**II.1.3:**

|  |  |
| --- | --- |
| Registration to the system | **Title** |
| Guest | **Actors** |
| The details are not exists in the system | **Preconditions** |
| The system adds the user to the member list | **Postconditions** |
| User details | **Parameters** |
| The user is added to the system as a member | **Expected result** |
| * The gust asks to register to the system. * The system asks for a unique details. * The guest enters the details. * The system adds the guest as a member to the system. | **Actions** |

**II.1.4:**

|  |  |
| --- | --- |
| logging in to the system | **Title** |
| User | **Actors** |
| The user exists in the system | **Preconditions** |
| The user logged in | **Postconditions** |
| User details | **Parameters** |
| The user logged in successfully | **Expected result** |
| * The user asks to log in. * The system asks the user to insert his details. * The user enters the details. * The system check if the details are correct, if they are correct the user enters to the system as a member. * If the details are wrong the system shows a message that the user with this specific details is not exist. | **Actions** |

**II.2.1:**

|  |  |
| --- | --- |
| Information about stores and products | **Title** |
| User | **Actors** |
| Store exist | **Preconditions** |
| None | **Postconditions** |
| * Store id/name * Product id/name | **Parameters** |
| The system display the relevant asked info . | **Expected result** |
| * The user asks about a store/product. * The system display the relevant data to the user. | **Actions** |

**II.2.2:**

|  |  |
| --- | --- |
| Search for product | **Title** |
| User | **Actors** |
| None | **Preconditions** |
| None | **Postconditions** |
| One of the following at least:   * Product name. * Product category. * Key words. | **Parameters** |
| Displaying a relevant products | **Expected result** |
| * The user search for a product. * The system shows a relevant products. * The user could filter the result according to different features. | **Actions** |

**II.2.3:**

|  |  |
| --- | --- |
| Saving the products into the bag | **Title** |
| User | **Actors** |
| Products are added to the bag. | **Preconditions** |
| The system remembers the product in the saved bag for some store | **Postconditions** |
| Products | **Parameters** |
| The products for a store are saved in the bag. | **Expected result** |
| * The user choose the products that he wants to save in the store bag. * The system saves these product in the bag. | **Actions** |

**II.2.4:**

|  |  |
| --- | --- |
| Cart checking and changing | **Title** |
| User | **Actors** |
| None | **Preconditions** |
| None | **Postconditions** |
| Cart | **Parameters** |
| The changes are saved in the system | **Expected result** |
| * The user asks for his cart. * The system display for the user the cart. * The user can make changes in the cart. * If the user makes change – the system save the changes that the user did. | **Actions** |

**II.2.5:**

|  |  |
| --- | --- |
| Cart purchase | **Title** |
| User | **Actors** |
| None Empty Cart | **Preconditions** |
| Relevant message from the system the contain approval or denying the purchasing process. | **Postconditions** |
| Cart and payment details | **Parameters** |
| The purchase done successfully and saved in the purchasing history | **Expected result** |
| * The user asks to complete his purchase. * The system checks the validity of the payment details . * The system checks that each product is within the stock at the purchasing moment. * Purchase is completed and saved via the system in the history. * The user gets a relevant message about the purchase. | **Actions** |

**II.3.1:**

|  |  |
| --- | --- |
| Logout | **Title** |
| Member | **Actors** |
| Member is logged in | **Preconditions** |
| The cart is saved | **Postconditions** |
| None | **Parameters** |
| The user state changed to guest | **Expected result** |
| * The member asks to logout * The system removes the user from the "activated" users. * The system changes the user to guest state. | **Actions** |

**II.3.2:**

|  |  |
| --- | --- |
| Store opening | **Title** |
| Member | **Actors** |
| Unique details of the new store | **Preconditions** |
| * Relevant message to the user either if it is successful opening or fail opening. * The store appears for each user so he could search for it. * The member gets "founder" role for this store | **Postconditions** |
| Store details | **Parameters** |
| The store is added to the system successfully and the member assigned as a founder for this store | **Expected result** |
| * The user asks to open a store. * The system asks the user to enter store details. * The user enters the details. * The system verify that the details are unique. * The system adds the store and sends a message to the user that the store added successfully. * The system assign the member as a "founder" for the store. | **Actions** |

**Store owner use cases:**

**II.4.1.1:**

|  |  |
| --- | --- |
| Adding product to store's stock | **Title** |
| 1. Store owner 2. Store manager (with relevant permission) | **Actors** |
| Product does not exist in the stock | **Preconditions** |
| Relevant message to the store owner either if it is successful adding or fail adding. | **Postconditions** |
| Product details  Store details | **Parameters** |
| The product is added to the stock | **Expected result** |
| * Store owner accesses his own store. * Store owner requests to add new product to the stock and enters the product details. * The system adds the product to the stock if does not exist, otherwise, system sends error message. | **Actions** |

**II.4.1.2:**

|  |  |
| --- | --- |
| Deleting product from store's stock | **Title** |
| 1. Store owner 2. Store manager (with relevant permission) | **Actors** |
| Product exists in the stock | **Preconditions** |
| Relevant message to the store owner either if it is successful deleting or fail deleting. | **Postconditions** |
| Product details  Store details | **Parameters** |
| The product is deleted from the stock | **Expected result** |
| * Store owner accesses his own store. * Store owner requests to delete a product from the stock and enters the product details. * The system searches and deletes the product from the stock if exists, otherwise, system sends error message. | **Actions** |

**II.4.1.3:**

|  |  |
| --- | --- |
| Change product details in store's stock | **Title** |
| 1. Store owner 2. Store manager (with relevant permission) | **Actors** |
| Product exists in the stock | **Preconditions** |
| Relevant message to the store owner either if it is successful changing or fail changing. | **Postconditions** |
| New product details  Store details | **Parameters** |
| The new product details are saved | **Expected result** |
| * Store owner accesses his own store. * Store owner requests to change a product details from the stock and enters the product details. * The system searches and changes the product details from the stock if exists, otherwise, system sends error message. | **Actions** |

**II.4.2.1:**

|  |  |
| --- | --- |
| Change payment policy | **Title** |
| 1. Store owner 2. Store manager (with relevant permission) | **Actors** |
| The owner owns this store. | **Preconditions** |
| Payment policy is changed if they do no conflict with the consistency constraints were defined by the founder, otherwise system sends error message. | **Postconditions** |
| New payment policy details  Store details | **Parameters** |
| The new payment policy is saved | **Expected result** |
| * Store owner accesses his own store. * Store owner requests to change the payment policy. * Store owner enters the new payment policy details. * The system changes the payment policy if they are valid, otherwise, system sends error message. | **Actions** |

**II.4.2.2:**

|  |  |
| --- | --- |
| Change discount policy | **Title** |
| 1. Store owner 2. Store manager (with relevant permission) | **Actors** |
| Product exists  The owner owns this store | **Preconditions** |
| Discount policy is updated if they do no conflict with the consistency constraints were defined by the founder, otherwise system sends error message. | **Postconditions** |
| Product details  New discount policy  Store details | **Parameters** |
| The new discount policy is saved | **Expected result** |
| * Store owner accesses his own store. * Store owner requests to update the discount policy for the product. * Store owner enters the updated discount policy details. * The system updates the discount policy for the product if they are valid, otherwise, system sends error message. | **Actions** |

**II.4.4:**

|  |  |
| --- | --- |
| Store owner appointment | **Title** |
| Store owner | **Actors** |
| He is a member (not a guest).  He is not a founder or manager or owner of this store | **Preconditions** |
| The new owner appointment is saved in the system if it was done successfully, otherwise system sends error message. | **Postconditions** |
| Member details  Store details | **Parameters** |
| The member has become a store owner | **Expected result** |
| * Store owner accesses his own store. * Store owner requests to add a new owner. * Store owner enters the member details. * The system defines the member as an owner if he is not a founder or manager or owner of the specified store, otherwise, system sends error message. | **Actions** |

**II.4.6:**

|  |  |
| --- | --- |
| Store manager appointment | **Title** |
| 1. Store owner 2. Store manager (with relevant permission) | **Actors** |
| He is a member (not a guest).  He is not a founder or manager or owner of this store. | **Preconditions** |
| The new manager appointment is saved in the system if it was done successfully, otherwise system sends error message.  The new manager only has permissions to receive information – use cases 4.12, 4.13 | **Postconditions** |
| Member details  Store details | **Parameters** |
| The member has become a store manager | **Expected result** |
| * Store owner accesses his own store. * Store owner requests to add a new manager. * Store owner enters the member details. * The system defines the member as a manager if he is not a founder or manager or owner of the specified store, otherwise, system sends error message. | **Actions** |

**II.4.7:**

|  |  |
| --- | --- |
| Change store manager permissions | **Title** |
| 1. Store owner 2. Store manager (with relevant permission) | **Actors** |
| The owner appointed the specified manager | **Preconditions** |
| The new permissions are saved in the system if they are valid, otherwise system sends error message. | **Postconditions** |
| Manager details  Store details  New permissions | **Parameters** |
| The permissions are changed successfully. | **Expected result** |
| * Store owner accesses his own store. * Store owner requests to change the permissions of a specified manager. * Store owner enters the member details and the new permissions. * The system searches for the manager and changes the permissions, otherwise, system sends error message. | **Actions** |

**II.4.9:**

|  |  |
| --- | --- |
| Closing store | **Title** |
| Store founder (he is also a store owner) | **Actors** |
| The founder founded this store. | **Preconditions** |
| * The store becomes inactive (closed). * The member who is not an owner or system manager can't receive information about this closed store. * Store owners and managers get notified but they still owners and managers. * The store products will not appear in the search. | **Postconditions** |
| Store details | **Parameters** |
| The Store is closed. | **Expected result** |
| * Store founder accesses his own store. * Store founder requests to close the store. * The system saves the roles. * The system defines the store as an inactive (closed). * The system notifies the owners and managers of the store. * The system does not show the products in search. | **Actions** |

**II.4.11:**

|  |  |
| --- | --- |
| Request roles' information | **Title** |
| 1. Store owner 2. Store manager (with relevant permission) | **Actors** |
| The owner owns this store  The member is an owner or manager of this store. | **Preconditions** |
| None | **Postconditions** |
| Store details  Member details | **Parameters** |
| The system will give the owner the information about the member, if the member is a manager, the system will also give him his permissions. | **Expected result** |
| * Store owner accesses his own store. * Store owner requests information about a member (owner or manager) * Store owner enters the member details. * The system searches for the member, if he is an owner or manager of this store, the system will show the information about him and if he is a manager, the system will also show his permissions, otherwise, system sends error message. | **Actions** |

**II.4.13 + II.6.4:**

|  |  |
| --- | --- |
| Request store purchase history | **Title** |
| 1. Store owner 2. Store manager (with relevant permission) 3. System manager | **Actors** |
| The store owner owns this store / the store manager manages this store. | **Preconditions** |
| None | **Postconditions** |
| Store details | **Parameters** |
| The system will show the purchase history | **Expected result** |
| * The actor accesses the store. * The actor requests the store purchase history. * The system will show the purchase history to the actor, otherwise, system sends error message. | **Actions** |