

Use case 1 – Add simple purchase policy to store

Use Case Description	
Name	Add simple purchase policy to store
Actor	Client
Precondition	<ol style="list-style-type: none"> 1. The client should be a store owner or store manager and have the permission to change the store policies 2. The client should be registered in the system 3. the store should be existed in the system 4. the system should be active
Parameters	quantity, price, date, at least/at most, weight, age, purchase type (category,user,product,shopping cart), registered user that has the permission, store , purchase method (immediate or offer), the logical purchase relation (and, or, if then)
Postconditions	<ol style="list-style-type: none"> 1. the simple purchase policy added to the store. 2. Purchase policy applied to the products in the time of purchase
Actions	<ol style="list-style-type: none"> 1. the user register to the system 2. the user log in and initialize store or another user add the user to his store and give him the permission to edit store policies without the need to login in 3. the user log in and add simple purchase policy to the relevant store 4. the simple purchase policy added to the store

Acceptance tests 1

Action	Data	Expected result
Add simple purchase policy to store – positive	The user is registered and have the permission to edit store policies with correct inputs and store is existed and the system is active	Success: the purchase policy added to the store
Add simple purchase policy to store – negative	The user is registered and have the permission to edit store policies with correct inputs but not for the store in the input	Failure: the system returns an exception because he is not employed at this store
Add simple purchase policy to store – alternative	The user is registered but don't have the permission to edit	Failure: the system returns an error and disapprove to edit the policy of the store because the

	store policies and store is existed and the system is active	user doesn't have the permission to edit this store policies
--	--	--

Use case 2 – Add complex logical purchase policy to store

Use Case Description	
Name	Add complex logical purchase policy to store
Actor	Client
Precondition	<ol style="list-style-type: none"> 1. The client should be a store owner or store manager and have the permission to change the store policies 2. The client should be registered in the system 3. the store should be existed in the system 4. the system should be active
Parameters	registered user that has the permission, store, logical purchase type (and, or, if then), existed logical purchase relation (and, or, if then)
Postconditions	<ol style="list-style-type: none"> 1. the complex logical purchase policy added to the store. 2. Purchase policy applied to the products in the time of purchase
Actions	<ol style="list-style-type: none"> 1. the user register to the system 2. the user log in and initialize store or another user add the user to his store and give him the permission to edit store policies without the need to login in 3. the user log in and add logical complex purchase policy to the relevant store 4. the simple purchase policy added to the store

Acceptance tests 2

Action	Data	Expected result
Add complex logical purchase policy to store – positive	The user is registered and have the permission to edit store policies with correct inputs and store is existed and the system is active	Success: the purchase policy added to the store
Add complex logical purchase policy to store – negative	The user is registered and have the permission to edit store policies with correct inputs but not for the store in the input	Failure: the system returns an exception because he is not employed at this store

Add complex logical purchase policy to store – alternative	The user is registered but don't have the permission to edit store policies and store is existed and the system is active	Failure: the system returns an error and disapprove to edit the policy of the store because the user doesn't have the permission to edit this store policies
--	---	--

Use case 3 – Add simple discount policy to store

Use Case Description	
Name	Add simple discount policy to store
Actor	Client
Precondition	<ol style="list-style-type: none"> 5. The client should be a store owner or store manager and have the permission to change the store policies 6. The client should be registered in the system 7. the store should be existed in the system 8. the system should be active
Parameters	Discount type (conditional/standard), conditionalPrice (if condidional), conditionalQuantity(if condidional), discountPercentage , store , registered user that has the permission , the logical/numerical purchase relation (and,xor, or, if then,add,at most),policy type (category,product,store)
Postconditions	<ol style="list-style-type: none"> 1. the simple discount policy added to the store. 2. Discount policy applied to the products in the time of purchase
Actions	<ol style="list-style-type: none"> 1. the user register to the system 2. the user log in and initialize store or another user add the user to his store and give him the permission to edit store policies without the need to login in 3. the user log in and add simple discount policy to the relevant store 4. the simple discount policy added to the store

Acceptance tests 3

Action	Data	Expected result
--------	------	-----------------

Add simple discount policy to store – positive	The user is registered and have the permission to edit store policies with correct inputs and store is existed and the system is active	Success: the discount policy added to the store
Add simple discount policy to store – negative	The user is registered and have the permission to edit store policies with correct inputs but not for the store in the input	Failure: the system returns an exception because he is not employed at this store
Add simple discount policy to store – alternative	The user is registered but don't have the permission to edit store policies and store is existed and the system is active	Failure: the system returns an error and disapprove to edit the policy of the store because the user doesn't have the permission to edit this store policies

Use case 4 – Add complex logical discount policy to store

Use Case Description	
Name	Add complex logical discount policy to store
Actor	Client
Precondition	<ol style="list-style-type: none"> 1. The client should be a store owner or store manager and have the permission to change the store policies 2. The client should be registered in the system 3. the store should be existed in the system 4. the system should be active
Parameters	registered user that has the permission, store, logical discount type (and,xor, or, if then,add,at most), existed logical discount relation (and,xor, or, if then,add,at most)
Postconditions	<ol style="list-style-type: none"> 1. the complex logical discount policy added to the store. 2. Discount policy applied to the products in the time of purchase
Actions	<ol style="list-style-type: none"> 1. the user register to the system 2. the user log in and initialize store or another user add the user to his store and give him the permission to edit store policies without the need to login in 3. the user log in and add logical complex discount policy to the relevant store 4. the simple discount policy added to the store

Acceptance tests 4

Action	Data	Expected result
Add complex logical discount policy to store – positive	The user is registered and have the permission to edit store policies with correct inputs and store is existed and the system is active	Success: the discount policy added to the store
Add complex logical discount policy to store – negative	The user is registered and have the permission to edit store policies with correct inputs but not for the store in the input	Failure: the system returns an exception because he is not employed at this store
Add complex logical discount policy to store – alternative	The user is registered but don't have the permission to edit store policies and store is existed and the system is active	Failure: the system returns an error and disapprove to edit the policy of the store because the user doesn't have the permission to edit this store policies

Use case 5 – accepted client's offer

Use Case Description	
Name	accepted client's offer
Actor	Clients
Precondition	<ol style="list-style-type: none"> 1. The client should add from the relevant store some products to his shopping cart 2. The products should be existed in the store (quantity>0) 3. The client should be registered in the system 4. the store should be existed in the system 5. the system should be active
Parameters	User that has the product in his shopping cart, product's store, the product, product's price, offer price
Postconditions	<ol style="list-style-type: none"> 1. the offer is accepted 2. the price of the product changes just in the user shopping cart 3. update shopping cart total price for this user
Actions	<ol style="list-style-type: none"> 1. the user register to the system 2. the user log in to the system 3. the user adds the product from store to his shopping cart 4. the user send an offer price to the store

	<ol style="list-style-type: none"> 5. all assigned store owners review the offer and accept it (don't calculate resigned owners/managers) 6. the price of the product changes in the user's shopping cart 7. the user gets the response from the store (accepted)
--	--

Acceptance tests 5

Action	Data	Expected result
accepted client's offer – positive	The user is registered and has the product in his shopping cart, the store is existed and has the product in his inventory	Success: the offer accepted and the price of the product is changed in user's shopping cart
accepted client's offer – negative	The user is registered but don't have the product in his shopping cart, the store is existed and has the product in his inventory	Failure: the system returns an exception because the user sends offer for a product that he don't have in his shopping cart
accepted client's offer – alternative	The user is registered and has the product in his shopping cart, the store is existed but have the product in his inventory	Failure: the system returns an exception because the user sends offer for a product that the store don't have in his inventory

Use case 6 – unaccepted client's offer

Use Case Description	
Name	unaccepted client's offer
Actor	Clients
Precondition	<ol style="list-style-type: none"> 1. The client should add from the relevant store some products to his shopping cart 2. The products should be existed in the store (quantity>0) 3. The client should be registered in the system 4. the store should be existed in the system 5. the system should be active
Parameters	User that has the product in his shopping cart, product's store, the product, product's price, offer price
Postconditions	<ol style="list-style-type: none"> 1. the offer is unaccepted 2. the price of the product doesn't change in user's shopping cart
Actions	<ol style="list-style-type: none"> 1. the user register to the system

	<ol style="list-style-type: none"> 2. the user logs in to the system 3. the user adds the product from store to his shopping cart 4. the user sends an offer price to the store 5. all store owners review the offer and accept it except at least one of them (don't calculate resigned owners/managers) 6. the price of the product doesn't change in the user's shopping cart 7. the user gets the response from the store (unaccepted)
--	--

Acceptance tests 6

Action	Data	Expected result
unaccepted client's offer – positive	The user is registered and has the product in his shopping cart, the store is existed and has the product in his inventory	Success: the offer unaccepted, the price of the product didn't change in user's shopping cart
unaccepted client's offer – negative	The user is registered but don't have the product in his shopping cart, the store is existed and has the product in his inventory	Failure: the system returns an exception because the user sends offer for a product that he doesn't have in his shopping cart
unaccepted client's offer – alternative	The user is registered and has the product in his shopping cart, the store is existed but have the product in his inventory	Failure: the system returns an exception because the user sends offer for a product that the store doesn't have in his inventory

Use case 7 – unaccepted client's offer and get new offer price from the store

Use Case Description	
Name	unaccepted client's offer and get new offer price from the store
Actor	Clients
Precondition	<ol style="list-style-type: none"> 1. The client should add from the relevant store some products to his shopping cart 2. The products should be existed in the store (quantity>0) 3. The client should be registered in the system 4. the store should be existed in the system 5. the system should be active
Parameters	User that has the product in his shopping cart, product's store, the product, product's price, offer price
Postconditions	<ol style="list-style-type: none"> 1. the offer is unaccepted 2. the price of the product doesn't change in user's shopping cart 3. the user gets new offer price for the product from the store and review it, after that he accept/unaccept it
Actions	<ol style="list-style-type: none"> 1. the user register to the system 2. the user logs in to the system 3. the user adds the product from store to his shopping cart 4. the user sends an offer price to the store 5. all store owners review the offer and accept it except at least one of them 6. they send to the user new offer price 7. the price of the product doesn't change in the user's shopping cart 8. the user gets the response from the store (unaccepted) 9. he also gets the offer price from the store 10. the user reviews the offer 11. if he accepted it -> the price of the product in his shopping cart will change to the offer price from the store 12. else -> the price of the product in his shopping cart will not change to the offer price from the store

Acceptance tests 6

Action	Data	Expected result
unaccepted client's offer and get new offer price from the store – positive	The user is registered and has the product in his shopping cart, the store is existed and has the product in his inventory	Success: the offer unaccepted, the price of the product didn't change in user's shopping cart, the user receives new offer price, he reviews it and accept/unaccept the price
unaccepted client's offer and get new offer price from the store – negative	The user is registered but don't have the product in his shopping cart, the store is existed and has the product in his inventory	Failure: the system returns an exception because the user sends offer for a product that he doesn't have in his shopping cart
unaccepted client's offer and get new offer price from the store –alternative	The user is registered and has the product in his shopping cart, the store is existed but have the product in his inventory	Failure: the system returns an exception because the user sends offer for a product that the store doesn't have in his inventory