

## Contents

Page 1 (Req 1.1 – 1.2)	2
Page 2 (Req 1.3 – 2.1)	3
Page 3 (Req 2.2 – 2.4)	4
Page 4 (Req 2.4 – 2.5)	5
Page 5 (Req 2.5)	6
Page 6 (Req 2.5)	7
Page 7 (Req 3.1 – 3.3)	8
Page 8 (Req 3.4 – 3.6)	9
Page 9 (Req 3.7 – 3.8)	10
Page 10 (Req 3.9)	11
Page 11 (Req 4.1)	12
Page 12 (Req 4.2)	13
Page 13 (Req 4.2)	14
Page 14 (Req 4.2)	15
Page 15 (Req 4.2)	16
Page 16 (Req 4.2)	17
Page 17 (Req 4.2)	18
Page 18 (Req 4.4)	19
Page 19 (Req 4.4 – 4.5)	20
Page 20 (Req 4.5 – 4.6)	21
Page 21 (Req 4.7 – 4.8)	22
Page 22 (Req 4.9 – 4.10)	23
Page 23 (Req 4.11 – 4.12)	24
Page 24 (Req 4.12 – 4.13)	25
Page 25 (Req 6.1 – 6.2)	26
Page 26 (Req 6.3)	27
Page 27 (Req 6.4)	28
Page 28 (Req 6.5)	29

<b>Use Case Name</b>	Starting up the system
<b>Requirement Number</b>	1.1
<b>Actors</b>	User
<b>Parameters</b>	None
<b>Preconditions</b>	1. The machine has the latest version of the system 2. The machine has a stable connection
<b>Actions</b>	The User runs the application
<b>Result</b>	The User is appointed Guest and receives an empty Shopping cart
<b>Data</b>	<b>Result</b>
The User has no connection to the server	Fail
The User has an older version of the client	Fail

<b>Use Case Name</b>	Starting up the system – First Boot
<b>Requirement Number</b>	1.1
<b>Actors</b>	User
<b>Parameters</b>	None
<b>Preconditions</b>	1. The machine has the latest version of the system 2. The machine has a stable connection 3. First boot of the system
<b>Actions</b>	1. The User runs the application. 2. The User enters the default admin username and password
<b>Result</b>	Home page is shown and The User is appointed Guest and receives an empty Shopping cart.
<b>Data</b>	<b>Result</b>
The User has no connection to the server	Fail
The User has an older version of the client	Fail

<b>Use Case Name</b>	Exiting the system
<b>Requirement Number</b>	1.2
<b>Actors</b>	User
<b>Parameters</b>	None
<b>Preconditions</b>	None
<b>Actions</b>	1. The User closes the client application 2. If the User is logged in, the System runs logging out use case
<b>Result</b>	None
<b>Data</b>	<b>Result</b>
A Member is exiting the System, his Shopping cart is reloaded	Success
A Guest is exiting the System, his Shopping cart is discarded	Success

<b>Use Case Name</b>	Registering
<b>Requirement Number</b>	1.3
<b>Actors</b>	Guest
<b>Parameters</b>	<i>Username, Password</i>
<b>Preconditions</b>	The <i>Username</i> is not already known in the System
<b>Actions</b>	<ol style="list-style-type: none"> <li>1. The User requests to register with some <i>Username</i> and <i>Password</i></li> <li>2. The System Verifies the <i>Password</i>'s security level and that the <i>Username</i> does not already exist</li> <li>3. If one of the previous verifications fails, the User is not registered and the System notifies the User</li> <li>4. Else, a new Member is registered in the System</li> </ol>
<b>Result</b>	A new member can now be logged in to in the System
<b>Data</b>	<b>Result</b>
The User enters a Username that's already in use	Fail
The User is trying to use an insecure password	Fail
The User is using a unique username with a secure password	Success

<b>Use Case Name</b>	Login
<b>Requirement Number</b>	1.4
<b>Actors</b>	Guest
<b>Parameters</b>	<i>Username, Password</i>
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. The requested user was registered in the system beforehand</li> <li>2. The system has connection</li> </ol>
<b>Actions</b>	<ol style="list-style-type: none"> <li>1. Guest inputs <i>Username</i> and <i>Password</i></li> <li>2. If the Guest is already logged in the process is stopped and an error message appears</li> <li>3. The System verifies the <i>Username, Password</i> combination</li> <li>4. If the System doesn't recognize the combination the Guest is notified with the appropriate error message</li> <li>5. The System loads the Member's saved Shopping Cart and notifications</li> </ol>
<b>Result</b>	The Guest receives logged in status for the requested Member, and the Shopping Cart receives earlier saved Products
<b>Data</b>	<b>Result</b>
The Username is unknown to the system	Fail
The password is incorrect	Fail
The username password combination is correct	Success

<b>Use Case Name</b>	Searching for a Store
<b>Requirement Number</b>	2.1
<b>Actors</b>	User
<b>Parameters</b>	<i>Store_name</i>
<b>Preconditions</b>	None
<b>Actions</b>	<ol style="list-style-type: none"> <li>1. User chooses "Search Store" option from the menu</li> <li>2. User enters the name of the store he is searching for</li> <li>3. The System presents the store with the name the user wanted</li> </ol>
<b>Result</b>	The System presents list of products (and prices) in the store with an "add to cart" button under them
<b>Data</b>	<b>Result</b>
The User searches for a store that doesn't exist	Success
The User searches for a specific known store	Success

<b>Use Case Name</b>	Searching for a Product
<b>Requirement Number</b>	2.2
<b>Actors</b>	User
<b>Parameters</b>	<i>Product_attributes</i>
<b>Preconditions</b>	None
<b>Actions</b>	1. User chooses "Search Products" option from the menu 2. User requests a Product with specific attributes from the System 3. The System presents all Products from any Store, which apply to those attributes
<b>Result</b>	The system presents list of products (with prices), a quantity input box, an "add to cart" button and "place a bid" button under the list
<b>Data</b>	<b>Result</b>
The User is searching for a product that doesn't exist	Success
The User is searching for a product that exists	Success

<b>Use Case Name</b>	Adding a Product to the Shopping Cart
<b>Requirement Number</b>	2.3
<b>Actors</b>	User
<b>Parameters</b>	<i>Product</i>
<b>Preconditions</b>	1. The Product is sold by some Store 2. The User has a Shopping Cart
<b>Actions</b>	1. The User finds a Product (either from use case 2.1 or 2.2) 2. The User chooses a product 3. The User presses the "add to cart" button with the product chosen
<b>Result</b>	The User's Shopping Cart receives the Product from the appropriate store
<b>Data</b>	<b>Result</b>
The User is adding more of the product than the store has to offer	Fail
The User is adding a product from a Store	Success
The User is adding a product found from searching a product	Success

<b>Use Case Name</b>	Inspecting Shopping Cart
<b>Requirement Number</b>	2.4
<b>Actors</b>	User
<b>Parameters</b>	<i>Username</i>
<b>Preconditions</b>	None
<b>Actions</b>	1. User requests for his Shopping Cart 2. The System presents the Shopping Cart
<b>Result</b>	A Shopping Cart is shown to the user with all the products in it
<b>Data</b>	<b>Result</b>
The User is inspecting his shopping cart when it has no Products	Success
The User is inspecting his shopping cart when it has 1 or more products	Success

<b>Use Case Name</b>	Removing a Product from the Shopping Cart	
<b>Requirement Number</b>	2.4	
<b>Actors</b>	User	
<b>Parameters</b>	<i>Product, Shopping_Cart</i>	
<b>Preconditions</b>	<i>Shopping_Cart</i> has at least 1 <i>Product</i>	
<b>Actions</b>	1. The User inspects his shopping cart 2. The User chooses “remove <product name> from cart” button for a specific product in his cart	
<b>Result</b>	<i>Product is removed from shopping cart</i>	
<b>Data</b>		<b>Result</b>
The User is attempting to remove a product from an empty cart		Fail
The User is attempting to remove a product that exists in his shopping cart		Success

<b>Use Case Name</b>	Purchasing a Shopping Cart, with verified delivery and payment	
<b>Requirement Number</b>	2.5	
<b>Actors</b>	User	
<b>Parameters</b>	<i>Shopping_Cart, Payment, Supplyment</i>	
<b>Preconditions</b>	1. The <i>Shopping_Cart</i> is not empty 2. For every <i>Product</i> in the <i>Shopping_Cart</i> the Store which sells it has the amount of that <i>Product</i> that is listed in the <i>Shopping_Cart</i> 3. The <i>Supplyment</i> and <i>Payment</i> succeeded	
<b>Actions</b>	1. The User requests from the System to buy his shopping cart with a specific <i>Payment</i> type and <i>Supplier</i> 2. The System verifies the <i>Payment</i> 3. The System verifies the <i>Supplyment</i> 4. The System triggers the payment and supply, updates the Store with the amounts and triggers the purchase event on the notification bus	
<b>Result</b>	The Purchase is completed, the relevant Members are notified and the relevant amounts on the Stores are updated	
<b>Data</b>		<b>Result</b>
The User is attempting to buy an empty shopping cart		Fail
The User is attempting to buy a shopping cart without selecting payment type or supplier		Fail
The User is attempting to buy a shopping cart with at least 1 product		Success

<b>Use Case Name</b>	Purchasing a Shopping Cart – cart validation	
<b>Requirement Number</b>	2.5	
<b>Actors</b>	User	
<b>Parameters</b>	<i>ShoppingCart, PaymentInformation, SupplyingInformation</i>	
<b>Preconditions</b>	-	
<b>Actions</b>	1. The User requests from the System to buy his shopping cart with his Payment information and Supplying information 2. The System validates the User's current cart 3. If the validation fails, the User is notified and the purchase process stops without any changes in the System, if it succeeds the purchase process continues ("External Systems" use cases)	
<b>Result</b>	The system will continue with the purchase process ("Purchasing a Shopping Cart – External Payment System" use case)	
<b>Data</b>		<b>Result</b>
The User is attempting to buy a product from a closed store		Fail
The User is attempting to buy an empty cart		Fail
The User is attempting to buy a quantity larger than what the store has to offer		Fail
The User is attempting to buy a product when he isn't allowed by the purchase policy		Fail
None of the above		Success

<b>Use Case Name</b>	Purchasing a Shopping Cart – External Payment System	
<b>Requirement Number</b>	2.5	
<b>Actors</b>	User, Payment System	
<b>Parameters</b>	Purchasing a Shopping Cart – cart validation use case	
<b>Preconditions</b>	1. Purchasing a Shopping Cart – cart validation use case (Success)	
<b>Actions</b>	1. The System calculates the Cart's price and asks the External Payment System to make Payment 2. If the payment operation fails, the purchase process stops and the user is notified. If the payment succeeds the purchase process continues ("Purchasing a Shopping Cart – External Supplyment System" use case)	
<b>Result</b>	The System will continue the purchase process	
<b>Data</b>		<b>Result</b>
The Payment system fails to make payment		Fail
The System has lost connection to the payment system		Fail
None of the above		Success

<b>Use Case Name</b>	Purchasing a Shopping Cart – External Supplyment System	
<b>Requirement Number</b>	2.5	
<b>Actors</b>	User, Supplyment System	
<b>Parameters</b>	Purchasing a Shopping Cart – External Payment System use case	
<b>Preconditions</b>	Purchasing a Shopping Cart – External Payment System use case (Success)	
<b>Actions</b>	1. The System checks which products from the cart should be delivered and uses the External Supplying system to book the supplyment with the User's input information 2. If the booking operation fails, the system performs a roll back on the User's payment, stops the purchase process and notifies the User. If the operation succeeds the purchase process continues ("Purchasing a Shopping Cart – Updating the Market" use case)	
<b>Result</b>	The System will continue the purchase process	
<b>Data</b>		<b>Result</b>
The Supplyment system fails to book		Fail
The System has lost connection to the Supplyment system		Fail
None of the above		Success

<b>Use Case Name</b>	Purchasing a Shopping Cart – Updating the Market	
<b>Requirement Number</b>	2.5	
<b>Actors</b>	User	
<b>Parameters</b>	Purchasing a Shopping Cart – External Supplyment System use case	
<b>Preconditions</b>	2. Purchasing a Shopping Cart – External Supplyment System use case (Success)	
<b>Actions</b>	1. The System Updates each store and product with new quantities. If it fails to update (quantities go below 0, product is removed from the store...), The System performs a rollback on the payment system and aborts the supply booking, stops the purchase process and notifies the User 2. The System saves the Shopping Baskets in the appropriate store's history 3. The System notifies relevant store staff of the purchase's success 4. The System reset's the User's cart	
<b>Result</b>	The User completed a purchase	
<b>Data</b>		<b>Result</b>
A Product's quantity goes below 0		Fail
A Product is not found in a store (got removed during earlier stages of the process)		Fail
The update process succeeds		Success

<b>Use Case Name</b>	Logging out	
<b>Requirement Number</b>	3.1	
<b>Actors</b>	Member	
<b>Parameters</b>	<i>Member</i>	
<b>Preconditions</b>	The <i>Member</i> is logged in	
<b>Actions</b>	1. The Member requests to log out from the System 2. The System saves his / her current Shopping Cart content 3. The System changes the Member's status to Guest	
<b>Result</b>	The Member becomes a Guest and the Shopping Cart is saved	
<b>Data</b>		<b>Result</b>
A Guest is attempting to log out		Fail
The Member is logging out with products in his cart		Success
The Member is logging out with no products in his cart		Success

<b>Use Case Name</b>	Opening a Store	
<b>Requirement Number</b>	3.2	
<b>Actors</b>	Member	
<b>Parameters</b>	None	
<b>Preconditions</b>	The Member is logged in	
<b>Actions</b>	1. The Member requests to open a Store with some attributes 2. The System verifies the attributes are legal, if it fails it notifies the Member 3. The System makes a new Store and appoints the Member as a Founder	
<b>Result</b>	A new Store is created	
<b>Data</b>		<b>Result</b>
A Guest is attempting to open a store		Fail
A Member is attempting to open a store with invalid store attributes		Fail
A Member is attempting to open a store with legal attributes		Success

<b>Use Case Name</b>	Writing a review for a Product	
<b>Requirement Number</b>	3.3	
<b>Actors</b>	Member	
<b>Parameters</b>	<i>Member_ID, Product, review_string</i>	
<b>Preconditions</b>	The Member previously Purchased the Product	
<b>Actions</b>	1. The Member requests to submit a review for a Product 2. The System verifies that the <i>Product</i> is still sold by the Store, and that the <i>review_string</i> is legal 3. If the System fails to verify one of the above, the review is not submitted, and the Member is notified	
<b>Result</b>	The System posts the review and notifies relevant Members	
<b>Data</b>		<b>Result</b>
A Member is trying to write a review with 500+ characters		Fail
A Member is trying to write a review for a product he did not purchase		Fail
A Member is trying to write a review for a product he did purchase		Success



<b>Use Case Name</b>	Rating a Product and Store	
<b>Requirement Number</b>	3.4	
<b>Actors</b>	Member	
<b>Parameters</b>	<i>Rating, Store, Product</i>	
<b>Preconditions</b>	The Member haven't rated this Store and Product before	
<b>Actions</b>	1. The Member requests from the System to rate with a specific rating 2. The System verifies the preconditions and that the <i>Rating</i> has a legal value 3. If the System fails to verify, the Member is notified, and the rating is not submitted	
<b>Result</b>	The rating is submitted	
<b>Data</b>		<b>Result</b>
A Member is trying to rate twice the same Store and Product		Fail
A Guest is trying to rate		Fail
A Member is trying to rate with illegal rating value		Fail
A Member is trying to rate with legal rating value		Success

<b>Use Case Name</b>	Sending a message to a Store	
<b>Requirement Number</b>	3.5	
<b>Actors</b>	Member	
<b>Parameters</b>	<i>Store, Subject_string, body_string</i>	
<b>Preconditions</b>	None	
<b>Actions</b>	1. The Member selects a store and requests to send a message with a <i>Subject_string</i> title and <i>body_string</i> 2. The System sends the Message to the relevant Members and notifies them	
<b>Result</b>	The relevant Members are notified and receive the message	
<b>Data</b>		<b>Result</b>
A Member is attempting to send a message with invalid body or title length		Fail
A Member is attempting to send a message to a Store that doesn't exist		Fail
A Member is attempting to send a message with legal values for title and subject		Success

<b>Use Case Name</b>	Filling complaint to System admin	
<b>Requirement Number</b>	3.6	
<b>Actors</b>	User	
<b>Parameters</b>	<i>title_string, body_string, purchase_id</i>	
<b>Preconditions</b>	None	
<b>Actions</b>	1. The user requests the system to send a complaint to the system administrator about a purchase 2. The system prompts the user to enter title, body and purchase id 3. the user enter <i>title_string</i> , <i>body_string</i> and <i>purchase_id</i> The system sends the message and notifies the system administrator	
<b>Result</b>	The relevant Member is notified about the message	
<b>Data</b>		<b>Result</b>
The User enters an invalid title / body		Fail
The user didn't purchase anything		Fail
The user purchased something and he filled valid title and body		Success

<b>Use Case Name</b>	Getting information about former purchases	
<b>Requirement Number</b>	3.7	
<b>Actors</b>	Member	
<b>Parameters</b>	<i>None</i>	
<b>Preconditions</b>	Member is logged in	
<b>Actions</b>	1. The member requests the system for his purchase history 2. The system fetches his purchase history	
<b>Result</b>	The relevant Member receive his purchase history	
<b>Data</b>		<b>Result</b>
The Member is not logged in		Fail
The Member's purchasing history is empty		Success
The Member has made some purchases		Success

<b>Use Case Name</b>	Viewing Member's information	
<b>Requirement Number</b>	3.8	
<b>Actors</b>	Member	
<b>Parameters</b>	<i>The identification parameter the user wants to change</i>	
<b>Preconditions</b>	Member is logged in	
<b>Actions</b>	1. The member requests the system to view his information 2. The system fetches the member's information and shows it to him	
<b>Result</b>	The relevant Member gets his information	
<b>Data</b>		<b>Result</b>
The Member is not logged in		Fail
The Member is logged in		Success

<b>Use Case Name</b>	changing password / username	
<b>Requirement Number</b>	3.8	
<b>Actors</b>	Member	
<b>Parameters</b>	<i>Password / username the user wants to change</i>	
<b>Preconditions</b>	Member is logged in	
<b>Actions</b>	1. The member requests the system to change password/username 2. The system prompts the member to enter new password/username 3. The member enters new password/username The system verifies the username/password. The System checks they are valid and changes them	
<b>Result</b>	The relevant Member is notified that the parameters have changed	
<b>Data</b>		<b>Result</b>
The Member's new identification are invalid (empty password/username, weak password)		Fail
The member is not logged in		Fail
The Member's new identifications are valid		Success

<b>Use Case Name</b>	Upgrading the account security	
<b>Requirement Number</b>	3.9	
<b>Actors</b>	Member	
<b>Parameters</b>	<i>answer_string</i>	
<b>Preconditions</b>	The member is logged in	
<b>Actions</b>	1. The Member requests the system to upgrade his account security 2. The Member enters Security question and answer The system saves the question and the member's answer	
<b>Result</b>	The relevant Member is notified the process has been completed successfully	
<b>Data</b>		<b>Result</b>
The member enters an empty string as an answer		Fail
Guest is trying to add Security questions		Fail
The member submits a valid answer		Success

<b>Use Case Name</b>	Add product to a store	
<b>Requirement Number</b>	4.1, 5	
<b>Actors</b>	Manager	
<b>Parameters</b>	<i>Store, Product</i>	
<b>Preconditions</b>	Store manager must have proper permission from store owner / founder	
<b>Actions</b>	1. The owner is logged in 2. The System verifies the Manager has the right permissions to perform the action 3. The System adds the product to the Store	
<b>Result</b>	The relevant Manager is notified about the change	
<b>Data</b>		<b>Result</b>
The Manager doesn't have the right permissions		Fail
The Manager trying to add a product that already exists in the store		Fail
The Manager has permissions and adds a product		Success

<b>Use Case Name</b>	Update product in a store	
<b>Requirement Number</b>	4.1, 5	
<b>Actors</b>	Manager	
<b>Parameters</b>	<i>Store, Product</i>	
<b>Preconditions</b>	Store manager must have proper permission from store owner / founder	
<b>Actions</b>	1. The owner is logged in 2. The System verifies the Manager has the right permissions to perform the action 3. The system updates the product in the Store	
<b>Result</b>	The relevant Manager is notified about the change	
<b>Data</b>		<b>Result</b>
The Manager doesn't have the right permissions		Fail
The Manager has permissions and updates a product		Success

<b>Use Case Name</b>	Remove product from a store	
<b>Requirement Number</b>	4.1, 5	
<b>Actors</b>	Manager	
<b>Parameters</b>	<i>Store, Product</i>	
<b>Preconditions</b>	Store manager must have proper permission from store owner / founder	
<b>Actions</b>	1. The owner is logged in 2. The System verifies the Manager has the right permissions to perform the action 3. The System removes the product from the Store	
<b>Result</b>	The relevant Manager is notified about the change	
<b>Data</b>		<b>Result</b>
The Manager doesn't have the right permissions		Fail
The Manager tries to remove a product that doesn't exists in the store		Fail
The Manager has permissions and removes a product		Success

<b>Use Case Name</b>	Set direct discount to product	
<b>Requirement Number</b>	4.2, 5	
<b>Actors</b>	Store owner	
<b>Parameters</b>	<i>Store name, until date, discount percent, product name</i>	
<b>Preconditions</b>	Store owner is an owner of the given store, until date is valid, product exists and discount percent is valid	
<b>Actions</b>	1. User inserts input to system 2. System sets the proper discount to the given product – removing any previous discounts 3. Success message is displayed to the user	
<b>Result</b>	The product is now associated with the new discount	
<b>Data</b>		<b>Result</b>
A store which is not owned by the user is inserted		Fail
Invalid date is inserted		Fail
Invalid discount percent is inserted		Fail
Non-existing product is inserted		Fail
All preconditions hold		Success

<b>Use Case Name</b>	Set secret discount to product	
<b>Requirement Number</b>	4.2, 5	
<b>Actors</b>	Store owner	
<b>Parameters</b>	<i>Store name, until date, discount percent, product name, secret code</i>	
<b>Preconditions</b>	Store owner is an owner of the given store, until date is valid, product exists, discount percent is valid and secret code isn't empty	
<b>Actions</b>	1. User inserts input to system 2. System sets the proper discount to the given product – removing any previous discounts 3. Success message is displayed to the user	
<b>Result</b>	The product is now associated with the new discount	
<b>Data</b>		<b>Result</b>
A store which is not owned by the user is inserted		Fail
Invalid date is inserted		Fail
Invalid discount percent is inserted		Fail
Non-existing product is inserted		Fail
All preconditions hold		Success

<b>Use Case Name</b>	Set discount to store	
<b>Requirement Number</b>	4.2, 5	
<b>Actors</b>	Store owner	
<b>Parameters</b>	<i>Store name, until date, discount percent, type of discount (condition)</i>	
<b>Preconditions</b>	Store owner is an owner of the given store, until date is valid, discount percent is valid and the type of discount is supported by the system	
<b>Actions</b>	1. User chooses condition for a store discount 2. User enters discount details (date + percent) 3. The system associates the store with the new discount, removing any previous discounts 4. The system displays a success message	
<b>Result</b>	The store is now associated with the new discount	
<b>Data</b>		<b>Result</b>
Any of the preconditions doesn't hold		Fail
All preconditions hold		Success

<b>Use Case Name</b>	Set composed discount to store	
<b>Requirement Number</b>	4.2, 5	
<b>Actors</b>	Store owner	
<b>Parameters</b>	<i>Store name, until date, discount percent, previous existing discounts, how to compose both discounts</i>	
<b>Preconditions</b>	Store owner is an owner of the given store, until date is valid, discount percent is valid, both discounts exist in the system	
<b>Actions</b>	1. User chooses 2 previous existing discounts in the system 2. User enters discount details (date + percent) 3. The system associates the store with the new discount, removing any previous discounts 4. The system displays a success message	
<b>Result</b>	The store is now associated with the new discount	
<b>Data</b>		<b>Result</b>
Any of the preconditions doesn't hold		Fail
All preconditions hold		Success

<b>Use Case Name</b>	Purchasing a cart with valid discounts	
<b>Requirement Number</b>	4.2, 5	
<b>Actors</b>	User / guest	
<b>Parameters</b>	<i>Payment and supplying info</i>	
<b>Preconditions</b>	Cart isn't empty and contains items from stores with discounts	
<b>Actions</b>	1. User attempts to purchase cart 2. The system calculates the price of the cart and takes into account all discount rules 3. <b><i>Purchase cart use case</i></b> is now activated	
<b>Result</b>	Same as purchase cart use case result	
<b>Data</b>		<b>Result</b>
Empty cart		Fail
All preconditions hold		Purchase has succeeded + payment was done for the price AFTER discounts

<b>Use Case Name</b>	Creating a new purchase condition and setting it to a store	
<b>Requirement Number</b>	4.2, 5	
<b>Actors</b>	Store owner	
<b>Parameters</b>	<i>Store name, condition type(s), condition info</i>	
<b>Preconditions</b>	Owner is an actual owner of the store	
<b>Actions</b>	1. User inserts store name and condition type(s) 2. User chooses how to compose the conditions (single, or, xor, and) 3. System associates store with the new purchase condition, removing any previous conditions	
<b>Result</b>	The store is now associated with the new purchase condition	
<b>Data</b>		<b>Result</b>
A non- owner triggers the use case		Fail
Precondition holds, valid conditions are inserted		Success

<b>Use Case Name</b>	Composing new purchase condition from previous conditions	
<b>Requirement Number</b>	4.2, 5	
<b>Actors</b>	Store owner	
<b>Parameters</b>	<i>Store name, condition ids, composing operator</i>	
<b>Preconditions</b>	Owner is an actual owner of the store	
<b>Actions</b>	1. Store owner inserts store name, and condition ids he wants to compose 2. Owner chooses operator to compose between conditions 3. System associates store with new purchase condition removing any previous conditions it was associated with	
<b>Result</b>	The store is now associated with the new purchase condition	
<b>Data</b>		<b>Result</b>
A non-owner triggers the use case		Fail
Precondition holds, valid conditions are inserted		Success

<b>Use Case Name</b>	A user tries to purchase a cart that violates a purchase condition	
<b>Requirement Number</b>	4.2, 5	
<b>Actors</b>	User/Guest	
<b>Parameters</b>	<i>Payment and supplying info</i>	
<b>Preconditions</b>	Cart isn't empty, and violates some purchase conditions	
<b>Actions</b>	1. User attempts to purchase a cart that violates some purchase conditions in some stores 2. The system displays a relevant message to the user stating that a purchase condition violation has occurred 3. Purchase use case ISN'T TRIGGERED	
<b>Result</b>	An appropriate response message is displayed	
<b>Data</b>		<b>Result</b>
A cart that violates at least 1 purchase condition		System displays error message

<b>Use Case Name</b>	Owner changes product's policy to bargaining	
<b>Requirement Number</b>	4.2, 5	
<b>Actors</b>	Store owner	
<b>Parameters</b>	<i>Store name, product name, starting price for bidding</i>	
<b>Preconditions</b>	Owner is a store owner, product exists	
<b>Actions</b>	1. Owner inserts store name, product name and bidding starting price 2. System removes item from any purchase cart of any user in the system 3. System associates product with new purchase policy 4. System displays relevant success message	
<b>Result</b>	Product is associated with the new policy	
<b>Data</b>		<b>Result</b>
Any of the conditions doesn't hold		Fail
All conditions hold		Product is associated with new purchase policy and all products are deleted from users' carts (direct purchase policy)

<b>Use Case Name</b>	User bids on product	
<b>Requirement Number</b>	4.2, 5	
<b>Actors</b>	Member	
<b>Parameters</b>	<i>Store name, product name, payment and supplying info</i>	
<b>Preconditions</b>	1. Triggering user is a member of the market 2. Product is up for bargaining	
<b>Actions</b>	1. User searches for a relevant product and presses "place a bid" 2. User enters bid price and payment & supplying info 3. System stores supply and payment info in its database 4. System creates a new bid (or updates the existing bid with the new price) 5. System displays success message to user	
<b>Result</b>	Bid is created, its status is set to waiting for approval and payment and supplying info is saved in the system for later user	
<b>Data</b>		<b>Result</b>
Any of the conditions doesn't hold		Fail
All conditions hold, no previous existing bids		A new bid is created for the user, relevant message is displayed
All conditions hold, there was a previous existing bid for the user		The bid is updated with the new bid the user has given and a relevant response message is displayed



<b>Use Case Name</b>	Store staff approves bid
<b>Requirement Number</b>	4.2, 5
<b>Actors</b>	Store staff
<b>Parameters</b>	<i>Store name, product name, bidding user to approve</i>
<b>Preconditions</b>	1. Triggering user is a staff of the store and has relevant permissions 2. User has made a bid for the product
<b>Actions</b>	1. User enters input details 2. System adds the approving user to the approvers' list. If the approving user is the last user to approve, the system restores the payment and supply info of the bidding user and executes <b><i>Purchase use case</i></b> with this data 3. Appropriate response message is displayed to user
<b>Result</b>	Approving user is added to approvers' list and / or purchase use case is executed (only if 2.1 is triggered)
<b>Data</b>	<b>Result</b>
Any of the conditions doesn't hold	Fail
All conditions hold, user isn't the last staff to approve	The user is added to the approver's list – purchase use case isn't yet executed
All conditions hold, user is the last staff to approve	The user is added to the approver's list – purchase use case is executed, and the bid is deleted from the system

<b>Use Case Name</b>	Store staff declines bid
<b>Requirement Number</b>	4.2, 5
<b>Actors</b>	Store staff
<b>Parameters</b>	<i>Store name, product name, bidding user to decline</i>
<b>Preconditions</b>	1. Triggering user is a staff of the store and has relevant permissions 2. User has made a bid for the product
<b>Actions</b>	1. User enters input details 2. System deletes bid from database 3. System notifies bidding user that the store has rejected its bidding 4. System displays relevant success message to user
<b>Result</b>	Bid is deleted from system and a relevant message is sent to the bidding user
<b>Data</b>	<b>Result</b>
Any of the conditions doesn't hold	Fail
All conditions hold	The bid is deleted from the system, proper message is sent to the declined user

<b>Use Case Name</b>	Change back to direct purchase policy
<b>Requirement Number</b>	4.2, 5
<b>Actors</b>	Store staff
<b>Parameters</b>	<i>Store name, product name, original price</i>
<b>Preconditions</b>	Triggering user is a staff of the store and has relevant permissions
<b>Actions</b>	<ol style="list-style-type: none"> <li>1. User enters input details</li> <li>2. System closes all ongoing bids</li> <li>3. System executes <b><i>Store staff declines bid</i></b> for any ongoing bid for the product</li> <li>4. The product is associated with the new direct purchase policy</li> <li>5. Relevant response message is displayed</li> </ol>
<b>Result</b>	All bids are deleted from system, relevant message is sent to all the bidding users, product is not up for bidding anymore
<b>Data</b>	<b>Result</b>
Any of the conditions doesn't hold	Fail
All conditions hold, no on-going bids	No messages are sent, product isn't up for bidding
All conditions hold, there are on-going bids	All bids are declined, relevant messages are sent to declined users. Product isn't up for bidding

<b>Use Case Name</b>	Approve store owner appointment request	
<b>Requirement Number</b>	4.4, 5	
<b>Actors</b>	Owner, Founder	
<b>Parameters</b>	<i>new_owner, Store</i>	
<b>Preconditions</b>	1. The Owner is logged in 2. Store manager must have proper permission from store owner	
<b>Actions</b>	1. The owner requests the system to approve the appointment of new_owner 2. The system verifies the Owner is the owner of the Store 3. The system verifies there is an open request to appoint new_owner to Owner 4. The system adds the vote to the list of Owners who approved 5. If all Owners approved the request then the new_owner is granted Owner permissions	
<b>Result</b>	The system updates the list of owners who approved the appointment and if all of the owners approved then the new_owner receives Owner Permissions	
<b>Data</b>		<b>Result</b>
The Owner is not the Store owner		Fail
There is no request to appoint new_owner to Owner		Fail
The owner is the store owner and there is a request to appoint new_owner		Success

<b>Use Case Name</b>	Decline store owner appointment request	
<b>Requirement Number</b>	4.4, 5	
<b>Actors</b>	Owner, Founder	
<b>Parameters</b>	<i>new_owner, Store</i>	
<b>Preconditions</b>	1. The Owner is logged in 2. Store manager must have proper permission from store owner	
<b>Actions</b>	1. The owner requests the system to decline the appointment of new_owner 2. The system verifies the Owner is the owner of the Store 3. The system verifies there is an open request to appoint new_owner to Owner 4. The system removes the request and notifies the requesting owner that the request has been declined	
<b>Result</b>	The system updates the list of owners who approved the appointment and if all of the owners approved then the new_owner receives Owner Permissions	
<b>Data</b>		<b>Result</b>
The Owner is not the Store owner		Fail
There is no request to appoint new_owner to Owner		Fail
The owner is the store owner and there is a request to appoint new_owner		Success

<b>Use Case Name</b>	Appoint store owner	
<b>Requirement Number</b>	4.4, 5	
<b>Actors</b>	Owner,Store Manager	
<b>Parameters</b>	<i>new_owner,Store</i>	
<b>Preconditions</b>	1. The Owner is logged in 2. Store manager must have proper permission from store owner	
<b>Actions</b>	1. The Owner requests the system to add a store owner to his Store 2. The System verifies the Owner is the Store owner 3. The system verifies the new_owner is not already the store owner/manager 4. The system adds a request to appoint new_owner and notifies all other owners about the request	
<b>Result</b>	The system adds a request to appoint new_owner and notifies all other owners about the request	
<b>Data</b>		<b>Result</b>
The Owner is not the Store owner		Fail
The new_owner already owns the Store		Fail
The owner is the store owner and the new_owner is not		Success

<b>Use Case Name</b>	Remove Store owner	
<b>Requirement Number</b>	4.5, 5	
<b>Actors</b>	Owner,Store Manager	
<b>Parameters</b>	<i>Store, owner_to_remove</i>	
<b>Preconditions</b>	1. The Owner is logged in 2. Store manager must have proper permission from store owner	
<b>Actions</b>	1. The Owner requests the system to remove a store owner from his Store 2. The system verifies the Owner is the Store owner 3. The System verifies owner_to_remove was appointed by the Owner 4. After removing the owner, the system checks if there is a pending owner appointment request that can now be approved	
<b>Result</b>	The system removes the permissions of owner_to_remove and all other members that were appointed by owner_to_remove and notifies the Owner	
<b>Data</b>		<b>Result</b>
The owner is not the Store owner		Fail
The owner_to_remove wasn't appointed by Owner		Fail
The Owner is the owner and owner_to_remove was appointed by him		Success

<b>Use Case Name</b>	Receive owner appointment requests	
<b>Requirement Number</b>	4.5, 5	
<b>Actors</b>	Owner, Founder	
<b>Parameters</b>	<i>Store</i>	
<b>Preconditions</b>	1. The Owner is logged in 2. Store manager must have proper permission from store owner	
<b>Actions</b>	1. The Owner / Founder requests the system for the open owner appointment requests he didn't vote on of Store 2. The system verifies the store exists 3. The system verifies the Owner / Founder is indeed Owner / Founder of Store	
<b>Result</b>	The system provides the Owner / Founder with all the Store's open Owner appointment requests that he didn't vote on	
<b>Data</b>		<b>Result</b>
The owner is not the Store owner		Fail
The Store does not exist		Fail
The Owner is the owner and Store exists		Success

<b>Use Case Name</b>	Appoint store manager	
<b>Requirement Number</b>	4.6, 5	
<b>Actors</b>	Owner	
<b>Parameters</b>	<i>new_manager, Store</i>	
<b>Preconditions</b>	The Owner is logged in	
<b>Actions</b>	1. The owner requests the system to add a store manager to his Store 2. The System verifies the Store exists 3. The System verifies the requester is the Store owner 4. The System verifies the new_manager is a valid member 5. The System makes the new_manager a manager with default permissions to the store	
<b>Result</b>	The system updates new_manager permissions and notifies the Owner	
<b>Data</b>		<b>Result</b>
The owner is not the Store owner		Fail
The new_manager is already the Store's manager / owner		Fail
The owner sets a User who is not the Store's manager / owner		Success

<b>Use Case Name</b>	Change store manager permissions
<b>Requirement Number</b>	4.7, 5
<b>Actors</b>	Owner, Store manager
<b>Parameters</b>	<i>manager, new_permissions, Store</i>
<b>Preconditions</b>	1. The Owner is logged in 2. Store manager must have proper permission from store owner
<b>Actions</b>	1. The Owner requests the system to manager permissions 2. The System verifies the Owner is the Store owner 3. The Owner enters the new_permissions
<b>Result</b>	The system updates the manager permissions and notifies the Owner
<b>Data</b>	
The owner is not the Store owner	Fail
The manager is not the Store's manager	Fail
The owner chooses correct permissions	Success

<b>Use Case Name</b>	Remove store manager
<b>Requirement Number</b>	4.8, 5
<b>Actors</b>	Owner, Store Manager
<b>Parameters</b>	<i>manager_to_remove, Store</i>
<b>Preconditions</b>	1. The Owner is logged in 2. Store manager must have proper permission from store owner
<b>Actions</b>	1. The Owner requests the system to remove manager_to_remove from his position as Store manager 2. The System verifies the Owner is the Store owner 3. The system verifies manager_to_remove was appointed by the Owner
<b>Result</b>	The system removes manager_to_remove permissions as store manager and notifies the Owner
<b>Data</b>	
The owner is not the Store owner	Fail
The manager_to_remove was not appointed by the Owner	Fail
The manager_to_remove was appointed by the Owner	Success

<b>Use Case Name</b>	Close a store
<b>Requirement Number</b>	4.9, 5
<b>Actors</b>	Store Founder
<b>Parameters</b>	Store ID
<b>Preconditions</b>	1. Store exists and in “open” status 2. User is the store’s founder
<b>Actions</b>	1. The store founder asks the system to close a store 2. The store founder inputs a store that he founded 3. The system verifies that the store exists and is not in “closed” state already 4. The system verifies the relevant user permissions 5. The system changes the store status to “closed” 6. The system sends the store staff a “Store closing message” 7. The system removes the store’s products from the product search view
<b>Result</b>	1. Store’s status is changed to “closed” 2. Store is only visible to admins and store owners 3. Store’s product are not visible on search 4. “Store closing message” is sent to store’s owner and manager
<b>Data</b>	<b>Result</b>
User enters an invalid store name	Fail
User enters a store that is already closed	Fail
User enters a store ID of a store he founded	Success

<b>Use Case Name</b>	Reopen a closed store
<b>Requirement Number</b>	4.10, 5
<b>Actors</b>	Store founder
<b>Parameters</b>	Store ID
<b>Preconditions</b>	1. Store exists and in “closed” status 2. User is the store’s founder
<b>Actions</b>	1. Store founder chooses to reopen a store 2. Store founder inputs a store he founded 3. The system verifies that the store exists and in not in “open” state already 4. The system verifies the relevant user permissions 5. The system changes store’s status to “open” 6. The system sends a “Store reopening message” to the store staff
<b>Result</b>	1. Store’s status is changed to “open” 2. Store’s manager and owner receive a “Store reopening message”
<b>Data</b>	<b>Result</b>
User enters an invalid store name	Fail
User enters a store that is already open	Fail
User enters a store ID of a store he founded	Success

<b>Use Case Name</b>	Get store employees information	
<b>Requirement Number</b>	4.11, 5	
<b>Actors</b>	Store founder, Store manager, Store Owner	
<b>Parameters</b>	<i>Store_name</i>	
<b>Preconditions</b>	1. Member must exist 2. Member ID must be of a store employee 3. Store manager must have proper permission from store owner	
<b>Actions</b>	1. Store founder asks the system to get information about the <i>store_name</i> employees 2. The system verifies the relevant user permissions 3. The system provides the store founder with the information about the staff including manager permissions	
<b>Result</b>	Store founder is shown the employee's information and for each manager it shows the permissions he has	
<b>Data</b>		<b>Result</b>
Member ID is of a member that doesn't have permissions to view store staff		Fail
Member ID is of a member that has the right permissions to view store staff		Success

<b>Use Case Name</b>	Receive and read member's questions	
<b>Requirement Number</b>	4.12, 5	
<b>Actors</b>	Store owner, Store manager	
<b>Parameters</b>	None	
<b>Preconditions</b>	Store manager must have proper permission from store owner	
<b>Actions</b>	1. The store owner chooses to read member's questions 2. The system verifies the relevant user permissions 3. The system shows the member's questions to the store owner	
<b>Result</b>	Store owner is shown the member's questions list	
<b>Data</b>		<b>Result</b>
User (Not a store owner) chooses the read member's questions option		Fail
User (Store owner) chooses the read member's questions option		Success



<b>Use Case Name</b>	Answer members' questions	
<b>Requirement Number</b>	4.12, 5	
<b>Actors</b>	Store owner, Store manager	
<b>Parameters</b>	<i>Answer text</i>	
<b>Preconditions</b>	Store manager must have proper permission from store owner	
<b>Actions</b>	1. Store owner chooses a member's questions 2. Store owner chooses to answer the question 3. Store owner inputs the answer text 4. The system verifies the text is not empty 5. The system verifies relevant user permissions 6. The system adds the answer to the member's question	
<b>Result</b>	The answer is added to the member's question	
<b>Data</b>		<b>Result</b>
User enters an empty string as answer		Fail
User enters a non-empty string as answer		Success

<b>Use Case Name</b>	Get store's purchase history	
<b>Requirement Number</b>	4.13, 5	
<b>Actors</b>	Store owner, Store manager	
<b>Parameters</b>	<i>Store ID</i>	
<b>Preconditions</b>	1. Store ID must be of a store that is owned by the store owner 2. Store manager must have proper permission from store owner	
<b>Actions</b>	1. The store owner asks the system to get store purchase history 2. The store owner inputs the store ID 3. The system verifies that the store exists 4. The system verifies the relevant user permissions 5. The system provides the purchase history for that store	
<b>Result</b>	Store's purchase history is provided to the store owner	
<b>Data</b>		<b>Result</b>
User enters a store ID that doesn't exist		Fail
User enters a store ID of a store he doesn't own		Fail
User enters a store ID of the store he owns		Success

<b>Use Case Name</b>	Closing store permanently	
<b>Requirement Number</b>	6.1	
<b>Actors</b>	Admin, Store Owner, Store Manager	
<b>Parameters</b>	<i>Store ID</i>	
<b>Preconditions</b>	Store exists	
<b>Actions</b>	1. The admin inputs a store to close 2. The system verifies that the store exists 3. The system verifies the relevant user permissions 4. The system sends "Store closing message" to store owners and managers 5. The system cancels the membership of the store owners and managers	
<b>Result</b>	The store is no longer available to users and store managers and owners are no longer members	
<b>Data</b>		<b>Result</b>
User enters a store ID of a store that doesn't exist		Fail
User enters a store ID of a store that is already closed		Fail
User enters a store ID of an open store		Success

<b>Use Case Name</b>	Canceling membership	
<b>Requirement Number</b>	6.2	
<b>Actors</b>	Admin, Member	
<b>Parameters</b>	<i>Member ID (user name)</i>	
<b>Preconditions</b>	Member exists and registered	
<b>Actions</b>	1. The admin inputs a member ID he wishes to remove 2. The system verifies that the member exists 3. The system verifies the relevant user permissions 4. The system cancels all permissions and roles of the member	
<b>Result</b>	The membership is canceled	
<b>Data</b>		<b>Result</b>
User enters a member ID of a member that doesn't exist		Fail
User enters a character that is not a number		Fail
User enters an empty string		Fail
User enters a member ID of an existing member		Success

<b>Use Case Name</b>	Read and comment to complaints	
<b>Requirement Number</b>	6.3	
<b>Actors</b>	Admin, Member	
<b>Parameters</b>	<i>Comment text</i>	
<b>Preconditions</b>	None	
<b>Actions</b>	1. The admin gets information and reads the members complaints 2. The admin chooses a complaint 3. The admin comments on the member complaint 4. The system verifies the text is not empty 5. The system verifies the relevant user permissions	
<b>Result</b>	A comment is added to the member's complaint	
<b>Data</b>		<b>Result</b>
User enters an empty string as comment		Fail
User enters a non-empty string as comment		Success

<b>Use Case Name</b>	Send messages to buyers and sellers	
<b>Requirement Number</b>	6.3	
<b>Actors</b>	Admin, Buyer/Seller	
<b>Parameters</b>	<i>Message text</i>	
<b>Preconditions</b>	None	
<b>Actions</b>	1. The admin chooses a buyer or seller 2. The admin chooses to send the member a message 3. The admin inputs the message text 4. The system verifies the text is not empty 5. The system verifies the relevant user permissions 6. The system sends to message to the selected member	
<b>Result</b>	A message is sent to the selected member	
<b>Data</b>		<b>Result</b>
User enters an empty string as message		Fail
User enters a non-empty string as message		Success

<b>Use Case Name</b>	Receive messages from buyers and sellers	
<b>Requirement Number</b>	6.3	
<b>Actors</b>	Admin, Buyer/Seller	
<b>Parameters</b>	None	
<b>Preconditions</b>	None	
<b>Actions</b>	1. The system gets a message to send from a buyer or seller 2. The system notifies the admin that a message is waiting (If admin is not logged in he will get notified when he logs in) 3. The admin opens the message and reads it	
<b>Result</b>	A message is received and read by the admin	
<b>Data</b>		<b>Result</b>
No internet connection		Fail
A buyer sends a non-empty message to admin		Success

<b>Use Case Name</b>	Get purchase history of member	
<b>Requirement Number</b>	6.4	
<b>Actors</b>	Admin, Member	
<b>Parameters</b>	<i>Member ID</i>	
<b>Preconditions</b>	Member exists	
<b>Actions</b>	1. The admin chooses a member it wants to get its purchase history 2. The system verifies the member exists 3. The system verifies the relevant user permissions 4. The system sends the admin the saved history of the member	
<b>Result</b>	Member's purchase history is provided to the admin	
<b>Data</b>		<b>Result</b>
User enters a member ID that doesn't exist		Fail
User enters a character that is not a number		Fail
User enters a member ID of an existing member		Success

<b>Use Case Name</b>	Get purchase history of a store	
<b>Requirement Number</b>	6.4	
<b>Actors</b>	Admin	
<b>Parameters</b>	<i>Store ID</i>	
<b>Preconditions</b>	Store exists	
<b>Actions</b>	1. The admin chooses a store he wants to know the purchase history of 2. The system verifies the store exists 3. The system verifies the relevant user permissions 4. The system sends the admin the saved history of the store	
<b>Result</b>	Store's purchase history is provided to the admin	
<b>Data</b>		<b>Result</b>
User enters a store ID that doesn't exist		Fail
User enters a character that is not a number		Fail
User enters a store ID of an existing store		Success

<b>Use Case Name</b>	Admin attempts to view system stats	
<b>Requirement Number</b>	6.5	
<b>Actors</b>	System admin	
<b>Parameters</b>	<i>Date</i>	
<b>Preconditions</b>	Actor has to be a system admin, date has to be today or any earlier date	
<b>Actions</b>	1. Admin enters a date 2. System displays all stats of the system regarding the inserted date.	
<b>Result</b>	Relevant stats are displayed to the user	
<b>Data</b>		<b>Result</b>
Non admin tries to trigger this use case		Fail
Admin tries to trigger this use case		System displays relevant system stats
Admin inserts an invalid date		Fail

<b>Use Case Name</b>	Real time stats change	
<b>Requirement Number</b>	6.5	
<b>Actors</b>	System admin, system-event change trigger, publisher component	
<b>Parameters</b>	<i>None</i>	
<b>Preconditions</b>	Admin is currently viewing the system stats of TODAY	
<b>Actions</b>	1. An event that triggers system-stats change has happened 2. System notifies publisher that system stats have changed 3. Publisher posts updated stats to admin	
<b>Result</b>	Relevant stats are refreshed without the user doing anything	
<b>Data</b>		<b>Result</b>
Non trigger event happens		No change to system stats
Trigger event happens		System stats are updated and displayed to viewing admin(s)