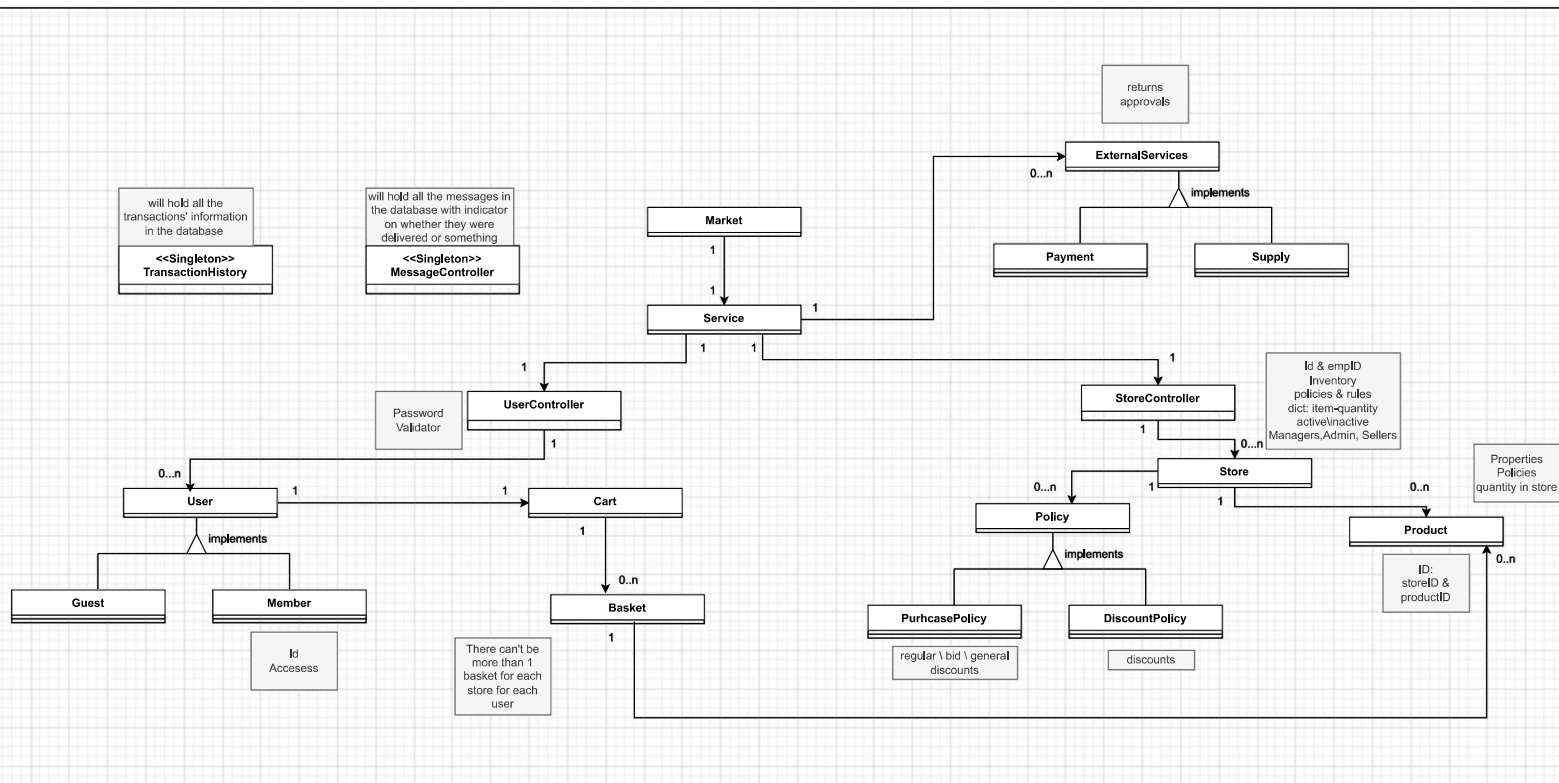


## White diagram:



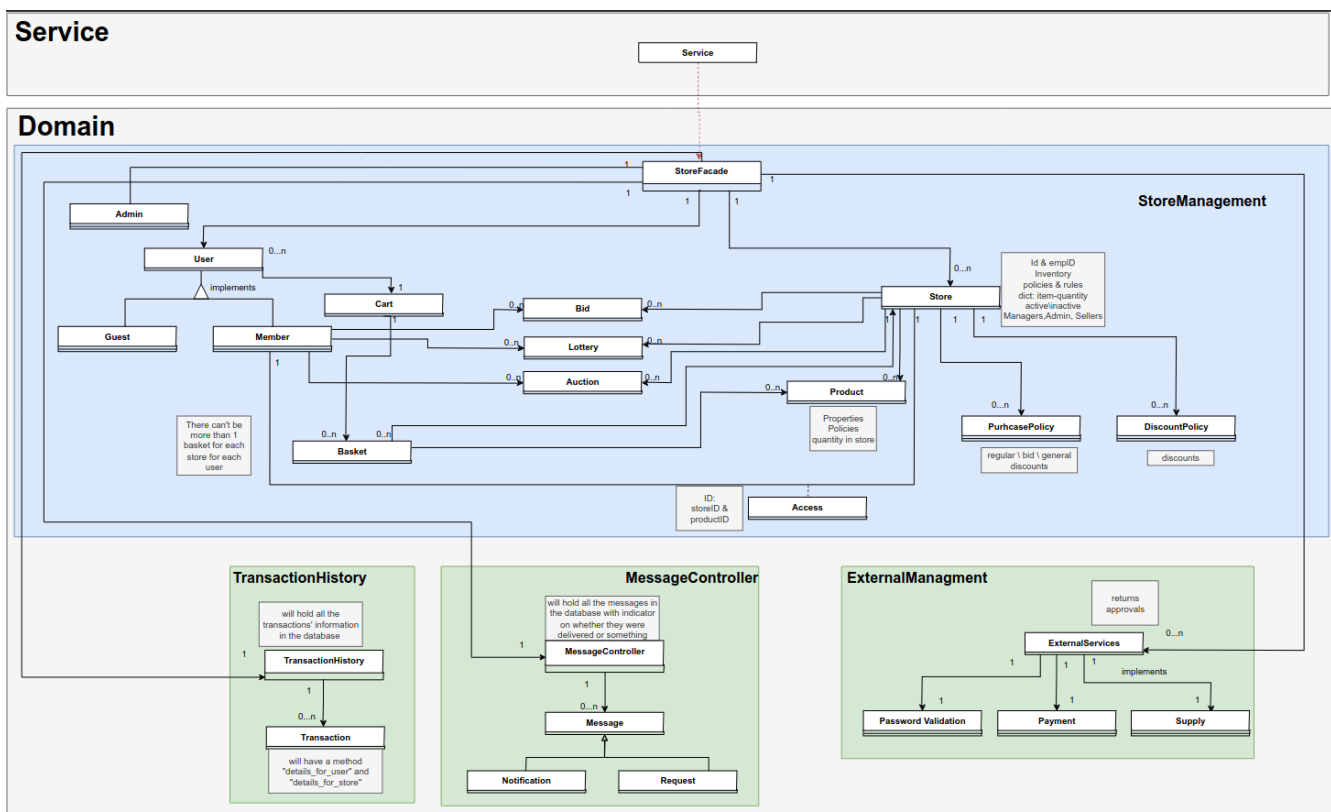
## Architecture diagram:

# System requirements:

## 1.1 starting the market system

Actor: Admin

Pre-conditions: only admin has system manager premissions, the given user exists in the database.



Post-conditions: None

Flow:

- An admin logging into the system using his password and username of an Admin type user
- The Admin request to open the market system
- The system checks if the user is an Admin
- The system sends a message to payment and supliment services
- a message is being return from those services to the system
- The system sends a message to the the user that the system had started

Tests:

- Positive: the admin succeed to start the system
- Negative: the system says this user doesn't have system Admin permissions

## **1.2 change/replace/add a connection with external services**

Actor: Admin

Pre-conditions: at least one external service is connected to the system while the system is running.

Post-conditions: None.

### **a. change:**

Flow:

- The Admin will select a connected external service to change
- The system will show all the available fields for changing
- The admin will change the selected fields
- The system will update the selected fields.
- The admin will receive a message "The connection has updated successfully"

Tests:

- Positive: the selected fields of the external service connection will be updated
- Negative: the selected fields of the external service connection will not update

### **b. replace:**

Flow:

- The admin will select a connected external service to replace
- The admin will choose a second (not connected) external service
- The system will send a connection request to the second external service
- The system will establish a connection with the second external system
- The system will disconnect from the first external system
- The admin will receive a message "The replacement has accomplished successfully"

Tests:

- Positive: the second selected external service will be connected to the system and the first selected external service will no longer be connected
- Negative: the replacement will fail – the second selected external service will not be connected to the system or the first selected one will still be connected to the system.

### **c. add:**

Flow:

- The admin will select a connected external service to add
- The system will send a connection request to the selected external service
- The system will establish a connection with the selected external system
- The admin will receive a message "The connection has established successfully"

Tests:

- Positive: the selected external service will be connected to the system
- Negative: the selected external service will not connect to the system

## **1.3 payment**

Actor: System

Pre-condition: an unpaid transaction details exist in the database, and a payment system is connected to the the market system.

Post-condition: None.

Flow:

- the system checks the connection to the payment system
- The system receive a message from the payment system it's connected to the system
- The system sends to the payment service the transaction details
- The system receive from the payment system a message that the payment has set successfully

Test:

- Positive: The system connects to the payment system successfully
- Negative: The system doesn't receive a message back from the payment system and the connection fails.

## **1.4 suppliment**

Actor: System

Pre-conditions: at least one product suppliment system is connected to the market system.

Post-conditions: None.

Flow:

- The system checks the connection to the suppliment system
- The system recive a message from the suppliment system it's connected to the system
- The system send to the suppliment system a package details and customer details
- The system recive from the suppliment system an approval for the suppliment of the given package

Tests:

- Positive: the system recives an approval message for the suppliment of the given package from the suppliment system.
- Negative: the system doesn't recive an approval message from the suppliment system about the given package.

### **1.5 real time notifications**

Actor: System

Pre-conditions: the shop owner is logged into the system.

Post-conditions: None.

Flow A:

- Customer buys a product from the shop owner's shop
- The system fetch all the logged shop owners
- The system sends an notification to each of them which says "a customer bought a <product name> from your store!"

Tests:

- Positive: each shop owner recives a notification to his screen.
- Negative: at least one shop owner of this shop doesn't recive a notification from the system about this transaction.

Flow B:

- The shop is being shut down
- The system fetch all the logged shop owners of this specific shop
- The system sends an notification to each of them which saying "the shop has successfully closed"

Tests:

- Positive: each shop owner receives a notification to his screen.
- Negative: at least one shop owner of this shop doesn't receive a notification from the system about the shop being shut down.

Flow C:

- A shop has opened
- The system fetch all the logged shop owners
- The system sends an notification to each of them which saying "the store has successfully opened".

Tests:

- Positive: each shop owner receives a notification to his screen.
- Negative: at least one shop owner of this shop doesn't receive a notification from the system about the shop being opened.

Pre-conditions: The store has at least 2 shop owners, the removed shop owner as no nominators under it.

Post-conditions: None.

Flow:

- The system remove the nomination of the shop owner
- The system sends to the former shop owner a message "your nomination in <name of the shop> has been removed".

Tests:

- Positive: the former shop owner receives a notification to his screen.
- Negative: the former shop owner is no longer a owner of this store but no notification was sent by the system.

Actor: User

Pre-conditions: At least 2 users connected to the system

Post-conditions: None.

Flow:

- User 1 sends a message to user2 using the system
- The system receives the message and checks if user2 is connected

- The system send a notification with the message to user2

Tests:

- Positive: user2 will receive the message.
- Negative: user2 will not receive the message.

## **1.6 Delayed notifications**

Actor: System

Pre-conditions: a non-logged user with delayed notifications in the database

Post-conditions: the user is logged in and with no delayed notifications in the database

Flow:

- User logging into the system
- The system check if there are any delayed messages belong to this user in the database
- The system fetch the messages and send them as notifications to the user
- The user receives a notification in his screen

Tests:

- Positive: the user receive his delayed notifications.
- Negative: the user have delayed notification but he doesn't receive them.

## **2 -Users**

### **2.1.1 Guest Visit**

Actor: User

Pre-conditions: None

Post-conditions: New guest have a cart and exists in the system's online-guests list.

Flow:

- The guest enters the market system.
- The system checks if the type of the current user is Guest.
- If OK:
  - The system assigns a cart for the user.
- Else:
  - The system redirects to an error page.

Tests:

- Positive: User add products to his basket.
- Negative: User gets an error trying to enter the market.

### **2.1.2 Guest Exits**

Actor: User

Pre-conditions: Guest is connected to the system.



Post-conditions: the guest doesn't exist in the system any more.

Flow:

- The guest exits the system.
- The system removes the guest's cart and the guest from its connected users.

Tests:

- Positive: the guest doesn't exist in the system's guest list.
- Negative: the guest still exist in the system.

### **2.1.3 Registration to the system**

Actor: User

Pre-conditions: No user with the tested username exist in the system.

Post-conditions: a new member created in the system with a new member id and the registration information

Flow:

- The user enter username, password and email and send a registration request.
- The system validates that there is no user with the same username or email.
- If OK:
  - The system creates a new user and sends a confirmation to the user.
- else:
  - The system returns an appropriate error.

Tests:

- Positive: A guest user fills the sign-up info sheet with a unique user-name, and a new member account is created in the system.
- Negative: A guest user fills the sign-up info sheet with an already used user-name, and an error message pops and says "this user name already exists."

### **2.1.4 Logging in the system**

Actor: User

Pre-conditions: None

Post- conditions: A new guest exist in the system's online guests list

Flow:

- The user login as guest to the system.
- The system create a new guest instance in the system.
- The system add the new guest instance to the online-guests list
- the system returns a new guest instance

Tests:

- Positive: the guest successfully enters the system
- Negative: the guest successfully enters the system but doesn't exist in the system online-guests list

### **2.2.1 Guest Information fetching**

Actor: User

Pre-conditions: A guest is connected to the system

Post-conditions: None

Flow:

- The guest enters a store in the market.
- The system fetches the requested store.
- If OK:
  - The system returns to the user all the products information from the selected store.
- Else:
  - The system returns an error that no such store exists.

Tests:

- Positive: Guest enters a hair products store and get its products information.
- Negative: Guest enters a hair products store and receive no results for products in this store (or the requested store doesn't exists)

### **2.2.2 Guest product search**

Actor: User

Pre-conditions: A guest is connected to the system

Post-conditions: None.

Flow:

- The guest types a certain product name/category/keyword on the search bar.
- The system iterates over the stores and looks for a product that fits the description.
- If OK:

- The system returns the results of all the products that fits the description.
- Else:
  - The system returns a message that there are no products with that description.

Tests:

- Positive: Guest searches for specific brand name t-shirts and the search results shows all the t-shirts available for the brand name.
- Negative: Guest types a specific brand name t-shirts and the search results shows a different brand

### **2.2.3 Guest adds product to the basket.**

Actor: User

Pre-conditions: guest is connected to the system

Post-conditions: the guest basket is filled with the products he added.

Flow:

- Guest adds a specific product to the basket.
- The system looks for the product id that corresponds to the store id.
- The system checks if it the product is available to buy.
- If OK:
  - The system adds the product to the basket.
- Else:
  - The system returns an error.

Tests:

- Positive: Guest adds an existing product to his basket, when viewing the basket the product appears.
- Negative: Guest adds an existing product to his basket, but the product doesn't adds to the basket

### **2.2.4.a Guest views the basket**

Actor: User

Pre-conditions: Guest is connected to the system

Post-conditions: Guest basket contains all the products he added.

Flow:

- The guest enters his cart and choose basket from his existing baskets.
- The system fetches the basket the associates with the specific guest.
- If OK:

- The system returns all the basket with the products in them.
- Else:
  - The system sends an error that no basket exists.

Tests:

- Positive: User adds a product, then view the basket and the added appears.
- Negative: Same as positive, only that the product doesn't appear.

#### **2.2.4.b Guest changes the basket**

Actor: User

Pre-conditions: A connected user have at least 1 product in his basket.

Post-conditions: The changes in the user's basket appear.

Flow:

- User enters his cart.
- The system fetches the basket the associates with the specific user.
- If OK:
  - The system returns all the basket with the products in them.
  - The user chooses a product to remove/change quantity from a specific basket.
  - The system looks for the specific product id
  - If OK:
    - The system removes the product from the user basket.
  - Else:
    - The system sends an error that no such product exists.
- Else:
  - The system sends an error that no basket exists.

Tests:

- Positive: User removes a product, then the product doesn't appear in the basket.
- Negative: User changes the quantity of a product, but the quantity stays the same.

#### **2.2.5.a Guest immediate purchase**

Actor: User

Pre-conditions: A connected user's basket have at least 1 product that he can buy according to the buying policies.

Post-conditions: User's purchase completed successfully.

Flow:

- The user request for immediate purchase of his products
- The system goes to each basket in the cart and does:
  - Checks if all items in basket are in stock. If they are – update them to new values.

- Sends payment request to external system.
- If payment is succeeded: cart is being emptied and user gets OK message. Else: items quantities are reversed and user gets error message.

Tests:

- Positive: User adds a product that eligible for immediate purchase, then purchase the product by clicking immediate purchase, then getting a purchase confirmation.
- Negative: User tries to buy an out of stock product through immediate purchase, and receive an error message.

### **2.2.5.b Guest bid request**

Actor: User

Pre-conditions: A connected user have at least 1 product that eligible for bidding.

Post-conditions: The user placed the desired bid successfully.

Flow:

- User choose a product for bidding.
- User place a bid.
- User waits for a response from the shop owner if:
  1. Owner accepts the offer – therefore he can buy the product
  2. Owner rejects – therefore he cannot buy the product.
  3. Owner responds with another offer.

Test:

- Positive: User chooses a product for bidding, the store owner rejects the offer and the user cannot purchase the product.
- Negative: User places a bid, owner accepts the offer but the user can't buy the product.

### **2.2.5.c Guest immediate purchase with secret discount**

Actor: User

Pre-conditions: A connected user basket have at least 1 product that he can buy according to the buying policies.

Post-conditions: The user's purchase completed successfully.

Flow:

- The system goes to each basket in the cart and does:
  - Checks if all items in basket are in stock. If they are – update them to new values ().
  - Sends payment request to external system.
  - If payment is succeeded: cart is being emptied and user gets OK message. Else: items quantities are reversed and user gets error message.

Tests:

- Positive: User adds a product that eligible for immediate purchase, then purchase the product with immediate purchase, then getting a purchase confirmation.
- Negative: User enters a coupon code, a message appears that the code is correct but the price doesn't change.

### **3 Registered user**

**Purchase Actions:**

**mostly the same as guest**

#### **3.1.1 Logging in the system**

Actor: member

Pre-conditions: A member with login status false exist in the system

Post- conditions: The member's login status changed to true

Flow:

- The member login to the system.
- The system checks if the member's username exist in the system
- if OK:
  - the system sends the member's password to the External service to confirm the member's password is valid
  - if OK:
    - the system checks if the member's login status is false
    - if OK:
      - the system changes the member's login status to true
    - else:
      - the system sends back an error message claimig the requested member is already logged in
- else:

- the system return an error message claiming the password isn't valid.
- else:
  - the system sends back an error message claiming there is no user with that username in the system.

Tests:

- Positive: The member successfully enters the system
- Negative: The member successfully enters the system with an incorrect password

### 3.1.2 Logging out the system

Actor: member

Pre-conditions: A registered member login status is true

Post-conditions: The member's login status changed from true to false.

Flow:

- The member sends a request to the system to logout.
- The system validates that the user status is logged-in.
- The system changes the user status to logged-out.
- The system returns a message with confirmation that the logout operation succeeded.

Tests:

- Positive: A member request to logout and his login status changes to false successfully
- Negative: A member request to logout and receives an error message -Log out has failed

### 3.2 Opening a shop

Actor: member

Pre-conditions: A member is logged in

Post-conditions: A new shop is created in the system and the member is defined as its first shop owner.

Flow:

- The member send a request to open a store.
- The system validates that the member is a registered member.
- If OK:
  - The system prompt the user for the appropriate information of the store.
  - The member fills the information about his new store.
  - The system creates a new store associated with the member that opened it.
  - If ok:
    - The system sets the access of the member as a Shop Owner.
  - Else:

- the system return an error that a shop could not be created, and that the member should contact admin
- Else:
  - the system returns an error that the user is not a member

Tests:

- Positive: A member request to open a new shop, fills the shop information and a new shop is created successfully.
- Negative: A member request to open a new shop, fills the information of the shop and receives an error message "a shop could not be created, please contact an admin".

## 4.1 Inventory Management

### a. adding a new product.

Actor: Shop owner

Pre-conditions: the product doesn't exist in the shop's product list, and the shop owner has logged to the system.

Post-conditions: the product does exist in the shop's product list.

Flow:

- The shop owner request to add a new product
- The system checks if the shop owner has login into the system.
- The system checks if shop owner has the right privileges for a specific store.
- If OK: The shop owner fills the relevant fields of a new product.
- The product is now added to the shop's product list.

Tests:

- Positive: After filling in the details by the shop owner, the product is now in the shop's item list.
- Negative: After filling in the details by the shop owner, the product is not in the shop's item list.



### **b. deleting a product.**

Actor: Shop owner

Pre-conditions: the product exists in the shop's product list, and the shop owner has logged to the system

Post-conditions: the product doesn't exist in the shop's product list.

Flow:

- The shop owner request to delete an existing product
- The system checks if the shop owner has login into the system.
- The system checks if the shop owner has the right privileges for a specific store.
- If OK: The shop selects the product to be deleted.
- The product is now deleted from the shop's product list.

Tests:

- Positive: After selecting a product to remove and selecting the removal option, the product is now not in the shop's item list.
- Negative: After selecting a product to remove and selecting the removal option, the product is still in the shop's item list.

### **c. Changing an existing product.**

Actor: Shop owner

Pre-conditions: the product exists in the shop's product list, and the shop owner has logged to the system

Post-conditions: None

Flow:

- The shop owner request to change an existing product
- The system checks if the shop owner has login into the system.
- The system checks if user has the right privileges for a specific store.
- If OK: The shop owner changes the relevant fields of a product.
- The product is now updated.

Tests:

- Positive: Shop owner changed the price to another positive number, change succeeded.

- Negative: Shop owner changed the price to a negative number, change failed.

## **4.2 Purchase policy and discounts**

### **a. defining purchase policy for the entire shop.**

Actor: Shop-owner

Pre-conditions: The shop owner has logged into the system

Post-conditions: The required purchase policy is now one of the shop's purchase policies.

Flow:

- The shop-owner chooses the required shop.
- The system checks if the shop owner has login into the system.
- The system checks if user has the right privileges for a specific store.
- If OK: The shop owner fills the relevant details.
- The new purchase policy is now one of the shop's purchase policies.

Tests:

- Positive: The shop-owner fills out a new purchase policy with compatible details and the policy is added to the shop's policies.
- Negative: The shop-owner fills out a new purchase policy with dates that have already passed, and an error message is presented, and the operation fails.

### **b. defining discount for the entire shop.**

Actor: Shop-owner

Pre-conditions: The shop owner has logged into the system

Post-conditions: The required discount is now one of the shop's discounts.

Flow:

- The shop-owner chooses the required shop.
- The system checks if the shop owner has login into the system
- The system checks if user has the right privileges for a specific store.
- If OK: The shop owner fills the relevant details.
- The new discount policy is now one of the shop's purchase policies.

Tests:

- Positive: The shop-owner fills out a new discount with compatible details and the discount is added to the shop's discounts.
- Negative: The shop-owner fills out a new discount with a negative discount value (+20%) and an error message is presented, and the operation fails.

### **c. changing purchase policy for the entire shop.**

Actor: Shop-owner

Pre-conditions: There's a purchase policy for the shop and the shop owner has logged into the system.

Post-conditions: The changed purchase policy is a part of the purchases policies.

Flow:

- The shop-owner chooses the required shop.
- The system checks if the shop owner has login into the system.
- The system checks if user has the right privileges for a specific store.
- If OK: The shop owner selects the purchase policy to be changed.
- The shop-owner fills out the required details.
- The policy is now with the updated details.

Tests:

- Positive: The shop-owner fills out the updated purchase policy with compatible details and the policy is updated and part of the shop's policies.
- Negative: The shop-owner fills out the updated purchase policy with dates that have already passed. An error message is presented to the user, and the operation fails.

#### **d. changing discount for the entire shop**

Actor: Shop-owner

Pre-conditions: The required discount is part of the shop's discounts, and the shop owner has logged into the system.

Post-conditions: The required discount is part of the shop's discounts.

Flow:

- The shop-owner chooses the required shop.
- The system checks if the shop owner has login into the system.
- The system checks if user has the right privileges for a specific store.
- If OK: The shop owner selects the discount policy to be changed.
- The shop-owner fills out the required details.
- The discount is now with the updated details.

Tests:

- Positive: The shop-owner chooses an existing discount and updates it with compatible details and the changes are saved.
- Negative: The shop-owner chooses an existing discount and updates it with a negative discount value (+20%) and an error message is presented, and the operation fails.

#### **e. defining purchase policy for specific product.**

Actor: Shop-owner

Pre-conditions: The shop's product list is not empty, and the shop owner has logged into the system.

Post-conditions: The required product includes the wanted purchase policy.

Flow:

- The shop-owner chooses the required shop.
- The system checks if the shop owner has login into the system.
- The system checks if user has the right privileges for a specific store.
- If OK: The shop owner chooses the desired product.
- The shop-owner fills the relevant details.
- The new purchase policy is now one of the shop's purchase policies.

Tests:

- Positive: The shop-owner fills out a new purchase policy with compatible details and the policy is added to the product's policies.
- Negative: The shop-owner fills out a new purchase policy with dates that have already passed, and an error message is presented, and the operation fails.

#### **f. defining discount for specific product.**

Actor: Shop-owner

Pre-conditions: The shop's product list is not empty, and the shop owner has logged into the system.

Post-conditions: The required product includes the wanted discount.

Flow:

- The shop-owner chooses the required shop.
- The system checks if the shop owner has login into the system.
- The system checks if user has the right privileges for a specific store.
- If OK: The shop owner chooses the desired product.
- The shop-owner fills the relevant details.
- The new discount policy is now one of the shop's discount policies.

Tests:

- Positive: The shop-owner fills out a new discount with compatible details and the discount is added to the product's discounts.
- Negative: The shop-owner fills out a new discount with a negative discount value (+20%) and an error message is presented, and the operation fails.

#### **g. changing purchase policy for a specific product.**

Actor: Shop-owner

Pre-conditions: There's a purchase policy for the product, and the shop owner has logged into the system.

Post-conditions: The required product includes the changed purchase policy.

Flow:

- The shop-owner chooses the required shop.
- The system checks if the shop owner has login into the system.
- The system checks if user has the right privileges for a specific store.
- If OK: The shop owner chooses the desired product and policy.
- The shop-owner changes the relevant details.
- The new purchase policy is now one of the shop's purchase policies.

Tests:

- Positive: The shop-owner chooses a purchase policy to update and fill it with compatible details and the policy is changed and is a part of the product's policies.
- Negative: The shop-owner chooses a purchase policy to update and fill it with dates that already passed, and the policy is changed and is a part of the product's policies.

#### **h. changing discount for a specific product.**

Actor: Shop-owner

Pre-conditions: There's a discount for the product, and the shop owner has logged into the system.

Post-conditions: The updated discount is part of the shop's discounts.

Flow:

- The shop-owner chooses the required shop.
- The system checks if the shop owner has login into the system.
- The system checks if user has the right privileges for a specific store.
- If OK: The shop owner chooses the desired product and discount.
- The shop-owner changes the relevant details.
- The new discount policy is now one of the shop's discount policies.

Tests:

- Positive: The shop-owner chooses an existing discount and updates it with compatible details and the changes are saved.
- Negative: The shop-owner chooses an existing discount and updates it with a negative discount value (+20%) and an error message is presented, and the operation fails.

#### **4.4 Nomination of a shop-owner**

Actor: Store owner

Pre-conditions: Nominated user is not a store owner, and a shop owner has logged into the system.

Post-conditions: Nominated user is a store owner.

Flow:

- The store owner request to nominate a new store owner

- The system checks if the shop owner has login into the system.
- The system checks if user has the right privileges for a specific store.
- If OK: Store owner selects a user to be a store owner.
- User becomes a store owner – if user isn't a store owner already.

Tests:

- Positive: Store owner tries to appoint a new user, not a store owner already, to be store owner.
- Negative: Store owner tries to appoint a new user, that is a store owner already, to be store owner.

#### **4.6 Nomination of a store manager**

Actor: Store owner

Pre-conditions: Nominated user is not a store owner or a store manager, and a shop owner has logged into the system.

Post-conditions: Nominated user is a store manager.

Flow:

- The store owner request to nominate a new store manager
- The system checks if the shop owner has login into the system.
- The system checks if user has the right privileges for a specific store.
- If OK: Store owner selects a user to be a store manager.
- User becomes a store manager– if user isn't a store owner/manager already.

Tests:

- Positive: Store owner tries to appoint a new user, not a store owner/manager already, to be store manager.
- Negative: Store owner tries to appoint a new user, that is already a store owner/manager, to be store manager.

#### **4.7 Permissions addition/change of store managers**

##### **a. Adding permissions for a store manager**

Actor: Store owner

Pre-conditions: The store manager has been assigned by the store owner (actor), and the shop owner has logged into the system.

Post-conditions: Added permissions for the store owner.

Flow:

- The store owner request to add a store manager permissions

- The system checks if the shop owner has login into the system.
- The system checks if user has the right privileges for a specific store.
- If OK: Store owner selects a store manager that he appointed.
- Store owner adds permissions to that store manager.

Tests:

- Positive: A store owner selects a store manager that has been appointed by him and adds permissions to him.
- Negative: A store owner selects a store manager that hasn't been appointed by him and adds permissions to him.

#### **b. Changing permissions for a store manager**

Actor: Store owner

Pre-conditions: The store manager has been assigned by the store owner (actor), and the shop owner has logged into the system.

Post-conditions: None.

Flow:

- The store owner request to change a store manager permissions
- The system checks if the shop owner has login into the system.
- The system checks if user has the right privileges for a specific store.
- If OK: Store owner selects a store manager that he appointed.
- Store owner selects permissions to that store manager to be changed.
- Store owner changes permissions to that store manager.

Tests:

- Positive: A store owner selects a store manager that has been appointed by him and changes his permission.
- Negative: A store owner selects a store manager that hasn't been appointed by him and changes his permission.

### **4.9 Store closing**

Actor: Store-founder

Pre-conditions: Store is opened and the store founder is connected

Post-conditions: Store is in-active.

Flow:

- Store founder requests to close his store.
- The system checks if the store founder has login into the system.
- The system checks if the store founder is the real store founder of the specific store

- if OK:
  - the system change to store's status to closed
- else:
  - the system returns an error that the user isn't the store founder of the requested store

Tests:

- Positive: Store founder requests to close his store and the store status changed to closed.
- Negative: Store founder requests to close a store that was opened by another user.

#### 4.11

##### a. Request store staff info

Actor: Store owner

Pre-conditions: The store owner has logged into the system.

Post-conditions: None

Flow:

- The store owner request to see a info about the staff of the store.
- the system checks if the store owner login to the system
- The system checks if user has the right privileges for a specific store.
- If OK: Store owner receives the info
- if no: error occurs.

Tests:

- Positive: Store owner requests staff info of his store.
- Negative: Store owner requests staff info a store that he isn't store owner of.

##### b. Request store managers permissions

Actor: Store owner

Pre-conditions: The store owner has logged into the system.

Post-conditions: None

Flow:

- The store owner request to see permissions of the store staff.
- the system checks if the store owner login to the system
- The system checks if user has the right privileges for a specific store.
- If OK: Store owner receives the permissions.
- if no error occurs.

Tests:

- Positive: Store owner requests managers permissions of his store.



- Negative: Store owner requests manager permissions of a store that he isn't store owner of.

#### **4.13 Receiving purchase history of a store**

Actor: Store owner

Pre-conditions: The store owner has logged into the system.

Post-conditions: None

Flow:

- The store owner request to see the purchase history of his store.
- the system checks if the store owner login to the system
- The system checks if user has the right privileges for a specific store.
- If OK: Store owner receives the purchases
- if no: error occurs.

Tests:

- Positive: Store owner requests purchase history of his store.
- Negative: User requests purchase history of a store that he isn't an owner of.

#### **5. Making actions based on privileges**

Actor: Shop manager

Pre-conditions: The store owner has logged into the system.

Post-conditions: None

Flow:

- Shop manager chooses an action to perform that is allowed by his privileges.
- the system checks if the store owner login into the system
- The system checks if user has the right privileges for a specific store.
- If OK: Action is performed.
- If not OK: Action is not performed.

Test:

- Positive: Shop manager tried to perform an action that is in his privileges.
- Negative: Shop manager tried to perform an action that is not in his privileges.

#### **6.4 Obtain purchase history of a store and buying users**

### **a. Obtain purchase history of a store**

Actor: Admin

Pre-conditions: The Admin has logged into the system.

Post-conditions: None.

Flow:

- The Admin requests purchase history of a certain store.
- The system checks if the admin login into the system
- The system checks if user has the right privileges for a specific store.
- If OK:
  - the admin receive all the purchase history of the buying users from the specific store
- else:
  - the admin receive a message he doesn't have the right privileges for this action

Test:

- Positive: System manager requests purchase history of an existing store.
- Negative: System manager requests purchase history of a non-existing store.

### **b. Obtain purchase history of a buyer**

Actor: Admin

Pre-conditions: The Admin has logged into the system.

Post-conditions: None.

Flow:

- The Admin requests purchase history of a certain buyer (user).
- The system checks if the admin login into the system
- The system checks if user has the right privileges for a specific store.
- If OK:
  - the admin receive the requested information
- else:
  - the admin receive a message he doesn't have the right privileges for this action

Tests:

- Positive: System manager requests purchase history of an existing user.
- Negative: System manager requests purchase history of a non-existing user.

## **Terms Dictionary**

- User – User of the system, could be a guest, a member or an admin.

- Guest – a User that hasn't logged in the system and does not have the same privileges and access of system functionality that member has.
- Member – a User that logged in the system and could benefit of the full system functionality as a member.
- Member ID – an identifier to every logged in Member in our system it will be the members name.
- Admin – a User that logged in the system and acts as an administrator of the of the system.
- Password validator – an external service that checks if a password passes certain criterias.
- Payment – a bridge to the payment method that will be connected to the system.
- Store – an object that the Member can create, will hold products and will have a policy in regards to the store.
- Store founder – a state of a member who has opened a store, which lets him manage his store to its full functionality.
- Store owner – a state of a member who got permission from a store founder to be an owner at his store, doesn't have all the store founder functionality.
- Store manager – a state of a member who got permission from a store owner to be a manager at his store. The state lets the member certain access to the store management functionality.
- Cart – an object used by the User to hold Baskets.
- Basket – an Object that gathers products from a specific store.
- Product – an Item that can appear in stores to be sold.
- policy – an object in every store that dictates how the store will behave in both Purchase and Discount policies.
- Purchase policy – a set of rules that dictates how the store will behave in regards of the rules of the Purchases.
- Discount policy – a set of rules that dictates how the store will behave in regards to Discounts on products.
- Access – a set of permissions that a user has based on his status of a specific store.
- Purchase – a standard Purchase between a user and a store

- Auction – a way of engaging a transaction by the store of a limited time Purchase for the highest bidder, the highest Member bidder on that Product will win the Purchase and can claim it.
- Transaction – a successful Purchase between a guest or a member and a store.
- Transaction history – a set of successful Purchases that happened, stating their date and their details, each successful Purchase is saved in the transaction history.
- Message - a string of Notification or Request that is sent in order to communicate.
- Notification - a Message that is sent from a store to user to notify him when certain event is happening.
- Request – a Message that is send from the user to the store when the user wants to purchase certain item from the store.
- External Management – a group of external services that the system is holding in order to preform actions in the system.
- Bid – an offer that a User of type Member can offer to a store, the store needs its owners' approval to approve the offer, if approved, the member can claim the purchase.
- Lottery - a way of engaging a transaction by the store, the store uploads a product and make it a lottery product, Members of the system can place their bid on the product, when the sum of all bidding reached the product price all the bidders has a chance to win the purchase with their bid on the item.