**System requirements:**

**1.1 starting the market system -**

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

An admin logging into the systrem using his password and username of a system manager type user,

the system checks if the user exists in the database, if it does, the system checks if the user is a system\_manager type of user and if it does the system sends a message to payment and supliment services - if the connection succeed a message is being return from those services to the system and the system sends a message to the the user that he had connected successfuly.

pre conditions: a non\_system\_manager user exists in the database

a regular user which isn’t an admin logging into the system using his password and username, the system checks if the user exists in the database, if it does the system checks if th user is a system\_manager type of user and returns an error message that this user doesn’t have premmision to start the system – only a system manager type of user can start this system.

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

pre conditions: each external service connected to the system has at least one connection available while the system is running.

Change - ???

**1.3 payment -**

pre condition: an unpaid transaction details exist in the database

the system checks the connection to the payment system – if the payment service sends back a message it’s connected to the system the system sends to the payment service the transaction details.

The payment service send back to the system a message that the payment has set successfuly

**1.4 suppliment - ????**

**1.5 real time notifications -**

Shop owner -

pre condition: the shop owner is logged in to the system

after a customer buys a product from the shop owner’s shop, the system fetch all the logged shop owners and sends an notification to each of them which says "a customer bought a <product name> from your store!"

after the shop has shut down, the system fetch all the logged shop owners and sends an notification to each of them which saying "the shop has successfuly closed"

after a shop has opened, the system fetch all the logged shop owners and sends an notification to each of them which saying "the store has successfuly opened".

pre condition: the store has at least 2 shop owners, the removed shop owner as no nominators under it.

the system remove the nomination of the shop owner – if it has removed succesfully the system sends to the former shop owner a message "your nomination in <name of the shop> has been removed".

User-

Pre condition: 2 users connected to the system

user 1 sends a message to user2 using the system - the system recives the message and checks if user2 is connected. If he does, the system send a notification with the message to user2.

**1.6 Delayed notifications -**

pre conditions: a not logged user with delayed notifications in the database

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

after the user login to the system, the system check if there are any delayed messages belong to this uer in the database – if there are any, the system fetch the messages and send them as notifications to the user.

**2 -Users**

**a) guest**

**2.1.1 Guest Visit**

Actor: User

Pre-condition:

Post-condition: User defined as a guest.

Flow:

* User enters the market.
* User gets a basket assigned.

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-condition: User is crawling the market.

Post-condition: User loses his basket.

Flow:

* User closing the website.

Tests:

* Positive: User adds a product to his basket, exiting the market and then returns to an empty basket.
* Negative: Same as the positive test, but the basket stays the same.

**2.1.3 Registration to the system**

Actor: User

Pre-conditions: User is defined as a guest.

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

Flow:

A guest user press on the Sign-up button fills the information needed and a new member is created in the system.

Tests:

Success: A guest user fills the sign-up info sheet with a unique user-name, and a new member account is created in the system.

Fail: 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: User is defined as a guest and has already registered to the system.

Post- conditions: The User status is changed from Guest to Member.

Flow:

A guest user press on the Sign-in button fills the user-name and password bars and log in his account

Tests:

Success: A guest clicks on the sign-in button, fills username and passwords bars and successfully log in to his account.

Fail: A guest clicks on the sign-in button, fills username and passwords bars incorrectly and receives an error message "User name or password are incorrect".

**b)**

**2.2.1 Guest Information fetching**

Actor: User

Pre-condition: User defined as a guest.

Post-condition: User gets the information about the stores and products.

Flow:

* User queries information about certain products or stores.

Tests:

* Positive: User search for hair products and gets related results.
* Negative: User search for a specific store but gets a wrong result.

**2.2.2 Guest product search**

Actor: User

Pre-condition: User defined as guest.

Post-condition: User gets the appropriate result according to his search.

Flow:

* User types a certain product name/category/keyword on the search bar.
* User gets the information about the products.

Tests:

* Positive: User searches for specific brand name t-shirts and the search results shows all the t-shirts available for the brand name.
* Negative: User 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-condition: User defined as a guest.

Post-condition: User basket filled with the products he added.

Flow:

* User searches for a product.
* User adding a product by pressing "Add to basket" button.

Tests:

* Positive: User adds a product to his basket, when viewing the basket the product appears.
* Negative: Same as the positive, but the product doesn’t adds to the basket

**2.2.4.a Guest views the basket**

Actor: User

Pre-condition: User defined as a guest.

Post-condition: User basket contains all the productse he added.

Flow:

* User clicks on the "Basket" button.
* User can view all the products he added.

Tests:

* Positive: User adds a product, then clicks the "Basket" button where the product he added appears.
* Negative: Same as positive, only that the product doesn’t appear.

**2.2.4.b Guest changes the basket**

Actor: User

Pre-condition: User have atleast 1 product in his basket.

Post-condition: User changes to the basket success.

Flow:

* User press the "Basket" button.
* User removes/changes the quantity of a product

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-condition: User basket have at least 1 product that he can buy according to the buying policies.

Post-condition: User purchase completed successfully.

Flow:

* User clicks on immediate purchase button.
* User fill all the information needed for the payment method.
* User gets a confirmation that his purchase succeeded.

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 a product through immediate purchase but the product is out of stock.

**2.2.5.b Guest bid request**

Actor: User

Pre-condition: User have at least 1 product that eligible for bidding.

Post-condition: 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 reject the offer and the user cannot purchase the product.
* Negative: User places a bid, owner accepts the offer but the user cant buy the product.

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

Actor: User

Pre-condition: User basket have at least 1 product that he can buy according to the buying policies.

Post-condition: User purchase completed successfully.

Flow:

* User clicks on immediate purchase button.
* User enter the coupon code that he possesses.
* A message displayed if the coupon code eligible or not.
* User fill all the information needed for the payment method.
* User gets a confirmation that his purchase succeeded.

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 enters a coupon code, a message appears that the code is correct but the price doesn’t change.

**3 registered user**

**3.1 Logging out the system**

Actor: User

Pre-conditions: User is in a member status and is logged in

Post- conditions: The User status is changed from Member to Guest.

Flow:

A member user press on the Log out button and his status changed from Member to User.

Tests:

Success: A member clicks on the Log out button and exits his state as a Member Successfully.

Fail: A member clicks on the Log out button and receives an error message "Log out has failed

**3.2 Opening a shop**

Actor: User

Pre-conditions: User is in a member status and is logged in

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

Flow:

A member user clicks on the create shop button fills in the info about the shop and a new shop is created and the first shop owner is the member

Tests:

Success: A member clicks on the Create shop button, fills the shop information and a new shop is created successfully.

Fail: A member clicks on the Create-shop button fills the information of the shop and receives an error message "a shop could not be created, please contact an admin".

ניהול מלאי: בעל-חנות יכול לנהל את מלאי המוצרים של חנות שבבעלותו. ניהול מלאי כולל הוספה והסרת מוצרים, ושינוי פרטיהם.

**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.

Post-Conditions: the product does exist in the shop’s product list.

Main flow:

1. The shop owner selects the adding option.
2. The shop owner fills the relevant fields of a new product.
3. The product is now added to the shop’s product list..

Alternative flow:

1. The shop owner selects the adding option.
2. The shop owner fills the relevant fields of a new product.
3. The system alerts the owner that one or more of the fields is illegal and doesn’t add the product.

Positive test: After filling in the details by the shop owner, the product is now in the shop’s item list.

Negative test: 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.

Post-Conditions: the product doesn’t exist in the shop’s product list.

Main flow:

1. The shop owner selects the item he wishes to delete.
2. The shop owner selects the deleting option.
3. The product is now not on the shop’s product list.

Positive test: After selecting a product to remove and selecting the removal option, the product is now not in the shop’s item list.

Negative test: 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.

Post-Conditions: None

Main flow:

1. The shop owner selects the changing option on the wanted product.
2. The shop owner changes the relevant fields of the product with compatible data.
3. The product is now changed with the new info from the owner.

Alternative flow:

1. The shop owner selects the changing option on the wanted product.
2. The shop owner changes the relevant fields of the product with incompatible data.
3. The product is not being changed and an error message is displayed to the user.

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

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

.שינוי מדיניות קניה והנחה של חנות: בעל-חנות רשאי להגדיר ולשנות את סוגי הרכישה וההנחה האפשריים בחנות שבבעלותו, ואת מדיניות הקניה וההנחה עבור מוצרים בחנות.

אילוץ: כללי הרכישה של חנות לא סותרים אילוצי עקיבות )consistency )שנקבעו על ידי המייסד

**4.2 Purchase policy and discounts**

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

Actor: Shop-owner

Pre-Conditions: None

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

Main flow:

1. The shop-owner chooses the required shop.
2. The shop-owner chooses the option of defining a new purchase policy.
3. The shop-owner fills out the required details.
4. The new purchase policy is now one of the shop’s purchase policies.

Positive test: The shop-owner fills out a new purchase policy with compatible details and the policy is added to the shop’s policies.

Negative test: 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: None

Post-Conditions: The required discount is now one of the shop’s discounts.

Main flow:

1. The shop-owner chooses the required shop.
2. The shop-owner chooses the option of defining a new shop’s discount.
3. The shop-owner fills out the required details.
4. The new discount is now one of the shop’s discounts.

Positive test: The shop-owner fills out a new discount with compatible details and the discount is added to the shop’s discounts.

Negative test: 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.

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

Main flow:

1. The shop-owner chooses the required shop.
2. The shop-owner chooses the purchase policy and the option of update its fields.
3. The shop-owner fills out the required details.
4. The policy is now with the updated details.

Positive test: 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 test: 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.

Post-Conditions: The required discount is part of the shop’s discounts.

Main flow:

1. The shop-owner chooses the required shop.
2. The shop-owner chooses the option of updating a new shop’s discount.
3. The shop-owner fills out the required details.
4. The discount is now with the updated details.

Positive test: The shop-owner chooses an existing discount and updates it with compatible details and the changes are saved.

Negative test: 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.

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

Main flow:

1. The shop-owner chooses the required shop and the required item.
2. The shop-owner chooses the option of defining a new purchase policy.
3. The shop-owner fills out the required details.
4. The new purchase policy is now one of the product’s purchase policies.

Positive test: The shop-owner fills out a new purchase policy with compatible details and the policy is added to the product’s policies.

Negative test: 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.

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

Main flow:

1. The shop-owner chooses the required shop and the required product.
2. The shop-owner chooses the option of defining a new product’s discount.
3. The shop-owner fills out the required details.
4. The new discount is now one of the product’s discounts.

Positive test: The shop-owner fills out a new discount with compatible details and the discount is added to the product’s discounts.

Negative test: 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.

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

Main flow:

1. The shop-owner chooses the required shop and the required item.
2. The shop-owner chooses the option of updating an existing purchase policy.
3. The shop-owner fills out the required details.
4. The updated purchase policy is now one of the product’s purchase policies.

Positive test: 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 test: 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.

Post-Conditions: The updated discount is part of the shop’s discounts.

Main flow:

1. The shop-owner chooses the required shop and the required product.
2. The shop-owner chooses the option of updating an existing product’s discount.
3. The shop-owner fills out the required details.
4. The discount is now with the updated details.

Positive test: The shop-owner chooses an existing discount and updates it with compatible details and the changes are saved.

Negative test: 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 store-owner**

Actor: Store owner

Pre-conditions: Nominated user is not a store owner.

Post-conditions: Nominated user is a store owner.

Flow:

1. Store owner selects a user to be a store owner.
2. User becomes a store owner – if user isn’t a store owner already.

Positive test: Store owner tries to appoint a new user, not a store owner already, to be store owner.

Negative test: Store owner tries to appoint a new user, that is a store owner already, to be store owner.

**4.6**

Actor: Store owner

Pre-conditions: Nominated user is not a store owner or a store manager.

Post-conditions: Nominated user is a store manager.

Flow:

1. Store owner selects a user to be a store manager.
2. User becomes a store manager– if user isn’t a store owner/manager already.

Positive test: Store owner tries to appoint a new user, not a store owner/manager already, to be store manager.

Negative test: 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).

Post-conditions: Added permissions for the store owner.

Flow:

1. Store owner selects a store manager that he appointed.
2. Store owner adds permissions to that store manager.

Positive test: A store owner selects a store manager that has been appointed by him and adds permissions to him.

Negative test: 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).

Post-conditions: None.

Flow:

1. Store owner selects a store manager that he appointed.
2. Store owner selects a permission to change.
3. Store owner changes permissions to that store manager.

Positive test: A store owner selects a store manager that has been appointed by him and changes his permission.

Negative test: 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.

Post-conditions: Store is in-active.

Flow:

1. Store founder requests to close his store.

Positive test: Store founder requests to close his store.

Negative test: 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: None

Post-conditions: None

Flow:

1. Store owner requests info about the staff of the store.
2. Store owner receives the info if no error occurs.

Positive test: Store owner requests staff info of his store.

Negative test: Store owner requests staff info a store that he isn’t store owner of.

**b. Request store managers permissions**

Actor: Store owner

Pre-conditions: None

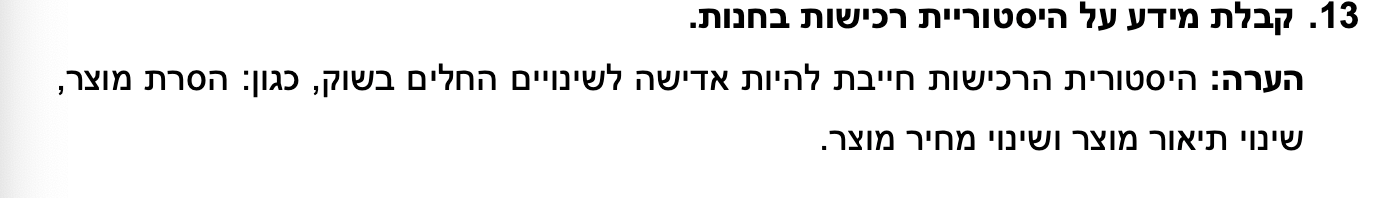
Post-conditions: None

Flow:

1. Store owner requests the permissions of the store staff.
2. Store owner receives the permissions if no error occurs.

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

Negative test: 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: None

Post-conditions: None

Flow:

1. Store owner requests the store purchase history.
2. Store owner receives the purchases if no error occurs.

Positive test: Store owner requests purchase history of his store.

Negative test: -

**5. Making actions based on privileges**

Actor: Shop manager

Pre-conditions: None

Post-conditions: None

Flow:

1. Shop manager chooses an action to perform that is allowed by his privileges.
2. Action is performed.

Alternative flow:

1. Shop manager chooses an action to perform that is not allowed by his privileges.
2. Action is not performed.

Positive test: Shop manager tried to perform an action that is in his privileges.

Negative test: 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: System Manager

Pre-conditions: None.

Post-conditions: None.

Flow:

1. System manager requests purchase history of a certain store.

Positive test: System manager requests purchase history of an existing store.

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

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

Actor: System Manager

Pre-conditions: None.

Post-conditions: None.

Flow:

1. System manager requests purchase history of a certain buyer (user).

Positive test: System manager requests purchase history of an existing user.

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