Sadna Express

סדנא ליישום פרויקט תוכנה

מגישים:

שי קרסנר - 209120278

נוגה שוורץ - 207687849

דינה אגפוב - 207181629

טל גלמור - 318416575

גיל חאיס – 207919374

רדואן גאנם - 322509951

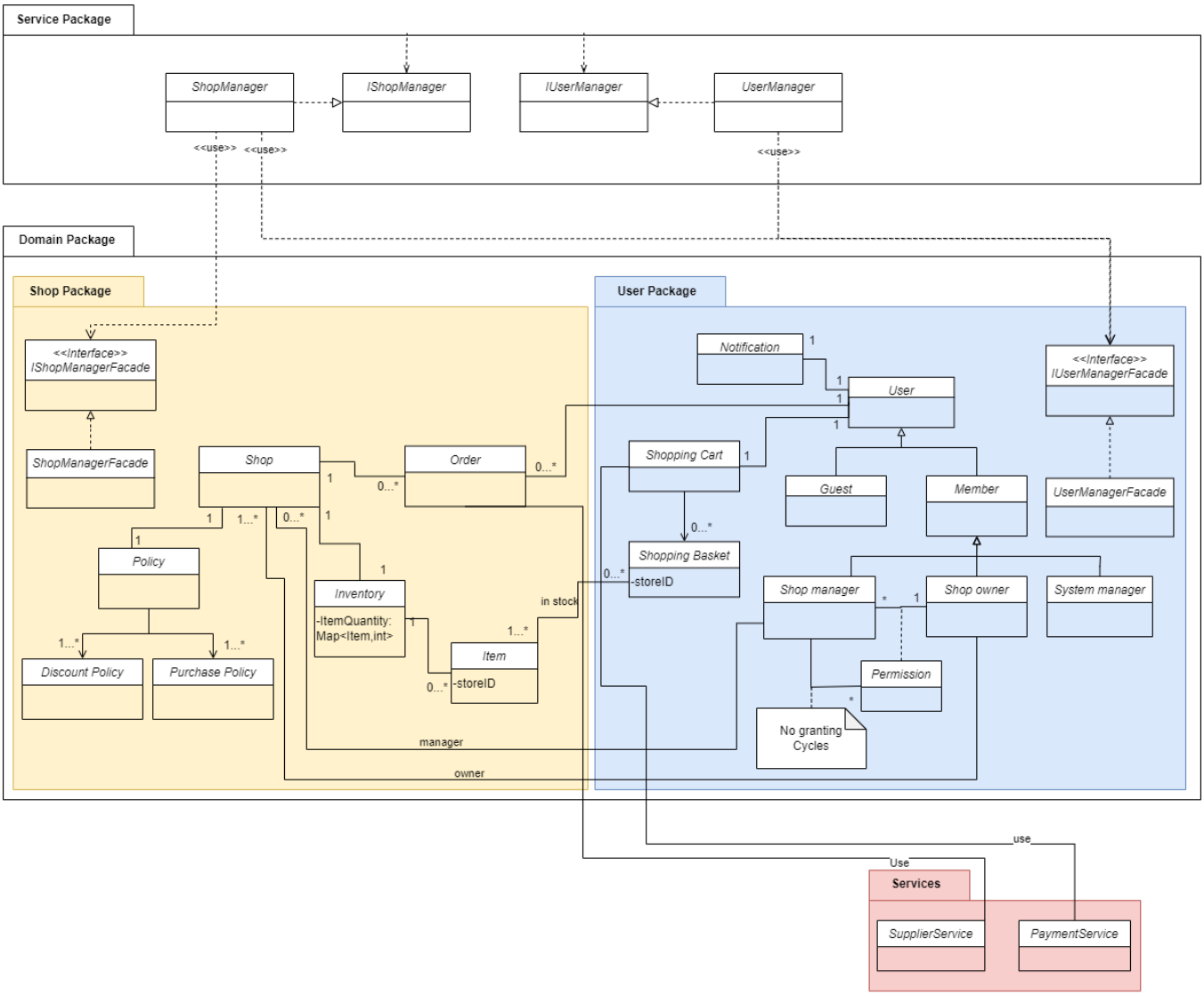
תוכן עניינים

מילון מונחים - Glossary:

מילון מושגים עבור תרחישי השימוש במערכת.

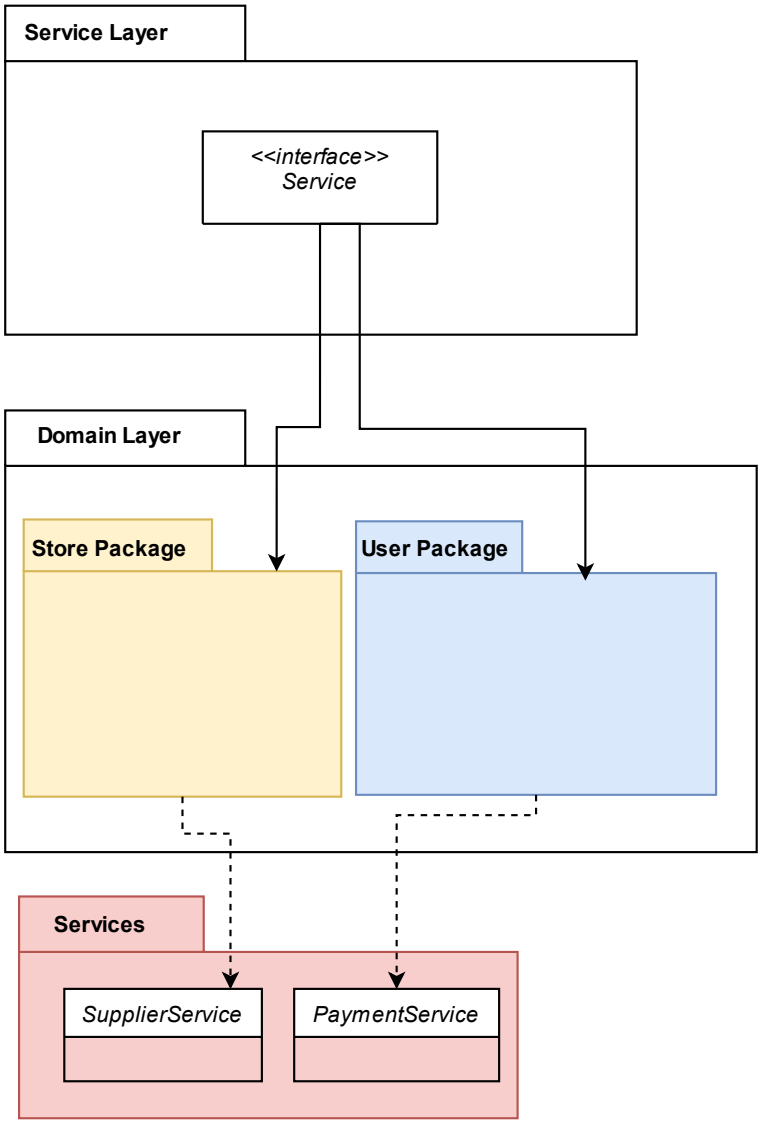
1. Trading system - מערכת המסחר, האתר המרכזי שמציג את כלל החנויות.
2. Login - כניסה למערכת עבור מנויים בעזרת פרטי זיהוי שנקבעו מראש.
3. User - משתמש במערכת - יכול להיות אורח/מנוי
4. Guest - מבקר באתר שאין לו חשבון.
5. Member - משתמש הרשום למערכת המסחר. משתמש מנוי היכול לשמש בתפקידים שונים בשוק - בעל-חנות, מנהל-חנות.
6. System manager - מנהל מערכת המסחר. אחראי על הניהול השותף של המערכת ובעל הרשאות אדמין באתר.
7. Store owner - בעל חנות והוא גם מנהל בחנות עם כל ההרשאות.
8. Store manager - מנהל בחנות עם הרשאות מוגבלות.
9. Store Founder - מייסד החנות , בעל החנות הראשון.
10. User notification center - מקום בו מאוחסנים כלל ההתראות עבור משתמש.
11. User shopping cart - עגלת הקניות של משתמש, המקום בו מאוחסנים כלל המוצרים שהמשתמש מעוניין לקנות, אוסף של סלי קניות מחנויות שונות.
12. User shopping basket - סל קניות של משתמש עבור חנות ספציפית במערכת המסחר.
13. Store policy – מכיל את המדיניות של החנות.
14. Discount policy – מפרט הנחה מסוימת שקיימת ברכישה.
15. Purchase policy – מפרט הגבלה מסוימת שקיימת ברכישה.
16. Inventory – המלאי של חנות ספציפית, כלומר מכיל את הזמינות של המוצרים של החנות.
17. Item – פריט של חנות ספציפית. מכיוון שלכל חנות יש תיאור ומחיר משלה לכל מוצר, אז המוצרים של חנויות שונות יהיו אובייקטים שונים זה מזה.
18. External services - שירותים חיצוניים שמערכת המסחר נעזרת בהן על מנת לנהל אותה - כמו שירותי אספקה ושירותי רכישה ותשלומים.

מודל מחלקות לבן:



Sadna Express

דיאגרמת ארכיטקטורה:



הטלת אחריות לקיום אילוצי נכונות

מס' אילוץ	אילוץ נכונות	הטלת אחריות לקיום האילוץ
1	למנוי יש שם יחיד המזהה אותו במערכת.	תרחיש שימוש - Register To the system
2	יש למערכת לפחות מנהל אחד. מנהל- מערכת חייב להיות מנוי (עבר תהליך רישום).	תרחיש שימוש – Initialization of the trading system
3	בעל-חנות או מנהל-חנות חייב להיות מנוי.	תרשים מחלקות - לפי יחס הירושה
4	פעולות בשוק מבוצעות רק ע"י משתמשים המבקרים בשוק.	תרחישי שימוש – כלל פעולות השוק הרלוונטיות למנויים מותנות ברישום למערכת
5	לחנות פעילה (שאינה סגורה) חייב להיות לפחות בעל-חנות אחד.	תרשים מחלקות - לפי קרדינליות הקשר (לפחות אחד)
6 א	לחנות חייבים להיות מוגדרים תהליכי קנייה (רכישה) והנחה. יתכן ויש ברירות מחדל עבור אופן קנייה ועבור סוג הנחה.	תרשים מחלקות – אובייקט store חייב להחזיק policy
6 ב	לחנות חייבים להיות מוגדרים מדיניות רכישה ומדיניות הנחה. תיתכן ברירת מחדל של כללי רכישה או חישוב הנחה כמו למשל "אין מגבלות רכישה או הנחה".	תרשים מחלקות – אובייקט store חייב להחזיק policy
7	לקונה יש עגלת קניות יחידה, המורכבת מאוסף כל סלי הקנייה שלו. לקונה יש לכל היותר סל קניות יחיד עבור חנות כלשהי.	תרשים מחלקות - לפי קרדינליות הקשר (קשר יחיד ליחיד) שדות storeID בItem,ShoppingBasket
8	עגלת הקניות של קונה (אורח או מנוי) הינה בבעלותו הבלעדית ואינה ניתנת לשינוי על ידי שום משתמש אחר.	תרשים מחלקות – shopping cart מקושרת לuser אחד בלבד
9	ניתן לקנות פריטים של מוצר בחנות לכל היותר בכמות הקיימת במלאי.	תרשים מחלקות – inventory מחזיקה את הitems לפי כמותם.
10 א	ניתן לגבות כסף מקונים רק עבור עסקאות שבוצעו ולגבות רק את הסכומים שהוצהרו.	תרחיש שימוש מערכת : payment
10 ב	תהליך קנייה מסתיים בהצלחה רק אם שולם הסכום הדרוש ורק אם האספקה אושרה	תרחיש שימוש מערכת : payment, supply
10 ג	מוכר יכול לקבל תשלום רק כתוצאה מתהליך קנייה מוצלח	לאחר שתהליך קניה יסתיים בהצלחה (10 ב) המערכת תבצע העברת תשלום למוכר

11 א	נדרש לפחות קשר אחד לשירות גביית כספים	תרשים מחלקות - קיים אובייקט payment service עם קרדינליות של לפחות אחד
11 ב	נדרש לפחות קשר אחד לשירות אספקה	תרשים מחלקות - קיים אובייקט supplier service עם קרדינליות של לפחות אחד

תרחישי שימוש – Use Cases :

System use cases:

Use case: Initialization of the trading system.

- Actor: user
- Precondition: user is logged in.
- Postcondition: trading system is initialized.
- Parameter: none
- Actions:
    1. User request to open (initialize) the trading system.
    2. System checks that the user has permissions to perform this operation.
    3. System if user has permissions – checks whether there is a connection to payment and supply services and there is a system manager.
    4. System if action succeeded – opens the trading system for clients.

Participants	Parameters	Expected Result	Scenario
Member	User is system Manager	Trading system is initialized	Good
Member	User isn't System Manager	Trading system staying closed	Bad
External services	There is a connection to payment and supply services	Trading system is initialized	Good
External services	There isn't a connection to payment and supply services	Trading system staying closed	Bad

Use case: Update connection with external services.

- Actor: System Manager
- Precondition: user is logged in
- Postcondition: external services are connected to the trading system.
- Parameter: old external service, new external service
- Actions:
    1. System checks that the user has permissions to perform this operation.
    2. System if user has permissions – checks whether is it possible to connect to the new external service.
    3. System if action succeeded – checks that old and new services have the same API.
    4. System if action succeeded – waits until there is no use in old service.
    5. System replaces the old service with the new one.
    6. System sends an indication to the user about the exchange.

Participants	Parameters	Expected Result	Scenario
System Manager	old external service, new good external service	Service replaces successfully	Good
System Manager	old external service, new bad external service (unavailable service/service doesn't exist)	Service is not replaced	Bad

Use case: Payment.

- Actor: System
- Precondition: user triggered a request to make a payment , User shopping cart is valid and not empty.
- Postcondition: Store received the payment from the system.
- Parameter:
  - Transaction details
  - User details
  - User shopping cart details
- Actions:
  1. System sends request to payment external service.
  2. System waits for service response.
  3. System if action succeeded –
     - Sends user shopping cart and user details to supply external service.
     - Sends positive indication for the user and emptying the user shopping cart.
     - Store received the payment from the system.
  4. System if action failed–Send negative indication to the user.

Participants	**Parameters**	**Expected Result**	**Scenario**
user	Good transaction details	Payment completed successfully	Good
user	Bad transaction details (for example wrong ccv)	Payment is not completed	Bad

Use case: Supply.

- Actor: System
- Precondition: Payment completed (for some shopping cart)
- Postcondition: Supply process started (for some shopping cart)
- Parameter:
  - User details
  - User shopping cart details
- Actions:
  1. System sends user shopping cart and user details to supply external service.
  2. System waits for service response.
  3. System if action succeeded sends positive indication for the user that supply process has begun successfully.
  4. System if action failed– sends negative indication for the user.

Participants	**Parameters**	**Expected Result**	**Scenario**
System	Good user shopping cart and user details	Receiving a supply confirmation from the service	Good
System	Bad user shopping cart and user details (errors in user details - bad address ext.)	Receiving error from the supply service	Bad

Use case: User online notifications.

- Actor: System
- Precondition:
  - Message has received or
  - purchase has occurred or
  - Store opened or closed or
  - Subscription has been removed.
- Postcondition: Message received by the user if logged in, or added to the user notification system if not logged in.
- Parameter:
  - User Message with notification info
  - User details
- Actions:
  1. System checks if the user with the user details is logged in.
  2. System if logged in - sends the message to the user.
  2. System if not logged in - adds the message to the user notification center which the user will receive when logs into the system.

Participants	**Parameters**	**Expected Result**	**Scenario**
System	purchase has occurred to a specific store; store manager is connected to the trading system	The store manager receives a notification about the purchase	Good
System	purchase has occurred to a specific store; store manager is connected to the trading system	The store manager does not receive a notification about the purchase	Bad

Use case: User offline notifications.

- Actor: System
- Precondition:
  - Message has received or
  - purchase has occurred or
  - Store opened or closed or
  - Subscription has been removed.
- Postcondition: Message received by the user if logged in, or added to the user notification system if not logged in.
- Parameter:
  - User Message with notification info
- Actions:
  1. System checks if the user with the user details is logged in.
  2. System if logged in - sends the message to the user.
  3. System if not logged in - adds the message to the user notification center which the user will receive when logs into the system.

Participants	Parameters	Expected Result	Scenario
System	purchase has occurred to a specific store; store manager is not connected to the trading system.	The store manager logs in to the system and accepts a notification about the purchase	Good
System	purchase has occurred to a specific store; store manager is not connected to the trading system	The store manager logs in to the system and does not accept a notification about the purchase	Bad

Users use cases:

Guest user:

Use case: Enter the system.

- Actor: User
- Precondition: User not logged in
- Postcondition: -
- Parameter: none
- Actions:

1. User entering the home page of the system.
2. System Defines user as Guest-user.
3. System assigns the user shopping cart.

Participants	Parameters	Expected Result	Scenario
User	The user entered to the system, and the system is initialized	Successfully entered the system and accepts a shopping cart	Good
User	The user entered to the system, and the system is not initialized	The system did not upload correctly to the user	Bad

Use case: Exit from the system.

- Actor: User
- Precondition: User not logged in
- Postcondition: -
- Parameter: none
- Actions:

  1. User exits from the system.
  2. System Delete the user's shopping cart.
  3. System Delete User from Guest's list.

Participants	**Parameters**	**Expected Result**	**Scenario**
User	-	Successfully exit the system and his shopping cart deleted	Good
User	-	Successfully exit the system and did not delete from Guests's list	Bad

Use case: Register To the system.

- Actor: User
- Precondition: User not logged in and not registered to the system before
- Postcondition: System registered the user to the system.
- Parameter: Identifying details
- Actions:

  1. User enters the system.
  2. User register to the system by giving Identifying details.
  3. System validates username is unique in the system.
  4. System creates new member.
  5. System assigns registered user a new empty shopping cart.

Participants	**Parameters**	**Expected Result**	**Scenario**
User	Unique username , strong password , valid id	Successfully registered to the system	Good
User	username that already in the system, strong password , valid id	Username already used in the system , msg will be thrown	Bad

Use case: Login To the system.

- Actor: User
- Precondition: User not logged(guest) in and registered to the system before
- Parameter: username and password
- Actions:

1. User enters the system.
2. User login to the system using the username and password.
3. System defines user as member.
4. System will update the shopping cart , according to member's shipping history.
5. System will show all offline notification of the user.

Participants	**Parameters**	**Expected Result**	**Scenario**
User	Unique username , strong password	Successfully logged in to the system	Good
User	Username and password that don't match the username and password from the system's db.	Wrong password / username entered and the log in didn't work	Bad

Use case: Getting information about stores in the market and the products in the stores.

- Actor: User
- Precondition: User not logged in
- Postcondition: User received the info.
- Parameter: none
- Actions:
    1. User asked from the system to accept information about specific store.
    2. System if store doesn't exist , raises appropriate message.
    3. System displays information

Participants	Parameters	Expected Result	Scenario
User	-	System sends user info he requested	Good
User	-	System didn't send user info he requested	Bad

Use case: Search Products by general search or filters.

- Actor: User
- Precondition: User not logged in
- Postcondition: User received the list of items according to the search.
- Parameter: Info of product
- Actions:
  1. User asks from the system to search according to the info of the product he has.
  2. System if store doesn't exist , raises appropriate message.
  3. System returns result based on the info provided.

Participants	Parameters	Expected Result	Scenario
User	Search on known product in the searching tool of the system	System shows list of products according to the info provided	Good
User	Enter product that didn't exists	System shows list of products that unrelated to the info provided	Bad

Use case: Saving item in the shopping cart for some store.

- Actor: User
- Precondition: User not logged in
- Postcondition: System saved the item in shopping cart.
- Parameter: -
- Actions:

  1. User puts items in the shopping cart.
  2. System updates the shopping cart.

Participants	**Parameters**	**Expected Result**	**Scenario**
User	-	System saves the shopping cart of the user for next time	Good
User	-	System don't save the shopping cart of the user and he lost it	Bad

Use case: Checking the content of the shopping cart and making changes.

- Actor: User
- Precondition: User not logged in.
- Precondition: User makes changes in his shopping cart.
- Parameter: Shopping list
- Actions:

  1. User puts item in his shopping cart.
  2. System Displays the shopping cart.

Participants	**Parameters**	**Expected Result**	**Scenario**
User	-	when checking the content of the cart , the item appears	Good
User	-	when checking the content of the cart , the item doesn't appear	Bad

Use case: Making a purchase of the shopping cart.

- Actor: User
- Precondition: User not logged in, User's Shopping cart is not empty, payment method is provided.
- Postcondition: System validate the order and remove the purchased items from the store they brought from.
- Parameter: Shopping list
- Actions:

1. User puts items in his shopping cart.
2. User starts the process of checking the shopping cart.
3. System processes the purchase.
4. User inserts payment method.
5. System validates payment.
6. System send receipt to the user.

Participants	**Parameters**	**Expected Result**	**Scenario**
User	Valid payment information	Successful purchase: Payment is accepted and products shipping to the user	Good
User	Valid payment information	Payment is not received, and the purchase is not made	Bad

Member:

Use case: Exit from the system.

- Actor: Member
- Precondition: User logged in
- Postcondition: -
- Parameter: none
- Actions:

1. User exits from the system.
2. System saving the shopping cart of the user.
3. System define user as guest.

Participants	Parameters	Expected Result	Scenario
User	-	Successfully exit the system and his shopping cart saved	Good
User	-	Successfully exit the system and his shopping cart deleted	Bad

Use case: Logout.

- Actor: Member
- Precondition: User logged in
- Postcondition: User not logged in.
- Parameter: none
- Actions:

  1. User logs out from the system.
  2. System define user as guest.

Participants	Parameters	Expected Result	Scenario
User	-	Successfully logged out	Good
User	-	User logged out but the system recognizes the user as log in	Bad

Use case: Opening a store.

- Actor: Member
- Precondition: User logged in.
- Postcondition: New store is opened.
- Parameter: Store's detailed information
- Actions:

1. User creates store in the system.
2. User filling the store's information.
3. System marks the user as the 'Store Founder' of the store.
4. System set the store to 'open.'

Participants	**Parameters**	**Expected Result**	**Scenario**
User	Good and unique store name	Store created Successfully	Good
User	Invalid store name -	Store didn't create	Bad

Use case: Writing a review on items the user purchased.

- Actor: Member
- Precondition: User logged in and he has purchased products before
- Postcondition: Written review appears in the item’s comment section
- Parameter: The product the user wants to write a review about
- Actions:

1. User search for the product
2. User inserts his review.
3. System adds the review to the item’s comment section.

Participants	Parameters	Expected Result	Scenario
User	Well written review	The users' review is added to the comments on the product	Good
User	A review that is inappropriate, containing rude words	The users' review is not appearing in the comments on the product	Bad

Use case: Rating item and store by user.

- Actor: Member
- Precondition: User logged in ; user has purchased the item before / user has purchased from the store before.
- Postcondition: Rating appears in the item/store's ratings section.
- Parameter: The item/ store the user wants to rank
- Actions:

1. User search for the item/ store he wants to rate.
2. User inserts his ranking.
3. System inserts the rating on the item/ store ratings section.

Participants	Parameters	Expected Result	Scenario
User	A rating that meets the site's rating standard	The users' ranking is added to the rankings on the product/ store	Good
User	A rating that does not meet the site's rating standard	The users' ranking is not appearing in the rankings of the product/ store	Bad

Use case: User sending messages to store.

- Actor: User
- Precondition: User logged in; store exists.
- Postcondition: Written message appears in store's messages section.
- Parameter: store details , message
- Actions:

1. User search for the store he wants to send message to
2. User writing the message and send it.
3. System processes the message and send it to the store.
4. Store receiving the message.

Participants	Parameters	Expected Result	Scenario
User	Well written message	The user's message received in the store's messages	Good
User	A message that is inappropriate, containing rude words	The user's message didn't receive in the store's messages	Bad

Use case: Filing a complaint by a user about a purchase.

- Actor: User , System Managers
- Precondition: User logged in; user purchased before something
- Postcondition: complaint appears in the store's complaints section.
- Parameter: The purchase , complaint
- Actions:

1. User search for the purchase in the 'past purchases' section
2. User chooses the option of 'complaint' in the purchase.
3. User describes the reason for complaining about the purchase.
4. User sends the complaint.
5. System processes the complaint and send it to the store.
6. System Managers receiving the complaint.

Participants	**Parameters**	**Expected Result**	**Scenario**
User , System Managers	Well written complaint	The user's message received in the store's messages	Good
User , System Managers	A complaint that is inappropriate, containing rude words	The user's message didn't receive in the store's messages	Bad

Use case: Receiving information about personal purchase history.

- Actor: User
- Precondition: User logged in
- Postcondition: User receives info about his personal purchase history.
- Parameter: -
- Actions:

1. User enters his personal user profile.
2. User chooses the 'purchases' history' section.
3. System display purchases' history'

Participants	**Parameters**	**Expected Result**	**Scenario**
User	User purchased some product in different purchases	The user's purchases' history printed and matching the user's purchases	Good
User	User purchased 1 product only	The system shows 0 purchases	Bad

Use case: Editing identifying details of user.

- Actor: User
- Precondition: User logged in
- Postcondition: User's details updated in the system.
- Parameter: The info that the user wants to update
- Actions:

1. User enters his personal user profile.
2. User update the info he wants.
3. System updates the details.

Participants	Parameters	Expected Result	Scenario
User	new email address	New updated email ad	Good
User	Username that exists in the system	Username updated in the user's profile	Bad

Use case: Registration security for the trading system.

- Actor: User
- Precondition: User logged in.
- Postcondition: User is requested to answer questions when security checks are needed.
- Parameter: Questions and Answers
- Actions:

  1. User enters his personal user profile.
  2. User enters 'registration security' section.
  3. System adds Questions and Answers.

Participants	**Parameters**	**Expected Result**	**Scenario**
User	Good Q & A - only user knows	System updated the new Q & A	Good
User	Bad Q & A - everybody knows	System don't update the new Q & A	Bad

Store owner:

Use case: product management.

- Actor: store owner
- Precondition: user is logged in; store must exist, and the user is the owner of the store
- Postcondition: new item appears in the inventory of the store.
- Parameter: product data(name, price, etc.)
- Actions:
    1. User chooses to edit the store inventory.
    2. System displays him with options:
        a. Add a new item.
        b. Remove an item.
        c. Edit an item.
    3. User enters the new item details
    4. System adds the item to the store.

Participants	Parameters	Expected Result	Scenario
Store owner	Valid product data	The item is added to store	good
Store owner	Invalid product price	The item was not added to the store, and an error is displayed	bad

Use case: changing store policy.

- Actor: store owner
- Precondition: user is logged in; store must exist, and the user is the owner of the store
- Postcondition: discount policy of the store updated.
- Parameter: discount policy(items affected, discount amount…)
- Actions:
    1. User chooses to edit the store policy.
    2. User chooses to change an existing discount policy.
    3. User enters the new discount policy details.
    4. System changes the discount policy of the store.

Participants	**Parameters**	**Expected Result**	**Scenario**
Store owner	Valid discount policy	The discount policy of the store is changed	good
Store owner	Invalid discount policy	The discount policy was not changed, and an error is displayed	bad

Use case: appointing a new store owner.

- Actors: store owner, member
- Precondition: store owner is logged in, member is registered, store must exist, store owner is the owner of the store, and member is not the owner of the store
- Postcondition: store owner updated.
- Parameter: member's identifying details.
- Actions:
    1. store owner chooses to add new store owner.
    2. The system asks for identifying details of member.
    3. store owner enters the identifying details of member.
    4. System adds member as the owner of the store, and sets the appointer of member as store owner.

Participants	**Parameters**	**Expected Result**	**Scenario**
store owner, member	member is registered and is not yet the store's owner	member is added as the owner of the store, and sets the appointer of member as store owner.	good
store owner, member	member is not registered	The operation fails and an error is displayed	bad

Use case: appointing a new store manager.

- Actors: store owner, member
- Precondition: store owner is logged in, member is registered, store must exist, store owner is the owner of the store, and member is not the owner or manager of the store
- Postcondition: store manager updated.
- Parameter: member 's identifying details.
- Actions:
    1. store owner chooses to add new store manager.
    2. The system asks for identifying details of member.
    3. Store owner enters the identifying details of member.
    4. System adds member as the manager of the store, and sets the appointer of member as store owner.

Participants	**Parameters**	**Expected Result**	**Scenario**
store owner, member	Member is registered and is not yet the store's owner or manager	member is added as the manager of the store, and sets the appointer of member as store owner.	good
store owner, member	The user is not registered (not member)	The operation fails and an error is displayed	bad

Use case: changing a store manager permission

- Actors: store owner, store manager in the store
- Precondition: store owner is logged in, store manager is registered and is a manager in the store, store must exist, store owner is the owner of the store, and store owner is the appointer of store manager.
- Postcondition: store manager permission updated.
- Parameter: store manager's identifying details.
- Actions:
    1. Store owner chooses to edit the permissions store manager.
    2. The system asks for identifying details of store manager.
    3. Store owner enters the identifying details of store manager.
    4. Store owner enters new permissions of the store manager.
    5. System adds the permission to store manager.

Participants	**Parameters**	**Expected Result**	**Scenario**
store owner, store manager	The store manager of the specific store and the store owner is the appointer of the store manager	store manager permissions are updated.	good
store owner, store manager	store manager is not a manager of this store	The operation fails and an error is displayed	bad

Use case: closing a store.

- Actors: store owner.
- Precondition: user is logged in, store must exist and the user is the owner of the store
- Postcondition: store closed.
- Parameter: store id
- Actions:
    1. Store owner chooses to close his store.
    2. The system makes the store inactive.
    3. The System notifies all the managers and owners that the store is inactive.

Participants	**Parameters**	**Expected Result**	**Scenario**
store owner	Store id is valid	Store is now inactive, and all the managers and owners got notified	good
store owner	Store id is invalid	The operation fails and an error is displayed	bad

Use case: request store employees' information.

- Actors: store owner.
- Precondition: user is logged in, store must exist and the user is the owner of the store
- Postcondition: store owner receives the info.
- Parameter: store id
- Actions:
    1. Store owner requests store employee information.
    2. The system displays all the managers and owners of the store as well as their permissions.

Participants	**Parameters**	**Expected Result**	**Scenario**
store owner	Store id is valid	System displays the details	good
store owner	Store id is invalid	The operation fails and an error is displayed	bad

Use case: request store purchase history.

- Actors: store owner.
- Precondition: user is logged in, store must exists and the user is the owner of the store
- Postcondition: store owner receives the info.
- Parameter: store id
- Actions:
    1. Store owner requests the store purchase information.
    2. The system displays the purchase history of the store.

Participants	**Parameters**	**Expected Result**	**Scenario**
store owner	Store id is valid	System displays the details	good
store owner	Store id is invalid	The operation fails and an error is displayed	bad

Store manager:

Use case: Store manager read users complaints and respond.

- Actor: User
- Precondition: User logged in that is store manager
- Postcondition: complaint received on the store manager notifications, his response received by the user.
- Parameter: Complaint
- Actions:

1. User sends complaint about purchase.
2. System adds the complaint to the store manager to messages list.
3. System sends notification to the store manager.
4. Store manager writes response to the user's complaint.
5. System sends the response.

Participants	Parameters	Expected Result	Scenario
User, store, system manager	Well written complaint and response	Complaint received and response sent successfully	Good
User, store, system manager	The user complained on store that he never bought from before.	Complaint didn't get sent, the user got an error.	Bad

System Manager use cases:

Use case: Purchases information history.

- Actor: System Manager user
- Precondition: System Manager is logged in.
- Precondition: System Manager receives store purchases information history.
- Parameter: Store details
- Actions:
    1. User request to see store purchases information history.
    2. System checks that the user has permissions to perform this operation.
    3. System if user has permissions – asks from the user for store details
    3. User inserts store details
    4. System checks that store exist
    5. System if action succeeded – search for Store purchases information history.
    6. System if search succeeded – present Store purchases information history . if not - show an appropriate message

Participants	Parameters	Expected Result	Scenario
System, Manager user	Good and valid Store details	Store purchases information history is presented	Good
System, Manager user	Bad and invalid Store details	Store purchases information history is not presented	Bad