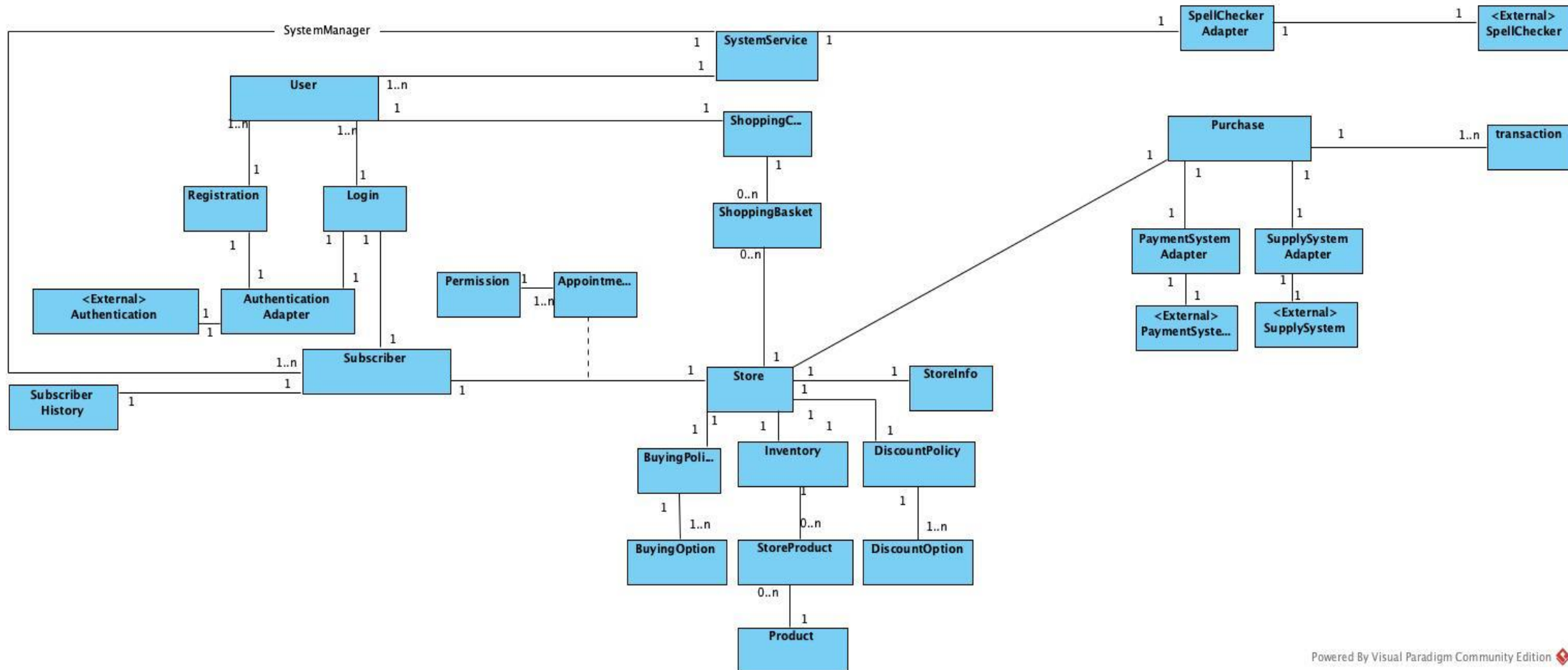


## **Table of Content – Version 0**

1. [White Diagram Module](#)
2. [Architecture Diagram](#)
3. [Correctness constraints](#)
4. [Glossary](#)
5. [Use Cases](#)
6. [Acceptance Tests](#)

## White Diagram:



SystemService: המחלקה אחראית על ניהול המערכת, אתחול המערכת וקיום כל הישויות הנדרשות ע"פ אילוצי הנכונות למשל מערכת המסחר, מערכת התשלומים, מנהל המערכת.

UserManagmentSystem: המחלקה האחראית על ניהול המשתמשים במערכת, כניסתם, יציאתם, רישומם כמנויים בעזרת הזנת פרטים מזהים וקביעת סיסמא, כניסה מזוהה (login) למנויים באמצעות פרטים מזהים וסיסמא ויציאה מהמערכת כמנוי (logout).

StoreManagerSystem: המחלקה אחראית לאפשר למשתמשים פתיחת חנות חדשה במערכת, ניהול מלאי של חנות ע"י מנהלי החנות, עריכה וקבלה של מידע אודות סוגי הרכישה וההנחה האפשריים בחנות ומדיניות הקניה וההנחה עבור מוצרים בחנות, אפשרור מינוי מנהלי ובעלי חנות נוספים והסרתם, ניהול ההרשאות עבור המנהלים השונים ע"י בעלי החנות, קבלת מידע על בעלי התפקידים השונים בחנות, קבלת מידע על היסטוריית ברכישות בחנות.

BuyerService: המחלקה אמונה על חיפוש המוצרים השונים בלי קשר לחנויות ספציפיות לפי שם המוצר קטגוריה או מילות מפתח, שמירת מוצרים בסל הקניות של לקוח כלשהו על מנת שיוכל לחזור אליהם ולקנות בכל שלב שיחפוץ, חישוב עלויות המוצרים בהתאם לסוגי ההנחה של החנות הרלוונטית, קבלת מידע היסטוריית רכישה אישית לקונה,

PurchaseService: המחלקה אחראית על רכישה של מוצרים, בדיקת היתכנות לרכישה מבחינת תנאי התשלום וכמות המוצרים במלאי החנות הרלוונטית, הודעה על רכישה גם ללקוח שקונה וגם לחנות שצריכה לבצע את משלוח המוצר לבעליו.

User: המחלקה מייצגת קונה בלי הרשאות של מנוי, המחלקה אחראית לבצע פעולות הקשורות במשתמש ספציפי כלשהו.

UserCart: המחלקה מייצגת סל קניות של משתמש כלשהו ומורכבת מכלל עגלות החנות השונות של המשתמש.

StoreCart: המחלקה מייצגת סל קניות של חנות ספציפית, היא מחזיקה רשימה של מוצרים אותם הלקוח מעוניין לשמור, לכל משתמש יש לכל היותר storeCart יחיד לכל חנות.

Subscriber: המחלקה מייצגת לקוח שרשום כמנוי, היא מאפשרת לו הרשאות מיוחדות כמו חזרה לאותו מצב לאחר התנתקות והתחברות למערכת ופתיחת חנות חדשה.

Store: המחלקה מייצגת חנות כלשהי במערכת, היא אחראית על שמירה של צוות החנות והרשאותיהם, רשימת מוצרים ומחיריהם, אופן התשלום וההנחות השונות של המוצרים.

BuyingPolicy: המחלקה מייצגת את מדיניות הרכישה שמבוצעת בחנות ע"י שילוב של אפשרויות רכישה וחוקים שונים.

BuyingOption: המחלקה מייצגת את אפשרות הרכישה בחנות ומגדירה את החוק, מנהל החנות יכול לשנות את אפשרות זו, למחוק ולהוסיף על מנת ליצור מדיניות רכישה כזו או אחרת.

Product: המחלקה מייצגת מוצר הנרכש בחנות ע"י שם, תיאור, מחיר, כמות, קטגוריה, דירוג ועוד.

SupplySystem: המחלקה מתארת את המערכת החיצונית האמונה על סיפוק של מוצרים חדשים לחנויות.

DiscountPolicy: המחלקה מייצגת את מדיניות ההנחות שמבוצעות החנות ע"י שילוב של אפשרויות הנחה וחוקים שונים עבור משתמשים ומוצרים כאלו ואחרים.

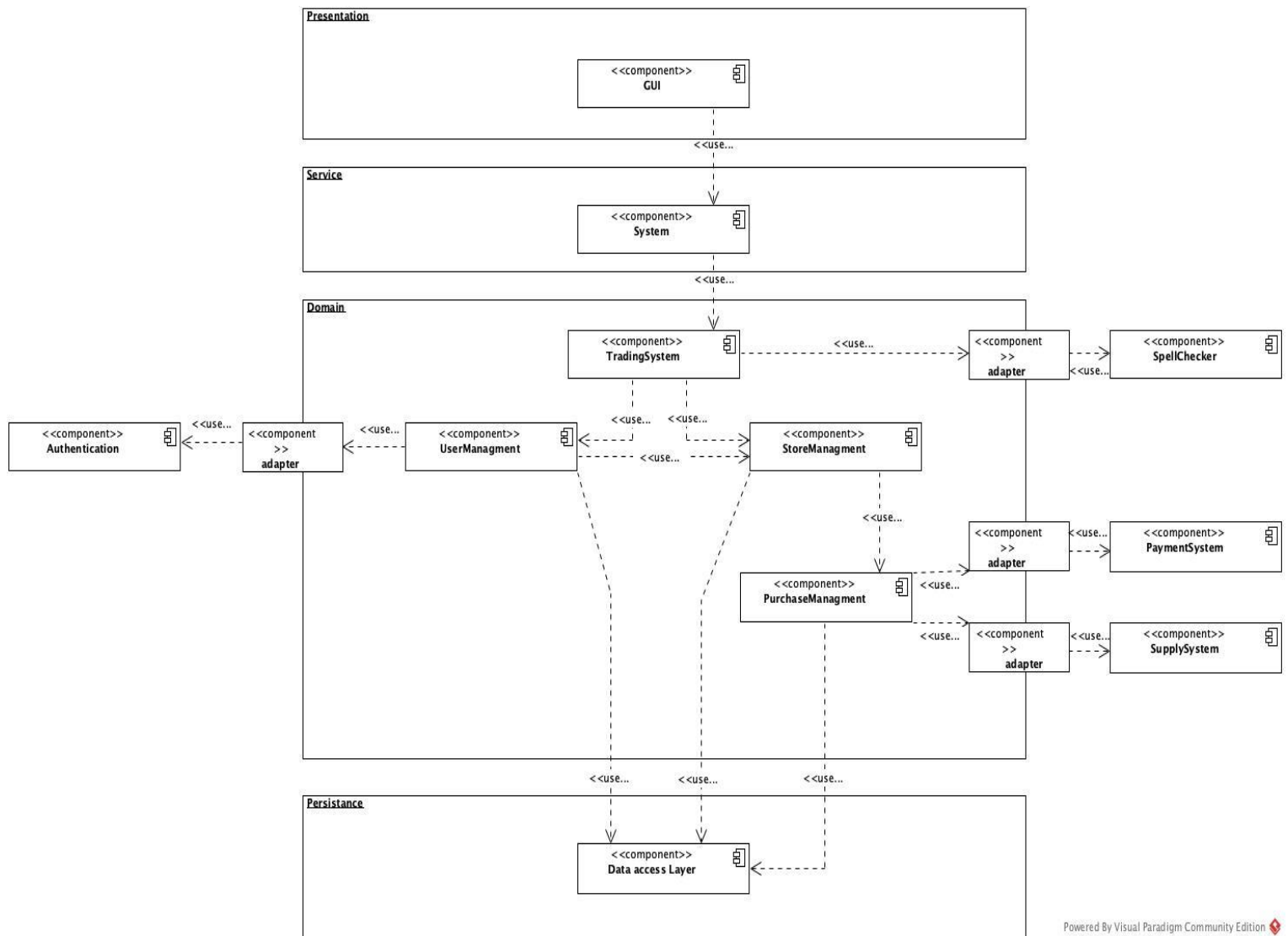
DiscountOption: המחלקה מייצגת אפשרות הנחה כלשהי שמנהל החנות יכול להוסיף להסיר ולשלב על מנת ליצור מדיניות הנחה כזו או אחרת.

Role: המחלקה מייצגת את תפקידו של מנכ"ל בחנות כלשהי, מגדירה את ההרשאות שאותו מנכ"ל רשאי להן, מגדירה האם הוא מנהל או בעל חנות ואיזה פעולות ביכולתו לבצע.

PaymentSystem: המחלקה מייצגת את המערכת החיצונית שאחראית על ביצוע התשלום עצמו והעברת הכסף לנמען.

Transaction: המחלקה אחראית על הגדרת טרנזקציה מוצלחת, שומרת את הקונה והמוכר, את כמות הכסף שעבר, זמן הביצוע ועוד.

## Architecture diagram:



## Correctness constraints:

1. לכל משתמש יש שם יחיד המזהה אותו המערכת – אין מספר משתמשים עם אותו שם.

תרחיש כישלון:

Action – יצירת משתמש בעל שם זהה למשתמש רשום קיים.

Data – משתמש שלא רשום במערכת.

Expected Result – הודעת שגיאה שמודיעה על שם משתמש תפוס.

תרחיש הצלחה:

Action - יצירת משתמש בעל שם שלא קיים למשתמש במערכת.

Data - משתמש שלא רשום במערכת.

Expected Result – המשתמש נרשם בהצלחה.

2. יש למערכת לפחות מנהל אחד. מנהל-מערכת חייב להיות מנוי (עבר תהליך רישום).

מהבניה – על מנת להיות מנהל מערכת על המשתמש להיות רשום.

תרחיש כישלון:

Action – אתחול מערכת ללא מנהל.

Data – אין.

Expected Result – המערכת נכשלה באתחול.

תרחיש הצלחה:

Action - אתחול מערכת באמצעות מנהל רשום.

Data – מנוי מנהל המערכת.

Expected Result – המערכת אותחלה בהצלחה.

3. בעל-חנות או מנהל-חנות חייב להיות מנוי .

מהבניה - רק מנויים במערכת רשאים לקבל תפקיד (Role). בעל חנות ומנהל חנות הם תפקידים במערכת.

4. לחנות פעילה (שאינה סגורה) חייב להיות לפחות בעל -חנות אחד.

מהבניה – לכל חנות יש לפחות מנוי אחד, מנוי זה הוא מייסד החנות (בעל החנות הראשון).

5. לחנות חייבים להיות מוגדרים :

- סוגי קניה (רכישה) וסוגי הנחה. תיתכן ברירת מחדל של סוג אחד.
- מדיניות רכישה ומדיניות הנחה. תיתכן ברירת מחדל של "אין מגבלות רכישה או הנחה".

מהבניה – לכל חנות יש מדיניות רכישה, הנחה וסוגי קניה והנחה. במידה ולא הוזנה מדיניות רכישה/הנחה, השדה יאותחל בערך דיפולטיבי.

תרחיש הצלחה:

Action – פתיחת חנות עם סוגי קניה וסוגי הנחה.  
Data – פרטי החנות וסוגי הקניה וההנחה הנתמכים בה.  
Expected Result – החנות נפתחה בהצלחה ותומכת אך ורק בסוגי הקניה וההנחה שהוגדרו לה.

תרחיש כישלון:

Action – פתיחת חנות ללא סוג קניה.  
Data – פרטי החנות.  
Expected Result – החנות נפתחת עם סוג קניה דיפולטיבי המוגדר במערכת המסחר.

תרחיש הצלחה:

Action – פתיחת חנות עם מדיניות רכישה.  
Data – פרטי החנות ופרטי מדיניות רכישה.  
Expected Result – החנות נפתחה בהצלחה ומקיימת את מדיניות הרכישה שהוגדרה.

תרחיש כישלון:

Action – פתיחת חנות ללא מדיניות רכישה.  
Data – פרטי החנות.  
Expected Result – החנות נפתחת עם מדיניות רכישה דיפולטיבי המוגדר במערכת המסחר.

6. לקונה יש עגלת הקניות יחידה, המורכבת מאוסף כל סלי הקניה שלו. לקונה יש לכל היותר סל קניות יחיד לחנות.

מהבניה – למשתמש יש עגלת קניות יחידה המורכבת מסלי הקניה שלו.

7. עגלת הקניות של קונה (אורח או מנוי) הינה בבעלותו הבלעדית ואינה ניתנת לשינוי על ידי שום משתמש אחר.

מהבניה – קשר 1 ל-1 בין משתמש לבין עגלת קניות, לאף משתמש אין גישה לעגלת קניות של משתמש אחר.

8. כללי יושרה :

- לא ניתן לגבות כסף מקונים עבור עסקאות שלא ביצעו או בסכומים שונים מהסכומים שהוצגו.

תרחיש הצלחה:

Action – ביצוע תשלום עבור עסקה שבוצעה ע"י מנוי.  
Data – פרטי תשלום של המנוי ועגלת הקניות שלו.  
Expected Result – השלמת הטרנזקציה עם פרטי עגלת הקניות המוזנים.

תרחיש כישלון:

Action – ביצוע תשלום עבור עסקה שלא ביצע מנוי.  
Data – פרטי תשלום של המנוי ועגלת הקניות של העסקה.

Expected Result – הודעת שגיאה, על המנוי לאשר את העסקה.

- לא ניתן לסיים בהצלחה תהליך קנייה בלי לשלם את הסכום הדרוש עבור המוצר.

תרחיש הצלחה:

Action – ביצוע טרנזקציה.

Data – פרטי מנוי, תשלום ופרטי האמצעי התשלום.

Expected Result – התקבל אישור ממערכת התשלומים על קבלת הסכום הדרוש.

תרחיש כישלון:

Action – ביצוע טרנזקציה.

Data – פרטי מנוי, תשלום ופרטי האמצעי התשלום ללא הסכום הדרוש.

Expected Result – הודעת שגיאה, לא ניתן להשלים את הטרנזקציה עקב חסר הימצאות הסכום הדרוש באמצעי התשלום.

- מוכר לא יכול לקבל תשלום שלא כתוצאה מתהליך קנייה מוצלח .

תרחיש הצלחה:

Action – העברת תשלום למוכר החנות.

Data – פרטי טרנזקציה תקינים ופרטי מוכר החנות.

Expected Result – התשלום התקבל אצל מוכר החנות בהצלחה.

תרחיש כישלון:

Action - העברת תשלום למוכר החנות.

Data – פרטי טרנזקציה שטרם הושלמה ופרטי מוכר החנות.

Expected Result – הודעת שגיאה, יש להשלים את הטרנזקציה לפני העברת התשלום למוכר.

- הרשאת ניהול של מנהל- מערכת/בעל-חנות/מנהל -חנות לא יכולה לסתור את כללי היושרה. לדוגמה, מנהל-מערכת לא יכול לעדכן עגלת קניות של קונה שאינו עצמו .

מהבניה – לא קיימות הרשאות שמאפשרות לסתור את כללי היושרה.

9. חיבורים למערכות חיצוניות

- נדרש ערוץ פניה למערכת תשלומים.
- נדרש ערוץ פניה למערכת אספקה.

נאכף ע"י הבנייה - למערכת ניהול החנויות יש קשר למערכת תשלומים ולמערכת אספקה.



## Glossary:

- ביקורת- הערה חיובית או שלילית על החנות או מוצריה שנכתבת ע"י מנוי שביצע בה רכישה.
- הגשת הצעת קנייה- המשתמש רשאי להציע מחיר עבור מוצר. מנהל החנות בוחר האם לקבל את הצעתו או לדחותה.
- דירוג חנות – ציון בין 1-5 המביע את שביעות רצון הקונה מהחנות.
- דירוג מוצר - ציון בין 1-5 המביע את שביעות רצון הקונה מהמוצר.
- הודעה – פנייה בין הקונה לחנות ולהפך ומאפשרת תקשורת ביניהם.
- היסטוריית רכישות אישית- רשימת הרכישות האישיות שנתבצעו ע"י המשתמש.
- הנחה גלויה- מורכבת מאחוז הנחה וממשך ההנחה. הקונים רואים את המחיר המקורי ואת מועד סיומה.
- הנחה מותנית - מורכבת ממשך הנחה, אחוז הנחה ותנאי הנחה. מוצגת לקונים, אך מגדירה תנאים בהם צריך לעמוד כדי לקבלה.
- הנחה נסתרת - מורכבת ממשך הנחה, אחוז הנחה וקוד הנחה. הנחה זו אינה גלויה לקונים.
- התראה מושהית- הודעה המוצגת למנוי רק בעת כניסתו למערכת, ולא קודם לכן.
- התראת זמן אמת- הודעה אשר נשלחת לבעל חנות/קונה בזמן ביצוע הפעולה.
- חנות- ישות הנפתח ע"י בעל החנות. מוצעים בה מוצרים למכירה למשתמשים.
- טרנזקציה- מתארת פרטי רכישה שמשתמש מעוניין לבצע, מכילה את פרטי המשתמש, פרטי החנות, סכום הרכישה ועוד..
- מדיניות הנחה- מדיניות הנחה מגדירה מי רשאי לקבל הנחה ובאיזה אופן, בשעת קניית מוצרים בחנות, מהם סוגי ההנחה, ומהם הכללים החלים עליהם.
- מדיניות קנייה- מדיניות קניה מגדירה מי רשאי לקנות מוצרים בחנות, מהם מסלולי הרכישה האפשריים, ומהם הכללים החלים עליהם.
- מוכר בעל החנות – מנוי שפתח חנות מסחר או שמונה ע"י בעל חנות אחר. רשאי למנות בעלי חנות נוספים ולהסירם. בעל הרשאות גבוהות בנוגע לחנות וניהולה.
- מוכר מנהל חנות – רשאי לבצע כל פעולה שהקצה לו בעל החנות.
- מוצר- פריט המוצע למכירה בחנות כלשהי.
- מייסד החנות- בעל החנות הראשון. אין לו ממנים.
- מלאי- כלל המוצרים הזמינים בחנות.
- מנהל מערכת המסחר – אחראי על התנהלות השוטפת של המערכת. ביכולתו להסיר מנויים וחנויות ממערכת המסחר.
- משתמש- לקוח אשר משתמש בשירותי המערכת.
- סל קניות- מוצרים אותם בחר הלקוח מחנות מסוימת.
- עגלת קניות- יחידה עבור כל לקוח, מורכבת מסך כל סלי הקניות של הלקוח.
- קונה אורח - משתמש במערכת שאינו מחובר כמנוי. בעל הרשאות מוגבלות.
- קונה מנוי – משתמש אשר נרשם למערכת ומחובר באמצעות פרטי הזדהות ייחודיים. בנוסף להרשאותיו של קונה אורח, יש לו הרשאות נוספות ייחודיות לו.
- קנייה מיידי- המשתמש נדרש לשלם את הסכום הנקוב במוצר על מנת לקנות אותו באופן מיידי.
- רכישה בהגרלה- המנוי זכאי לרכוש סיכויי זכייה למוצר. כל זאת בתנאי שסכום התשלומים מהמשתתפים אינו גדול ממחיר המוצר.

- רכישה פומבית- מנויים יכולים להגיש הצעות מחיר עבור הפריט הרצוי, רק הצעות גדולות מהמחיר ההתחלתי ומכל הצעה קודמת שהתקבלה, יתקבלו.
- תלונה – פנייה הנשלחת למנהלי המערכת במקרה של הפרת אילוץ יושרה.

## Use Cases:

1. System Use Cases
  - 1.1. Initialize System
2. Guest User Use Cases
  - 2.1. Enter System
  - 2.2. Exit System
  - 2.3. Register
  - 2.4. Login
  - 2.5. View Store
  - 2.6. Search Products
  - 2.7. Add to Store-cart
  - 2.8. View/Edit Shopping Cart
    - 2.8.1. View Shopping cart
    - 2.8.2. Edit Shopping Cart
  - 2.9. Purchase
3. Subscribed Buyer Use Cases
  - 3.1. Logout
  - 3.2. Open Store
  - 3.7. Get purchase history
4. Store Owner Use Cases
  - 4.1. Edit store inventory
  - 4.2. Edit store discount / buying policy
  - 4.3. Appoint store owner
  - 4.5. Appoint store manager
  - 4.6. Edit store manager permissions
  - 4.7. Remove store manager appointment
  - 4.9. Get information on staff
  - 4.11. Get store purchase history
5. Store Manager Use Cases
  - 5.1. Perform manager actions
6. System Manager Use Cases
  - 6.4. Get user/store history
7. Payment
8. Supply

# 1-System

## 1.1: Initializes System

Actor: Any User

Precondition:

- There exists a System Manager in the System
- The System is connected to the External Payment System

Post Condition: The system is online and ready for user actions.

Parameters: None

Actions:

1. The user starts the system
2. If the preconditions are satisfied, the system is successfully initialized.
3. Otherwise the process fails.

Main:

The user opens the system. The system successfully verifies that there exists a system manager and that the payment system is available. The user is identified as a guest and can perform the actions defined for guest users.

Alternative:

The user opens the system. The System cannot find a system manager or verify that the payment system is available. The user is notified of the issue and the system closes

## 2- Guest Buyer

### 2.1: Enter system

Actor: Any User

Precondition: User is offline

Post condition: User is online as a guest.

Parameters: none

Main scenario:

1. User enters the system
  2. The System initiates (use case 1.1).
  3. The user is defined as guest.
- Alternate:
    - 2.1. The system initialization fails
    - 2.2. The process terminates

### 2.2: Exit system

Actor: Any User

Precondition: User is online

Post Condition: User is offline

Parameters: none

Main scenario:

1. Guest user exits the system
  2. The system closes the connection to the external systems
  3. The system saves all unsaved data in its tables
- Alternate:
    - 2.1. Connection to external system is lost
    - 2.2. Exit process is stopped

## 2.3: Register

Actor: Guest User

Precondition: username is not already used

Post Condition: username and password were added to the users.

Parameters: username , password

Main scenario:

1. Guest user chooses to register and enters his username and password
  2. The system checks the username is not already used
  3. The system checks for the username and password validity
  4. A new user is created and saved in the system.
- Alternate:
    - 2.1. System finds out the username is already used
    - 2.2. The user is requested to enter different username
    - 2.3. Restart use case 2.3
  - Alternate:
    - 3.1. The user name or password contains invalid characters
    - 3.2. The user is requested to insert different username/password
    - 3.3. Restart user case 2.3

## 2.4: Login

Actor: Guest user

Precondition: no logged on user in current session, user is already registered

Post Condition: User is logged on as Subscriber

Parameters: username and password

Main scenario:

1. Guest user enters his username and password
  2. The system checks for a match between the username and password
  3. The guest user changes his state to logged on
- Alternate:
    - 2.1. The username and the password doesn't match
    - 2.2. Relevant message is sent to the user

## 2.5: View Store

Actor: Any User

Precondition: The store exists in the system

Post Condition: relevant data is available

Parameters: The store's id

Main scenario:

1. The user chooses a store
  2. The system searches for relevant store
  3. The system returns all details about the store and the products in the store
- Alternate:
    - 2.1. The selected store isn't in the system
    - 2.2. Relevant message is sent to the user
  - Alternate:
    - 2.1. The selected store is closed and can't be viewed
    - 2.2. Relevant message is sent to the user

## 2.6: Search products

Actor: Any User

Precondition: none

Post Condition: only data filtered by the parameters is presented

Parameters: one or more of: product name, category and keywords.

optional : any number of product's attributes such as: price range, ranking , etc.

Main scenario:

1. Guest user asks to see details about products according to one of the following:  
product name, category and keywords
2. The trading system searches all stores and return only products that correspond to the parameters given.
3. If optional parameters are given the system filters the results from the previous stage according to these parameters.
4. The filtered data is returned

## 2.7: Add products to store-cart.

Actor: Any User

Precondition: The store's inventory contains all requested products.

Post Condition: The products has been added to the users old shopping cart.

Parameters: products and quantities to add.

Main scenario:

1. The user asks to add products to his store-cart
  2. The system checks the store's inventory
  3. The products are added to the user's store-cart
  4. Otherwise the products are added to his old store cart
  5. If the user is a subscriber the store-cart is saved in the system.
- Alternate:
    - 2.1. The store doesn't contain the amount of requested product
    - 2.2. Relevant message is sent to the user.
  - Alternate:
    - 3.1. If the user doesn't have store-cart to this store, a new cart is created.
    - 3.2. Continue main scenario

### 2.8.1: View shopping cart.

Actor: Any user

Precondition: none

Post Condition: none

Parameters: none

Actions:

1. The user asks to see his shopping-cart
2. The system returns a list of all of his store-carts

## 2.8.2: Edit shopping-cart

Actor: Any user

Precondition: none

Post Condition: the shopping-cart has changed according to the user's request.

Parameters: list of <product, quantity> to be changed

Main scenario:

1. The user views his shopping-cart (use case 2.8.1) and chooses a store-cart to edit
  2. The user enters list of products with their quantities
  3. The system checks for the product availability in the store
  4. The products are saved in the store-cart
  5. The store-cart is saved in the system's tables
- Alternate:
    - 3.1. The user entered negative number of products
    - 3.2. Error message is sent to the user
  - Alternate:
    - 3.1. The store's inventory doesn't contain the amount of requested products
    - 3.2. Error message is sent to the user

## 2.9: Purchase

Actor: Any user

Precondition: The user have at least one none empty store-cart

Post Condition: The users' store-cart and the stores' inventory are updated according to the amount of purchased products

Parameters: products, amounts and purchase type

Main scenario:

1. User asks to buy products from a store
  2. The User indicates the purchase type he want to perform
  3. The system verifies that all products are in the store's inventory
  4. The system verifies that the combination of <products, purchase type> is legal.
  5. The purchase continues according to the purchase type (use cases 2.9.1 – 2.9.4)
- Alternate:
    - 3.1. The store doesn't hold the amount of requested products.
  - Alternate:
    - 4.1. One of the combinations <product, purchase type> is not legal according to the stores' policy

### 2.9.1: Immediate Purchase

Actor: Any user

Precondition: The products are open for immediate purchase

Post Condition: The users' store-cart and the stores' inventory are updated according to the amount of purchased products

Parameters: products, amounts

Main scenario:

1. The system calculates discounts according to the stores' discount policies
2. The system informs the user of the final price and requests his approval
3. The system transfers payment request to the external payment system and waits for its response (use case 7)
4. Supply request was sent to the supply system with the users' information

- 5. The users' cart and the stores' inventory are updated
- Alternate:
  - 4.2. The payment request was refused
  - 4.3. Relevant message is sent to the user
- Alternate:
  - 4.3. The supply request was refused
  - 4.4. The system sends cancellation request to payment system
  - 4.4. Relevant message is sent to the user

### 2.9.2: Bid offer Request purchase

Actor: Any user

Precondition: The products are open for bid purchase

Post Condition: The users' store-cart and the stores' inventory are updated according to the amount of purchased products

Parameters: list of <product, amount >

Main scenario:

1. For each product the user makes a bid request which is sent to the authorized staff in the store
  2. The user waits for the bid approval
  3. The user gets bid approval and confirms the deal
  4. The system transfers payment request to the external payment system and waits for its response (use case 7)
  5. Supply request was sent to the supply system with the users' information
  6. The users' cart and the stores' inventory are updated
- Alternate:
    - 3.1. The staff doesn't approve the bid offer
    - 3.2. The User chooses to make a new offer or abort the purchase
  - Alternate:
    - 4.5. The payment request was refused
    - 4.6. Relevant message is sent to the user
  - Alternate:
    - 5.1. The supply request was refused
    - 5.2. The system sends cancellation request to payment system
    - 5.3. Relevant message is sent to the user

### 2.9.3: Public auction purchase

Actor: Any user

Precondition: The products are open for public auction purchase

Post Condition: The users' store-cart and the stores' inventory are updated according to the amount of purchased products

Parameters: list of <product, amount>

Main scenario:

1. For each product the user views the initial price and all previous offers
2. The user makes offer for the items he wants to purchase
3. The user receives updates about offers made by different users
4. Repeat steps 2-3 until the auction is finished
5. The user made the highest price and wins the auction
6. The system calculates discounts according to the stores' discount policies



7. The system informs the user of the final price
  8. The system transfers payment request to the external payment system and waits for its response (use case 7)
  9. Supply request was sent to the supply system with the users' information
  10. The users' cart and the stores' inventory are updated
- Alternate:
    - 2.1. The user makes smaller offer than the highest offer
    - 2.2. The user is requested to make a new offer
  - Alternate:
    - 2.1. The user decides this auction went out of his budget and leaves the sale
  - Alternate:
    - 8.1. The payment request was refused
    - 8.2. Relevant message is sent to the user
  - Alternate:
    - 9.1. The supply request was refused
    - 9.2. The system sends cancellation request to payment system
    - 9.3. Relevant message is sent to the user

#### 2.9.4: Lottery purchase

Actor: Any user

Precondition: The products are open for lottery purchase

Post Condition: The users' store-cart and the stores' inventory are updated according to the amount of purchased products

Parameters: list of <product, amount>

Main scenario:

1. For each product the user makes a price offer
  2. After the products reach its price boundary the lottery takes place
  3. The user wins the lottery and gets to buy the product
  4. The system calculates discounts according to the stores' discount policies
  5. The system informs the user of the final price
  6. The system transfers payment request to the external payment system and waits for its response (use case 7)
  7. Supply request was sent to the supply system with the users' information
  8. The users' cart and the stores' inventory are updated
- Alternate:
    - 1.1. The users' offer makes the sum of offers exceed the products' price
    - 1.2. The user is requested to make a smaller offer
  - Alternate:
    - 3.1. The user lost the lottery
    - 3.2. The system charges the user for his price offer
  - Alternate:
    - 6.1. The payment request was refused
    - 6.2. Relevant message is sent to the user

- Alternate:
  - 7.1. The supply request was refused
  - 7.2. The system sends cancellation request to payment system
  - 7.3. Relevant message is sent to the user

## 3- Subscribed Buyer

### 3.1: Logout

Actor: Subscriber

Precondition: User is subscribed

Post Condition: User is logged off as Guest

Parameters: none

Main scenario:

1. Subscribed user logs into the system
2. After the user finished he decides to logout from the system
3. The system backs up all unsaved data.
4. The system marks the user as logged out

### 3.2: Open store

Actor: Subscriber

Precondition: User is subscribed and logged on, parameters are valid

Post Condition: A new store was created in the system, the subscribed user is its only store owner

Parameters: Name, Description, policies, products and amounts.

Main scenario:

1. A Subscribed user decides to open a store
  2. He enters the stores name, description, policies and products
  3. The system verifies the parameters validity
  4. A new store is created, and the user is marked as the store's owner and gets all available permissions over the store
- Alternate:
    - 3.1. The policies entered by the user are illegal
    - 3.2. Relevant message sent to the user

### 3.7: Get purchase history

Actor: Subscriber

Precondition: User is subscribed and logged on

Post Condition: none

Parameters: none

Main scenario:

1. A Subscribed user logs into the system
2. The user requests to view his purchase history
3. The system returns the purchase history to the user

## 4-store owner

### 4.1: Edit inventory.

Actor: Store Owner

preconditions: User is already logged in as a store owner.

Post conditions: Inventory is changed according to the user's request.

Parameters: new inventory to update.

Main scenario:

1. Store owner changes his store's inventory.
2. Store inventory changed accordingly

### 4.2.1: Edit discount

Actor: Store owner

preconditions: User is already logged in as a store owner.

Post conditions: Discount is changed according to the user's request.

Parameters: new discount/ discount to update.

Main scenario:

1. Logged in store owner views the discount options.
2. Store owner changes the store's discount policy.
3. store's discount policy is changed accordingly.

### 4.2.2: Edit buying policies

Actor: Store owner

preconditions: User is already logged in as a store owner.

Post conditions: Buying policies are changed according to the user's request.

Parameters: new buying policy/ policy to update.

Main scenario:

1. Logged in store owner views the buying options.
2. Store owner changes the store's Buying policy.
3. store's Buying policy is changed accordingly.

## 4.3: Appoint store owner

Actor: Store owner

1. Otherwise, the process fails, and the subscriber is not added as a store owner.

preconditions:

- User is already logged in as a store owner.
- The new store owner is not already an owner of the store.

Post conditions: A new store owner is added to the store and get all the abilities of the store owner.

Parameters: name of subscriber.

Main scenario:

1. store owner adds a new store owner
  2. the system checks if the subscriber is not already an owner.
  3. The subscriber is available the system adds the subscriber as a new store owner.
- Alternate:
    - 2.1 The subscriber is already a store owner.
    - 2.2 The process fails.

## 4.5: Appoint store manager

Actor: Store owner

preconditions:

- User is already logged in as a store owner.
- The new store owner is not already the store manager nor store owner

Post conditions: A new store manager is added to the store and gets basic permissions.

Parameters: name of subscriber.

Main scenario:

1. Store owner tries to add a new store manager
  2. The system checks if the subscriber is not the store manager nor store owner.
  3. The subscriber is available, the system adds the new store manager to the store.
- Alternate:
    - 2.1 The subscriber is already a store manager.
    - 2.2 The process fails.

## 4.6: Edit store manager permissions

Actor: Store owner

preconditions: User is already logged in as a store owner.

Post conditions: User can view and edit the manager permissions in the store

Parameters: name of subscriber, permission to edit.

Main scenario:

1. Store Owner tries to change a store's manager permissions.
2. the store's manager permissions are changed accordingly

## 4.7: Remove store manager

Actor: Store owner

preconditions:

- User is already logged in as a store owner.
- Subscriber is already a store manager.

Post conditions: Subscriber is no longer a store manager.

Parameters: name of subscriber.

Main scenario:

1. store owner tries to remove a subscriber from being a store manager.
2. the system checks if the subscriber is already a store manager.

3. The subscriber is already a store manager the system removes the subscriber's permissions.
- Alternate:
  - 2.1 The subscriber is not a store manager.
  - 2.2 The process fails.

## 4.9: Get Staff Information.

Actor: Store owner

preconditions: User is already logged in as a store owner.

Post conditions: User can watch the store's staff info

Parameters: None.

Main scenario:

1. The store manager looks for a particular piece of information.
2. The system finds that information and delivers.

## 4.11: Get store purchase history

Actor: Store owner

preconditions: User is already logged in as a store owner.

Post conditions: User can watch the store's purchase history.

Parameters: None.

Main scenario:

1. The store owner tries to receive the store's purchase history.
2. The system finds the store's purchase history and deliver it back to the store owner.

# 5-Store manager

## 5.1: Perform a management action

Actor: Store Manager

Precondition:

- The Store Manager has been granted permission by the store owner to perform the action
- Additionally, the same preconditions that are stated for the Store Owner performing the action must hold

Post Condition: the same post conditions that are stated for the Store Owner performing the action must hold

Parameters: same as parameters stated for the Store Owner performing the same action

Actions:

1. If the Store Manager has been granted permission, then the same as the actions stated for the Store Owner performing the same action
2. Otherwise, the system presents a "Permission Denied" message

Main:

The store manager attempts to perform an action that an owner of the store has granted him permission to perform. The action completes successfully (as defined for store owners)

Alternatives:

- The store manager attempts to perform an action that he has not been granted permission to perform. The store manager is informed that permission has been denied.
- The store owner attempts to perform an action that he has been granted permission to perform by the store owner. Alternative cases for the same actions that pertain to the store owner, pertain to the store manager as well.

## 6-system manager

### 6.4: Receive transaction history of a buyer or store

Actor: System Manager

Precondition:

- The Entity (Buyer/ Store) exists in the system

Post Condition: All transactions pertaining to the entity are presented to the System Manager

Parameters: the Entity

Actions:

1. The System manager selects the entity of interest
2. The system presents the list of transactions (Buyer's info, payment method, time of purchase, items, pricings, Store's info)

Positive:

The system manager selects a subscribed user. The system successfully verifies the existence of said user. The user's transaction history is presented to the system manager.

Negative:

The System Manager selects a user. The system cannot identify the user in the system. The system informs the system manager of the issue and allows him to select another user.

## 7: Payment

Actor: The System, External Payment System (EPS)

Precondition:

- Payment information for the purchase has been provided

Post Condition: the payment status is reported to the system and the user

Parameters: Total Price, Payment info

Actions:

1. The System provides the EPS with the payment info
2. The EPS notifies the System whether the payment was successful or not

Main:

The System provides the EPS with the payment info. The EPS charges the appropriate amount and reports back to the system that the transaction was successful.

Alternative:

- The System provides the EPS with invalid payment info. The EPS reports back to the system that the transaction could not be completed.
- The System provides the EPS with valid payment info, however the payment could not be completed (insufficient funds, credit not approved, etc.). The EPS reports back to the system that the transaction could not be completed.

## 8: Supply

Actor: The System, External Supply System (ESS)

Precondition:

- Information of the user and the items purchased have been provided

Post Condition: the system and the user receive notice that the request for the items has been received

Parameters: the user and his cart

Actions:

3. The System provides the ESS with the users info and the user's cart's contents
4. The ESS notifies the System that the request has been received

Main:

The System provides the ESS with the user and cart info. The ESS notifies the system that the request has been successfully received. The system passes the notification along to the user.

Alternative:

- The System provides the ESS with invalid user info (i.e. invalid shipping address, no name, etc) . The ESS reports back to the system that the supply request could not be completed. The system passes the notification along to the user.
- The System provides the ESS with an invalid cart (i.e. items that do not exists, items not in supply). The ESS reports back to the system that the supply request could not be completed. The system passes the notification along to the user.

## Acceptance Tests:

1.1: Initializes System

2.1: Enter system

2.2: Exit system

2.3: Register

2.4: Login

2.5: View Store

2.6: Search products

2.7: Add products to store-cart.

2.8.1: View shopping cart.

2.8.2: Edit shopping-cart

2.9: Purchase

3.1: Logout

3.2: Open store

3.7: Get purchase history

4.1: Edit inventory.

4.2: Edit discount / buying policies

4.3: Appoint store owner

4.5: Appoint store manager

4.6: Edit store manager permissions

4.7: Remove store manager

4.9: Get Staff Information.

4.11: Get store purchase history

5.1: Perform a management action

6.4: Receive transaction history of a buyer or store

7: Payment

8: Supply



### 1.1: Initializes System

- The system is initialized. There exists a system manager and all external connections are verified. The current user is a guest.
- The system is initialized. No system manager exists. An exception is thrown.
- The system is initialized. The External Payment/Supply System's connection could not be verified. An Exception is thrown.

### 2.1: Enter system

- A user enters the system, a system manager was defined previously and all external connections were set successfully. The Operation succeeded.
- A user enters the system, no system manager was defined and the operation fails.

### 2.2: Exit system

- A user exits the system successfully.

### 2.3: Register

- A guest user registers the system, he enters the username "avi" and password "1234456789", after a check for the username uniqueness a new user is created.
- A guest user tries to register with the username "avi" and password "1234", the username is already taken and the registration fails.
- A guest user tries to register with the user name "avi2" and the password "\_-\*&", the system informs the user the characters are invalid and requests to enter new password.

### 2.4: Login

- A user who was previously registered to the system tries to login. He enters the username "avi" and password "12346789". The username and password matches, the login succeeds and the user can see his old shopping cart.

### 2.5: View Store

- A guest user requests to view a store, he enters the stores' name "Aluf Hasport". The system checks and finds out the store isn't present in the system.
- A guest user requests to view a store, he enters the stores' name "Mega". The System checks and finds out that the store was closed and can't be viewed.
- A guest user requests to view a store, he enters the stores' name "Shufersal". The System returns all relevant data about the store and the products it includes.

### 2.6: Search products

- A guest user requests to search the system by product name, he enters "basketball" and requests to get only items within the price range 30-100 NIS. The system searches all stores and returns 3 different basketballs from "Shufersal".

- A guest user requests to search the system by category, he enters "sport" and requests to get only items ranked with 4 stars and above. The system searches all stores and return only 2 items a basketball from "Shufersal" and tennis racquet from "Mega sport".

## 2.7: Add products to store-cart.

- A guest user request to add 1 camping tent to his "Rikushet" store-cart, the camping tent is added to his old store-cart which contained one sleeping bag.
- A guest user requests to add 1 Plastic bottle to his store-cart "LaMetayel", the system didn't have cart to this store so a new cart is created and the plastic bottle is added to it.
- A guest user requests to add 35 tennis balls to his "Aluf Hasport" store-cart. The store only had 20 tennis balls available, the user gets notified and the action fails.

### 2.8.1: View shopping cart.

- A guest user requests to view his shopping cart. This user opened store-carts in 2 stores: "Aluf Hasport" and "Rikushet", the system returns the store-carts to the user

### 2.8.2: Edit shopping-cart

- A guest user requests to edit his "Aluf hasport" store-cart. Currently the store-cart contains 5 tennis balls, the user requests to edit the amount of tennis balls to 3. The process succeeds.
- A guest user requests to edit his "Aluf hasport" store-cart. Currently the store-cart contains 5 tennis balls, the user requests to edit the amount of tennis balls to 35. The store doesn't have the amount of tennis balls and the process fails
- A guest user requests to edit his "Aluf hasporst" store-cart. Current the store-cart contains 5 tennis balls, the user requests to edit the amount of tennis balls to -5. The system notifies the user that this amount is illegal and the process fails.

## 2.9: Purchase

- A guest user requests to buy 5 tennis balls from "Aluf Hasport" by immediate purchase, apparently the tennis balls are out of stock. The process fails and the stores' inventory is unchanged.
- A guest user requests to buy 5 tennis balls from "Aluf Hasport" by public auction. The stores' policy doesn't allow to buy tennis balls by public auction, the process fails and the stores' inventory is unchanged.

### 2.9.1: Immediate Purchase

- A guest user requests to buy 1 Lego cart from "ToysRUs" by immediate purchase, after discount calculation the final price is 75 NIS. The user agrees to pay the price but the payment request was refused. The process fails and the stores' inventory is unchanged.
- A guest user requests to buy 1 Teddy bear from "ToysRUs" by immediate purchase, after discount calculation the final price is 30 NiS. The user agrees to pay the price, the payment request was approved, but supply request was

rejected. The process fails, the payment was cancelled and the stores' inventory is unchanged.

#### 2.9.2: Bid offer request purchase

- A guest user requests to buy 1 computer screen from "KSP" by bid offer request purchase. He offers to pay 750 NIS on the screen, after 10 minutes he gets a message that the store approved the price and allows him to buy the screen. Payment and supply requests were sent and approved, and the stores' inventory was updated.
- A guest user requests to buy 1 T-shirt from "Pull & Bear" by bid offer request purchase. He offers to pay 35 NIS on the shirt, after 10 minutes he receives a message that the store rejected his bid. The process fails and the stores' inventory is unchanged.

#### 2.9.3: Auction purchase

- A guest user request to buy 1 Grey parrot from "Animal Shop" by auction purchase. The initial price was 1500 NIS and the last offer was 1600 NIS, the user offers 1700 NIS and waits. A lot of people are making offers on this parrot and the price went to 2500 NIS, the user decides the sale went out of his budget and decides to quit. No charges were made.
- A guest user decides to buy 1 wireless headphones from "Ivory" by auction purchase. The Initial price was 200 NIS and the last offer was 275 NIS. The user offers 300 NIS and waits. The auction has ended, the user made the highest offer and wins the auction. Payment and supply requests were successfully sent, and the stores' inventory is updated.

#### 2.9.4: Lottery purchase

- A guest user decides to buy 1 iPhone stand from "Walmart" by lottery purchase. The item costs 75 NIS and the user wants to buy a ticket to the lottery with 20 NIS. After the item reaches its price the lottery takes place. The user loses the lottery and is charged by 20 NIS.
- A guest user decides to buy 1 Samsung charger from "Walmart" by lottery purchase. The item costs 80 NIS and the user wants to buy a ticket to the lottery with 35 NIS. After the item reaches its price the lottery takes place. The user wins the lottery, payment and supply request were successfully sent and the stores' inventory is updated.

### 3.1: Logout

- The logged user "avi" requests to logout. After the logout the online user defined as guest, and avis' cart and details are no longer available.

### 3.2: Open store

- The logged on user "avi" requests to open a store, he enters the stores' name, description, policies, products and amounts. The store opening was successful, avi is the only store owner and granted all permissions.
- The logged on user "avi" requests to open a store, he enters the stores' name, description, policies, products and amounts. The system notifies avi he inserted negative amount to one of the products and requests to correct it.

### 3.7: Get purchase history

- The logged user "avi" requests to view his shopping history. He receives the history which contained 2 purchases, one from "Aluf Hasport" and one from "Rikushet".

### 4.1: Edit inventory.

- The logged in user "Shimrit" which serve as a store owner of "Aluf Hasport" edit a new item "soccer ball" with quantity of 10 to the store inventory, the store inventory is changed accordingly.
- The logged in user "Shimrit" which serve as a store owner of "Aluf Hasport" edit a new item "soccer ball" with quantity of -3 to the store inventory, the process fails.

### 4.2: Edit discount / buying policies

- The logged in user "Shimrit" which serve as a store owner of "Aluf Hasport" edit the buying policy to be immediate purchase and the buying policy changed accordingly.
- The logged in user "Shimrit" which serve as a store owner of "Aluf Hasport" edit the discount policy to be that every costumer that buy more than two products deserve discount of 10% off the original value. the discount policy changed accordingly.
- The logged in user "avi" which have no permissions at "Aluf Hasport" tries to edit the discount policy to be that every costumer that buy more than two products deserve discount of 10% off the original value. the process fails.

### 4.3: Appoint store owner

- The logged in user "Shimrit" which serve as a store owner of "Aluf Hasport" appoints the user "avi" which have no permissions at "Aluf Hasport" to be a store owner, the user "avi" now have all the permissions regarding policies as a store owner.
- The logged in user "Shimrit" which serve as a store owner of "Aluf Hasport" appoints the user "avi" which serve as a store owner at "Aluf Hasport" to be a store owner, the process fails .

### 4.5: Appoint store manager

- The logged in user "Shimrit" which serve as a store owner of "Aluf Hasport" appoints the user "avi" which have no permissions at "Aluf Hasport" to be a store manager and chooses the permissions he has, the user "avi" now have permissions as a store manager.
- The logged in user "Shimrit" which serve as a store owner of "Aluf Hasport" appoints the user "avi" which serve as a store owner at "Aluf Hasport" to be a store manager and chooses the permissions he has, the process fails .

### 4.6: Edit store manager permissions

- The logged in user "Shimrit" which serve as a store owner of "Aluf Hasport" edit the permissions of the user "avi" which have permissions at "Aluf Hasport" and is a store manager, the user "avi" permissions changed accordingly.

#### 4.7: Remove store manager

- The logged in user "Shimrit" which serve as a store owner of "Aluf Hasport" remove the user "avi" which serve as a store manager at "Aluf Hasport" from being a manager. the user "avi" now have no permissions as a store manager.
- The logged in user "Shimrit" which serve as a store owner of "Aluf Hasport" remove the user "avi" which does not serve as a store manager at "Aluf Hasport" from being a manager. The process fails.

#### 4.9: Get Staff Information.

- The logger in user "Shimrit" which serve as a store owner of "Aluf Hasport" watches the information of the store's staff.

#### 4.11: Get store purchase history

- The logger in user "Shimrit" which serve as a store owner of "Aluf Hasport" watches the information of the store's purchase history.

#### 5.1: Perform a management action

- Bob is appointed as a manager of store1, but not granted any permissions. Bob attempts actions (4.1)-(4.11) and fails to perform all of them.
- Bob is appointed as a manager of store1, and is granted permission to get staff information. Bob is not a manager or owner of store2. Bob attempts to get staff information for store2. Bob is denied permission.
- Bob is appointed manager of store1 with partial permissions. Bobs attempts all actions (4.1-4.11) and only succeeds on those that he has been granted permission.

#### 6.4: Receive transaction history of a buyer or store

#### 7: Payment

- Bob attempts to pay with a non existent credit card number. The payment is not approved.
- Bob attempts to pay with an expired credit card. The payment is not approved.
- Bob attempts to pay with valid payment info. The payment is approved.

#### 8: Supply

- Bob's cart contains only a PlayStation 4 sold by store1. Store1 is out of Playstation 4's. The supply request is rejected and the user is notified.
- Bob's cart contains only a PlayStation 4 sold by store1. Store1 has 10 Playstation 4's in stock. The supply request is approved and the user is notified. Store1's stock for Playstation 4's is now 9.
- Bob's cart contains only a "Real Life Pikachu" sold by store1. Store1 does not sell "Real Life Pikachu"s. The supply request is rejected and the user is notified.