**Table of Contents**

System Boot Requirement: 1.1 page 3

Update External Systems Requirement: 1.2 page 3

Payment Requirement: 1.3 page 4

Supply Requirement: 1.4 page 5

Real time alerts Requirement: 1.5 page 5

Delayed alerts Requirement: 1.6 page 6

Enter System Requirement: 2.1.1 page 6

Exit System Requirement: 2.1.2 page 7

Register Requirement: 2.1.3 page 7

Login Requirement: 2.1.4 page 8

Receive Store Info Requirement: 2.2.1 page 9

Search Store/Item Requirement: 2.2.2 page 10

Save Item - Direct Requirement: 2.2.3 page 10

Save Item – Bid Requirement: 2.2.3 page 10

View ShoppingCart Items Requirement: 2.2.4 page 11

Purchase ShoppingCart Requirement: 2.2.5 page 12

Purchase ShoppingCart – Supply Requirement: 2.2.5 page 13

Purchase ShoppingCart – Payment Requirement: 2.2.5 page 13

Logout – Registered User Requirement: 2.3.1 page 14

Open Store Requirement: 2.3.2 page 14

Stock Management – Add New Item Requirement: 2.4.1 page 15

Stock Management – Remove Item Requirement: 2.4.1 page 15

Stock Management – Change Item Information Requirement: 2.4.1 page 16

Set Store Policy Requirement: 2.4.2 page 16

Appoint Store Owner Requirement: 2.4.4 page 17

Appoint Store Manager Requirement: 2.4.6 page 18

Set Store Manager Permissions Requirement: 2.4.7 page 18

Close Store Requirement: 2.4.9 page 19

Get Store Staff List Requirement: 2.4.11 page 20

Get Store Sale History Requirement: 2.4.13 page 20

Store Manager Actions Requirement: 2.5 page 21

Get Store Sale History – System Admin Requirement: 2.6.4 page 21

Purchase ShoppingCart – discount and policy Requirement: 2.2.5 page 13

**Usecase Scenarios:**

1. Name: System Boot

Requirement: 1.1

Actors: System Administrator

Preconditions: None

Postconditions: System is running.

Input Parameters: PROPERTIES SOURCE file of external systems, JSON state file.

Actions Scenario:

1. System Administrator starts server.
2. The server reads the JSON state file and loads the objects into memory as necessary.
3. The system runs integrity checks:

* Checks for the existence of at least one administrator account.
* Creates connection with the external systems.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| פתיחה של מערכת המסחר | 1.1 | מנהל מערכת פותח את מערכת המסחר והמערכת מבצעת בדיקות יושרה | קובץ תצורה JSON בעלי נתונים נכונים לאתחול | תרחיש שמח: כל הבדיקות עוברות והמערכת עולה. |
| קובץ תצורה JSON בעל שגיאה בהגדרות בקובץ כך שהנתונים לא מותאמים לאתחול | תרחיש עצוב: שגיאה בהגדרות בקובץ JSON שליחת שגיאה |

1. Name: Update External Systems

Requirement: 1.2

Actors: System Administrator.

Preconditions: None

Postconditions: New/updated external system is online.

Input Parameters: PROPERTIES SOURCE configurations file.

Action Scenario:

1. Add:

* Create new config file for the new system.

1. Change Existing System:

* Change the PROPERTIES SOURCE file of the existing system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| עדכון של קשר עם שירותים חיצוניים | 1.2 | הוספה/שינוי של קשר | קובץ תצורה PROPERTIES SOURCE בעלי נתונים נכונים לעדכון | תרחיש שמח: השינוי עבר בהצלחה ועודכן במערכת. |
| קובץ תצורה Properties Source בעל שגיאה בהגדרות בקובץ כך שהנתונים לא מותאמים | תרחיש עצוב : השינוי נכשל ולא עודכן במערכת וקבלת שגיאה. |

1. Name: Payment

Requirement: 1.3

Actor: Client, external payment system .

Preconditions: Existing connection with an external purchase system.

Input Parameters: Transaction information.

1. Action Scenario: User enters his/her credit card information.
2. The system forwards the payment request to the external purchase system through the PaymentController object – handshake and pay.
3. If the payment is successful return confirmation number, otherwise return error message.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| תשלום | 1.3 | לקוח משלם בעבור המוצרים בעגלת הקניות שלו | כל הפריטים בעגלה זמינים בחנות, המחיר תשלום זהה לנצפה, מערכת התשלום פתוחה ומזהה את האשראי | תרחיש שמח: פרטי עסקה  התשלום בוצע בהצלחה והלקוח מקבל אישור על התשלום. |
| מערכת התשלום אינה פתוחה | תרחיש עצוב: התשלום נכשל ולקוח מקבל הודעה על שגיאה-  שפרטי העסקה שגויים. |
| קיים פריט בעגלה לא זמין בחנות |
| המחיר תשלום איננו זהה לנצפה |
| מערכת התשלום אינה מזהה את האשראי כתקין |

1. Name: Supply

Requirement: 1.4

Actor: External supply system

Preconditions: Existing connection with an external supply system.

Postconditions: Supply confirmation received.

Input Parameters: Item information and client information.

Action Scenario:

1. User enters his Shipping Address into the system or confirms .
2. The system passes the information to the external supply system through the SupplyController object –by handshake and supply.
3. The external system returns its response - success or failure.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| אספקה | 1.4 | פנייה למערכת אספקת מוצרים | פרטי חבילה ולקוח עם כתובת תקינה. | תרחיש שמח: המערכת מחזירה אישור עם האספקה בוצעה |
| פרטי חבילה ולקוח עם כתובת או לקוח לא תקינים או ריקים. | תרחיש עצוב: המערכת מחזירה הודעה שהאספקה נכשלה.  פרטי החבילה/לקוח שגויים. |

1. Name: Real time alerts

Requirement: 1.5

Actor: Client A, Client B (Optional)

Preconditions: Notification-worthy event happens for Client A, Client A is online

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| התראות בזמן אמת | 1.5 | המערכת או משתמש אחר שולחים התראה למשתמשים על אירועים בהם נדרשת תשומת ליבם | פרטי האירוע: מערכת ההתראות מקבלת את פרטי האירוע | תרחיש שמח: מערכת ההתראות מקבלת את פרטי האירוע ושולחת התראה למשתמש הרלוונטי. |
| פרטי האירוע: מערכת ההתראות לא מקבלת את פרטי האירוע | תרחיש עצוב: המשתמש לא מקבל התראה על אירוע שהיה אמור לקבל. |

Postconditions: Client A receives notification.

Input Parameters: Event

Action Scenario:

1. System/Client B causes an event that requires Client A to receive a notification.
2. The NotificationController receives the event and creates a notification object and sends it to the client.
3. Client A checks the notification queue and receives the message.
4. Name: Delayed alerts

Requirement: 1.6

Actor: Registered User A, Registered User B (Optional)

Preconditions: Notification-worthy event happens for Registered User A, Registered User A is offline

Postconditions: Registered User A’s notification queue receives the notification.

Input Parameters: Event

Action Scenario:

1. System/ Registered User B causes an event that requires Registered User A to receive a notification.
2. The NotificationController receives the event and creates a notification object and adds it the user’s queue.
3. When Registered User A becomes online the system will check the queue and the user will receive the notification.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| התראות מושהות | 1.6 | התראה למשתמש שלא מחובר בזמן קבלת ההתראה תתקבל כאשר המשתמש יתחבר | פרטי האירוע: מערכת מקבלת את פרטי האירוע | תרחיש שמח: המשתמש מקבל את ההתראה בעת כניסתו. |
| פרטי האירוע: מערכת לא מקבלת את פרטי האירוע | תרחיש עצוב: המשתמש לא קיבל התראה כאשר נכנס למערכת. |

1. Name: Enter System

Requirement: 2.1.1

Actor: Client

Preconditions: None

Postconditions: None

Input Parameters: None

Action Scenario:

1. Client enters the system via the specific interface.
2. UserController object logs the entry to the system and creates a new client object, along with a ShoppingCart object.
3. The client is now able to interact with the system

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| כניסה למערכת | 2.1.1 | משתמש נכנס למערכת המסחר. הוא מוגדר כאורח. עם כניסתו, האורח מקבל עגלת קניות, ויכול לתפקד כקונה . | אין קלט ומערכת המסחר עלתה | תרחיש שמח: המשתמש נכנס כאורח למערכת המסחר. |
| אין קלט ומערכת המסחר לא עלתה | תרחיש עצוב: מערכת המסחר לא עלתה. |

1. Name: Exit System

Requirement: 2.1.2

Actor: Client

Preconditions: None

Postconditions: None

Input Parameters: None

Action Scenario:

1. Client chooses to exit the system by closing his/her interface.
2. If the client had an active ShoppingCart then it is deleted.
3. UserController deletes the client object and updates the DB appropriately.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| יציאה מהמערכת | 2.1.2 | אורח עוזב את השוק. עם עזיבתו, הוא מאבד את עגלת הקניות שלו ואינו מוגדר כמבקר. | המשתמש עזב את המערכת. | תרחיש שמח: המשתמש עזב את המערכת ואינו מוגדר כמבקר. |
| המשתמש עזב את המערכת. | תרחיש רע: האורח עדיין במערכת. |
| תרחיש רע: האורח יצא אך עגלת הקניות לא אבדה. |

1. Name: Register

Requirement: 2.1.3

Actor: Client

Preconditions: Client is not logged in.

Postconditions: New Registered User is added to the system.

Input Parameters: Identifying information

Action Scenario:

1. Client chooses the registration option.
2. Client enters identifying information.
3. UserController checks that the unique identifiers aren’t present in the system:

* Success: The identifiers are found to be unique and the registration is successful. UserController passes the password to the SecurityController to be encrypted and saved, and passes the client a success message.
* Failure: The identifiers aren’t unique – failure message sent to client.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| רישום למערכת המסחר | 2.1.3 | אורח נרשם ע"י מסירת פרטים מזהים ייחודיים. | פרטי זיהוי: בעל שם ייחודי וסיסמה תקינה | תרחיש שמח: האורח נרשם במערכת כמנוי. |
| פרטי זיהוי: בעל שם ייחודי וסיסמה לא תקינה | תרחיש עצוב: קבלת הודעת שגיאה שהסיסמה איננה תקינה |
| פרטי זיהוי: בעל שם שקיים וסיסמה תקינה | תרחיש עצוב: קבלת הודעת שגיאה שהשם איננו תקן |

1. Name: Login

Requirement: 2.1.4

Actor: Client

Preconditions: Client is not logged in.

Postconditions: Client is logged in.

Input Parameters: Username, password

Action Scenario:

1. Client chooses the login option.
2. Client enters username and password and confirms.
3. UserController dispatches call to SecurityController with the username and password for validation.
4. Response:

* Success: SecurityController validates the login request successfully and returns a success response. UserController marks the user as logged in and returns a success message to the client.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| כניסה מזוהה | 2.1.4 | משתמש מתחבר למערכת | שם משתמש, סיסמא: שם משתמש וסיסמה תואמת הנמצאים במערכת כאורח שרשום למערכת | תרחיש שמח: המשתמש מזוהה כמבקר-מנוי. |
| שם משתמש, סיסמא: שם משתמש שאינו נמצא במערכת כאורח שרשום למערכת | תרחיש עצוב: ההתחברות נכשלה, המשתמש לא זוהה ונשאר אורח. |
| שם משתמש, סיסמא: שם משתמש וסיסמה שאיננה תואמת לשם המשתמש הנמצאים במערכת כאורח שרשום למערכת | תרחיש עצוב: ההתחברות נכשלה, המשתמש לא זוהה ונשאר אורח. |

* Failure: SecurityController finds that there is a mismatch between the username and the given password or does not find the specified username and returns a failure response. UserController returns a failure message to the client.

1. Name: Receive Store Info

Requirement: 2.2.1

Actor: Client

Preconditions: None

Postconditions: None

Input Parameters: Store/Item information.

Action Scenario:

1. Client chooses a store/item to view.
2. The StoreController object receives the request and returns the item list (if store) or specific item information.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| קבלת מידע על החנות | 2.2.1 | המשתמש מבקש מידע על חנות ועל המוצרים בה | מידע על החנות/מוצר הקיימים | תרחיש שמח: המשתמש מקבל את כל המידע הנדרש על החנות והמוצרים. |
| מידע על החנות/מוצר שלא קיימים | תרחיש שמח: המשתמש מקבל שאין מידע (מידע ריק) |
| מידע על חנות/מוצר הקיימים | תרחיש רע: המשתמש לא מקבל את כל המידע הנדרש. |
| מידע על חנות/מוצר הקיימים | תרחיש רע: המשתמש מקבל מידע שגוי. |

1. Name: Search Store/Item

Requirement: 2.2.2

Actor: Client

Preconditions: None

Postconditions: None

Input Parameters: Store/Item information to search.

Action Scenario:

1. Client enters search parameters to the system.
2. The SearchController object receives the request and searches the database/cache for items or stores that match the parameters.
3. SearchController returns and indexed list of relevant results.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| חיפוש חנות/מוצר | 2.2.2 | המשתמש מחפש חנות או מוצר | מידע על החנות/מוצר הנדרש לחיפוש (מילות מפתח) הקיימות במערכת | תרחיש שמח: הלקוח מקבל רשימה של החנויות והמוצרים הרלוונטיים לחיפוש. |
| מידע על החנות/מוצר הנדרש לחיפוש (מילות מפתח) כשמילות המפתח לא קיימות במערכת | תרחיש שמח: הלקוח מקבל רשימה ריקה |
| מידע על החנות/מוצר הנדרש לחיפוש (מילות מפתח) הקיימות במערכת | תרחיש עצוב: הלקוח מקבל רשימה עם ערכים שגויים שלא מתאימים לחיפוש\ הלקוח מקבל רשימה עם ערכים חסרים. |

1. Name: Save Item – Direct Purchase

Requirement: 2.2.3

Actor: Client

Preconditions: Item is a direct purchase type item.

Postconditions: Item is added to a new/existing ShoppingBasket.

Input Parameters: Direct purchase item

Action Scenario:

1. Client finds an item.
2. Store object checks the quantity of the item and shows to client.
3. Client chooses amount of the item to add to cart and confirms.
4. If the ShoppingCart object does not have a ShoppingBasket object of the item’s store, it creates one.
5. Either way, the item is added to the relevant ShoppingBasket.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| שמירת מוצר  (Direct) | 2.2.3 | המשתמש שומר מוצר בעגלת הקניות | מוצר מחנות קיימת | תרחיש שמח: המוצר נוסף לרשימת הקניות בהצלחה לסל המוצרים הנכון. |
| מוצר מחנות קיימת | תרחיש עצוב: המוצר לא נוסף לרשימת הקניות. |
| מוצר מחנות קיימת | תרחיש רע: המוצר לא נוסף לסל המוצרים הנכון. |

1. Name: Save Item – Bid Purchase

Requirement: 2.2.3

Actor: Client, Store owners

Preconditions: Item is a bid purchase type item.

Postconditions: Item is added to a new/existing ShoppingBasket.

Input Parameters: Bid purchase item

Action Scenario:

1. Client finds an item.
2. Client adds bid to the item.
3. Store Owners receive a notification that a new bid on an item was received.
4. Owners accept bid.
5. Client receives notification that the bid was accepted and adds to cart and confirms.
6. If the ShoppingCart object does not have a ShoppingBasket object of the item’s store, it creates one.
7. Either way, the item is added to the relevant ShoppingBasket.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| שמירת מוצר  (Bid) | 2.2.3 | המשתמש מוסיף הצעת מחיר, הבעלי חנות מאשרים והמשתמש שומר מוצר בעגלת הקניות | מוצר מחנות קיימת, הצעת מחיר | תרחיש שמח: המוצר נוסף לרשימת הקניות בהצלחה לסל המוצרים הנכון. |
| מוצר מחנות קיימת, הצעת מחיר | תרחיש עצוב: המוצר לא נוסף לרשימת הקניות. |
| מוצר מחנות קיימת, הצעת מחיר | תרחיש רע: המוצר לא נוסף לסל המוצרים הנכון. |

1. Name: View ShoppingCart Items

Requirement: 2.2.4

Actors: Client

Preconditions: None

Postconditions: None

Input Parameters: None

Action Scenario:

1. Client requests to view the shopping cart.
2. If the shopping cart contains items, it will show them sorted by store.
3. Otherwise, the client will receive a message that the shopping cart is empty.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| צפייה בעגלת הקניות | 2.2.4 | הלקוח מסתכל בעגלת הקניות שלו | ללא קלט | תרחיש עצוב : רשימת הקניות לא מופיעה או מופיעה עם פרטים שגויים. |
| ללא קלט | תרחיש שמח: רשימת הקניות מופיעה עם כל המוצרים שנשמרו עם הכמויות הנכונות והפרטי החנות הנכונים. |

1. Name: Purchase ShoppingCart

Requirement: 2.2.5

Actors: Client

Preconditions: ShoppingCart is not empty

Postconditions: Items purchased do not appear in ShoppingCart, Client Purchase History contains the items purchased and Store Sale History contains the items sold.

Input Parameters: Items to purchase, client shipping address and payment information.

Action Scenario:

1. Client selects the Purchase Cart option.
2. Client enters shipping address and confirms.
3. SupplyController passes the information to the external supply system for verification (see Usecase #16). On success:
4. Client enters payment information and confirms.
5. PaymentController passes the information to the external payment system for verification (see Usecase #17). On success:
6. Client gives final confirmation for all information and the confirmation finalizes the purchase with regards to the Payment and Supply systems.
7. StoreController object creates new Purchase & Sale objects, saves them to the database and deletes the purchased items from the user’s cart.
8. Name: Purchase ShoppingCart – Supply

Requirement: 2.2.5

Actors: External Supply System

Preconditions: Supply system connected to market system

Postconditions: Supply request confirmed by external supply system.

Input Parameters: Items to purchase, client shipping address.

Action Scenario:

1. StoreController passes supply information to SupplyController.
2. SupplyController verifies the passed information and creates a verification request from the external system.
3. If the request is successful and the information is verified, the SupplyController saves the information internally and returns the success response to the StoreController.
4. StoreController passes final confirmation to the SupplyController which sends an order request to the external supply system.
5. SupplyController returns the appropriate response to the StoreController.
6. Name: Purchase ShoppingCart – Payment

Requirement: 2.2.5

Actors: External Payment System

Preconditions: Payment system connected to market system

Postconditions: Payment request confirmed by external payment system.

Input Parameters: Items to purchase, client payment information.

Action Scenario:

1. StoreController passes payment information to PaymentController.
2. SupplyController verifies the passed information and creates a verification request from the external system.
3. If the request is successful and the information is verified, the PaymentController saves the information internally and returns the success response to the StoreController.
4. StoreController passes final confirmation on the payment to the PaymentController which then sends a payment request to the external payment system.
5. PaymentController returns the appropriate response to the StoreController.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| רכישת עגלת קניות | 2.2.5 | הלקוח רוכש את המוצרים בעגלת הקניות שלו | כל הפריטים בעגלה זמינים בחנות, המחיר תשלום זהה לנצפה, מערכת התשלום פתוחה ומזהה את האשראי ומערכת האספקה מאשרת את הכתובת | הלקוח מקבל הודעת אישור לרכישה ותשלום, מחויב ומקבל הודעה על רכישה מוצלחת ולאחר מכן עגלת הקניות מתרוקנת והמוצרים מסופקים ללקוח. |
| כל הפריטים בעגלה זמינים בחנות, המחיר תשלום זהה לנצפה, מערכת התשלום פתוחה ומזהה את האשראי ומערכת האספקה מאשרת את הכתובת | תרחיש רע: הלקוח קיבל מוצרים אך לא חויב. |
| כל הפריטים בעגלה זמינים בחנות, המחיר תשלום זהה לנצפה, מערכת התשלום פתוחה ומזהה את האשראי ומערכת האספקה מאשרת את הכתובת | תרחיש עצוב: הרכישה הצליחה אך עגלת הקניות לא התרוקנה. |
| קיים פריט בעלה שלא זמין בחנות, המחיר תשלום זהה לנצפה, מערכת התשלום פתוחה ומזהה את האשראי ומערכת האספקה מאשרת את הכתובת | תרחיש שמח: הרכישה נכשלה. |
| קיים פריט בעגלה שלא זמין בחנות, המחיר תשלום זהה לנצפה, מערכת התשלום פתוחה ומזהה את האשראי ומערכת האספקה מאשרת את הכתובת | תרחיש רע: הרכישה נכשלה ועגלת הקניות התרוקנה. |
| כל הפריטים בעגלה זמינים בחנות, המחיר תשלום זהה לא זהה לנצפה, מערכת התשלום פתוחה ומזהה את האשראי ומערכת האספקה מאשרת את הכתובת | תרחיש שמח: הרכישה נכשלה. |

1. Name: Logout – Registered User

Requirement: 2.3.1

Actor: Registered User

Preconditions: Registered User is logged in

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| התנתקות מהמערכת | 2.3.1 | משתמש מתנתק ממערכת המסחר | המשתמש יוצא מהמערכת | תרחיש רע: המשתמש התנתק אך עדיין מצליח לבצע פעולות. |
| המשתמש יוצא מהמערכת | תרחיש שמח: המשתמש לא רשום כמבקר ולא רשאי לבצע פעולות. |

Postconditions: Registered User is logged out and is now a client only.

Input Parameters: None

Action Scenario:

1. Logged in user selects the logout option.
2. UserController removes the user from the LoggedInUsers dictionary.
3. The user will receive a message stating that the logout action was successful.
4. Name: Open Store

Requirement: 2.3.2

Actor: Registered User

Preconditions: User is logged in

Postconditions: New store opened

Input Parameters: Store name, store description

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| פתיחת חנות | 2.3.2 | המשתמש פותח חנות חדשה ונהייה מייסד חנות | שם החנות, תיאור החנות | תרחיש רע: החנות נפתחה והמשתמש לא נרשם כמייסד החנות. |
| שם החנות, תיאור החנות | תרחיש עצוב : החנות לא נפתחה. |
| שם החנות, תיאור החנות | תרחיש שמח: החנות נפתחה והמשתמש הוא מייסד החנות. |

Action Scenario:

1. Registered User chooses the option to create a new store.
2. Registered User enters store name and description.
3. StoreController creates a new store instance with the inputted parameters.
4. StoreController saves the store to the database and returns to the user object a new StoreFounder Role object.
5. The user receives a message notifying him that the store opened successfully.
6. Name: Stock Management – Add New Item

Requirement: 2.4.1

Actor: Store Owner

Preconditions: User has an open store.

Postconditions: New item added to store.

Input Parameters: Item information

Action Scenario:

1. Store owner selects store.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| ניהול מלאי - הוספת מוצר | 2.4.1 | המשתמש מוסיף מוצר לחנות | מידע על המוצר | תרחיש רע: המוצר נוסף אך הפרטים שלו שגויים. |
| מידע על המוצר | תרחיש שמח: המוצר נוסף לחנות. |

1. Store Owner selects the Add New Item option.
2. Store Owner enters relevant item information.
3. StoreController checks item information entered for faults:

* Success: StoreController adds item to database.
* Failure: Store Owner receives error message.

1. Name: Stock Management – Remove Item

Requirement: 2.4.1

Actor: Store Owner

Preconditions: User has an open store and an item.

Postconditions: Item removed from store.

Input Parameters: Item ID

Action Scenario:

1. Store Owner selects store.
2. Store Owner selects item.
3. Store Owner selects the Remove Item option.
4. StoreController checks item information entered for faults:

* Success: StoreController removes item from the database.
* Failure: Store Owner receives error message.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| ניהול מלאי – מחיקת מוצר | 2.4.1 | המשתמש מוחק מוצר מהחנות | מזהה המוצר | תרחיש שמח: לא נשארים מוצרים מסוג מזהה המוצר בחנות. |
| מזהה המוצר | תרחיש עצוב: נשארים מוצרים של מזהה המוצר בחנות. |

1. Name: Stock Management – Change Item Information

Requirement: 2.4.1

Actor: Store Owner

Preconditions: User has an open store and an existing item.

Postconditions: Item information changed.

Input Parameters: Item information to change.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| ניהול מלאי- שינוי פרטי מוצר | 2.4.1 | המשתמש עורך את פרטי המוצר | המידע שצריך לשנות | תרחיש עצוב: פרטי המוצר לא שונו. |
| המידע שצריך לשנות | תרחיש שמח: פרטי המוצר שונו בהצלחה. |

Action Scenario:

1. Store Owner selects store.
2. Store Owner selects the Change Item Information option.
3. Store Owner edits item information.
4. StoreController checks item information entered for faults:

* Success: StoreController modifies item in database.
* Failure: Store Owner receives error message.

1. Name: Add Store Purchase Term

Requirement: 2.4.2

Actor: Store Owner

Preconditions: Store exists and open.

Postconditions: New Purchase Term applied.

Input Parameters: Policy parameters

Action Scenario:

1. Store Owner selects store.
2. Store Owner selects the Add Store Purchase Term option for a specific store.
3. Store Owner inputs the policy parameters.
4. Store object checks its previous policies and Founder policy for inconsistencies:

* Success: Store policy is saved to database by StoreController object.
* Failure: Inconsistency found and error message shown – go to step b.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| שינוי מדיניות קניה והנחה של חנות | 2.4.2 | המשתמש משנה את מדיניות הקניה וההנחה של החנות | פרטי המדיניות | תרחיש עצוב: שינוי המדיניות נכשל ולא עודכן. |
| פרטי המדיניות | תרחיש שמח : המדיניות שונתה בהצלחה. |

1. Name: Appoint Store Owner

Requirement: 2.4.4

Actors: Store Owner, Registered User, other Store Owners (if any)

Preconditions: Store exists and is open, Registered User isn’t one of the store’s owners.

Postconditions: Registered User is now an owner of the store under the appointing owner.

Input Parameters: None

Action Scenario:

1. Store Owner selects store.
2. Store Owner selects the Appoint Store Owner.
3. Store Owner selects the Registered User to appoint.
4. StoreController object checks that the user is not already an owner of the store.
5. Store Owners receive notification that a new petition for ownership was applied.
6. Store Owners may accept the petition.
7. If all owners accept the petition: StoreController object returns the user object a StoreOwner Role assigned to the store and saves the role in the database. New owner receives notification.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| מינוי בעל חנות | 2.4.4 | מינוי של משתמש (שאיננו בעל החנות) להיות בעל חנות | נתוני בעל החנות החדש וממנה תקינים כך שהממנה בעל החנות והחדש איננו בעל תפקיד | תרחיש שמח: המשתמש נשמר בתור בעל חנות וקיבל את ההרשאות של בעל חנות. |
| המשתמש כבר בעל החנות | תרחיש שמח : קבל הודעה שהמשתמש כבר בעל התפקיד |
| נתוני בעל החנות החדש וממנה תקינים כך שהממנה בעל החנות והחדש איננו בעל תפקיד | תרחיש רע: המשתמש נשמר בתור בעל חנות אך לא קיבל הרשאות של בעל חנות |

1. Name: Appoint Store Manager

Requirement: 2.4.6

Actors: Store Owner, Registered User

Preconditions: Store exists and is open.

Postconditions: Registered User is a manager of the store.

Input Parameters: None

Action Scenario:

1. Store Owner selects store.
2. Store Owner selects the Appoint Store Manager option.
3. Store Owner searches for the user to appoint.
4. Store Owner selects the user and confirms the selection.
5. StoreController object checks that the user isn’t already a manager of the store.
6. Registered User receives the appointment, which he may confirm.
7. StoreController object attaches a StoreManager Role (with information reading permissions only) to the user object that is assigned to the store and saves the role and permissions in the database.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| מינוי מנהל חנות | 2.4.6 | מינוי של משתמש (שאיננו מנהל החנות) להיות מנהל חנות | נתוני מנהל החנות החדש וממנה תקינים כך שהממנה מנהל החנות והחדש איננו בעל תפקיד זה | תרחיש רע: המשתמש נשמר בתור מנהל חנות אך לא קיבל הרשאות של מנהל חנות. |
| נתוני מנהל החנות החדש וממנה תקינים כך שהממנה מנהל החנות והחדש איננו בעל תפקיד זה | תרחיש טוב : המשתמש נשמר בתור מנהל חנות וקיבל את ההרשאות של מנהל חנות. |
| נתוני מנהל החנות החדש וממנה תקינים כך שהממנה מנהל החנות והחדש איננו בעל תפקיד זה | תרחיש עצוב: המשתמש לא נשמר בתור מנהל חנות |

1. Name: Set Store Manager Permissions

Requirement: 2.4.7

Actors: Store Manager, Store Owner

Preconditions: None

Postconditions: Store Manager Permissions changed.

Input Parameters: Permission parameters.

Action Scenario:

1. Store Owner selects store.
2. Store Owner selects the Set Store Manager Permissions option.
3. Store Owner selects the desired manager out of a list.
4. Store Owner inputs the new permissions for the manager.
5. Store Owner confirms the selection.
6. StoreController object adds the new permissions to the manager’s Role object’s Permissions collection.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| שינוי הרשאות של מנהל חנות | 2.4.7 | בעל חנות משנה את הרשאות מנהל חנות שהוא מינה | פרטי הרשאות תקינים של בעל החנות ושל הממונה | תרחיש רע: השינוי נכשל והמנהל לא רשאי לבצע את הפעולות החדשות. |
| פרטי הרשאות תקינים של בעל החנות ושל הממונה | תרחיש שמח: הרשאות המנהל עודכנו בהצלחה. |

1. Name: Close Store

Requirement: 2.4.9

Actors: Store Founder

Preconditions: Store is open.

Postconditions: Store is closed.

Input Parameters: None

Action Scenario:

1. Store Founder selects store.
2. Store Founder selects the Close Store option and confirms it.
3. StoreController object updates the database and cache: Sets the store to closed and all its items as hidden.
4. All managers and owners receive a notification that the store has been closed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| סגירת חנות | 2.4.9 | מייסד החנות סוגר חנות שפתח | אין קלט | תרחיש עצוב: החנות נסגרה אך עדיין משתמשים מצליחים לקבל מידע.בעלי ומנהלי החנות לא קיבלו התראה. |
| אין קלט | תרחיש רע: החנות נסגרה, משתמשים שאינם בעלי החנות או מנהלי מערכת לא יכולים לקבל מידע, בעלי ומנהלי החנות מקבלים התראה על הסגירה. החנות נשארה פתוחה |
| אין קלט | תרחיש שמח : החנות נסגרה, משתמשים שאינם בעלי החנות או מנהלי מערכת לא יכולים לקבל מידע, בעלי ומנהלי החנות מקבלים התראה על הסגירה. |

1. Name: Get Store Staff List

Requirement: 2.4.11

Actors: Store Owner

Preconditions: None

Postconditions: None

Input Parameters: None

Action Scenario:

1. Store Owner selects store.
2. Store Owner selects the Get Store Staff List option.
3. StoreController object searches the database for all users with a role assigned to the store ID and returns the list along with the users’ permissions for viewing.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| בקשה למידע על תפקידים בחנות | 2.4.11 | בעל החנות מבקש מידע על בעלי תפקידים בחנות שבבעלותו ואילו הרשאות יש למנהלים | אין קלט | תרחיש רע: בעל החנות קיבל מידע חלקי/שגוי. |
| אין קלט | תרחיש שמח: תרחיש שמח:בעל החנות מקבל את כל המידע שביקש בהצלחה. |
| אין קלט | תרחיש עצוב: בעל החנות לא קיבל את המידע |
| משתמש שאינו בעל חנות או מנהל עם הרשאות מבקש מידע | אין קלט | תרחיש עצוב: המשתמש לא מקבל את המידע |
| תרחיש רע: המשתמש מקבל את המידע |

1. Name: Get Store Sale History

Requirement: 2.4.13

Actors: Store Owner

Preconditions: None

Postconditions: None

Input Parameters: None

Action Scenario:

1. Store Owner selects store.
2. Store Owner selects Get Store Sale History option.
3. StoreController object queries the database for all sales made by the store and compiles a collection of sales and returns them to the Store Owner for viewing.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| קבלת מידע על היסטוריית רכישות בחנות (בעל החנות או מנהל עם הרשאות מתאימות) | 2.4.13 | בעל החנות או מנהל מבקש מידע על היסטוריית הרכישות בחנות | אין קלט | תרחיש רע: בעל החנות קיבל מידע חלקי/שגוי. |
| אין קלט | תרחיש שמח: בעל החנות מקבל את כל המידע שביקש בהצלחה. |

1. Name: Store Manager Actions

Requirement: 2.5

Actors: Store Manager

Preconditions: None

Postconditions: None

Input Parameters: None

Action Scenario: All Store Manager actions are to be consistent with his permissions. All actions permitted to him work as if the Store Owner performs them, and so for each particular action see Use Cases for Store Owner Requirements (2.4.x).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| פעולות של מבקר-מנוי בתפקידו כמנהל חנות | 2.5 | מבקר בתפקיד מנהל חנות מבצע פעולות ניהול בחנות בהתאם להרשאות שהוא קיבל מבעל החנות שמינה אותו | נתוני מנהל החנות החדש וממנה תקינים כך שהממנה מנהל החנות והחדש איננו בעל תפקיד זה | תרחיש רע: המשתמש נשמר בתור מנהל חנות אך לא קיבל הרשאות של מנהל חנות. |
| נתוני מנהל החנות החדש וממנה תקינים כך שהממנה מנהל החנות והחדש איננו בעל תפקיד זה | תרחיש טוב : המשתמש נשמר בתור מנהל חנות וקיבל את ההרשאות של מנהל חנות. |
| נתוני מנהל החנות החדש וממנה תקינים כך שהממנה מנהל החנות והחדש איננו בעל תפקיד זה | תרחיש עצוב: המשתמש לא נשמר בתור מנהל חנות |

1. Name: Get Store Sale History – System Admin

Requirement: 2.6.4

Actors: System Admin

Postconditions: None

Input Parameters: Store ID

Action Scenario:

1. System Admin selects store.
2. System Admin selects Get Store Sale History option.
3. StoreController object collects the store that has the passed Store ID as its ID’s sales from the database and returns them to the admin for viewing.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| קבלת מידע על היסטוריית רכישות בחנות (מבקר-מנוי בתפקיד מנהל מערכת מסחר) | 6.4 | מנהל של השוק מבקש לקבל מידע על היסטוריית הרכישות בחנות | מזהה חנות | תרחיש שמח: המנהל מקבל את כל המידע שביקש בהצלחה. |
| מזהה חנות | תרחיש עצוב : המנהל לא קיבל את המידע. |

1. Name: Purchase ShoppingCart – discount and policy

Requirement: 2.2.5

Actors: Client

Preconditions: ShoppingCart is not empty

Postconditions: Items purchased do not appear in ShoppingCart, Client Purchase History contains the items purchased and Store Sale History contains the items sold.

Input Parameters: Items to purchase, client shipping address and payment information.

Action Scenario:

1. Client selects the Purchase Cart option.
2. Client enters for payment
3. Purchase Controller passes the information and return the payment according to discount policy.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סיפור משתמש | מספר דרישה | פעולה | קלט | תוצאה |
| רכישת עגלת קניות - חישוב מחיר סופי בהתאם למדיניות הנחה | אקסטרה | בתהליך הקנייה | עגלת קניות | תרחיש עצוב: קבלת הנחה לא נכונה |
| עגלת קניות | תרחיש שמח : קבלת המחיר המותאם עפ"י ההנחה |
| עגלת קניות | תרחיש עצוב: קבלת מחיר מלא ללא ההנחה |
| עגלת קניות | תרחיש רע: קבלת הנחה כאשר סל הקנייה איננו עונה על הדרישה להנחה |
| עגלת קניות | תרחיש רע: אי קבלת הנחה כאשר סל הקניות עונה על הדרישה להנחה |
| עגלת קניות | תרחיש רע: קבלת הנחה מקסימלית כאשר חנות נתנה הנחה עבור כל ההחנות (ולהפך) |