

Use case name	#1.1 System initiation
Actors	SystemInitializer, TradeSystem
Pre-conditions	None.
Post-conditions	<ol style="list-style-type: none"> 1. SystemManager exists with admin username and password 2. 1 UserManager was created 3. 1 SystemManager was created 4. 1 PurchaseManager was created 5. There is communications with the Shipment system 6. There is communications with the Payment system
Input Parameters	None.
Main success scenario	<ol style="list-style-type: none"> 1. SystemInitializer: creates new TradeSystem 2. TradeSystem: creates new UserManager 3. TradeSystem: creates new PurchaseManager 4. TradeSystem: creates new SystemManager 5. SystemInitializer: make a connection with Shipment system 6. SystemInitializer: make a connection with PaymentCollection system
Alternatives/Extensions	<p>~5. Display Error ("The connection to the external Shipment system has been failed").</p> <p>~6. Display Error ("The connection to the external Payment system has been failed").</p>
Trigger	SystemInitializer get the order to open the trade system

Action	Data	Expected Result
#1.1 - System initiation (external systems communication)	Insert Valid communication details with Payment and Shipment external systems	Success: Trading system activated. Allows transactions including payment and shipping
	Insert invalid communication details with Payment and Shipment external systems	Failure: receive an error message that the system could not be open due to communication with the external systems.

Use case name	#2.2 Registration to the system
Actors	GuestUser
Pre-conditions	1. Current state - GuestUser.
Post-conditions	1. New RegisteredUser with input parameters(Username, Password) exists in the system
Input Parameters	Username, Password.
Main success scenario	<ol style="list-style-type: none"> 1. TradeSystem: presents two text-fields, for Username and Password And two options, "Register", "Cancel" 2. GuestUser: If user selects "Register" <ol style="list-style-type: none"> a. TradeSystem: if the details are valid, Register a new user to the system 3. GuestUser: If user selects "Cancel" <ol style="list-style-type: none"> a. TradeSystem: Displays the last menu
Alternatives/Extensions	~2.a. Display Error ("Invalid registration details!, try again").
Trigger	GuestUser requests to register to the system.

Action	Data	Expected Result
#2.2 - Registration to the system	Insert Valid user details that does not yet exist in the system	Success: The user, with the details entered, is saved to the system and a message is received that the user has successfully registered.
	Insert invalid user details that does not yet exist in the system	Failure: receive an error message that the entered information is incorrect or missing.

Use case name	#2.3 User Login to the system
Actors	GuestUser
Pre-conditions	1. Current state - GuestUser.
Post-conditions	1. GuestUser
Input Parameters	Username, Password.
Main success scenario	<ol style="list-style-type: none"> 1. TradeSystem: presents two text-fields, for Username and Password And two options, "Login", "Cancel" 2. GuestUser: If user selects "Login" <ol style="list-style-type: none"> a. TradeSystem: checks if the are exists and correct in the system b. TradeSystem: returns acknowledge message ("Hello @Username, you are now login to the system")
Alternatives/Extensions	~1. If user selects cancel, returns to the previous menu ~2.a. Display Error ("Invalid details!").
Trigger	GuestUser requests to login to the system

Action	Data	Expected Result
#2.3 - User Login to the system	Inserts valid user details that match RegisteredUser on the system.	Success: The system connects the user to the system the user goes from GuestUser to RegisteredUser.
	Inserts incorrect, missing, or nonexistent user details.	Failure: An error message is displayed according to the type of error (incorrect or missing details or user does not exist), Remains in GuestUser mode.

Use case name	#2.4 View information of store and the products of the store
Actors	GuestUser
Pre-conditions	None.
Post-conditions	None.
Input Parameters	StoreID or None.
Main success scenario	<ol style="list-style-type: none"> 1. TradeSystem: presents option: ("provide store id") 2. GuestUser: provide store id 3. TradeSystem: checks if the storeId exists 4. TradeSystem: display store information and products
Alternatives/Extensions	~1. If user selects cancel, returns to the previous menu ~3. If the storeId does not exist or is missing, Display Error("incorrect storeId").
Trigger	GuestUser Ask to see information about the store and its products.

Action	Data	Expected Result
#2.4 View information of store and the products of the store	Inserts good and existing storeId.	Success: The system returns store details and store product details.
	Inserts incorrect, missing, or nonexistent storeId.	Failure: An error message is displayed according to the type of error (incorrect or missing details or Store does not exist).

*System sequence diagram of 2.5 UC attached separately.

Use case name	#2.5.a Browse for products by different options
Actors	GuestUser
Pre-conditions	None.
Post-conditions	All the products that are presented are compatible with the chosen option.
Input Parameters	searchOptions, searchText
Main success scenario	<p>1. TradeSystem: presents options: (“by product name”, “by product category”, “by keywords”, “cancel”)</p> <p>2.1 GuestUser: If User selects “by product name”:</p> <p> a. TradeSystem: Checks if the text that was inserted is valid</p> <p> b. TradeSystem: Shows all products related to this name</p> <p>2.2 GuestUser: if User selects “by product category”:</p> <p> a. TradeSystem: Checks if the text that was inserted is valid</p> <p> b. TradeSystem: Shows all products related to this category</p> <p>2.3 GuestUser: if User selects “by product keywords”:</p> <p> a. TradeSystem: Checks if the text that was inserted is valid</p> <p> b. TradeSystem: Shows all products related to this keywords</p> <p>3. GuestUser: Choosing by filters the products that will be displayed (use-case 2.5.b).</p>
Alternatives/Extensions	<p>~1. If user selects cancel, returns to the previous menu</p> <p>~2.1.a Display Error (“Invalid Name!”).</p> <p>~2.2.a Display Error (“Invalid Category!”).</p> <p>~2.3.a Display Error (“Invalid Keywords!”).</p>
Trigger	GuestUser ask to browse for products related to different options.

Action	Data	Expected Result
#2.5.a (AT1) Browse for products by product name	Inserts a valid name	Success: Products associated with their name to the input are displayed.
	Inserts incorrect, missing, or nonexistent name.	Failure: An error message is displayed according to the type of error (incorrect or missing details or name does not exist).

Action	Data	Expected Result
#2.5.a (AT2) Browse for products by product category	Inserts a valid category	Success: Products associated with their category to the input are displayed.
	Inserts incorrect, missing, or nonexistent category.	Failure: An error message is displayed according to the type of error (incorrect or missing details or category does not exist).

Action	Data	Expected Result
#2.5.a (AT3) Browse for products by keywords	Inserts a valid keywords	Success: Products associated with their keywords to the input are displayed.
	Inserts incorrect, missing, or nonexistent keywords.	Failure: An error message is displayed according to the type of error (incorrect or missing details or keyword does not exist).

Action	Data	Expected Result
#2.5.a (AT4) Browse for products by Spelled wrong keywords	Inserts words with grammatical or letter errors	Success: find products according to similar words.
	Inserts incorrect, missing, or nonexistent keywords.	Failure: An error message is displayed according to the type of error (incorrect or missing details or keyword does not exist).

Use case name	#2.5.b Filter results according to characteristics
Actors	GuestUser
Pre-conditions	1. The user has a product list by query he did
Post-conditions	1. List of products according to the user's filtering characteristics
Input Parameters	productsList, filtersList
Main success scenario	<p>1. TradeSystem: presents options filter by: ("Price Range", "Product Rating", "Category", "Store Rating", (maybe more), "Cancel")</p> <p>2.1 GuestUser: If User selects to filter by "Price Range":</p> <p>a. TradeSystem: Checks if the input that was inserted is valid</p> <p>b. TradeSystem: Shows all products from the productsList in the price range</p> <p>2.2 GuestUser: If User selects to filter by "Product Rating":</p> <p>a. TradeSystem: Checks if the text that was inserted is valid</p> <p>b. TradeSystem: Shows all products from the productsList in requested rating and above.</p> <p>2.3 GuestUser: If User selects to filter by "Category":</p> <p>a. TradeSystem: Checks if the text that was inserted is valid</p> <p>b. TradeSystem: Shows all products from the productsList in the requested category.</p> <p>.</p> <p>.</p> <p>.</p> <p>(Could be more filters and combinations of them)</p>
Alternatives/Extensions	<p>~1. If user selects cancel, returns to the previous menu</p> <p>~2.1.a Display Error ("Invalid input!").</p> <p>~2.2.a Display Error ("Invalid input!").</p> <p>~2.3.a Display Error ("Invalid input!").</p>
Trigger	GuestUser ask to filter results by different characteristics.

Action	Data	Expected Result
#2.5.b (AT2) Filter results by Price Range	Insert valid price range and a valid products list	Success: The inserted list is filtered and only products in the price range that have been inserted remain.
	Inserts incorrect, missing, or nonexistent price range or missing a products list.	Failure: An error message is displayed according to the type of error (incorrect or missing details or no products in that price range.

Action	Data	Expected Result
#2.5.b (AT2) Filter results by Product Rating	Insert valid product rating and a valid products list	Success: The inserted list is filtered and only products within the rating and above remain.
	Inserts incorrect, missing, or nonexistent rating or missing a products list.	Failure: An error message is displayed according to the type of error (incorrect or missing details or no products in that rating range.

Action	Data	Expected Result
#2.5.b (AT3) Filter results by Category	Insert valid product Category and a valid products list	Success: The inserted list is filtered and only products within the Category remain.
	Inserts incorrect, missing, or nonexistent Category or missing a products list.	Failure: An error message is displayed according to the type of error (incorrect or missing details or no products in that Category.

Use case name	#2.6 Saving products in User's basket for specific store
Actors	GuestUser
Pre-conditions	1. GuestUser has a products list from 2.5 use-case.
Post-conditions	1. GuestUser has a basket with saved products.
Input Parameters	productsList.
Main success scenario	<ol style="list-style-type: none"> 1. GuestUser: Choose product from products list 2. TradeSystem: display product's page 3. TradeSystem: presents 2 options "Add to basket", "Go back" 4. GuestUser: if the GuestUser choose "Add to the basket": <ol style="list-style-type: none"> a. TradeSystem: adds to the user's-store basket the chosen product b. TradeSystem: displays message to the GuestUser that the product have been added to the basket c. Go back to step 1 5. TradeSystem: displays message to the GuestUser that the basket have been saved with the added products
Alternatives/Extensions	~1. Go to step 5. ~2. Display Error ("Product not available!"). ~4. Go back to step 1. ~4.a Display Error("couldn't add the product"), Go back to step 1 ~5. Display Message ("No products added")
Trigger	GuestUser choose product from the products list.

Action	Data	Expected Result
#2.6 Saving products in User's basket for specific store	Selects and saves a product/s to a specific store's basket	Success: Displaying a success message, the products were saved to the basket.
	Select product that currently not available and tries to add it to the basket	Failure: Display error message, the product could not be saved in the basket.

Use case name	#2.7 View and edit shopping cart
Actors	GuestUser
Pre-conditions	None.
Post-conditions	1. GuestUser's cart edited according to his actions
Input Parameters	None.
Main success scenario	1. TradeSystem : returns and displays GuestUser's cart 2. TradeSystem : presents the option "Edit" or "Go Back" 3 GuestUser : If GuestUser selects "Edit" <ul style="list-style-type: none"> a. TradeSystem: Displays the cart and the edit options (delete, change quantity, and more) b. GuestUser: Choose products to edit, and what to change c. TradeSystem: The system makes the changes requested by the GuestUser and displays message "Cart was Edited"
Alternatives/Extensions	~3. Go back to the previous state
Trigger	GuestUser ask to edit his shopping cart

Action	Data	Expected Result
#2.7 View and edit shopping cart	Choose the product and how to edit it	Success: The cart is finished editing, the product or products have changed accordingly and notified that the change has been made
	Select product that could not be edited or did not select any product to edit	Failure: Display error message, the product could not be edited or no product has been chosen.

*System sequence diagram of 2.8 UC attached separately.

Use case name	#2.8.a Purchase products according to the types of purchase
Actors	GuestUser
Pre-conditions	1. GuestUser has a products list from 2.5 use-case.
Post-conditions	1. A receipt was received for the purchase with the full purchase details
Input Parameters	productsList
Main success scenario	<ol style="list-style-type: none"> 1. GuestUser: Asks to buy a product from the productsList 2. TradeSystem: The system displays the buying options for each product ("Immediate Purchase", "Auction", "Lottery purchase") 3. GuestUser: Choose 1 option: <ol style="list-style-type: none"> 3.1. If the GuestUser choose "Immediate Purchase": <ol style="list-style-type: none"> a. TradeSystem: Ask Payment Details from the GuestUser b. GuestUser: Provides personal payment information and approves the purchase. c. TradeSystem: Issues and returns a receipt for the purchase 3.2. If the GuestUser choose "Auction": <ol style="list-style-type: none"> a. TradeSystem: Showing the highest bid so far b. GuestUser: Submits the highest bid c. TradeSystem: The auction time ends and the TradeSystem announces that GuestUser has won. Ask Payment Details from the GuestUser d. GuestUser: Provides personal payment information and approves the purchase. e. TradeSystem: Issues and returns a receipt for the purchase 3.3. If the GuestUser choose "Lottery purchase": <ol style="list-style-type: none"> a. TradeSystem: Asking the GuestUser for his bid for the lottery. b. GuestUser: Offers offer and provides payment details c. TradeSystem: When reached the amount set first for the product, announces the winner, charges GuestUser and issues a receipt for the purchase.
Alternatives/Extensions	~3. Go back to the previous state ~3.2.c Display Error("Another user won the auction") ~3.3.c Display Error("The total amount collected did not reach the amount initially set for the product")
Trigger	GuestUser ask to buy a product

Action	Data	Expected Result
#2.8.a (AT1) Immediate Purchase	Buying Product on Immediate Purchase, provides proper payment details	Success: The purchase is made, the payment goes between the buyer and the store with external systems and the user receives a receipt with the transaction details
	Buying Product on Immediate Purchase, provides invalid payment details, or the product is not available in Immediate Purchase	Failure: Display error message, the product could not be sold in Immediate Purchase or payment details are missing.

Action	Data	Expected Result
#2.8.a (AT2) Auction	Buying Product on Auction, submits the highest bid on a product that sold in auction	Success: The purchase is made, the payment goes between the buyer and the store with external systems and the user receives a receipt with the transaction details
	Buying Product on Auction. submits bid, but not the highest bid. payment details are missing, or the product is not available in auction.	Failure: Display error message, the product could not be sold in auction or payment details are missing or someone submitted a higher bid.

Action	Data	Expected Result
#2.8.a (AT3) Lottery purchase	Buying Product on Lottery purchase, submits his bid on a product that sold in Lottery purchase. The total amount that was set, has reached.	Success: The user has won. The purchase is made, the payment goes between the buyer and the store with external systems and the user receives a receipt with the transaction details
	Buying Product on Lottery purchase. submits a bid, but someone else wins the lottery. payment details are missing, or the product is not available in auction.	Failure: Display error message, the product could not be sold in Lottery or payment details are missing or someone else won the lottery .

Use case name	#2.8.b Purchase products according to discount types
Actors	GuestUser
Pre-conditions	1. GuestUser has a products list from 2.5 use-case.
Post-conditions	1. A receipt was received for the purchase with the full purchase details
Input Parameters	productsList
Main success scenario	<p>1. GuestUser: Asks to buy a product from the productsList, with "Visible Discount":</p> <ol style="list-style-type: none"> TradeSystem: Displays the current product price, discount price, and discount details. Ask Payment Details from the GuestUser. GuestUser: Provides personal payment information and approves the purchase. TradeSystem: Issues and returns a receipt for the purchase. <p>2. GuestUser: Asks to buy a product from the productsList, with "Conditional Discount":</p> <ol style="list-style-type: none"> If the user meets the discount conditions: <ol style="list-style-type: none"> The system displays the standard product price, discounted product price, discount duration, discount percentage and discount conditions. Ask Payment Details from the GuestUser. GuestUser: Provides personal payment information and approves the purchase. TradeSystem: Issues and returns a receipt for the purchase. <p>3. GuestUser: Asks to buy a product from the productsList, with "Hidden discount":</p> <ol style="list-style-type: none"> GuestUser: Enters coupon code TradingSystem: If the code is correct the TradingSystem displays the updated price of the product after a discount, the discount, the percentage of the discount and its time. Ask Payment Details from the GuestUser. GuestUser: Provides personal payment information and approves the purchase. TradeSystem: Issues and returns a receipt for the purchase.
Alternatives/Extensions	~3. Go back to the previous state
Trigger	GustUser ask to buy a product

Action	Data	Expected Result
#2.8.b (AT1) Visible Discount	Buying Product under Visible Discount, provides proper payment details	Success: The purchase is made, the payment goes between the buyer and the store with external systems and the user receives a receipt with the transaction details
	Buying Product under Visible Discount, provides invalid payment details, or the product is not available under Visible Discount	Failure: Display error message, the product could not be sold under Visible Discount or payment details are missing.

Action	Data	Expected Result
#2.8.b (AT2) Conditional Discount	Buying Product under Conditional Discount, user meets the conditions for the product discount. provides proper payment details	Success: The purchase is made, the payment goes between the buyer and the store with external systems and the user receives a receipt with the transaction details
	Buying Product under Conditional Discount, user does not meet the conditions to get the discount	Failure: Display error message for the user that said that he is not meet the conditions for the discount.

Action	Data	Expected Result
#2.8.b (AT3) Hidden Discount	Buying Product under Hidden Discount, provides valid coupon and provides proper payment details	Success: The purchase is made, the payment goes between the buyer and the store with external systems and the user receives a receipt with the transaction details
	Buying Product under Hidden Discount, provides invalid coupon code.	Failure: Display error message, the product could not be sold under Hidden Discount the coupon is wrong.

Differences between GuestUser and RegisteredUser.

Use Case **2.4** performs the same for RegisteredUser.

Use Case **2.5** performs the same for RegisteredUser.

Use Case **2.6** performs differently for RegisteredUser: the Basket and the cart has been saved in his watch list for later date.

Use Case **2.7** performs the same for RegisteredUser.

Use Case **2.8** performs the same for RegisteredUser.

Use case name	#3.1 Logout
Actors	RegisteredUser
Pre-conditions	Have valid registration, User state- Logged in.
Post-conditions	User state become GuestUser
Input Parameters	None.
Main success scenario	1. TradeSystem ask: "are you sure" If you press yes, the system logs out the RegisteredUser , the RegisteredUser becomes unlogged.
Alternatives/Extensions	User press cancel, the RegisteredUser stays logged in.
Trigger	The RegisteredUser wanted to logout.

Action	Data	Expected Result
#3.1 AT - Logout	None	User Logged out
	None	

Use case name	#3.2 Open store
Actors	RegisteredUser
Pre-conditions	<ol style="list-style-type: none"> 1. Begin logged in. 2. Store name doesn't exist.
Post-conditions	<ol style="list-style-type: none"> 1. New store opens to buy the RegisteredUser. 2. Registered RegisteredUser become the main StoreOwner.
Input Parameters	<ol style="list-style-type: none"> 1. Store name. 2. Store purpose.
Main success scenario	<ol style="list-style-type: none"> 1. RegisteredUser: enter new store name 2. TradeSystem: checks Store name doesn't exist. <ol style="list-style-type: none"> 2.1 TradeSystem: asks RegisteredUser's identifications . 2.2 RegisteredUser: enter the identifications. 3. New Store opens in the system 4. RegisteredUser becomes a StoreOwner in this store.
Alternatives/Extensions	<ol style="list-style-type: none"> 1. RegisteredUser: enter same StoreName that exists already <ol style="list-style-type: none"> 1.1 TradeSystem: Display Error ("Store name already exists") 1.2 TradeSystem asks RegisteredUser for new store name.
Trigger	RegisteredUser want to open a store.

Action	Data	Expected Result
#3.2 AT - Open Store	Free to use store name	New Store opens with the chosen name
	User choose name that is already exists	Failure: getting an error message about the name.

Use case name	#3.7 Purchase history
Actors	RegisteredUser
Pre-conditions	None.
Post-conditions	None.
Input Parameters	RegisteredUserID, specific dates of purchase history
Main success scenario	<ol style="list-style-type: none"> 1. RegisteredUser asks the system his purchase history. 2. TradeSystem displays the user's purchase history.
Alternatives/Extensions	<ol style="list-style-type: none"> 1. RegisteredUser purchase history is empty.
Trigger	RegisteredUser asks to view his own purchase history.

Action	Data	Expected Result
#3.7 AT - Purchase history	Specific date range	Display Purchase history
	Wrong date – not real calendar date	Failure: error message about the fake date

Use case name	#4.1 StorageManaging
Actors	StoreOwner
Pre-conditions	None.
Post-conditions	Store Storage changes.
Input Parameters	1. StoreOwner
Main success scenario	1. StoreOwner Add new product to store. 2. StoreOwner Delete exists product from the store. 3. StoreOwner Edit product details.
Alternatives/Extensions	1.There are no items to delete. 2.The store is empty, no items to edit.

Action	Data	Expected Result
#4.1 AT - StorageManaging	Add item to store	Item successfully added
	Delete non exist item	Failure: error message

Use case name	#4.3 Add new StoreOwner.
Actors	StoreOwner

Pre-conditions	The StoreOwner should be StoreOwner in his Store.
Post-conditions	An exist RegistertUser become StoreOwner.
Input Parameters	RegisterdUserID of the new StoreOwner.
Main success scenario	1. StoreOwner : enter the user's details. 2. RegisteredUser : become StoreOwner.
Alternatives/Extensions	1.The User is already StoreOwner.
Trigger	StoreOwner asks to change an RegisteredUser to StoreOwner.

Action	Data	Expected Result
#4.3 AT – Add new StoreOwner	UserID to become StoreOwner	User is now the new StoreOwner
	UserID of someone who is already StoreOwner	Failure: an error message about invalid UserID

Use case name	#4.5 Add new StoreManager.
Actors	StoreOwner
Pre-conditions	The user should be StoreOwner in his Store.
Post-conditions	An exist RegisteredUser become StoreManager.
Input Parameters	RegisteredUserID of the new StoreManager.

Main success scenario	1. StoreOwner enter the RegisteredUser's details. 2. RegisteredUser become StoreManager.
Alternatives/Extensions	1.The User is already StoreManager
Trigger	StoreOwner asks to change an RegisteredUser to StoreManager.

Action	Data	Expected Result
#4.5 AT- add new store manager	UserID to become StoreManager	User is now the new StoreManager
	UserID of someone who is already StoreManager.	Failure: an error message about invalid UserID

Use case name	#4.6 Change StoreManager permissions.
Actors	StoreOwner
Pre-conditions	StoreManger exists.
Post-conditions	StoreManager permissions changed.
Input Parameters	StoreManagerId, New permissions to add, permissions to delete.
Main success scenario	1. StoreOwner Enter StoreManagerID. 1.1 StoreOwner Adds permissions 1.2 StoreOwner Deletes permissions.
Alternatives/Extensions	1. There is no StoreManager in the store. 1.1 Display Error("No manager found")

Trigger	StoreOwner asks to change an StoreManager permissions.
---------	--

Action	Data	Expected Result
#4.6 AT - Change StoreManager permissions.	StoreManagerId, New permissions to add	New permissions added to the StoreManager.
	StoreManagerId ,permissions to that doesn't exits.	Failure: an error message about invalid permission to delete.

Use case name	#4.7 Delete StoreManager
Actors	StoreOwner
Pre-conditions	At least 1 Store manager in the store.
Post-conditions	StoreManager become RegisteredUser.
Input Parameters	StoreManagerID.
Main success scenario	1. StoreOwner Enter StoreManagerID 1.1 StoreOwner deletes RegisteredUser .
Alternatives/Extensions	1. The StoreManager is not created by this StoreOwner
Trigger	StoreOwner asks to delete an StoreManger from his store.

Action	Data	Expected Result
#4.7 AT - Delete StoreManager	StoreManagerID	StoreManager deleted .
	StoreManagerID that not created by the actor.	Failure: an error message about invalid StoreManager to delete.

Use case name	#4.10 Watch Store purchase history
Actors	StoreOwner
Pre-conditions	None
Post-conditions	None
Input Parameters	None
Main success scenario	1. TradeSystem : show a list of all purchase history 2. StoreOwner : choose to see all history or part of it.
Alternatives/Extensions	1. StoreOwner have no purchase history. 2. StoreOwner : choose specific time to view 2.1 TradeSystem Display Error("no purchase history")
Trigger	StoreOwner asks to view the purchase history in his own store.

Action	Data	Expected Result
--------	------	-----------------

#4.10 AT - Watch Store purchase history	History of specific exist date	Display the purchase history of the store on the specific date
	History of specific non exists date	Failure: an error message about invalid Specific date to view.

Use case name	#5.1 Manage Store
Actors	StoreManager
Pre-conditions	<ol style="list-style-type: none"> 1. Current state – StoreManager 2. StoreManager must be authorized to carry out management action
Post-conditions	None
Input Parameters	None
Main success scenario	<ol style="list-style-type: none"> 1. StoreManager carries out management action. 2. Store is successfully updated
Alternatives/Extensions	~1. StoreManager is not authorized to carry out management action. Display Error (“Unauthorized action”).
Trigger	Appointed StoreManager attempts to update store

Action	Data	Expected Result
#5.1 Manage Store	StoreManager attempts to carry out management action he is authorized to do	Action is carried out
	StoreManager attempts to carry out management action he is not authorized to do	Failure: an error message about attempting unauthorized action..

Use case name	#6.4 View Purchase History
Actors	SystemManager
Pre-conditions	1. Current state - SystemManager
Post-conditions	1. Purchase history is displayed
Input Parameters	StoreID or Username
Main success scenario	<p>1. TradeSystem: presents 2 options: (“provide store ID” or “provide Username”)</p> <p>2.1 SystemManager:</p> <p> If SystemManager selects “provide store ID”</p> <p> a. TradeSystem: checks if the storeID exists</p> <p> b. TradeSystem: Displays store’s purchase history</p> <p>2.2 SystemManager:</p> <p> if SystemManager selects “provide Username”</p> <p> a. TradeSystem: checks if the Username exists</p> <p> b. TradeSystem: Displays user’s purchase history</p>
Alternatives/Extensions	<p>~2.1.a. Display Error (“StoreID not found!”).</p> <p>~2.2.a. Display Error (“Username not found!”).</p>
Trigger	SystemManager looks up purchase history

Action	Data	Expected Result
#6.4 View Purchase History using Username	Valid Username	Display User’s purchase history
	Invalid Username	Failure: an error message about invalid Username.
#6.4 View Purchase History using StoreID	Valid StoreID	Display store’s purchase history
	Invalid StoreID	Failure: an error message about invalid StoreID.

#6.4 View Purchase History	Null	Failure: an error message about invalud input.
----------------------------	------	--

Use case name	#7 Payment Collection
Actors	PurchaseManager, PaymentCollection
Pre-conditions	1. Transaction details are valid
Post-conditions	1. Payment processed for exact amount written in transaction details
Input Parameters	Transaction details
Main success scenario	1. PurchaseManager transfers transactions to PaymentCollection 2. PaymentCollection charges user's account 3. PaymentCollection returns confirmation of payment.
Alternatives/Extensions	~2. Payment cannot be processed due to: ~ Transaction details are invalid Display Error("Payment cannot be processed")
Trigger	A Transaction has taken place, need to charge user

Action	Data	Expected Result
#7 Payment Collection	Valid Transaction details	Confirmation that package shipment request was received by Shipment
	Invalid Transaction details	Failure: an error message about invalid Transaction details.

Use case name	#8 Package Shipment
Actors	PurchaseManager, Shipment
Pre-conditions	<ol style="list-style-type: none"> 1. Package details are valid 2. User details are valid
Post-conditions	none
Input Parameters	Package details, User details
Main success scenario	<ol style="list-style-type: none"> 1. PurchaseManager transfers package and user details to Shipment. 2. Shipment verifies details. 3. Shipment returns confirmation that shipment request has been accepted.
Alternatives/Extensions	~2.:Display Error ("Shipment Error") ~ User's info is incorrect ~ Package details are insufficient
Trigger	A transaction has taken place need to ship package to user

Action	Data	Expected Result
#8 Package Shipment	Valid User and Package details	Confirmation that package shipment request was received by Shipment
	Invalid User details and valid package details	Failure: an error message about invalid User details.
	Valid User details and invalid package details	Failure: an error message about invalid Package details.

