Use Case Name: Add Item to Cart

Scope: Store

Level: user goal

Primary Actor: customer

Stakeholders and Interests:

- Customer: wants to be able to purchase multiple different items
- Company: wants to be able to keep track of the items the customer purchases so that they can get their items in a timely manner

Preconditions: Customer wants to add an item to their cart

Success Guarantee (or Postconditions):

The item is added to the cart, and the subtotal is updated

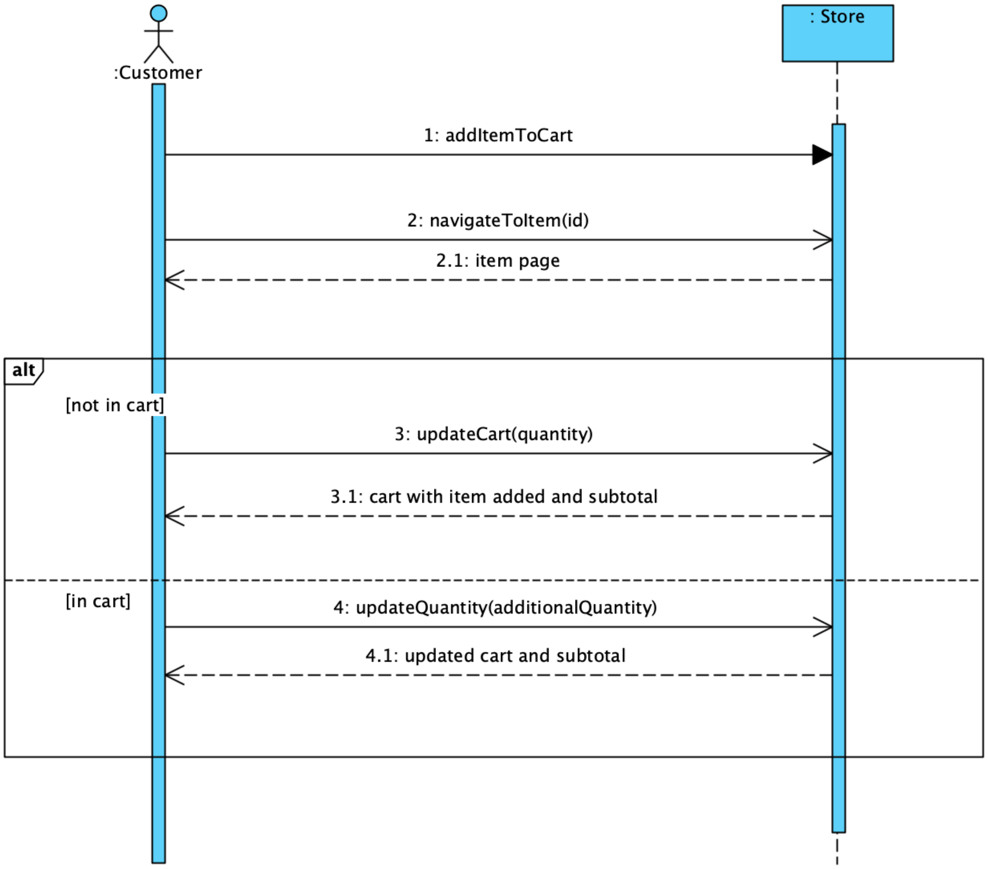
Main Success Scenario (or Basic Flow):

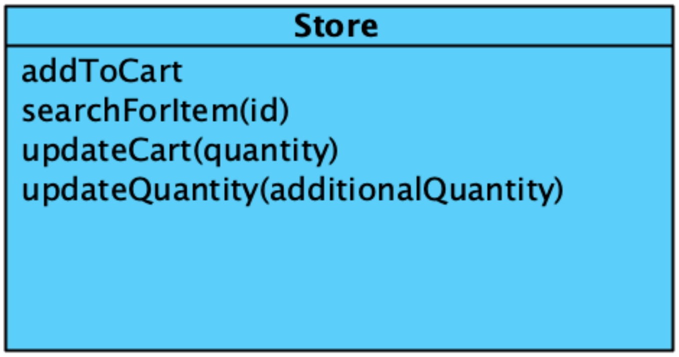
1. Customer navigates to the item
2. Customer clicks add to cart button
3. Customer enters the quantity they wish to purchase
4. Items are added to cart
5. Subtotal is updated
6. The customer is informed that the items have been successfully added to the cart

Extensions (or Alternative Flows):

2a. Item is already in cart:

1. Customer selects how many they wish to add to the cart
2. The customer's selection is added to their previous selection


:Customer
: Store
1: addItemToCart
2: navigateToItem(id)
2.1: item page
alt
[not in cart]
3: updateCart(quantity)
3.1: cart with item added and subtotal
[in cart]
4: updateQuantity(additionalQuantity)
4.1: updated cart and subtotal



Operation: addToCart
Cross References: Add Item to Cart
Preconditions: There is an instance of a cart created
Postconditions: The cart has an item added to it

Operation: searchForItem(id)
Cross References: Add Item to Cart
Preconditions: The customer knows what item they are searching for
Postconditions: The customer has navigated to the item page

Operation: updateCart(quantity)
Cross References: Add Item to Cart
Preconditions: The customer has navigated to the item page, and the item isn't in the cart
Postconditions: The item and its desired quantity is added to the cart

Operation: updateQuantity(additionalQuantity)
Cross References: Add Item to Cart
Preconditions: The customer has navigated to the item page, and the item is already in the cart
Postconditions: The quantity of the item in the cart is increased by additionalQuantity

Use Case Name: Remove Item from Cart

Scope: Store

Level: user goal

Primary Actor: customer

Stakeholders and Interests:

- Customer: wants to be able to keep track of the items to be purchased, and make adjustments to them (remove, alter number of) if needed.
- Company: wants to be able to properly track an order, including when items are removed, so that purchases are accurate.

Preconditions: Customer wants to remove an item from their cart

Success Guarantee (or Postconditions): The item the customer wanted removed from the cart will be removed and the subtotal calculated.

Main Success Scenario (or Basic Flow):

1. Customer navigates to their current shopping cart instance
2. The customer clicks on the item to be removed from the cart
3. The customer clicks on the UPDATE ORDER button
4. The customer clicks remove item
5. The machine shows a verification that the user wants to remove the item
6. The customer selects yes
7. The subtotal is updated

Extensions (or Alternative Flows):

4a. There are multiple of the same item in the cart

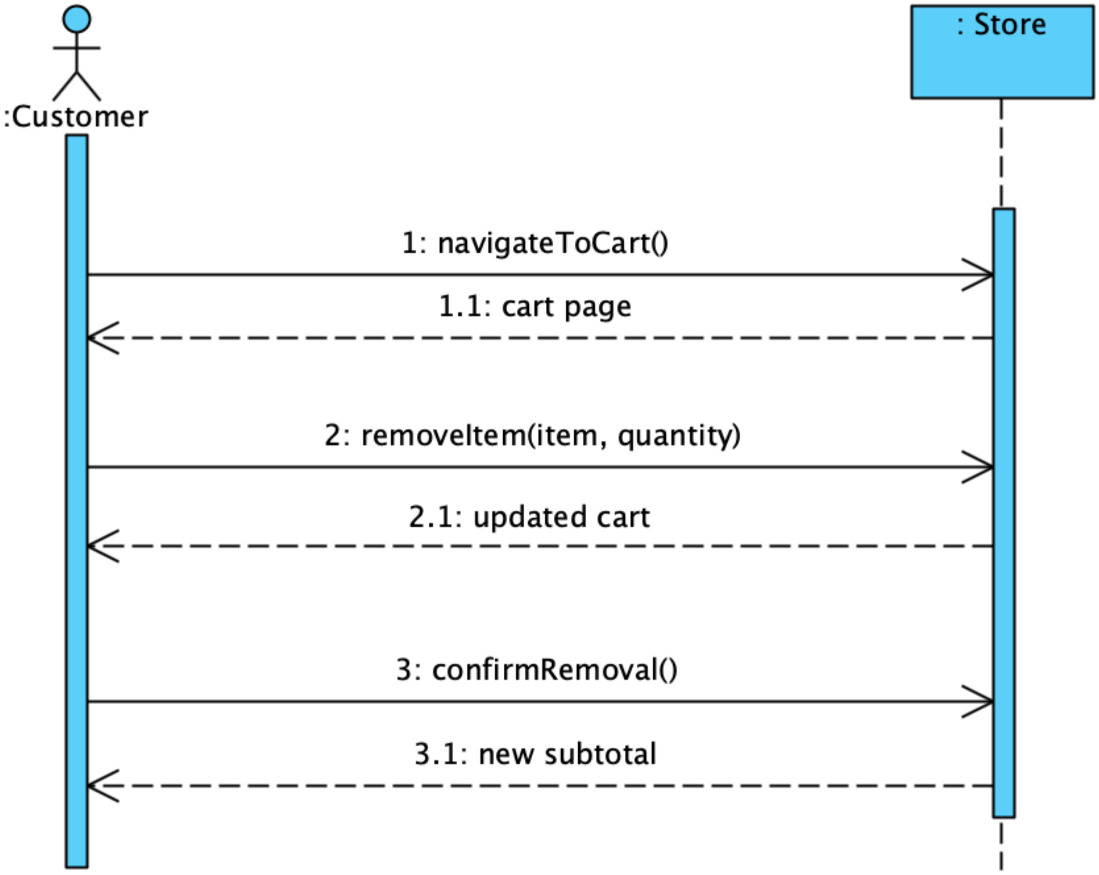
  1. Customer selects how many of the item they would like to remove
  2. Customer confirms the amount to remove

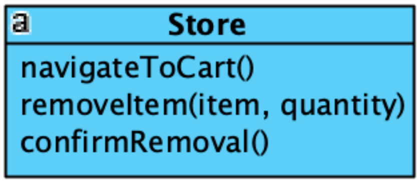
6a. The customer doesn't want to remove the item from cart

  1. The customer changes their mind and selects no

7a. There is no resulting change in the order

  1. The subtotal is not changed and thus not updated


:Customer
: Store
1: navigateToCart()
1.1: cart page
2: removeItem(item, quantity)
2.1: updated cart
3: confirmRemoval()
3.1: new subtotal



Operation: navigateToCart()
Cross References: Remove Item from Cart
Preconditions: The cart exists
Postconditions: The cart is displayed to the user

Operation: removeItem(item, quantity)
Cross References: Remove Item from Cart
Preconditions: there are at least "quantity" instances of "item" in the carty
Postconditions: the item is

Operation: confirmRemoval()
Cross References: Remove Item from Cart
Preconditions: The customer has selected to remove an item from their cart
Postconditions: If confirmed, the removal is finalized
Else, the removal is cancelled

Use Case Name: Make Account

Scope: Store

Level: user-goal

Primary Actor: Customer

Stakeholders and Interests:

- Customer: wants to store their information so that it's saved for next time
- Company: has to keep track of the customer's login information

Preconditions: Customer is asked if they want to create an account

Success Guarantee (or Postconditions): The customer's login information is saved and their cart is tied to their account

Main Success Scenario (or Basic Flow):

1. Customer selects they want to create an account
2. The customer is prompted for a username
3. The customer is prompted for a password

Repeat steps 2 and 3 until the login is validated

4. The user is asked to re-enter their username and password for validation
5. The customer is asked to confirm the account creation

Extensions (or Alternative Flows):

2a. The username is invalid

1. The customer is informed why the username is invalid (either it is taken or does not meet requirements)
2. The customer is asked to enter another username

3a. The password is invalid

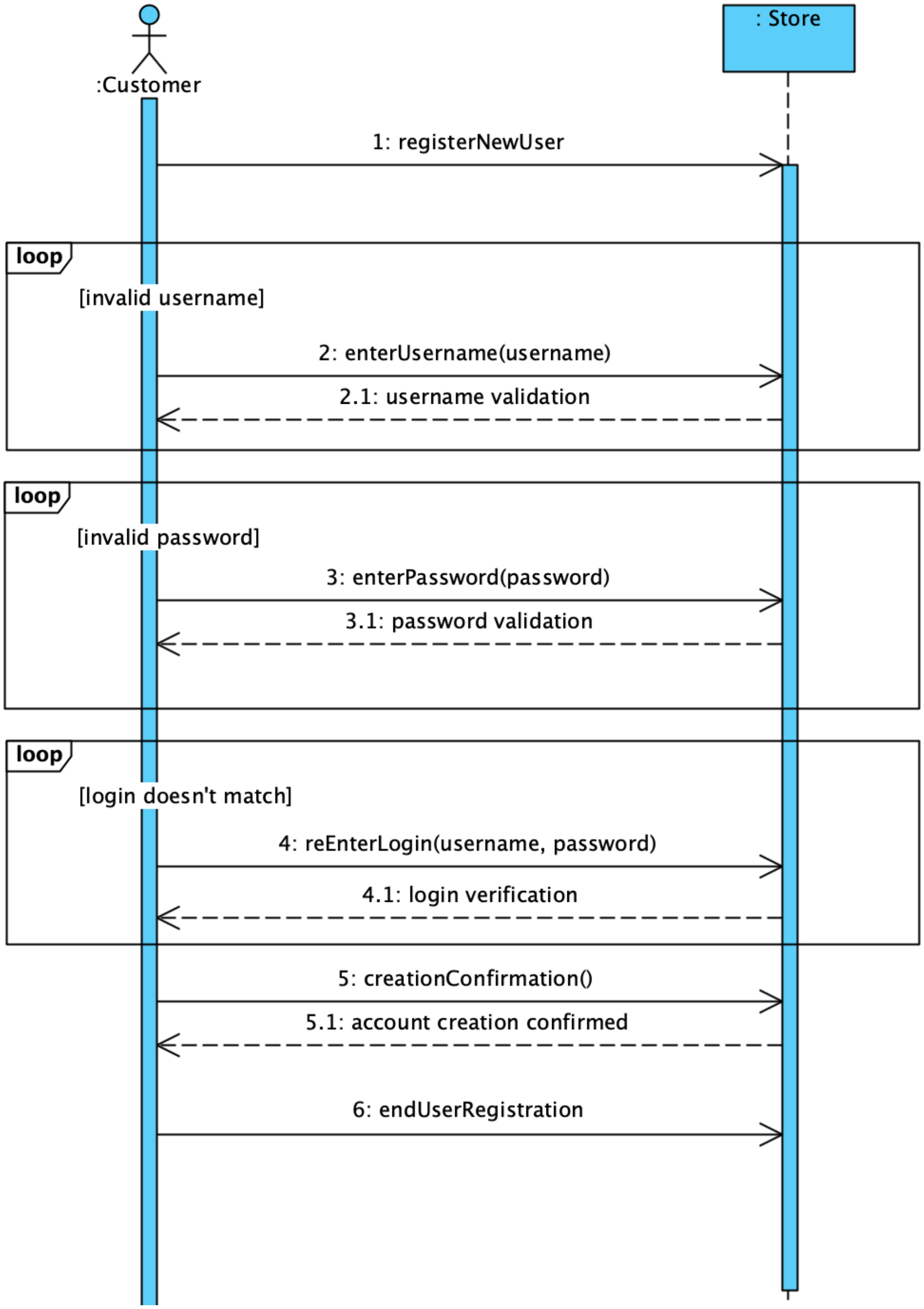
1. The customer is informed why the password is invalid (i.e. the requirements it violates)
2. The customer is asked to enter another password

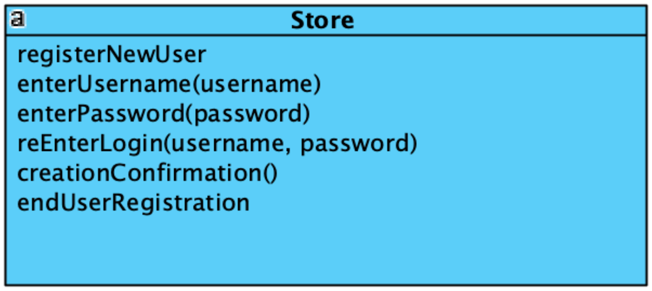
4a. The customer's username does not match

1. The customer is informed that their usernames don't match
2. The customer is prompted to try again

4b. The customer's password doesn't match

1. The customer is informed that their passwords don't match
2. The customer is prompted to try again


:Customer
: Store
1: registerNewUser
loop
[invalid username]
2: enterUsername(username)
2.1: username validation
loop
[invalid password]
3: enterPassword(password)
3.1: password validation
loop
[login doesn't match]
4: reEnterLogin(username, password)
4.1: login verification
5: creationConfirmation()
5.1: account creation confirmed
6: endUserRegistration



Operation: registerNewUser
Cross References: Make Account
Preconditions: The customer has elected to create an account.
Postconditions: A new account is registered.

Operation: enterUsername(username)
Cross References: Make Account
Preconditions: The customer is prompted to enter a username.
Postconditions: The username is validated.

Operation: enterPassword(password)
Cross References: Make Account
Preconditions: The customer is prompted to enter a password.
Postconditions: The password is validated.

Operation: reEnterLogin(username, password)
Cross References: Make Account
Preconditions: The customer is prompted to re-enter their valid username and password
Postconditions: The two logins are compared, then validated if they match

Operation: creationConfirmation()
Cross References: Make Account
Preconditions: The two logins matched
Postconditions: If the user confirms, the account is created
Else, account creation is canceled

Operation: endUserRegistration
Cross References: Make Account
Preconditions: The registration was sucessful
Postconditions: The process of creating an account ends