And I should have 14 pieces of 7 product

And I should have 2 products

And the overall cart amount should be 17

When I add 10 pieces of 5 product

Then I should have 13 pieces of 5 product

And I should have 14 pieces of 7 product

And I should have 2 products

And the overall cart amount should be 27

Scenario: Change product count in the cart

Given there is a cart with 5 pieces of 7 product

When I set 3 pieces for 7 product

Then I should have 3 pieces of 7 product

Scenario: Remove products from the cart

Given there is a cart with 5 pieces of 7 product

When I add 14 pieces of 7 product

And I clear cart

Then I should have empty cart

3. Add the storage test features/storage . feature file:

Feature: Shopping cart storage

I need to be able to put items into a storage

Scenario: Checking empty storage Given there is a clean storage Then I should have empty storage

Scenario: Save items into storage Given there is a clean storage

When I save 3 pieces of 7 product to the storage Then I should have 3 pieces of 7 product in the storage

4. Add implementation for all steps in the features/bootstrap/CartContext.php file: <?php

use Behat\Behat\Context\SnippetAcceptingContext; use book\cart\Cart;

use book\cart\features\bootstrap\storage\FakeStorage; use yii\di\Container; use yii\web\Application;

require\_once DIR . '/bootstrap.php';

class CartContext implements SnippetAcceptingContext {

/\*\*

\* @var Cart

\* \*/

private $cart;

/\*\*

\* @Given there is a clean cart \*/

public function thereIsACleanCart()

{

$this->resetCart();

}

/\*\*

\* @Given there is a cart with :pieces of :product product \*/

public function thereIsAWhichCostsPs($product, $amount)