

گزارش کار پروژه تست 6

محمدحسین عقیلی 810199576

سید علی امام زاده 810199377

شناسه آخرین کامیت: db38739b551f193970266b02adb02717bc96e928

بخش 1

سناریو گفته شده در صورت بویسله katalon ایجاد کردیم و کد اکسپورت شده در فایل های پروژه موجود است. (در فایل اپلود شده هم موجود است)

بخش 2

ابتدا فایل futere مربوط به cucumber و سناریو ها را مینویسیم:

```
Feature: User Actions
```

```
Scenario: User adds credit
```

```
Given the user has a credit of 100
```

```
When the user adds credit of 50
```

```
Then the new credit should be 150
```

```
Scenario: User removes item from buy list
```

```
Given the user has a buy list with item ID "1" and quantity 2
```

```
When the user removes 1 item with ID "1" from the buy list
```

```
Then the new quantity of item ID "1" in the buy list should be 1
```

```
Scenario: User withdraws credit
```

```
Given the user has a credit of 100
```

```
When the user withdraws credit of 30
```

```
Then the new credit should be 70
```

سپس کلاس stepdefinition را مینویسیم که سناریو های ما را به کد ترجمه کند.

در حقیقت هدف BDD و cucumber این است که تست خوانا برای کسانی که فنی نیستند داشته باشیم و تست ما human readable باشد.

کد stepdefinition :

```
package BDD;
```

```
import exceptions.CommodityIsNotInBuyList;
```

```

import exceptions.InsufficientCredit;
import exceptions.InvalidCreditRange;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
import model.Commodity;
import model.User;

import static org.junit.Assert.assertEquals;

public class UserStepDefinitions {
    private User user;

    @Given("the user has a credit of {float}")
    public void givenTheUserHasACreditOf(float initialCredit) {
        user = new User();
        user.setCredit(initialCredit);
    }

    @When("the user adds credit of {float}")
    public void whenTheUserAddsCreditOf(float creditToAdd) throws
InvalidCreditRange {
        user.addCredit(creditToAdd);
    }

    @When("the user withdraws credit of {float}")
    public void whenTheUserWithdrawsCreditOf(float creditToWithdraw) throws
InsufficientCredit {
        user.withdrawCredit(creditToWithdraw);
    }

    @Then("the new credit should be {float}")
    public void thenTheNewCreditShouldBe(float expectedCredit) {
        assertEquals(expectedCredit, user.getCredit(), 0.01);
    }

    @Given("the user has a buy list with item ID {string} and quantity
{int}")
    public void givenTheUserHasABuyListWithItemIDAndQuantity(String
itemId,int quantity) {
        user = new User();
        Commodity com1 = new Commodity();
        com1.setId(itemId);
        user.addBuyItem(com1);
        user.addBuyItem(com1);
    }

    @When("the user removes {int} item with ID {string} from the buy list")
    public void whenTheUserRemovesItemFromTheBuyList(int itemsToRemove,String
IdItemToRemove) throws CommodityIsNotInBuyList {
        Commodity commodity = new Commodity();
        commodity.setId(IdItemToRemove);
        for (int i = 0; i < itemsToRemove; i++) {
            try {
                user.removeItemFromBuyList(commodity);
            } catch (CommodityIsNotInBuyList ignored) {
            }
        }
    }
}

```

```

    }
}

@Then("the new quantity of item ID {string} in the buy list should be {int}")
public void thenTheNewQuantityInBuyListShouldBe(String itemId, int expectedQuantity) {
    assertEquals(expectedQuantity,
        user.getBuyList().get(itemId).intValue());
}
}

```

نتیجه:

```

Scenario: User adds credit # src/test/java/feature.feature:3
  Given the user has a credit of 100 # BDD.UserStepDefinitions.givenTheUserHasACreditOf(float)
  When the user adds credit of 50 # BDD.UserStepDefinitions.whenTheUserAddsCreditOf(float)
  Then the new credit should be 150 # BDD.UserStepDefinitions.thenTheNewCreditShouldBe(float)

Scenario: User removes item from buy list # src/test/java/feature.feature:8
  Given the user has a buy list with item ID "1" and quantity 2 # BDD.UserStepDefinitions.givenTheUserHasABuyListWithItemIDAndQuantity(int, int)
  When the user removes 1 item with ID "1" from the buy list # BDD.UserStepDefinitions.whenTheUserRemovesItemFromTheBuyList(int)
  Then the new quantity of item ID "1" in the buy list should be 1 # BDD.UserStepDefinitions.thenTheNewQuantityInBuyListShouldBe(java.lang.String, int)

Scenario: User withdraws credit # src/test/java/feature.feature:13
  Given the user has a credit of 100 # BDD.UserStepDefinitions.givenTheUserHasACreditOf(float)
  When the user withdraws credit of 30 # BDD.UserStepDefinitions.whenTheUserWithdrawsCreditOf(float)
  Then the new credit should be 70 # BDD.UserStepDefinitions.thenTheNewCreditShouldBe(float)

```

Share your Cucumber Report with your team at <https://reports.cucumber.io>

Activate publishing with one of the following:

```

src/test/resources/cucumber.properties: cucumber.publish.enabled=true
src/test/resources/junit-platform.properties: cucumber.publish.enabled=true
Environment variable: CUCUMBER_PUBLISH_ENABLED=true
JUnit: @CucumberOptions(publish = true)

```

More information at <https://cucumber.io/docs/cucumber/environment-variables/>

Disable this message with one of the following:

```

src/test/resources/cucumber.properties: cucumber.publish.quiet=true
src/test/resources/junit-platform.properties: cucumber.publish.quiet=true

```

Process finished with swift code 0