

## BDD in action

Login is possible with a username/password combination. In the past there was no awareness of security issues, but since the topic hits the media regularly the stakeholder wants to make sure that passwords are secure. Now the PO formulates a story, which could look something like this:

### Story

The user selects a password, during registration, which is secure.

**As a system administrator**

**I want** the users to register with a secure password

**so that** the user accounts are not easily hackable.

The acceptance criteria to this story could look like that:

- The password must have a length of eight or more characters.
- The password must have at least one number.
- The password must have at least one special character.
- The user should receive an error message if the above criteria are not met.

With this formulated story the PO calls for a "three amigo" meeting. After the PO introduces the story, the following conversation could take place:

**Dev:** A password with less than eight characters should be denied, for example "Secret."

**PO:** Correct. My understanding is that an algorithm exists, which can crack short passwords in no time.

**Dev:** Would the password "Password1!" work?

**OPs:** In my opinion, it would be correctly implemented, but it is a weak password. Dictionary words with one number and one special character are covered in the most basic brute force attack algorithms.

**Dev:** Ok, if "Password1!" is a weak password because it is a dictionary word, what about "GuineapigMonkey."

**OPs:** I would categorize it as a medium strength password.

**PO:** The suggested password contains no special characters or numbers and is considered a medium secure password.

What would a strong password look like?

**OPs:** A password like "Gu1neapig\$Monk3y!" would be a strong one in my opinion.

What happened in that conversation? We used examples to provide a better understanding of the problem to solve as well as we created awareness of what it means to provide a secure password nowadays. In the meantime, the acceptance criteria of the story had been adapted:

1. Passwords should be categorized into three types: Weak, Medium and Strong.

2. Only passwords  $\geq$  medium should be accepted as valid passwords.
3. The user should receive an error-message if the password is categorized as weak.

During the meeting, the QA began to formulate a data table:

Password	Category	Acceptance
Secret	Weak	No
Secret1!	Weak	No
Password1!	Weak	No
aBcDeFg1	Weak	No
GuineapigMonkey	Medium	Yes
djEzDip9	Medium	Yes
GuineapigMonkeyElephant	Strong	Yes
Gu1neapig\$Monk3y!	Strong	Yes

as well as the feature-file that contains the scenarios:

```

1 Feature: Create secure password during registration procedure
2   In order to protect the user more effectively from security breaches I,
3     as system administrator
4       want the user to create a secure password during the registration procedure
5 Scenario: A user tries to register with a weak password
6   Given the user has the registration form opened
7   And chooses the password <password>
8   Then the password should be categorised <category>
9   And the user should get the information, that the selected password is too weak
10 Scenario: A user tries to register with a valid password
11   Given the user has the registration form opened
12   And chooses the password <password>
13   Then the password should be categorised <category>
14   And the user can proceed with the registration
15

```

The biggest advantage of using an example or behavior-based approach is to analyze the requirements is that we minimize the loss of details during the “translation” from one business language into another, or even misinterpret requirements completely and implementing them erroneously. Simultaneously, using BDD from the beginning acts as a base to create a transparent automated test suite, where we can speak of executable specification and living documentation.

### **Feature: Adding Multiple Products to Cart (DemoBlaze)**

#### **Scenario1: Add two different items to cart**

Given the user is on the DemoBlaze home page

When the user selects the "Phones" category

And the user opens "Samsung galaxy s6"

And the user adds the product to the cart

And the user navigates back to the home page

And the user selects the "Laptops" category

And the user opens "Sony vaio i5"

And the user adds the product to the cart

Then the cart should contain 2 products

#### **Scenario2: Add same product multiple times**

Given the user is on the DemoBlaze home page

When the user opens "Nexus 6" from the Phones category

And the user adds the product to the cart

And the user adds the same product again to the cart

Then the cart should show 2 items of "Nexus 6"

### **Scenario3: Add products from Phones, Laptops, and Monitors**

Given the user is on the DemoBlaze home page

When the user adds "Nokia lumia 1520" from Phones to the cart

And the user adds "MacBook Pro" from Laptops to the cart

And the user adds "ASUS Full HD" from Monitors to the cart

Then the cart should contain 3 items

And each item should belong to a different category

### **More scenarios (edge cases, invalid flows, missing alerts, etc.)**

### **Scenario 4: Validate total price after adding multiple products**

Scenario: Validate total cart amount after adding two products

Given the user is on the DemoBlaze home page

When the user adds "Samsung galaxy s7" to the cart

And the user adds "Sony vaio i7" to the cart

Then the cart total should be equal to the sum of both product prices

### **Scenario 5: Cancel add-to-cart alert and verify cart remains unchanged**

Scenario: Canceling the add-to-cart alert keeps cart unchanged

Given the user is on the DemoBlaze home page

When the user opens "Samsung galaxy s6"

And the user clicks "Add to cart"

And the browser shows a confirmation alert

And the user dismisses the alert

Then the cart should contain 0 items

### **Scenario 6: Validate category navigation before adding product**

Scenario: Add product only after correct category navigation

Given the user is on the DemoBlaze home page

When the user clicks on the "Laptops" category

And the products displayed belong to "Laptops"

And the user selects "MacBook air"

And the user adds the product to the cart

Then the cart should contain "MacBook air"

#### **Scenario 7: Add product after searching it manually (scroll + click)**

Scenario: Scroll and select product manually

Given the user is on the DemoBlaze home page

When the user scrolls to find "ASUS Full HD" under Monitors

And the user opens the product details page

And the user adds the product to the cart

Then the cart should contain 1 product

#### **Scenario 8: Add products quickly one after another**

Scenario: Add products in quick succession

Given the user is on the DemoBlaze home page

When the user adds "Iphone 6 32gb" to the cart

And without delay the user adds "MacBook Pro" to the cart

Then the cart should show both products

#### **Scenario 9: User tries to add product when site alert is slow**

Scenario: Handle delay in add-to-cart alert

Given the user is on the DemoBlaze home page

When the user opens "Nokia lumia 1520"

And the user clicks add to cart

And there is a delay before the alert appears

Then the user should wait for the alert

And the alert should confirm product added

And the cart count should increase by 1

#### **Scenario 10: Remove an item after adding multiple products**

Scenario: Remove item from cart after adding multiple products

Given the user has added "Samsung galaxy s6" and "Sony vaio i5" to the cart

When the user opens the cart  
And the user removes "Sony vaio i5"  
Then the cart should contain only "Samsung galaxy s6"

#### **Scenario 11: Validate product details before adding to cart**

Scenario: Validate product details before confirming add to cart  
Given the user is on the product page of "Samsung galaxy s7"  
When the user verifies the price, description, and product image  
And the user adds the product to the cart  
Then the cart should contain "Samsung galaxy s7"

#### **Scenario 12: Add product to cart after login**

Scenario: Add product after user login  
Given the user logs in with valid credentials  
When the user opens "Sony xperia z5"  
And the user adds the product to the cart  
Then the cart should contain "Sony xperia z5"  
And the product should be linked to the logged-in session

#### **Scenario 13: Add product to cart before login (guest mode)**

Scenario: Guest user adds product to cart  
Given the user is not logged in  
When the user adds "Nexus 6" to the cart  
Then the cart should contain "Nexus 6"  
And the user should not be prompted for login

#### **Scenario 14: Add product to cart after page refresh**

Scenario: Add product after refreshing product page  
Given the user is on the "MacBook Pro" product page  
When the user refreshes the page  
And the user clicks add to cart  
Then the cart should contain "MacBook Pro"

### Scenario 15: Validate cart persists across categories

Scenario: Cart retains added items while browsing categories

Given the user adds "Iphone 6 32gb" to the cart

When the user navigates to Laptops

And the user navigates to Monitors

Then the cart should still contain "Iphone 6 32gb"

### 1. Smoke Test Scenarios (@smoke)

Smoke tests validate the **basic functionality** of the application.

@smoke @cart @add

Scenario: Add a single product to the cart

Given the user is on the DemoBlaze home page

When the user adds "Samsung galaxy s6" to the cart

Then the cart should contain "Samsung galaxy s6"

@smoke @navigation

Scenario: Validate category navigation

Given the user is on the DemoBlaze home page

When the user clicks on the "Laptops" category

Then products displayed should belong to "Laptops"

---

### 2. Sanity Test Scenarios (@sanity)

Sanity tests validate **small changes** or "touchpoints" after minor updates.

@sanity @cart @alert

Scenario: Alert appears when adding a product to the cart

Given the user opens "Sony xperia z5"

When the user clicks add to cart

Then an alert should confirm product added

@sanity @ui

Scenario: Validate product details before adding to cart

Given the user is on the product page of "Samsung galaxy s7"

When the user verifies price and description

Then the details should be displayed correctly

---

### 3. Regression Test Scenarios (@regression)

Regression tests validate that existing features still work after new changes.

@regression @cart @multiple

Scenario: Add multiple products and validate total price

Given the user is on the DemoBlaze home page

When the user adds "Samsung galaxy s7" and "Sony vaio i7" to the cart

Then the cart total should equal the sum of both product prices

@regression @cart @remove

Scenario: Remove an item after adding multiple products

Given the user has added "Samsung galaxy s6" and "Sony vaio i5" to the cart

When the user opens the cart

And the user deletes "Sony vaio i5"

Then the cart should contain only "Samsung galaxy s6"

@regression @session

Scenario: Cart retains items while navigating across categories

Given the user adds "Iphone 6 32gb" to the cart

When the user browses laptops and monitors

Then the cart should still contain "Iphone 6 32gb"

---

### 4. Functional Test Scenarios (@functional)

Functional tests validate end-user functionality.

@functional @login @cart

Scenario: Add product to cart after login

Given the user logs in with valid credentials

When the user adds "Sony xperia z5" to the cart

Then the cart should contain "Sony xperia z5"

@functional @guest @cart

Scenario: Guest user adds product to cart without login

Given the user is not logged in

When the user adds "Nexus 6" to the cart

Then the cart should contain "Nexus 6"

@functional @refresh

Scenario: Add product after refreshing the product page

Given the user is on "MacBook Pro" page

When the user refreshes the page and adds the product

Then the cart should contain "MacBook Pro"

---

## 5. End-to-End Workflow Scenarios (@e2e)

E2E tests simulate a full workflow that a real customer would perform.

@e2e @purchase

Scenario: End-to-end purchase flow using cart

Given the user logs in with valid credentials

And the user adds "Samsung galaxy s6" to the cart

When the user proceeds to place order

Then the order confirmation should be displayed

@e2e @cart

Scenario: Add products, verify totals, and remove one item

Given the user logs in

When the user adds "Iphone 6 32gb" and "MacBook Pro"

And removes "Iphone 6 32gb"

Then the cart total should match the remaining product

---

## 6. Negative Test Scenarios (@negative)

Tests validations, missing alerts, invalid behavior.

@negative @alert

Scenario: No alert displayed after adding product

Given the user opens "Nokia lumia 1520"

When the user clicks add to cart

But the alert does not appear

Then the system should show an error in logs

@negative @broken @navigation

Scenario: Incorrect category items displayed

Given the user clicks on "Laptops"

When unrelated products appear

Then the system should mark this as a failure

@negative @session

Scenario: Cart becomes empty after page reload unexpectedly

Given the user adds "Samsung galaxy s7" to the cart

When the user refreshes the home page

Then the cart should still contain the product

---

### ★ Execution Examples

#### Run only Smoke Tests

```
mvn test -Dcucumber.filter.tags="@smoke"
```

#### Run Smoke + Sanity

```
mvn test -Dcucumber.filter.tags="@smoke or @sanity"
```

#### Run only Regression

```
mvn test -Dcucumber.filter.tags="@regression"
```