**1. The regression set of your application consists of 2000 tests. You only have time for 200 test cases. Describe your course of action to create a regression set of cases. Result: a document 1-2 pages with list of your actions**

It’s important to understand which part of the application is affected by the most recent changes, so that we can prioritize test effort. The chances are higher the bug could be introduced either in the module where the changes were made or depending on modules.

First, we need to break down the regression of 2000 test cases into test suites, where each test suite includes a set of test cases that cover separate pieces of functionality. Once this is done, we need to define the test suites that cover the functionality of the affected and dependent modules.

Second, we prioritize other test suites depending on the importance of the functionality they cover, so that we can understand where most of the testing effort has to be put.

Third, within each test suite we need to break down the test cases into positive and negative.

The priority in selection is given to positive test cases because our main goal is to ensure that application works as expected under normal circumstances. To ensure that most crucial negative scenarios are covered, it’s suggested to choose 1 negative test case for 5 positive test cases within the suite.

Fourth, we prioritize test cases within the test suite. Test cases with the highest priority are to be executed first, however we must not forget about dependencies. In case the test case with the higher priority depends on the test case with lower priority, then the test case with lower priority must be executed first.

During the selection of test cases for the regression, we need to take into account the test cases that failed during previous regression runs, because chances are high they may fail again.

Once we understand the importance of test suites, test cases within test suites and dependencies between test cases, it’s suggested that we select 30% of test cases from the suites that are directly or indirectly affected by the implemented changes, 20% from the suites that cover crucial part of the functionality but not directly affected by the changes, 10% of test cases from other test suites.

2. **Three main e2e scripts for “Rozetka” online store.**

1. - Customer opens “Rozetka.com.ua”

- Go to product catalog

- Chooses the product

- Adds the chosen product to the basket

- Opens Basket

- Enters personal information

- Makes a payment

Expected result: Payment is successfully completed

1. - Customer opens “Rozetka.com.ua”

- Logs in

- Go to product catalog

- Chooses the products

- Adds the chosen products to the wishlist

- Opens wishlist and check

Expected result: Chosen products added to the wishlist

1. - Customer opens “Rozetka.com.ua”

- Logs in

- Go to wish list

- Choose the product

- Adds the chosen product to the basket

- Opens basket

- Makes an order

Expected result: An order is successfully made

I consider that it is possible to cover these main scripts with automation tests.

**3. Test Cases based on e2e scripts for smoke testing**

**ID - TC01**

Title - Log in on “Rozetka.com.ua”

Priority - High

Test data -

Login - test@dot.com

Password - test1

Steps:

* Go to “Rozetka.com.ua”
* Click on the “User icon”
* Enter the field “login” valid data
* Enter the field “Password” valid data
* Click on the button “Log in”

Expected result: User is successfully logged in

**ID - TC02**

Title - Add chosen products to the basket

Priority - High

Test data:

Login - test@dot.com

Password - test1

Steps:

* Go to “Rozetka.com.ua”
* Log in
* Go to product catalog
* Choose some product
* Add chosen product to the basket
* Check the basket

Expected result: the chosen product was added to the basket

**ID - TC03**

Title - Make a payment for the chosen product

Priority - High

Test data:

Card number - 4242424242424242

CVV - 222

Exp.Date - 11/25

Pre-condition - user has already logged in

Steps:

* Go to “Rozetka.com.ua”
* Go to the basket with the chosen product
* Choose the payment method
* Make a payment with the test data

Expected result: Payment is successfully completed with the test data

**ID - TC04**

Title - Add chosen product to the wishlist

Priority - High

Pre-condition - user has already logged in

Steps:

* Go to “Rozetka.com.ua”
* Go to the product catalog
* Choose product
* Add chosen product to the wishlist
* Go to the wish list and check

Expected result: Chosen product is added to the wishlist

**ID - TC05**

Title - Make a payment for the product from the wishlist

Priority - High

Test data:

Card number - 4242424242424242

CVV - 222

Exp.Date - 11/25

Pre-condition - user has already logged in

Steps:

* Go to “Rozetka.com.ua”
* Go to the wishlist
* Click on the chosen product
* Click on the button “Buy”
* Choose the payment method
* Make a payment with the test data

Expected result: Chosen product from the wish list is successfully ordered and paid