

**❑**

**KOLEJ PROFESIONAL MARA BERANANG**

**DIPLOMA IN COMPUTER SCIENCE**

|  |  |  |
| --- | --- | --- |
| **COURSE NAME** | : | SOFTWARE APPLICATION TESTING |
| **COURSE CODE** | : | CSC 1273 |
| **ACADEMIC SESSION** | : | 3 2023/2024 |
| **TYPE OF ASSESSMENT** | : | FINAL ASSIGNMENT |
| **DURATION** | : | 13 FEBRUARY 2024 – 05 MARCH 2024 |

**CLO3: Propose appropriate testing tools to conduct software testing. (C3, PLO6)**

**INSTRUCTION TO CANDIDATES:**

1. Late submissions after given due date will not be accepted.

2. Report should be written in using:

a. Font type: Arial

b. Size :11 pts

c. Line Spacing : 1.5

|  |  |
| --- | --- |
| **Personal Details** | |
| **Name** | Muhammad Eqmal Bin Noor Raniz |
| **I/D Number** | BCS2211-026 |
| **Class** | DCS 4B |
| **Lecturer** | Puan Shahellin Binti Huang |

|  |  |
| --- | --- |
| **Section / Question No.** | **Marks** |
| Task 1 | /8 |
| Task 2 | /42 |
| **Total** | **/50** |

**Scenario:**

Automation testing is a software testing technique in which software tools and scripts are used to perform the testing of software applications. In automation testing, specialized software tools or scripts are created to perform test cases, compare actual outcomes with expected outcomes, report test results, and generate detailed test reports. Katalon Recorder is one of the specialized tools that can simulate user interactions, such as clicking buttons, entering data, and navigating through screens to automate testing on the browser. In order to organize and manage automated test cases seamlessly, TestLink serves as a robust solution for managing and organizing automated test cases effectively.

**Task 1**

Create a comprehensive suite of automated test cases using **Katalon Recorder** on selected social media that you previously tested manually. Conduct automated test cases using Katalon Recorder for essential functionalities:

1. **Login** test case
2. **Search** test case
3. **Post** test case
4. **Profile** test case

The following are the steps on how to use Katalon Recorder:

1. Install the Katalon Recorder plugin using your preferred web browser such as Google Chrome or Mozilla Firefox.
2. Start recording your actions by performing the actions in the browser to execute the test steps.
3. After the recording is done, you can edit or add assertions to enhance the script.
4. Execute the automated test cases and ensure that the automated tests produce the test results either pass or fail.
5. Capture the test results and any defects encountered during the test execution.
6. Provide screenshots of the test results, including pass or fail status for each test case.

**Task 2**

Produce test plan using **TestLink** management tool for:

1. **Unit** Testing (Positive test case)
2. **Integration** Testing (Positive test case)
3. **System** Testing (Negative test case)
4. **Acceptance** Testing (Negative test case)

Each positive/negative test case must includes the following details:

1. Test case title
2. Test case summary
3. Preconditions
4. Step actions
5. Expected results
6. Execution notes
7. Execution status
8. Screenshots of actual results

**Assessment Rubrics:**

| **Task 1** | | **Attainment Scale** | | | | **Marks Weight** | **Student Attainments** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1**  **Very Weak** | **2**  **Weak** | **3**  **Fair** | **4**  **Good** | **Scale Marks** |
| Conduct automated test cases using Katalon Recorder for essential functionalities:   1. **Login** test case 2. **Search** test case 3. **Post** test case 4. **Profile** test case | 1. **Login** test case | Test case fulfills any **ONE** of the requirements:   * Test case execution is successful automated. * Include all necessary command, target, and value. * Test result is accurate, including pass or fail status. * Relevant screenshots or logs provided. | Test case fulfills any **TWO** of the requirements:   * Test case execution is successful automated. * Include all necessary command, target, and value. * Test result is accurate, including pass or fail status. * Relevant screenshots or logs provided. | Test case fulfills any **THREE** of the requirements:   * Test case execution is successful automated. * Include all necessary command, target, and value. * Test result is accurate, including pass or fail status. * Relevant screenshots or logs provided. | Test case fulfills any **ALL** of the requirements:   * Test case execution is successful automated. * Include all necessary command, target, and value. * Test result is accurate, including pass or fail status. * Relevant screenshots or logs provided. | 0.5 |  |
| 1. **Search** test case | Test case fulfills any **ONE** of the requirements:   * Test case execution is successful automated. * Include all necessary command, target, and value. * Test result is accurate, including pass or fail status. * Relevant screenshots or logs provided. | Test case fulfills any **TWO** of the requirements:   * Test case execution is successful automated. * Include all necessary command, target, and value. * Test result is accurate, including pass or fail status. * Relevant screenshots or logs provided. | Test case fulfills any **THREE** of the requirements:   * Test case execution is successful automated. * Include all necessary command, target, and value. * Test result is accurate, including pass or fail status. * Relevant screenshots or logs provided. | Test case fulfills any **ALL** of the requirements:   * Test case execution is successful automated. * Include all necessary command, target, and value. * Test result is accurate, including pass or fail status. * Relevant screenshots or logs provided. | 0.5 |  |
| 1. **Post** test case | Test case fulfills any **ONE** of the requirements:   * Test case execution is successful automated. * Include all necessary command, target, and value. * Test result is accurate, including pass or fail status. * Relevant screenshots or logs provided. | Test case fulfills any **TWO** of the requirements:   * Test case execution is successful automated. * Include all necessary command, target, and value. * Test result is accurate, including pass or fail status. * Relevant screenshots or logs provided. | Test case fulfills any **THREE** of the requirements:   * Test case execution is successful automated. * Include all necessary command, target, and value. * Test result is accurate, including pass or fail status. * Relevant screenshots or logs provided. | Test case fulfills any **ALL** of the requirements:   * Test case execution is successful automated. * Include all necessary command, target, and value. * Test result is accurate, including pass or fail status. * Relevant screenshots or logs provided. | 0.5 |  |
| 1. **Profile** test case | Test case fulfills any **ONE** of the requirements:   * Test case execution is successful automated. * Include all necessary command, target, and value. * Test result is accurate, including pass or fail status. * Relevant screenshots or logs provided. | Test case fulfills any **TWO** of the requirements:   * Test case execution is successful automated. * Include all necessary command, target, and value. * Test result is accurate, including pass or fail status. * Relevant screenshots or logs provided. | Test case fulfills any **THREE** of the requirements:   * Test case execution is successful automated. * Include all necessary command, target, and value. * Test result is accurate, including pass or fail status. * Relevant screenshots or logs provided. | Test case fulfills any **ALL** of the requirements:   * Test case execution is successful automated. * Include all necessary command, target, and value. * Test result is accurate, including pass or fail status. * Relevant screenshots or logs provided. | 0.5 |  |
| **Total Marks Earned** | | | | | | | **/8** |

| **Task 2** | | **Attainment Scale** | | | | **Marks Weight** | **Student Attainments** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1**  **Very Weak** | **2**  **Weak** | **3**  **Fair** | **4**  **Good** | **Scale Marks** |
| Produce test plan using TestLink management tool for:   1. **Unit** Testing (Positive test case) | 1. Test Case Title and Test Case Summary | Provide simple test case title.  **OR**  Provide simple test case summary. | Provide brief description of the test case title and summary. | Provide detail description of the test case title and summary. | Provide specific detail description of test case title and summary with clear, well-defined and understandable. | 0.5 |  |
| 1. Preconditions | The preconditions are incomplete and missing important details for the test case execution. | The preconditions are partially complete but missing important details for the test case execution. | The preconditions are mostly complete but may lack some important details for the test case execution. | The preconditions are clear and comprehensive, covering all important details for the successful execution of the test case. | 0.5 |  |
| 1. Step Actions | The test step actions are incomplete and missing important details for the test case execution. | The test step actions are partially complete but missing important details for the test case execution. | The test step actions are mostly complete but may lack some important details for the test case execution. | The test step actions are clear and comprehensive, covering all important details for the test case execution. | 0.5 |  |
| 1. Expected Results | The expected results incomplete and missing necessary details for the expected outcome of the test step. | The expected results are partially complete but missing necessary details for the expected outcome of the test step. | The expected results are mostly complete but may lack some necessary details for the expected outcome of the test step. | The expected results are clear and comprehensive, covering all necessary details for the expected outcome of the test step. | 0.5 |  |
| v. Execution Notes and Execution Status | The execution notes are incomplete and the execution status is inaccurate. | The execution notes are partially complete but the execution status is inaccurate.  **OR**  The execution notes are partially incomplete but the execution status is accurate. | The execution notes are mostly complete and the execution status is accurate. | The execution notes are clear and complete and the execution status accurately reflects the outcome of the test execution. | 0.5 |  |
| 1. **Integration** Testing   (Positive test case) | 1. Test Case Title and Test Case Summary | Provide simple test case title.  **OR**  Provide simple test case summary. | Provide brief description of the test case title and summary. | Provide detail description of the test case title and summary. | Provide specific detail description of test case title and summary with clear, well-defined and understandable. | 0.5 |  |
| 1. Preconditions | The preconditions are incomplete and missing important details for the test case execution. | The preconditions are partially complete but missing important details for the test case execution. | The preconditions are mostly complete but may lack some important details for the test case execution. | The preconditions are clear and comprehensive, covering all important details for the successful execution of the test case. | 0.5 |  |
| 1. Step Actions | The test step actions are incomplete and missing important details for the test case execution. | The test step actions are partially complete but missing important details for the test case execution. | The test step actions are mostly complete but may lack some important details for the test case execution. | The test step actions are clear and comprehensive, covering all important details for the test case execution. | 0.5 |  |
| 1. Expected Results | The expected results incomplete and missing necessary details for the expected outcome of the test step. | The expected results are partially complete but missing necessary details for the expected outcome of the test step. | The expected results are mostly complete but may lack some necessary details for the expected outcome of the test step. | The expected results are clear and comprehensive, covering all necessary details for the expected outcome of the test step. | 0.5 |  |
| v. Execution Notes and Execution Status | The execution notes are incomplete and the execution status is inaccurate. | The execution notes are partially complete but the execution status is inaccurate.  **OR**  The execution notes are partially incomplete but the execution status is accurate. | The execution notes are mostly complete and the execution status is accurate. | The execution notes are clear and complete and the execution status accurately reflects the outcome of the test execution. | 0.5 |  |
| 1. **System** Testing   (Negative test case) | 1. Test Case Title and Test Case Summary | Provide simple test case title.  **OR**  Provide simple test case summary. | Provide brief description of the test case title and summary. | Provide detail description of the test case title and summary. | Provide specific detail description of test case title and summary with clear, well-defined and understandable. | 0.5 |  |
| 1. Preconditions | The preconditions are incomplete and missing important details for the test case execution. | The preconditions are partially complete but missing important details for the test case execution. | The preconditions are mostly complete but may lack some important details for the test case execution. | The preconditions are clear and comprehensive, covering all important details for the successful execution of the test case. | 0.5 |  |
| 1. Step Actions | The test step actions are incomplete and missing important details for the test case execution. | The test step actions are partially complete but missing important details for the test case execution. | The test step actions are mostly complete but may lack some important details for the test case execution. | The test step actions are clear and comprehensive, covering all important details for the test case execution. | 0.5 |  |
| 1. Expected Results | The expected results incomplete and missing necessary details for the expected outcome of the test step. | The expected results are partially complete but missing necessary details for the expected outcome of the test step. | The expected results are mostly complete but may lack some necessary details for the expected outcome of the test step. | The expected results are clear and comprehensive, covering all necessary details for the expected outcome of the test step. | 0.5 |  |
| v. Execution Notes and Execution Status | The execution notes are incomplete and the execution status is inaccurate. | The execution notes are partially complete but the execution status is inaccurate.  **OR**  The execution notes are partially incomplete but the execution status is accurate. | The execution notes are mostly complete and the execution status is accurate. | The execution notes are clear and complete and the execution status accurately reflects the outcome of the test execution. | 0.5 |  |
| 1. **Acceptance** Testing   (Negative test case) | 1. Test Case Title and Test Case Summary | Provide simple test case title.  **OR**  Provide simple test case summary. | Provide brief description of the test case title and summary. | Provide detail description of the test case title and summary. | Provide specific detail description of test case title and summary with clear, well-defined and understandable. | 0.5 |  |
| 1. Preconditions | The preconditions are incomplete and missing important details for the test case execution. | The preconditions are partially complete but missing important details for the test case execution. | The preconditions are mostly complete but may lack some important details for the test case execution. | The preconditions are clear and comprehensive, covering all important details for the successful execution of the test case. | 0.5 |  |
| 1. Step Actions | The test step actions are incomplete and missing important details for the test case execution. | The test step actions are partially complete but missing important details for the test case execution. | The test step actions are mostly complete but may lack some important details for the test case execution. | The test step actions are clear and comprehensive, covering all important details for the test case execution. | 0.5 |  |
| 1. Expected Results | The expected results incomplete and missing necessary details for the expected outcome of the test step. | The expected results are partially complete but missing necessary details for the expected outcome of the test step. | The expected results are mostly complete but may lack some necessary details for the expected outcome of the test step. | The expected results are clear and comprehensive, covering all necessary details for the expected outcome of the test step. | 0.5 |  |
| v. Execution Notes and Execution Status | The execution notes are incomplete and the execution status is inaccurate. | The execution notes are partially complete but the execution status is inaccurate.  **OR**  The execution notes are partially incomplete but the execution status is accurate. | The execution notes are mostly complete and the execution status is accurate. | The execution notes are clear and complete and the execution status accurately reflects the outcome of the test execution. | 0.5 |  |
| Attach appropriate screenshots of the actual results for each test case conducted. | 1. Unit Testing 2. Integration Testing 3. System Testing 4. Acceptance Testing | Provide screenshots of the actual result for **ONE** of the following testing:   * Unit Testing * Integration Testing * System Testing * Acceptance Testing | Provide screenshots of the actual result for **TWO** of the following testing:   * Unit Testing * Integration Testing * System Testing * Acceptance Testing | Provide screenshots of the actual result for **THREE** of the following testing:   * Unit Testing * Integration Testing * System Testing * Acceptance Testing | Provide screenshots of the actual result for **ALL** of the following testing:   * Unit Testing * Integration Testing * System Testing * Acceptance Testing | 0.5 |  |
| **Total Marks Earned** | | | | | | | **/42** |
| **Grand Total** | | | | | | | **/50** |
| **Total Percentage (30%)** | | | | | | |  |

**Task 1:** Automation Testing (Katalon Recorder)

Login Test Case

A screenshot of a computer

Description automatically generated

Steps:

1) User Open Twitter Website Url: <https://twitter.com/?lang=en>\

2) User Then Click On Login Button.

3) User Will Enter Their Registered Name

4) User Will Enter the Registered Password.

5) User Will Click on Sign In Button

6) The System Will Display Home Page

Search Test Case

A screenshot of a computer

Description automatically generated

Steps:

1) User Open Twitter Website Url: https://twitter.com/?lang=en

2) User Then Click On Login Button.

3) User Will Enter Their Registered Name

4) User Will Enter the Registered Password.

5) User Will Click on Sign In Button

6) The System Will Display Home Page

7) User Then Click On Search Button At The Homepage.

8) User Then Will Go To The Search Page.

9) User Then Will Enter Their Desired Content Or Text That They Want To Search At The Search Page.

10) User Then Will Click On Search Button At The Search Page.

11) User Then Will Be Able To See Their Desired Content That They Search Earlier Correctly.

Post Test Case

A screenshot of a computer

Description automatically generated

Steps:

1) User Open Twitter Website Url: <https://twitter.com/?lang=en>

2) User ThenClick On Login Button.

3) User Will Enter Their Registered Name

4) User Will Enter the Registered Password.

5) User Will Click on Sign In Button

6) The System Will Display Home Page

7) User Then Click On Post Button At The Homepage.

8) User Then Will Go To The Post Page

9) User Then Will Enter Their Desired Text Or Image At The Post Page.

10) User Then Will Click On Post Button At The Page.

11) User Then Will Be Directed At The Homepage And Will See Their Posted Content.

Edit Profile Test Case

A screenshot of a computer

Description automatically generated

Steps:

1) User Open Twitter Website Url: <https://twitter.com/?lang=en>

2) User ThenClick On Login Button.

3) User Will Enter Their Registered Name

4) User Will Enter the Registered Password.

5) User Will Click on Sign In Button

6) The System Will Display Home Page

7) User Click On Profile Button At The Homepage Of The Twitter

8) User Will Go To The Profile Page And Click On Edit Profile Button.

9) User Will Go To The Edit Profile Page And will Edit Their Profile Following Their Preference.

10) After User Satisfied With Their Edit User Will Click On Save Button At The Edit Profile Page.

11) The System Will Display The Edited Profile Page Of The User.

**Task 2:** Produce test plan using TestLink

Intergration Test: Bookmark Functionality (Positive Test Case)

TestLink Community [configure $tlCfg->document\_generator->company\_name]

Test Plan Execution Report

Test Project: Twitter Test Project  
Test Plan: Twitter Integration Test Plan  
Test Suite: Test Suite For Intergration Test Plan For Twitter

Printed by TestLink on 03/03/2024

2012 © TestLink Community

# Table Of Contents

[**1.1.Test Suite For Intergration Test Plan For Twitter**](#toc_1_1)

[TWTR-8: Bookmark Functionality (Positive Test Case)](#toc_tc26)

# Test Project: Twitter Test Project

The project is to create a test plan for unit, system, integration,  and acceptance testing in order to develop a responsive and engaging Twitter app and website.

# Test Plan: Twitter Integration Test Plan

This is an integration test plan for Twitter that consists of one test case, which is Bookmark functionality (a positive test case) to test whether the user could Bookmark their desired content or picture at the Homepage Of The Twitter.

# 1.1.Test Suite : Test Suite For Intergration Test Plan For Twitter

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case TWTR-8: Bookmark Functionality (Positive Test Case) [Version : 1]** | | | | | |
| Author: | eggmoll | | | | |
| Summary:  This test case ensures the bookmark functionality on Twitter works as expected. It starts by verifying the login process to ensure the tester is logged in. Then, it checks the visibility and functionality of the bookmark button on both the homepage and the "More" page. The tester confirms that the bookmark button is easily visible and functions properly, including checking for visual cues like color changes upon clicking. After bookmarking, the test moves to confirm that the saved content is displayed correctly on the bookmark page, ensuring users can access their saved content later without issues. Optionally, the test may also include checking compatibility across devices and browsers. Additionally, negative testing scenarios may be included, such as trying to bookmark unsupported content types or attempting to bookmark while offline, to ensure appropriate error messages are displayed. | | | | | |
| Preconditions:   |  | | --- | | 1) Verify User Is Able to Login To Twitter Website Url:  <https://twitter.com/?lang=en>  2) Verify User Is Able to Click on Sign in Button.  3) Verify User Is Able To Enter Registered Email And Correct Password At Login Page.  4) Verify User Is Able to Click Sign in Button.  5) Verify User Is Able to Go To The Homepage After Successfully Login.  6) Verify User Is Able To See Image Or Text At The Homepage.  7) Verify User Is Able To See Bookmark Button At The Bottom Of The Picture Or Image.  8) Verify User Is Able To Click On Bookmark Button At The Picture Or Text.  9) Verify User Is Able To Click On More Button At The Homepage.  10) Verify User Is Able To See Bookmark Button At The More Page.  11) Verify User Is Able To Click On Bookmark Button At More Page.  12) Verify User Is Able To Go To The Bookmark Page.  13) Verify User Is Able The Bookmarked Picture Or Image That They Bookmark Earlier At The Bookmark Page. | | | | | | |
| #: | Step actions: | Expected Results: | Execution notes: | Execution Status: |  |
| 1 | 1) User Open Twitter Website Url: https://twitter.com/?lang=en  2) User Then Click On Login Button.  3) User Will Enter Their Registered Name  4) User Will Enter the Registered Password.  5) User Will Click on Sign In Button  6) The System Will Display Home Page  7) User Then Will See Picture Or Text At The Homepage .  8) User Then Will See Bookmark Button At The Picture Or Homepage And Will Click On Bookmark Button.  9) User Then Will Click On More Button At The Homepage.  10) User Then Will Click Bookmark Button At The More Page.  11) User Then Will Be Able To See Their Bookmarked Image Or Text Earlier At The Bookmark Page. | |  | | --- | | 1) Verify User Is Able to Login To Twitter Website Url:  <https://twitter.com/?lang=en>  (step 1)  2) Verify User Is Able to Click on Sign in Button.  (step 2)  3) Verify User Is Able To Enter Registered Email And Correct Password At Login Page.  (step 3)  4) Verify User Is Able to Click Sign in Button.  (step 4)  5) Verify User Is Able to Go To The Homepage After Successfully Login.  (step 5)  6) Verify User Is Able To See Image Or Text At The Homepage.  (step 6)  7) Verify User Is Able To See Bookmark Button At The Bottom Of The Picture Or Image.  (step 7)  8) Verify User Is Able To Click On Bookmark Button At The Picture Or Text.  (step 8)  9) Verify User Is Able To Click On More Button At The Homepage.  (step 9)  10) Verify User Is Able To See Bookmark Button At The More Page.  (step 10)  11) Verify User Is Able To Click On Bookmark Button At More Page.  (step 11)  12) Verify User Is Able To Go To The Bookmark Page.  (step 12)  13) Verify User Is Able The Bookmarked Picture Or Image That They Bookmark Earlier At The Bookmark Page  (step 13) | | https://twitter.com/PunchingCat/status/1764116551731699916 | Passed |  |
| Execution type: | Manual | | | | |
| Estimated exec. duration (min): |  | | | | |
| Priority: | Medium | | | | |
|  | | | | | |
| Requirements | None | | | | |
| Keywords: | None | | | | |
| **Execution Details** |  | | | | |
| Build | Twitter Build For Intergration Test Plan | | | | |
| Tester | eggmoll | | | | |
| Execution Result: | **Passed** | | | | |
| Execution Mode: | **Manual** | | | | |
| Execution duration (min): |  | | | | |

**Actual Result**

Before Bookmark Picture Or Text That Desired:

A screenshot of a video game

Description automatically generated

Bookmarked Picture Or Text

A screenshot of a video chat

Description automatically generated

At Boomark Page:

A screenshot of a video game

Description automatically generated

Unit Testing: Tweeting Functionality (Positive Test Case)

TestLink Community [configure $tlCfg->document\_generator->company\_name]

Test Plan Execution Report

Test Project: Twitter Test Project  
Test Plan: Twitter Unit Test Plan

Printed by TestLink on 03/03/2024

2012 © TestLink Community

# Table Of Contents

[**1.1.Test Suite For Unit Testing For Twitter**](#toc_1_1)

[TWTR-5: Tweeting Functionality (Positive Test Case)](#toc_tc19)

# Test Project: Twitter Test Project

The project is to create a test plan for unit, system, integration,  and acceptance testing in order to develop a responsive and engaging Twitter app and website.

# Test Plan: Twitter Unit Test Plan

This Is Unit Test Plan For Twitter That Consists Of 1 Test Case Which Are Tweeting Functionality (positive test case) To Test User Could Create Tweets Or Post Message And Picture At Twitter Properly.

# 1.1.Test Suite : Test Suite For Unit Testing For Twitter

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case TWTR-5: Tweeting Functionality (Positive Test Case) [Version : 1]** | | | | | |
| Author: | eggmoll | | | | |
| Summary:  This test case rigorously validates the user's ability to log in, navigate the homepage, create a post with text and images, and ensures visibility on both the homepage and the user's profile, across multiple devices and screen sizes, including checking image integrity, link functionality, and platform performance under load. | | | | | |
| Preconditions:   |  | | --- | | 1) Verify User Is Able to Login To Twitter Website Url:  <https://twitter.com/?lang=en>  2) Verify User Is Able to Click on Sign in Button.  3) Verify User Is Able To Enter Registered Email And Correct Password In Log In Page.  4) Verify User Is Able to Click Sign in Button.  5) Verify User Is Able to Go To The Homepage After Successfully Login.  6) Verify User Is Able To Click On Post Button At The Homepage.  7) Verify User Is Able To Go To The Post Page And Enter Their Desired Text And Image At The Post Page.  8) Verify User Is Able To Click On Post Button On the Post Page And Able To Go To The Homepage And See Their Posted Content. | | | | | | |
| #: | Step actions: | Expected Results: | Execution notes: | Execution Status: |  |
| 1 | 1) User Open Twitter Website Url: <https://twitter.com/?lang=en>  2) User ThenClick On Login Button.  3) User Will Enter Their Registered Name  4) User Will Enter the Registered Password.  5) User Will Click on Sign In Button  6) The System Will Display Home Page  7) User Then Click On Post Button At The Homepage.  8) User Then Will Go To The Post Page  9) User Then Will Enter Their Desired Text Or Image At The Post Page.  10) User Then Will Click On Post Button At The Page.  11) User Then Will Be Directed At The Homepage And Will See Their Posted Content. | |  | | --- | | 1) Validate User Is Able to Login To Twitter Website Url:  <https://twitter.com/?lang=en>  (step 1)  2) Validate User Is Able to Click on Sign in Button.  (step 2)  3) In Sign in Page Validate User Is Able To Enter Registered Email And Correct Password.  (step 3 to 4)  4) Validate User Is Able to Click Sign in Button.  (step 5)  5) Validate User Is Able to Go To The Homepage After Successfully Login.  (step 6)  6) Validate User Is Able To Click On Post Button At The Homepage.  (step 7)  7) Validate User Is Able To Go To The Post Page And Enter Their Desired Text And Image At The Post Page.  (step 8 and 9)  8) Validate User Is Able To Click On Post Button On the Post Page And Able To Go To The Homepage And See Their Posted Content.  (step 10 and 11) | | Caption: Test | Passed |  |
| Execution type: | Manual | | | | |
| Estimated exec. duration (min): |  | | | | |
| Priority: | Medium | | | | |
|  | | | | | |
| Requirements | None | | | | |
| Keywords: | None | | | | |
| **Execution Details** |  | | | | |
| Build | Twitter Build For Unit Test Plan | | | | |
| Tester | eggmoll | | | | |
| Execution Result: | **Passed** | | | | |
| Execution Mode: | **Manual** | | | | |
| Execution duration (min): |  | | | | |

Actual Result:

Homepage And User Click On Post Button

A screenshot of a video chat

Description automatically generated

Post Page

A screenshot of a computer

Description automatically generated

Homepage After User Post Picture Or Text

A screenshot of a computer

Description automatically generated

Acceptance Test: Changes Language Functionality (Negative Test Case)

TestLink Community [configure $tlCfg->document\_generator->company\_name]

Test Plan Execution Report

Test Project: Twitter Test Project  
Test Plan: Twitter Acceptance Test Plan  
Test Suite: Test Suite For Acceptance Test Plan For Twitter

Printed by TestLink on 03/03/2024

2012 © TestLink Community

# Table Of Contents

[**1.1.Test Suite For Acceptance Test Plan For Twitter**](#toc_1_1)

[TWTR-7: Changes Language Functionality (Negative Test Case)](#toc_tc23)

# Test Project: Twitter Test Project

The project is to create a test plan for unit, system, integration,  and acceptance testing in order to develop a responsive and engaging Twitter app and website.

# Test Plan: Twitter Acceptance Test Plan

This is an acceptance test plan for Twitter that consists of 1 test case that changes language functionality (a negative test case). The test user could choose the language they want to change while not connected to the internet, but the Twitter web did not change the language that the user chose.

# 1.1.Test Suite : Test Suite For Acceptance Test Plan For Twitter

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case TWTR-7: Changes Language Functionality (Negative Test Case) [Version : 1]** | | | | | |
| Author: | eggmoll | | | | |
| Summary:  This test case aims to verify the functionality of changing the language setting on Twitter. It covers both successful and negative scenarios, particularly focusing on the inability to change the language when there is no internet connection. | | | | | |
| Preconditions:   |  | | --- | | 1) Verify User Is Able to Login To Twitter Website Url:  <https://twitter.com/?lang=en>  2) Verify User Is Able to Click on Sign in Button.  3) Verify User Is Able To Enter Registered Email And Correct Password In Log In Page.  4) Verify User Is Able to Click Sign in Button.  5) Verify User Is Able to Go To The Homepage After Successfully Login.  6) Verify User Is Able To Click On More Button At The Homepage.  7) Verify User Is Able To Go To The More Page And Click On Settings Button.  8) Verify User Is Able To Go To The Settings Page And Click On Language Button.  9) Verify User Is Able To see The Desired Language And Click On The Language.  10) Verify User Is Not Able To Change The Language Due To Not Having Internet.  11) Verify User Is Able To See Pop Up No Internet At Twitter When The Connection Is Gone.  12) Verify user Is Able To  Reconnect To The Internet.  13) Verify User Is Able To Go To The Homepage After The Internet Back On And Not Have The Language Change That Made Earlier Due To Not Having Internet. | | | | | | |
| #: | Step actions: | Expected Results: | Execution notes: | Execution Status: |  |
| 1 | 1) User Open Twitter Website Url: https://twitter.com/?lang=en  2) User Then Click On Login Button.  3) User Will Enter Their Registered Name  4) User Will Enter the Registered Password.  5) User Will Click on Sign In Button  6) The System Will Display Home Page  7) User Then Click On More Button At The Homepage.  8) User Then Will Click On Settings At More Page  9) User Then Will Go To The Settings Page.  10) User Then Will Click Language Button At The Settings Page.  11) User Then Will Click On Desired Language Button And Press Apply Button Without Internet.  12) User Then Will Check On The Internet When User Receive No Internet Pop Up That Been Made From Twitter.  13) User Then Will Reconnect The Internet At Their Device To Execute The Language Change.  14) User Then Will Go To The Homepage After The Internet Is Back On And The Settings That The User Made Earlier Will Not Apply Due Not Having Internet | |  | | --- | | 1) Validate User Is Able to Login To Twitter Website Url:  <https://twitter.com/?lang=en>  (step 1)  2) Validate User Is Able to Click on Sign in Button.  (step 2)  3) In Sign in Page Validate User Is Able To Enter Registered Email And Correct Password.  (step 3 to 4)  4) Validate User Is Able to Click Sign in Button.  (step 5)  5) Validate User Is Able to Go To The Homepage After Successfully Login.  (step 6)  6) Validate User Is Able To Click On More Button At The Homepage.  (step 7)  7) Validate User Is Able To Go To The More Page And Click On Settings Button.  (step 8 and 9)  8) Validate User Is Able To Go To The Settings Page And Click On Language Button.  (step 10 and 11)  9) Validate User Is Able To see The Desired Language And Click On The Language.  (step 12)  10) Validate User Is Not Able To Change The Language Due To Not Having Internet.  (step 13)  11) Validate User Is Able To See Pop Up No Internet At Twitter When The Connection Is Gone.  (step 14)  12) Validate user Is Able To  Reconnect To The Internet.  (step 15)  13) Validate User Is Able To Go To The Homepage After The Internet Back On And Not Have The Language Change That Made Earlier Due To Not Having Internet.  (step 16) | | Caption: While switching languages, the Internet will be shut off. | Passed |  |
| Execution type: | Manual | | | | |
| Estimated exec. duration (min): |  | | | | |
| Priority: | Medium | | | | |
|  | | | | | |
| Requirements | None | | | | |
| Keywords: | None | | | | |
| **Execution Details** |  | | | | |
| Build | Twitter Build For Acceptance Test Plan | | | | |
| Tester | eggmoll | | | | |
| Execution Result: | **Passed** | | | | |
| Execution Mode: | **Manual** | | | | |
| Execution duration (min): |  | | | | |

Actual Result:

User Will Click on More Button At Homepage

A screenshot of a video chat

Description automatically generated

User Will Click On Settings

**A screenshot of a video chat

Description automatically generated**

User Will Click On Language And Change It While Not Connected To The Internet And Pop Up Will Come Out

A screenshot of a computer

Description automatically generated



User Will Go To The Homepage With Language Not Change Due To Not Having Internet Connection

A screenshot of a video chat

Description automatically generated

System Test: Messaging Functionality (Negative Test Case)

TestLink Community [configure $tlCfg->document\_generator->company\_name]

Test Plan Execution Report

Test Project: Twitter Test Project  
Test Plan: Twitter System Test Plan  
Test Suite: Test Suite For System Test Plan For Twitter

Printed by TestLink on 03/04/2024

2012 © TestLink Community

# Table Of Contents

[**1.1.Test Suite For System Test Plan For Twitter**](#toc_1_1)

[TWTR-9: Messaging Functionality (Negative Test Case)](#toc_tc28)

# Test Project: Twitter Test Project

The project is to create a test plan for unit, system, integration,  and acceptance testing in order to develop a responsive and engaging Twitter app and website.

# Test Plan: Twitter System Test Plan

This Is A System Test Plan For Twitter That Consists Of 1 Test Case Which Are Messaging Functionality (Negative Test Case) To Test Message Is Able To Type And Send Message While Not Connected To The Internet To The Wanted Person In Twitter.

# 1.1.Test Suite : Test Suite For System Test Plan For Twitter

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case TWTR-9: Messaging Functionality (Negative Test Case) [Version : 1]** | | | | | |
| Author: | eggmoll | | | | |
| Summary:  This negative test case simulates a situation where a user tries to send a message that is longer than the character limit in order to test Twitter's messaging feature. It is anticipated that Twitter will stop the message from being sent and give the user the relevant feedback. | | | | | |
| Preconditions:   |  | | --- | | 1) Verify User Is Able to Login To Twitter Website Url:  <https://twitter.com/?lang=en>  2) Verify User Is Able to Click on Sign in Button.  3) Verify User Is Able To Enter Registered Email And Correct Password In Login Page.  4) Verify User Is Able to Click Sign in Button.  5) Verify User Is Able to Go To The Homepage After Successfully Login.  6) Verify User Is Able To Click On Message Button At The Homepage.  7) Verify User Is Able To Go To The Message Page.  8) Verify User Is Able To Click On User Or People That They Want To Text.  9) Verify User Is Able To Enter Text At The Text Box At The Message Page.  10) Verify User Is Able To Click On Send Button At The Message Page.  11) Verify User Is Able To Receive Pop Up Saying Text Is Exceeded The Limit.  12) Verify User Is Not Able To Send The Text That Made Earlier At The Message Page.  13) Verify User Is Able To Go Back To The Text Box At The Message Page Where The Text Made Earlier. | | | | | | |
| #: | Step actions: | Expected Results: | Execution notes: | Execution Status: |  |
| 1 | 1) User Open Twitter Website Url: https://twitter.com/?lang=en  2) User Then Click On Login Button.  3) User Will Enter Their Registered Name  4) User Will Enter the Registered Password.  5) User Will Click on Sign In Button  6) The System Will Display Home Page  7) User Then Click On Message Button At The Homepage.  8) User Then Will Click On User Or People That They Want To Message At The Message Page.  9) User Then Will Type The Desired Text At The Text Box Of The People Or User That They Desired.  10) User Then Will Click Send Button At The Message Page To Send The Text.  11) User Then Will Receive An Pop Out Saying That Text Cannot Been Send Due To Text Exceeding The Limit.  12) User Then Will Go Back To The Message Page Where The Text Made Earlier That Couldn't Been Send. | |  | | --- | | 1) Validate User Is Able to Login To Twitter Website Url:  <https://twitter.com/?lang=en>  (step 1)  2) Validate User Is Able to Click on Sign in Button.  (step 2)  3) In Sign in Page Validate User Is Able To Enter Registered Email And Correct Password.  (step 3 to 4)  4) Validate User Is Able to Click Sign in Button.  (step 5)  5) Validate User Is Able to Go To The Homepage After Successfully Login.  (step 6)  6) Validate User Is Able To Click On Message Button At The Homepage.  (step 7)  7) Validate User Is Able To Go To The Message Page.  (step 8 )  8) Validate User Is Able To Click On User Or People That They Want To Text.  (step 9)  9) Validate User Is Able To Enter Text At The Text Box At The Message Page.  (step10)  10) Validate User Is Able To Click On Send Button At The Message Page.  (step 11)  11) Validate User Is Able To Receive Pop Up Saying Text Is Exceeded The Limit.  (step 12)  12) Validate User Is Not Able To Send The Text That Made Earlier At The Message Page.  (step 13)  13) Validate User Is Able To Go Back To The Text Box At The Message Page Where The Text Made Earlier.  (step 14) | | Captiom: Input Text That Exceeded The Limit | Passed |  |
| Execution type: | Manual | | | | |
| Estimated exec. duration (min): |  | | | | |
| Priority: | Medium | | | | |
|  | | | | | |
| Requirements | None | | | | |
| Keywords: | None | | | | |
| **Execution Details** |  | | | | |
| Build | Twitter Build For System Test Plan | | | | |
| Tester | eggmoll | | | | |
| Execution Result: | **Passed** | | | | |
| Execution Mode: | **Manual** | | | | |
| Execution duration (min): |  | | | | |

Actual Result:

User Will Click On Message On Twitter At Homepage

A screenshot of a video chat

Description automatically generated

User Will Enter Text That Exceeded The Text Limit

A screenshot of a black screen

Description automatically generated

User Will Go Back To Homepage

A screenshot of a video chat

Description automatically generated