National University of Computer and Emerging Sciences Chiniot-Faisalabad Campus



**metabase**

Software Quality Engineering

### Semester Project

### Phase 2

### BS (SE) 5A

## Team Name: zzwave-testing

## Team Member’s:

## 21F-9516 -> Suresh Kumar

## 21F-9519 -> Rai umer farooq

## 21f-9510 -> mian fahad akhtar

# **Test Cases For GUI: -**

1. **Login Validation:**

* *Test Case:* Verify that the login screen is displayed with fields for username and password.
* *Expected Result:* The user should see input fields for username and password.

1. **Successful Login:**

* *Test Case:* Log in with valid credentials.
* *Expected Result:* The user should be redirected to the Action Module dashboard.

1. **Invalid Login:**

* *Test Case:* Attempt to log in with invalid credentials.
* *Expected Result:* The system should display an error message indicating invalid login details.

1. **Logout Functionality:**

* *Test Case:* Check if the logout button logs the user out.
* *Expected Result:* The user should be redirected to the login screen.

1. **Action List Display:**

* *Test Case:* Ensure that the Action Module dashboard displays a list of available actions.
* *Expected Result:* The user should see a clear list of available actions.

1. **Action Execution:**

* *Test Case:* Execute a specific action from the list.
* *Expected Result:* The system should perform the action, and the user interface should reflect the changes.

1. **Confirmation Dialog:**

* *Test Case:* Verify that a confirmation dialog appears before executing a critical action.
* *Expected Result:* The user should see a confirmation dialog seeking confirmation for the action.

1. **Cancel Action:**

* *Test Case:* Attempt to cancel an action after initiating it.
* *Expected Result:* The system should revert any changes made by the action.

1. **Error Handling:**

* *Test Case:* Execute an action with incomplete or invalid data.
* *Expected Result:* The system should display appropriate error messages and prevent the action execution.

1. **Action History:**

* *Test Case:* Check if the system maintains a history log of executed actions.
* *Expected Result:* The user should access a log showing executed actions with timestamps.

1. **Filter Actions:**

* *Test Case:* Use filters to display specific types of actions.
* *Expected Result:* The system should accurately filter and display the selected type of actions.

1. **Responsiveness:**

* *Test Case:* Test the responsiveness of the Action Module on different screen sizes.
* *Expected Result:* The user interface should adapt to various screen sizes without loss of functionality.

1. **Tooltip Display:**

* *Test Case:* Hover over an action to check if a tooltip displays additional information.
* *Expected Result:* A tooltip should appear with relevant details about the action.

1. **Keyboard Shortcuts:**

* *Test Case:* Check if keyboard shortcuts (if applicable) perform the intended actions.
* *Expected Result:* Keyboard shortcuts should trigger the associated actions.

1. **Session Timeout:**

* *Test Case:* Leave the system idle for a session timeout period and attempt to perform an action.
* *Expected Result:* The system should prompt the user to log in again after a session timeout.

# **Gherkin For GUI: -**

Feature: MetaBase Action Module Functionality

* Scenario: Login Functionality
* Given the user is on the login screen
* When they enter valid credentials
* Then they should be redirected to the Action Module dashboard
* Scenario: Invalid Login Attempt
* Given the user is on the login screen
* When they enter invalid credentials
* Then an error message should be displayed
* Scenario: Logout Functionality
* Given the user is logged into the Action Module
* When they click on the logout button
* Then they should be redirected to the login screen
* Scenario: Display Action List
* Given the user is on the Action Module dashboard
* Then they should see a list of available actions
* Scenario: Execute Action
* Given the user is on the Action Module dashboard
* When they execute a specific action
* Then the system should perform the action
* Scenario: Confirmation Dialog for Critical Action
* Given the user is about to execute a critical action
* When they confirm the action
* Then the system should perform the action
* Scenario: Cancel Action
* Given the user is in the process of executing an action
* When they cancel the action
* Then any changes made by the action should be reverted
* Scenario: Error Handling for Action Execution
* Given the user is on the Action Module dashboard
* When they attempt to execute an action with incomplete or invalid data
* Then appropriate error messages should be displayed
* Scenario: Action History
* Given the user is on the Action Module dashboard
* When they check the action history log
* Then they should see a log of executed actions with timestamps
* Scenario: Filter Actions
* Given the user is on the Action Module dashboard
* When they apply filters to display specific types of actions
* Then the system should accurately display the selected type of actions
* Scenario: Responsiveness on Different Screen Sizes
* Given the user is on the Action Module dashboard
* When they access the dashboard on different screen sizes
* Then the user interface should adapt without loss of functionality
* Scenario: Tooltip Display
* Given the user is on the Action Module dashboard
* When they hover over an action
* Then a tooltip should appear with relevant details about the action
* Scenario: Keyboard Shortcuts
* Given the user is on the Action Module dashboard
* When they use keyboard shortcuts (if applicable)
* Then the associated actions should be triggered
* Scenario: Session Timeout
* Given the user is logged into the Action Module
* When they leave the system idle for a session timeout period
* Then the system should prompt the user to log in again