# SQE Project phase 2

# GUI test cases in java:

### Code:

| package gui\_testCases;  import org.openqa.selenium.By;  import org.openqa.selenium.WebDriver;  import org.openqa.selenium.WebElement;  import org.openqa.selenium.chrome.ChromeDriver;  import java.util.Scanner;  public class GuiTestMenu {  private static final int MAX\_ALLOWED\_DATA = 0;  public static void main(String[] args) {  try  {  Scanner scanner = new Scanner(System.in);  System.setProperty("webdriver.chrome.driver", "C:\\Users\\Abdullah Shahab\\Downloads\\chromedriver-win64\\chromedriver.exe");  WebDriver driver = new ChromeDriver();    int choice;  do {  System.out.println("Choose a test case to run:");  System.out.println("1. Data Page Opens When 'Add Data' is Clicked");  System.out.println("2. Error Message Displayed When Trying to Add Data with No Input");  System.out.println("3. Personal Dashboard is Successfully Added to 'Personal Collection'");  System.out.println("4. Error Message Displayed When Trying to Add a Personal Dashboard with No Name");  System.out.println("5. Maximum Allowed Data Can Be Added Without Errors");  System.out.println("0. Quit");  System.out.print("Enter your choice: ");  choice = scanner.nextInt();  scanner.nextLine();  switch (choice) {  case 1:    driver.get("https://www.google.com");  driver.findElement(By.linkText("Collection")).click();  driver.findElement(By.linkText("Add Data")).click();  WebElement dataPage = driver.findElement(By.id("data-page"));  assert dataPage.isDisplayed() : "Data page did not open as expected";  break;  case 2:  driver.get("https://www.google.com");  driver.findElement(By.linkText("Collection")).click();  driver.findElement(By.linkText("Add Data")).click();  driver.findElement(By.id("add-data-button")).click();  driver.findElement(By.id("save-button")).click();  WebElement errorMessage = driver.findElement(By.id("error-message"));  if (errorMessage.isDisplayed()) {  System.out.println("Error message was displayed as expected.");  } else {  System.out.println("Error message was not displayed.");  }  break;  case 3:  driver.get("https://www.google.com");  driver.findElement(By.linkText("Collection")).click();  driver.findElement(By.linkText("Personal Collection")).click();  driver.findElement(By.id("add-dashboard-button")).click();  driver.findElement(By.id("dashboard-name")).sendKeys("My Dashboard");  driver.findElement(By.id("dashboard-description")).sendKeys("Description for My Dashboard");  driver.findElement(By.id("save-dashboard-button")).click();  WebElement successMessage = driver.findElement(By.id("success-message"));  assert successMessage.isDisplayed() : "Personal dashboard was not added successfully";  break;  case 4:  driver.get("https://www.google.com");  driver.findElement(By.linkText("Collection")).click();  driver.findElement(By.linkText("Personal Collection")).click();  driver.findElement(By.id("add-personal-dashboard-button")).click();  driver.findElement(By.id("save-dashboard-button")).click();  WebElement errorMessage1 = driver.findElement(By.id("dashboard-error-message"));  assert errorMessage1.isDisplayed() : "Error message for dashboard name was not displayed";  break;  case 5:  driver.get("https://www.google.com");  driver.findElement(By.linkText("Collection")).click();  driver.findElement(By.linkText("Add Data")).click();  for (int i = 1; i <= MAX\_ALLOWED\_DATA; i++) {  WebElement dataInput = driver.findElement(By.id("data-input"));  dataInput.sendKeys("Data Point " + i);  driver.findElement(By.id("add-button")).click();  }  WebElement addedData = driver.findElement(By.id("added-data"));  assert addedData.isDisplayed() : "Maximum allowed data was not added";  break;  case 0:  break;  default:  System.out.println("Invalid choice. Please enter a valid option.");  }  } while (choice != 0);  driver.quit();  }  catch (NoClassDefFoundError e) {  // Handle or ignore the exception here  System.out.println("NoClassDefFoundError occurred but was ignored.");  }  }  } |
| --- |

### 

# API test cases in java:

### Code:

| package api\_testCases;  import io.restassured.RestAssured;  import io.restassured.http.ContentType;  import io.restassured.response.Response;  public class api\_testMenu {  public static void main(String[] args) {  RestAssured.baseURI = "https://api.testsite.org";  // Scenario 1: Successfully create a new user in the system  String createUserPayload = "{\"username\":\"newUser\",\"email\":\"newUser@example.com\",\"password\":\"password123\"}";  Response createUserResponse = RestAssured.given()  .contentType(ContentType.JSON)  .body(createUserPayload)  .post("/api/users");  if (createUserResponse.getStatusCode() == 201) {  System.***out***.println("User is created successfully");  } else {  System.***out***.println("User creation failed");  }  // Scenario 2: Successfully retrieve user details by ID  int userId = 1;  Response getUserResponse = RestAssured.get("/api/users/" + userId);  if (getUserResponse.getStatusCode() == 200) {  System.***out***.println("User details are successfully retrieved");  } else {  System.***out***.println("Failed to retrieve user details");  }  // Scenario 3: Successfully update user information  int updateUserPayload = "{\"email\":\"updatedEmail@example.com\"}";  Response updateUserResponse = RestAssured.given()  .contentType(ContentType.JSON)  .body(updateUserPayload)  .put("/api/users/" + userId);  if (updateUserResponse.getStatusCode() == 200) {  System.***out***.println("User information is successfully updated");  } else {  System.***out***.println("Failed to update user information");  }  // Scenario 4: Successfully delete a user from the system  Response deleteUserResponse = RestAssured.delete("/api/users/" + userId);  if (deleteUserResponse.getStatusCode() == 204) {  System.***out***.println("User is deleted successfully");  } else {  System.***out***.println("Failed to delete the user");  }  // Scenario 5: Handle invalid data when creating a user  String invalidUserPayload = "{\"username\":\"\",\"email\":\"invalidEmail\",\"password\":\"123\"}";  Response createInvalidUserResponse = RestAssured.given()  .contentType(ContentType.JSON)  .body(invalidUserPayload)  .post("/api/users");  if (createInvalidUserResponse.getStatusCode() == 400) {  System.***out***.println("Error is returned for invalid user data");  } else {  System.***out***.println("Invalid user data was accepted");  }  }  } |
| --- |