

Software Quality Management

Master of Information Technology

QA Assignment

MIT4201

Index No: 20550405

Name : H. A. K. S. Hettiarachchi

Question 01

a)

#	Product Risks	Criticality	Mitigation
1	Security flaws could arise from the application's improper handling of incorrect authentication attempts.	High, since it could risk employee data confidentiality.	Test authentication thoroughly using fictitious accounts and passwords. Put in place robust password restrictions and account lockout measures.
2	Unauthorized access or session hijacking could result from poor session management.	High, since it could risk user session integrity.	Use secure session methods, put in place session timeouts, and conduct frequent vulnerability assessments.
3	Injection attacks may arise due to inadequate input validation.	High, since it could risk the data integrity of the program.	Put in place robust input validation procedures, clean user input, and carry out frequent security audits.
4	The identification of security events or anomalies may be affected by inadequate logging.	Moderate to High because stopping security breaches requires prompt notice.	Put in place reliable logging procedures, keep a close eye on logs, and set up warnings for any suspicious activity.
5	Poor Policies for Passwords	Medium	Establish account lockout procedures, enforce strict password regulations, and provide users with password security education.

b)

1. Quality Goals and Range

Describe the project's overall quality goals, the testing parameters, and the methods for measuring and evaluating quality.

2. Roles and Responsibility

All team members participating in quality management, including those in development, testing, and management, should have their roles and duties well defined.

3. Management and Reporting of Defects

Specify the procedures for managing, monitoring, and reporting faults. Indicate the process for classifying, prioritizing, and informing stakeholders about faults.

4. Reporting & Metrics

Describe the metrics and key performance indicators (KPIs) that will be monitored to gauge the success of the testing procedure. Explain to stakeholders how and when they will receive reports on these measures.

5. Deliverables and Documentation

Describe the templates and documentation standards that will be applied to the project. Specify the outputs that should be produced at each stage of the testing procedure.

Question 02

2.

a). **Logging to the OrangeHRM system as an admin user** <https://opensource-demo.orangehrmlive.com/web/index.php/auth/login>

Feature file:

Feature: Admin Login to OrangeHRM

Scenario: Logging in as an admin user

Given I navigate to the OrangeHRM login page

When I enter username "Admin" and password "admin123"

And I click the login button

Then I should be logged in as an admin

And I should be redirected to the dashboard

Step Definitions:

```
public class OrangeHRMSteps {
```

```
    WebDriver driver;
```

```
    @Given("I navigate to the OrangeHRM login page")
```

```
    public void navigateToLoginPage() {
```

```
        System.setProperty("webdriver.chrome.driver", "path_to_chromedriver");
```

```
        ChromeOptions options = new ChromeOptions();
```

```
        options.addArguments("--start-maximized"); // Maximize the window
```

```
        driver = new ChromeDriver(options);
```

```
        driver.get("https://opensource-demo.orangehrmlive.com/");
```

```
    }
```

```

@When("I enter username {string} and password {string}")
public void enterCredentials(String username, String password) {
    WebElement usernameInput = driver.findElement(By.id("txtUsername"));
    WebElement passwordInput = driver.findElement(By.id("txtPassword"));

    usernameInput.sendKeys(username);
    passwordInput.sendKeys(password);
}

@And("I click the login button")
public void clickLoginButton() {
    WebElement loginButton = driver.findElement(By.id("btnLogin"));
    loginButton.click();
}

@Then("I should be logged in as admin")
public void verifyLoggedInAsAdmin() {
    WebDriverWait wait = new WebDriverWait(driver, 10);
    WebElement welcomeMessage =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("welcome")));
    assert welcomeMessage.getText().contains("Admin");
}

@And("I should be redirected to the dashboard")
public void verifyRedirectedToDashboard() {
    WebDriverWait wait = new WebDriverWait(driver, 10);
    wait.until(ExpectedConditions.urlContains("/dashboard"));
    driver.quit();
}
}

```

b) Click on Add user and get the add user pane.

Feature file:

Feature: Add User Functionality

Scenario: Clicking on Add user and getting the add user pane

Given I am logged in as an admin user on the OrangeHRM system

When I click on the "Add user" button

Then I should see the "Add User" pane displayed

Step Definitions:

```
public class AddUserSteps {
```

```
    WebDriver driver;
```

```
    @Given("I am logged in as an admin user on the OrangeHRM system")
```

```
    public void loginAsAdminUser() {
```

```
        System.setProperty("webdriver.chrome.driver", "path_to_chromedriver");
```

```
        ChromeOptions options = new ChromeOptions();
```

```
        options.addArguments("--start-maximized"); // Maximize the window
```

```
        driver = new ChromeDriver(options);
```

```
        driver.get("https://opensource-demo.orangehrmlive.com/");
```

```
        // Assuming you have login implemented and you're logged in as admin
```

```
        WebElement usernameInput = driver.findElement(By.id("txtUsername"));
```

```
        WebElement passwordInput = driver.findElement(By.id("txtPassword"));
```

```
        WebElement loginButton = driver.findElement(By.id("btnLogin"));
```

```
        usernameInput.sendKeys("Admin");
```

```
        passwordInput.sendKeys("admin123");
```

```
loginButton.click();  
}
```

```
@When("I click on the {string} button")  
public void clickOnButton(String buttonName) {  
    WebElement addButton = driver.findElement(By.id(buttonName));  
    addButton.click();  
}
```

```
@Then("I should see the {string} pane displayed")  
public void verifyAddUserPaneDisplayed(String paneName) {  
    WebDriverWait wait = new WebDriverWait(driver, 10);  
    WebElement addUserPane =  
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id(paneName)));  
    assert addUserPane.isDisplayed();  
}
```

```
@After  
public void closeBrowser() {  
    driver.quit();  
}}
```

c) Fill in mandatory fields and create a new user

Feature file:

Feature: Create a New User with Mandatory Fields

Scenario: Filling in mandatory fields and creating a new user

Given I am logged in as an admin user on the OrangeHRM system

When I click on the "Add user" button

And I fill in the mandatory fields with valid data

And I submit the form

Then I should see a success message indicating the user has been created

Step Definitions:

```
public class CreateUserSteps {
```

```
    WebDriver driver;
```

```
    @Given("I am logged in as an admin user on the OrangeHRM system")
```

```
    public void loginAsAdminUser() {
```

```
        System.setProperty("webdriver.chrome.driver", "path_to_chromedriver");
```

```
        ChromeOptions options = new ChromeOptions();
```

```
        options.addArguments("--start-maximized"); // Maximize the window
```

```
        driver = new ChromeDriver(options);
```

```
        driver.get("https://opensource-demo.orangehrmlive.com/");
```

```
        // Assuming you have login implemented and you're logged in as admin
```

```
        WebElement usernameInput = driver.findElement(By.id("txtUsername"));
```

```
        WebElement passwordInput = driver.findElement(By.id("txtPassword"));
```

```
        WebElement loginButton = driver.findElement(By.id("btnLogin"));
```

```
        usernameInput.sendKeys("Admin");
```



```
passwordInput.sendKeys("admin123");  
loginButton.click();  
}
```

```
@When("I click on the {string} button")  
public void clickOnButton(String buttonName) {  
    WebElement addButton = driver.findElement(By.id(buttonName));  
    addButton.click();  
}
```

```
@And("I fill in the mandatory fields with valid data")  
public void fillInMandatoryFields() {  
    WebElement usernameInput = driver.findElement(By.id("systemUser_userName"));  
    WebElement employeeNameInput =  
driver.findElement(By.id("systemUser_employeeName_empName"));  
    WebElement passwordInput = driver.findElement(By.id("systemUser_password"));  
    WebElement confirmPasswordInput =  
driver.findElement(By.id("systemUser_confirmPassword"));  
    WebElement saveButton = driver.findElement(By.id("btnSave"));  
  
    usernameInput.sendKeys("newuser");  
    employeeNameInput.sendKeys("John Doe");  
    passwordInput.sendKeys("password");  
    confirmPasswordInput.sendKeys("password");  
  
    // You may fill in other mandatory fields here if needed  
}
```

```
@And("I submit the form")
```

```
public void submitForm() {  
    WebElement saveButton = driver.findElement(By.id("btnSave"));  
    saveButton.click();  
}
```

```
@Then("I should see a success message indicating the user has been created")
```

```
public void verifyUserCreationSuccess() {  
    WebDriverWait wait = new WebDriverWait(driver, 10);  
    WebElement successMessage =  
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//div[contains(@class,'message success')]")));  
    assert successMessage.isDisplayed();  
}
```

```
@After
```

```
public void closeBrowser() {  
    driver.quit();  
}  
}
```

d) Validate that the new user created successfully

Feature file:

Feature: Validate New User Creation

Scenario: Verify that a new user is created successfully

Given I am logged in as an admin user on the OrangeHRM system

And a new user is created successfully

When I search for the newly created user

Then I should see the details of the newly created user

Step Definitions:

```
public class CreateUserSteps {
```

```
    WebDriver driver;
```

```
    @Given("I am logged in as an admin user on the OrangeHRM system")
```

```
    public void loginAsAdminUser() {
```

```
        System.setProperty("webdriver.chrome.driver", "path_to_chromedriver");
```

```
        ChromeOptions options = new ChromeOptions();
```

```
        options.addArguments("--start-maximized"); // Maximize the window
```

```
        driver = new ChromeDriver(options);
```

```
        driver.get("https://opensource-demo.orangehrmlive.com/");
```

```
        // Assuming you have login implemented and you're logged in as admin
```

```
        WebElement usernameInput = driver.findElement(By.id("txtUsername"));
```

```
        WebElement passwordInput = driver.findElement(By.id("txtPassword"));
```

```
        WebElement loginButton = driver.findElement(By.id("btnLogin"));
```

```
        usernameInput.sendKeys("Admin");
```

```
        passwordInput.sendKeys("admin123");
```

```
loginButton.click();  
}
```

```
@When("I click on the {string} button")  
public void clickOnButton(String buttonName) {  
    WebElement addButton = driver.findElement(By.id(buttonName));  
    addButton.click();  
}
```

```
@And("I fill in the mandatory fields with valid data")  
public void fillInMandatoryFields() {  
    WebElement usernameInput = driver.findElement(By.id("systemUser_userName"));  
    WebElement employeeNameInput =  
driver.findElement(By.id("systemUser_employeeName_empName"));  
    WebElement passwordInput = driver.findElement(By.id("systemUser_password"));  
    WebElement confirmPasswordInput =  
driver.findElement(By.id("systemUser_confirmPassword"));  
    WebElement saveButton = driver.findElement(By.id("btnSave"));  
  
    usernameInput.sendKeys("newuser");  
    employeeNameInput.sendKeys("John Doe");  
    passwordInput.sendKeys("password");  
    confirmPasswordInput.sendKeys("password");  
  
    // You may fill in other mandatory fields here if needed  
}
```

```
@And("I submit the form")  
public void submitForm() {
```

```
WebElement saveButton = driver.findElement(By.id("btnSave"));
saveButton.click();
}
```

```
@Then("I should see a success message indicating the user has been created")
public void verifyUserCreationSuccess() {
    WebDriverWait wait = new WebDriverWait(driver, 10);
    WebElement successMessage =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//div[contains(@class,'message success')]")));
    assert successMessage.isDisplayed();
}
```

```
@And("I navigate to the user list")
public void navigateToUserList() {
    WebElement adminTab = driver.findElement(By.id("menu_admin_viewAdminModule"));
    adminTab.click();
    WebElement userManagementTab =
driver.findElement(By.id("menu_admin_UserManagement"));
    userManagementTab.click();
    WebElement usersTab = driver.findElement(By.id("menu_admin_viewSystemUsers"));
    usersTab.click();
}
```

```
@Then("I should see the newly created user in the user list")
public void verifyNewUserInUserList() {
    WebDriverWait wait = new WebDriverWait(driver, 10);
    WebElement table =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("resultTable")));
}
```

```

List<WebElement> rows = table.findElements(By.tagName("tr"));
boolean userFound = false;
for (WebElement row : rows) {
    List<WebElement> cells = row.findElements(By.tagName("td"));
    for (WebElement cell : cells) {
        if (cell.getText().equals("newuser")) { // Assuming "newuser" is the username
            userFound = true;
            break;
        }
    }
    if (userFound) {
        break;
    }
}
assert userFound;
}

@After
public void closeBrowser() {
    driver.quit();
}
}

```