**Testing JAVA Applications and Writing Test Cases**

---------------------------------------------------------------------------------------------------------------------------

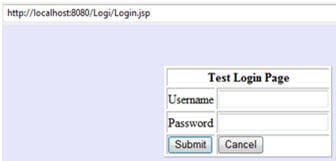
After completing this handson you will be able to **write Test cases** irrespective of any **testing framework**

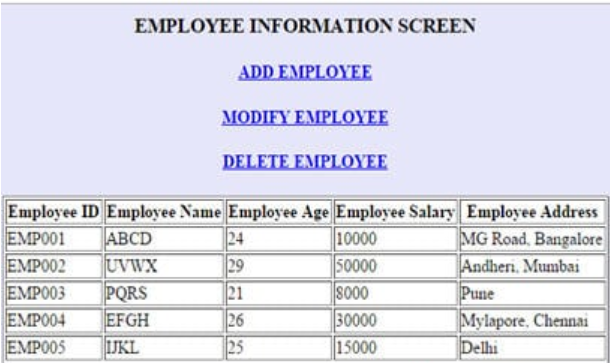
For eg: consider three screens.

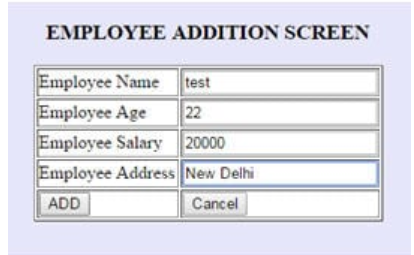
* A login screen
* An employee display screen, which lists all employees in the organization
* An employee modification/addition/removal screen.

The UI (User Interface) for these three screens are developed with JSP/HTML and the validations performed through JavaScript. Because it is a sample application, logic is in the Servlet and DAO (Data Access Object). DAO is a class for connecting to the database.

**Below are the sample screens:**







### **Manual Java Application Testing:**

During manual JAVA testing, a tester prepares the test cases from the detailed design document and tries to cover every scenario and code snippet possible.

**JAVA UNIT TESTING**

[Unit testing is a type of testing](https://www.softwaretestinghelp.com/unit-testing/) wherein a user needs to test the smallest of the code snippets for accuracy, correctness and meeting the requirements

Let us take the example of the login screen. The login screen has two text fields: username and password, and has two buttons: submit and cancel.

The test cases should cover all the loops and conditional statements. Test cases should display the expected results and the test data. Below are some of the general test cases that a user could execute manually in a login screen. The results are then noted down in the test case document.

**Below is a sample test case format for the login screen.**

| **S.No.** | **Test Case** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- |
| 1 | User checks the appearance of labels Username, Password | The labels should be correctly spelled and displayed in Normal sized font | The label username and password are displayed correctly | PASS |
| 2 | User checks the appearance of the button Submit and Cancel | The buttons should be displayed with the correct name | The buttons Submit and Cancel are displayed correctly | PASS |
| 3 | User checks the background color of the screen | The login form should be within a white table and the screen should be of background grey | The screen appearance does not match the requirements. | FAIL |
| 4 | The user leaves username textbox as Blank | Error message “Username cannot be empty” should be displayed | Error message “Username cannot be empty” is displayed | PASS |
| 5 | The user enters some value in the username textbox and leave the password textbox as Blank | Error message “Password cannot be empty” should be displayed | Error message “Password cannot be empty” is displayed | PASS |
| 6 | User enters username as “abcd” and password as “xxxx” | Error message “Invalid username password combination” should be displayed | Error message “Invalid username password combination” is displayed | PASS |
| 7 | User enters a username of more than 10 characters | Error message  “Username should not be more than 10 characters” should be displayed | Error message is not displayed | FAIL |
| 8 | User enters username as “testuser” and password as “password” and clicks the Submit button | The user should be able to see the “Employee details screen” | Employee details screen is displayed | PASS |
| 9 | User enters username as!@# | Error message “Username can’t contain special characters” | Error message is displayed | PASS |
| 10 | User enters a username of less than 6 characters | Error message “Username too short” | Error message is not displayed | FAIL |
| 11 | User enters correct combination of password and username | User should be taken to next page | User is taken to next page | PASS |
| 12 | User clicks on cancel button | The form should be reset | Cancel operation is performed | PASS |
| 13 | User clicks on Submit | Check for user permissions | Access given | PASS |

After going through the test cases, you may realize that you are mostly dealing with the testing of fields, buttons, functionality, and validations of a particular screen. This is accurate, as Unit Testing very keenly deals with the testing of every small code snippet and component. The same type of testing should be performed for all the screens.

**Test case format for the Employee Information screen.**

| **S.No.** | **Test Case** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- |
| 1 | User clicks on Add Employee | Redirect page to Employee addition screen | Employee Addition screen is displayed | PASS |
| 2 | User clicks on delete Employee | Redirect page to Employee Delete screen | Error Page not found | FAIL |
| 3 | User views added entries | Display added entries | Added entries are displayed | PASS |

**Test case format for the Employee Addition screen**

| **S.No.** | **Test Case** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- |
| 1 | User clicks on Add button without entering info in fields | Error message “Employee Name can’t be empty” should be displayed | Error message is displayed | PASS |
| 2 | User checks the appearance of labels Name,Age,Salary,Address | The labels should be correctly spelled and displayed in Normal sized font | The screen appearance does not match the requirements. | FAIL |
| 3 | User checks the appearance of the button Add and Cancel | The buttons should be displayed with the correct name | The buttons Add and Cancel are displayed correctly | PASS |
| 4 | The user enters some value in the name,age,salary textbox and leaves the address textbox as Blank | Error message “address cannot be empty” should be displayed | Error message “address cannot be empty” is displayed | PASS |
| 5 | User enters correct combination of Name,Age,Salary and Address | User should be taken to Employee info. page | User is taken to Employee info page | PASS |