**Testing JAVA Applications and Writing Test Cases**

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

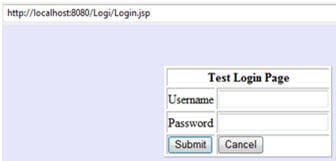
After completing this handson you will be able to **write Test cases** irrespective of any **testing framework**

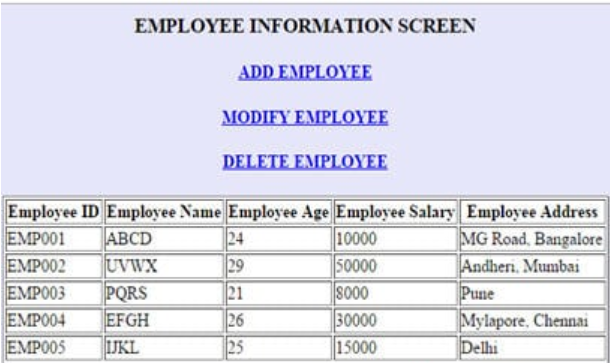
For eg: consider three screens.

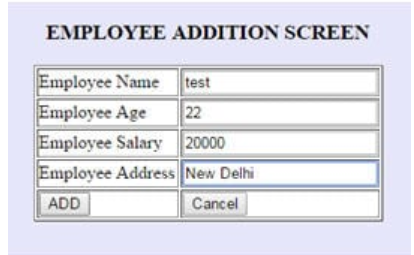
* A login screen
* An employee display screen, which lists all employees in the organization
* An employee modification/addition/removal screen.

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 | If User has entered the correct formats of username and Password | User should not get any alert message Regarding the format or the message can be ”Right User Name and Password is entered.” | User gets the message “Right User Name and Password is entered”. | PASS |
| 10. | To check whether after entering username and Password then it should forward to the next page. | The user should be able to see “Employee details screen” | Employee Details screen should be displayed. | PASS |
| 11. | Check whether user has entered null in username or password or not. | If both the fields are kept empty. Then it should show error near the textbox that “User Name or Password cannot be null”. | User should get error “User Name or Password cannot be null.” | PASS |
| 12. | Whether special character is available in username or not. | If user doesn’t enter the special characters in username then it should show error “Enter the correct User Name”. | User gets the message “Enter the correct User Name.” | PASS |
| 13. | Check if the cancel button works as per the requirement. | The user Login page should be shown again after entering cancel button. | Login page is displayed again. | 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 Cases of Add Employee :-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl. No. | Test Case | Expected Result | Actual Result | PASS/FAIL |
| 1. | Check whether clicking Add Employee Link Add Employee Page is displaying or not. | “Add Employee” Page should be displayed. | “Add Employee” Page is displayed. | PASS |
| 2. | To check whether the user has entered all the required details or not. | Will show error by displaying “All the fields are required.” | “All the fields are required” message is shown. | PASS |
| 3. | To check whether the details has entered in the correct format or not. | Will show error by displaying “Enter the details in the correct Format.” | “Enter the details in the correct Format” message is shown. | PASS |
| 4. | To check after clicking add button whether the particular employee details is displayed or not. | Employee Details page will be displayed. | Employee Details page has shown. | PASS |
| 5. | To check after clicking cancel button whether the blank addition page is showing or not. | Employee add page should be displayed. | Employee add page is displayed. | PASS |