



## Lab Final Report

Only for course Teacher						
		Needs Improvement	Developing	Sufficient	Above Average	Total Mark
Allocate mark & Percentage		25%	50%	75%	100%	40
Problem Understanding	10					
Analysis	15					
Implementation	10					
Task Efficiency	5					
Total obtained mark						
Comments						

**Semester: Fall 2024**

**Student Name: Nur Ahmed**

**Student ID: 222-35-1111**

**Batch: 38**

**Section: A2**

**Course Code: SE 313**

**Course Name: Software Quality Assurance & Testing Lab**

**Course Teacher Name: Md.**

**Suhag Ali**

**Designation: Lecturer**

**Submission Date: 29/11/2024**

# Table Of Contents

## Contents

<b>Class Work 1 (Test Case for Login)</b> .....	3
<b>Class Work 2 (Test Case for Lift)</b> .....	4
<b>Class Work 3 (Triangle Problem)</b> .....	5
<b>Equivalence Class Partitioning</b> .....	5
<b>Class Work 4 (Next Day Problem)</b> .....	7
<b>Boundary Value Analysis</b> .....	7
<b>Equivalence-class testing</b> .....	8
<b>Class Work 5 (Decision Table Based Testing)</b> .....	10
<b>Class Work 6 (Lab Performance 2)</b> .....	11
<b>Class Work 7 (Lab Performance 3)</b> .....	12

## Class Work 1 (Test Case for Login)

We have Learned how to perform ‘Test Case Analysis’ in this class and performed one for ‘Log In’.

We have performed one test case for successful log in and another for unsuccessful log in.

**Here is how I have created the test case for ‘Log In’:**

Project Name	Test Case design for login Page																		
Module Name	Login Page																		
Created By	Mur Ahmed																		
Creation Date	12/9/2024																		
Received By	SK																		
Received Date	12/9/2024																		
Test Scenario ID	Test Scenario Description	Test Case ID	Test Case	Test Steps	Preconditions	Test Data	Expected Result	Actual Result	Status	Comment									
TS_LOGIN_001	Verify login with valid credentials	TC_LOGIN_001	Verify successful login with valid credentials	1. Open the login page. 2. Enter valid username and password. 3. Click Login.	User is registered with valid credentials.	Username: murAhmed Password: mur@123	User is successfully logged in and redirected to the dashboard/home page.	Passed	Passed	Results verified successfully.									
TS_LOGIN_002	Verify error message for invalid credentials	TC_LOGIN_002	Verify login failure with invalid credentials	1. Open the login page. 2. Enter invalid username/password. 3. Click Login.		Username: invalid Password: mur@123	An error message "Invalid username or password" is displayed.	Passed	Passed	Functionality confirmed.									
TS_LOGIN_003	Verify login behavior with blank fields	TC_LOGIN_003	Verify login attempt with empty username/password fields	1. Open the login page. 2. Leave username and password blank. 3. Click Login.		Blank username and password	A validation message "Fields cannot be empty" is displayed.	Passed	Passed	Behavior as expected.									
TS_LOGIN_004	Verify password masking functionality	TC_LOGIN_004	Verify password input is masked	1. Open the login page. 2. Enter a password. 3. Check the input field for masking.	None	Password: mur@123	Password is displayed as dots or asterisks.	Passed	Passed	Password masking works well.									
TS_LOGIN_005	Verify login functionality with case-sensitive passwords	TC_LOGIN_005	Verify password case sensitivity	1. Open the login page. 2. Enter valid username and password in wrong case. 3. Click Login.	User is registered.	Username: murAhmed Password: mur@123	An error message "Invalid username or password" is displayed.	Passed	Passed	Case sensitivity verified.									
TS_LOGIN_006	Verify the "Remember Me" functionality	TC_LOGIN_006	Verify functionality of the "Remember Me" checkbox	1. Select "Remember Me" checkbox. 2. Login. 3. Logout and revisit the login page.	User is registered and logged out successfully.	Username: murAhmed Password: mur@123	The username and password fields are pre-filled upon returning to the login page.	Passed	Passed	Works as expected.									
TS_LOGIN_007	Verify SQL injection protection	TC_LOGIN_007	Verify SQL injection prevention	1. Open the login page. 2. Enter "OR 1=1" in the username field. 3. Click Login.	None	Username: OR 1=1	Application prevents SQL injection attempts and does not login.	Passed	Passed	Security intact.									
TS_LOGIN_008	Verify login behavior with multiple failed login attempts	TC_LOGIN_008	Verify CAPTCHA implementation	1. Attempt multiple failed logins. 2. Observe the login behavior.	None	Invalid credentials	CAPTCHA is displayed after a defined number of failed login attempts.	Passed	Passed	CAPTCHA functionality confirmed.									
TS_LOGIN_009	Verify error message for inactive accounts	TC_LOGIN_009	Verify login with an inactive user account	1. Enter credentials for an inactive account. 2. Click Login.	User account is deactivated.	Username: murAhmed Password: mur@123	An error message "Account is inactive" is displayed.	Passed	Passed	Error handling is accurate.									
TS_LOGIN_010	Verify responsiveness of the login page	TC_LOGIN_010	Verify the page layout on multiple devices	1. Open the login page on desktop, tablet, and mobile devices.	None	Various devices	The login page layout adjusts and displays properly on all screen sizes.	Passed	Passed	Responsive design verified.									

## Class Work 2 (Test Case for Lift)

We have Learned how to perform ‘Test Case Analysis’ in this class and performed one for a lift.

We have performed one test case for a lift, determining whether the lift will go up or not.

**Here is how I have created the test case for a lift:**

Project Name	Test Case for lift									
Module Name	Calling Lift									
Created by	Nur Ahmed									
Creation Date	8/9/2024									
Received by	SA									
Received date	10/9/2024									
Test Scenarion ID	Test Scenario Description	Test Case ID	Test case description	Test Steps	Pre Condition	Test Data	Expected Result	Actual Result	Status	Comment
TC101	Calling lift from Ground Floor	TCL101	Press the down button and the lift will come down	Calling lift	Existance of a lift in the building		Lift come down	Lift comes	Pass	No

## Class Work 3 (Triangle Problem)

We have learned how to perform 'Equivalence Class Partitioning' in this class and performed one for 'Triangle Problem', where we determined how different types of triangles are tested.

We have performed test cases for each different triangle and some more for the invalid triangle values.

**Here is how I have created the test case for 'Triangle Problem':**

### Equivalence Class Partitioning

**Problem:** The triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Assume that the upper limit for the size of any side is 10. Derive test cases for your program based on BVC, execute the test cases and discuss the results.

#### 1. Valid Triangle Sides

1. Equilateral Triangle: All three sides are equal, (5, 5, 5).
2. Isosceles Triangle: Two sides are equal, and the third is different, (5, 5, 3).
3. Scalene Triangle: All three sides are different, but the triangle inequality holds, (4, 5, 6).

#### 2. Invalid Triangle Sides

1. Not a Triangle: The three sides do not satisfy the triangle inequality theorem, (1, 2, 3).
2. Out of Range: Any side is outside the valid range (1 to 10), (11, 5, 6).

We have three variables a, b and c and each variables value ranges from 1 to 10.

Variables	Invalid	valid	Invalid
a	<1	2,5,9	>10
b	<1	2,5,9	>10
c	<1	2,5,9	>10

Test cases	Input			Expected Output
	A	B	C	
TC1	5	5	5	Equilateral Triangle
TC2	5	5	3	Isosceles Triangle
TC3	4	5	6	Scalene Triangle
TC4	10	10	10	Equilateral Triangle
TC5	0(Invalid)	3	5	Out of Range
TC6	7	4	11(Invalid)	Out of Range
TC7	3	4	8	Not a Triangle (Triangle cannot be formed)

### Description:

1. Equilateral Triangle Cases: TC1 and TC4 confirm the program correctly identifies equilateral triangles.
2. Isosceles Triangle Cases: TC2 test different configurations of two equal sides. Both cases should result in identifying an isosceles triangle.
3. Scalene Triangle Case: TC3 checks for all sides being different. The program should correctly recognize this as a scalene triangle.
4. Not a Triangle Cases: TC7 this tests case breaks the triangle inequality, and the program should recognize these as "not a triangle."
5. Out of Range Cases: TC5 and TC6 include sides outside the 1-10 range. The program should handle these as "out of range."

## Class Work 4 (Next Day Problem)

### Boundary Value Analysis

Considering Date program, we have three variables day, month and year.

Variables	Min	Min+	Nom	Max-	Max
day	1	2	15	30	31
month	1	2	6	11	12
year	1812	1813	1914	2014	2015

Test cases for Date program using Boundary Value Analysis

Test cases	Description	Inputs			Output	Comments
		DD	MM	YY		
BVA1	Enter the values for day(nom),month(nom) and year(min)	15	6	1812	16/6/1812	Valid
BVA2	Enter the values for day(nom),month(nom) and year(min+)	15	6	1813	16/6/1813	Valid
BVA3	Enter the values for day(nom),month(min) and year(nom)	15	6	1914	16/6/1914	Valid
BVA4	Enter the values for day(nom),month(nom) and year(max-)	15	6	2014	16/6/2014	Valid
BVA5	Enter the values for day(nom),month(nom) and year(max)	15	6	2015	16/6/2015	Valid
BVA6	Enter the values for day(nom),month(min) and year(nom)	15	1	1914	16/1/1914	Valid
BVA7	Enter the values for day(nom),month(min+) and year(nom)	15	2	1914	16/2/1914	Valid
BVA8	Enter the values for day(nom),month(max-) and year(nom)	15	11	1914	16/11/1914	Valid
BVA9	Enter the values for day(nom),month(max) and year(nom)	15	12	1914	16/12/1914	Valid
BVA13	Enter the values for day(max),month(nom) and year(nom)	31	6	1914	Day out of range for the month	Valid

## Equivalence-class testing

### Conditions:

D:  $1 < \text{Day} < 31$

M:  $1 < \text{Month} < 12$

Y:  $1800 < \text{Year} < 2048$

### Input classes:

Day:

D1: day between 1 to 28

D2: 29

D3: 30

D4: 31

Month:

M1: Month has 30 days

M2: Month has 31 days

M3: Month is February

Year:

Y1: Year is a leap year

Y2: Year is a normal year

### Output Classes:

Increment Day

Reset Day and Increment Month

Increment Year

Invalid Date



### Equivalence Class Test Cases:

Test Case ID	Day	Month	Year	Expected Output
E1	15	4	2004	16-4-2004
E2	15	4	2003	16-4-2003
E3	15	1	2004	16-1-2004
E4	15	1	2003	16-1-2003
E5	15	2	2004	16-2-2004
E6	15	2	2003	16-2-2003
E7	29	4	2004	30-4-2004
E8	29	4	2003	30-4-2003
E9	29	1	2004	30-1-2004
E10	29	1	2003	30-1-2003
E11	29	2	2004	1-3-2004
E12	29	2	2003	Invalid Date
E13	30	4	2004	1-5-2004
E14	30	4	2003	1-5-2003
E15	30	1	2004	31-1-2004
E16	30	1	2003	31-1-2003
E17	30	2	2004	Invalid Date

## Class Work 5 (Decision Table Based Testing)

We have learned how to perform 'Decision Table Based Testing' in this class and performed one for 'Upload Screen, where we determined how uploading different types of pictures are tested.

We have performed test cases for each different sizes of pictures and some more for the invalid operations.

### **Here is how I have created the test case for 'Upload Screen':**

#### **Problem:**

Let's now take another example, we consider the decision table and test scenarios for an Upload screen.

There is a dialogue box that will ask the user to upload a file.

Specification:

The file should be in .png format.

The size of the file should be less than 25kb.

The file resolution must be 132\*170px.

Now, let's create a decision table for the upload screen keeping the above specification in mind:

Conditions	1	2	3	4	5	6	7
Format	png	png	png	png	Not png	Not png	Not png
Size	<25kb	<25kb	>=25kb	>=25kb	<25kb	<25kb	>=25kb
Resolution	= 132*170px	!= 132*170px	= 132*170px	!= 132*170px	= 132*170px	!= 132*170px	= 132*170px
Action	Yes	No	No	No	No	No	No

## Class Work 6 (Lab Performance 2)

We have shown our lab performance on ‘One Search’ test cases where we showed how many test cases can be created for ‘One Search’.

**Here is how I have created the test cases for ‘One Search’:**

Project Name	Search functionality with one Search									
Module Name	Search Module									
Created By	Nur Ahmed									
Creation Date	3/11/2024									
Received By	SK									
Received Date	3/11/2024									
Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Pre Condition	Test Data	Expected Result	Actual Result	Status	Comment
TS_001	Verify search functionality with a single search term	TC_search_01	Execute a single search query	1. Open the search page. 2. Enter a search term (e.g., "apple"). 3. Click the search button or press Enter.	Search page is loaded and accessible	Search term: "apple"	Search results containing "apple" are displayed.	Search results containing "apple" are displayed.	Pass	No
TS_001	Verify no results message	TC_search_02	Search for a non-existent term	1. Open the search page. 2. Enter a search term (e.g., "xyz1234"). 3. Click the search button or press Enter.	Search page is loaded and accessible	Search term: "xyz1234"	A message such as "No results found for 'xyz1234'" is displayed.	A message such as "No results found for 'xyz1234'" is displayed.	Pass	No
TS_001	Verify case insensitive search	TC_search_03	Perform a case insensitive search	1. Open the search page. 2. Enter a search term in different cases (e.g., "APPLE" or "apple"). 3. Click the search button or press Enter.	Search page is loaded and accessible	Search term: "APPLE" or "apple"	Search results should include results with "apple" in any case.	Search results should include results with "apple" in any case.	Pass	No
TS_001	Verify search with partial matching	TC_search_04	Search using a partial term	1. Open the search page. 2. Enter a partial search term (e.g., "app"). 3. Click the search button or press Enter.	Search page is loaded and accessible	Search term: "app"	Results containing words that start with or include "app" (e.g., "apple," "application") are displayed.	Results containing words that start with or include "app" (e.g., "apple," "application") are displayed.	Pass	No
TS_001	Verify search with special characters	TC_search_05	Search using special characters	1. Open the search page. 2. Enter a search term with special characters (e.g., "apple!"). 3. Click the search button or press Enter.	Search page is loaded and accessible	Search term: "apple!"	Search functionality handles special characters properly, either showing results or a "No results found" message as appropriate.	Search functionality handles special characters properly, either showing results or a "No results found" message as appropriate.	Pass	No
TS_001	Verify search results order	TC_search_06	Verify relevance of search results	1. Open the search page. 2. Enter a search term (e.g., "apple"). 3. Click the search button or press Enter.	Search page is loaded and accessible	Search term: "apple"	Most relevant results appear at the top (e.g., items with "apple" in the title appear before items with "apple" in the description).	Most relevant results appear at the top (e.g., items with "apple" in the title appear before items with "apple" in the description).	Pass	No
TS_001	Verify empty search term	TC_search_07	Execute search with an empty input	1. Open the search page. 2. Leave the search bar empty. 3. Click the search button or press Enter.	Search page is loaded and accessible	Empty search term	An appropriate message or error (e.g., "Please enter a search term") is displayed.	An appropriate message or error (e.g., "Please enter a search term") is displayed.	Pass	No
TS_001	Verify search performance	TC_search_08	Measure time taken for search results	1. Open the search page. 2. Enter a search term (e.g., "apple"). 3. Click the search button or press Enter. 4. Measure time taken for results to appear	Search page is loaded and accessible	Search term: "apple"	Results appear within the acceptable response time limit (e.g., under 2 seconds).	Results appear within the acceptable response time limit (e.g., under 2 seconds).	Pass	No

## Class Work 7 (Lab Performance 3)

We have performed BVA on 'Valid Age Input' in this class and showed how many test cases can be created for valid age input from the user.

**Here is how I created Boundary Value Analysis on 'Valid Age Input':**

### Lab Performance 3

You are tasked with testing a simple function in a software system that accepts a user input age, where the valid age range is between 18 and 60 (inclusive). The system is supposed to return a message depending on whether the user is eligible for a service or not. Solve it using test case with BVA.

#### Solve:

Let's dive into Boundary Value Analysis (BVA) for testing the function that checks user eligibility based on age input. Here are some test cases derived using BVA:

1. **Lower Boundary:**
  - Age: 18 → Message: "Eligible" (since 18 is the lower limit and should be valid)
2. **Just Below Lower Boundary:**
  - Age: 17 → Message: "Not Eligible" (since 17 is below the valid range)
3. **Just Above Lower Boundary:**
  - Age: 19 → Message: "Eligible" (since 19 is within the valid range)
4. **Upper Boundary:**
  - Age: 60 → Message: "Eligible" (since 60 is the upper limit and should be valid)
5. **Just Below Upper Boundary:**
  - Age: 59 → Message: "Eligible" (since 59 is within the valid range)
6. **Just Above Upper Boundary:**
  - Age: 61 → Message: "Not Eligible" (since 61 is above the valid range)
7. **Nominal Value:**
  - Age: 40 → Message: "Eligible" (since 40 is well within the valid range)
8. **Just Inside Lower Boundary:**
  - Age: 18 → Message: "Eligible" (testing lower boundary again to ensure consistency)
9. **Just Inside Upper Boundary:**
  - Age: 60 → Message: "Eligible" (testing upper boundary again to ensure consistency)

### **Boundary value testing**

Boundary Value Analysis (Valid Range 18 to 60)

Valid	(Min, Min+1, Nominal, Max-1, Max)	Invalid
17	18, 19, 39, 59, 60	61

Test cases for Date program using Boundary Value Analysis.

Test cases	Inputs	Description	Expected Output
BVA1	17	Below the minimum Age	Invalid
BVA2	18	Exact the minimum Age	Valid
BVA3	19	Between the Age range	Valid
BVA4	39	Exact as the nominal Age	Valid
BVA5	59	Between the Age range	Valid
BVA6	60	Exact the maximum Age	Valid
BVA7	61	Above the minimum Age	Invalid