

## Table of Contents

Architecture Diagram to create a new customer in LoanApp .....	2
Presentation Layer: Client side .....	2
Business Logic Layer: Server Side.....	2
Data Layer: Server Side .....	2
System Process Flow Diagram .....	3
Code Coverage exercises .....	4
Exercise 1: .....	4
Exercise 2: .....	5
.....	5
Exercise 3: .....	6
Understanding HTML .....	<b>Error! Bookmark not defined.</b>

## Architecture Diagram to create a new customer in LoanApp

### Presentation Layer: Client side

Client Browser: Google Chrome/Mozilla Firefox/ Opera/ Internet Explorer

Add Customer Page

### Business Logic Layer: Server Side

<<device>>: Web Application server/ Apache Web Server

Customer's  
list

Customer Personal Box

Customer Financial Box

Loan Terms Box

### Data Layer: Server Side

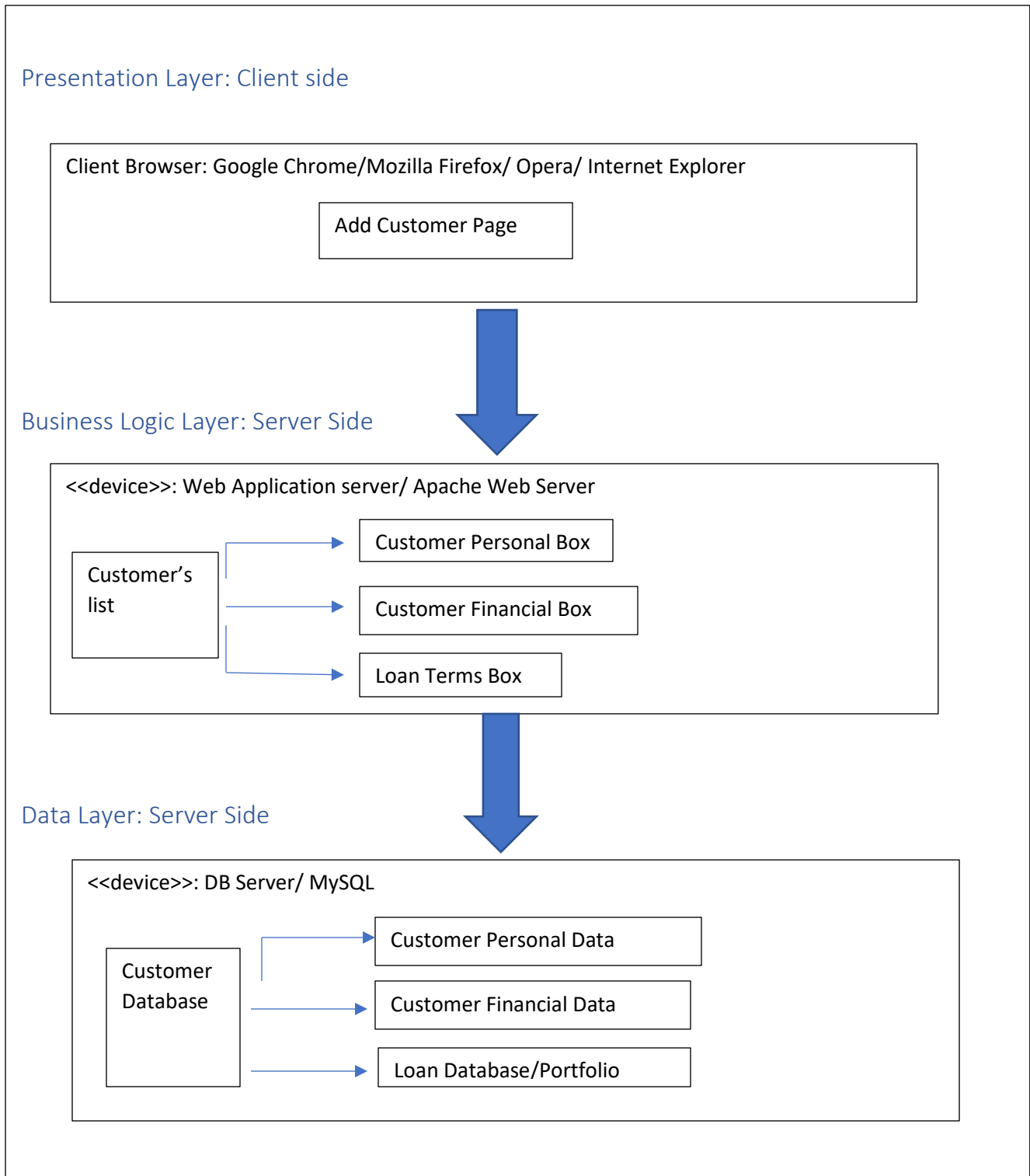
<<device>>: DB Server/ MySQL

Customer  
Database

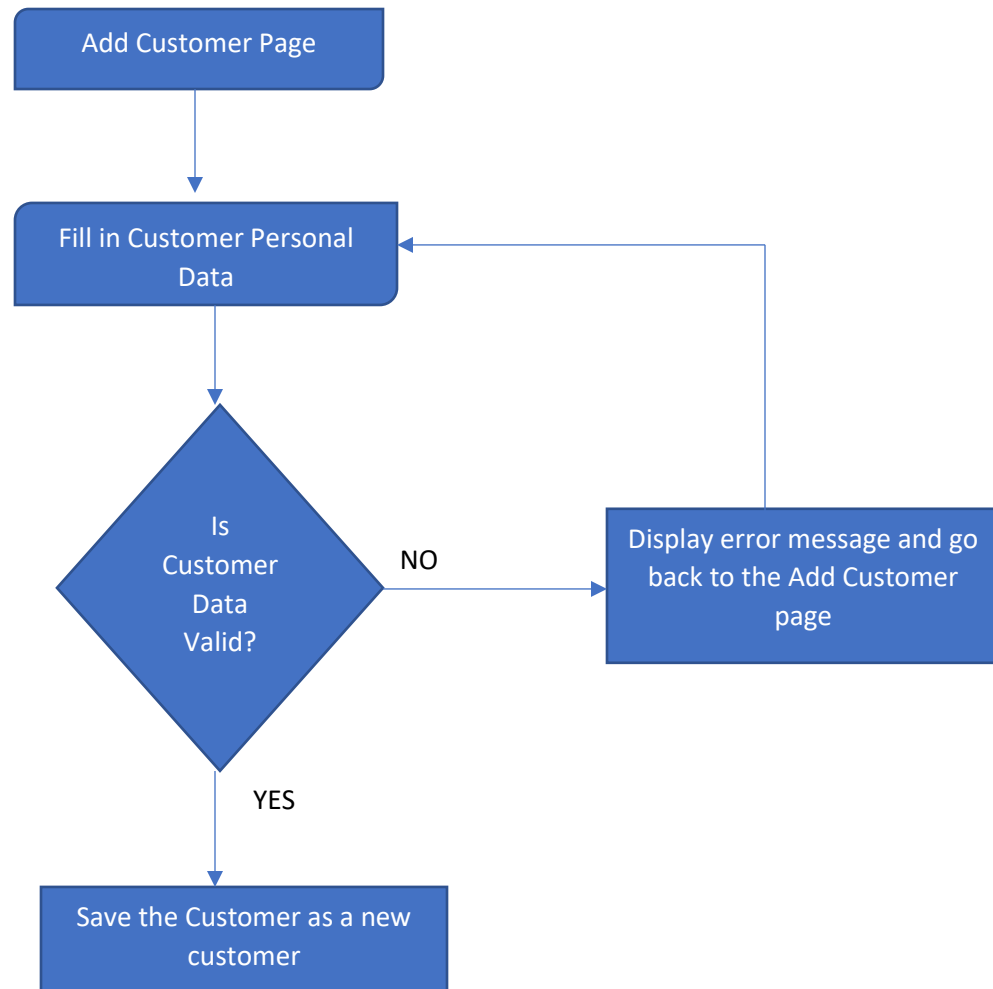
Customer Personal Data

Customer Financial Data

Loan Database/Portfolio

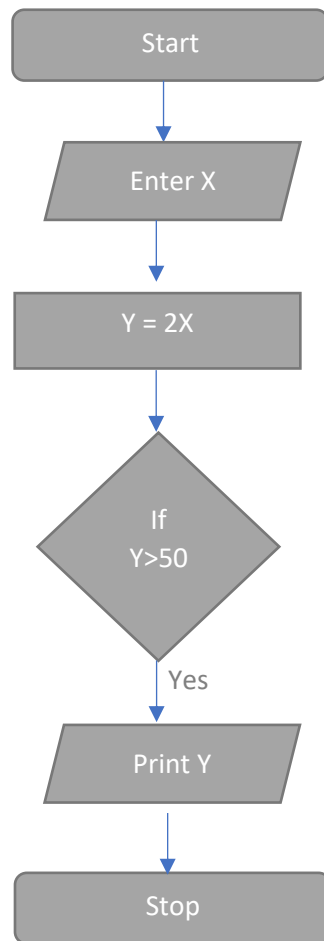


## System Process Flow Diagram



## Code Coverage exercises

### Exercise 1:

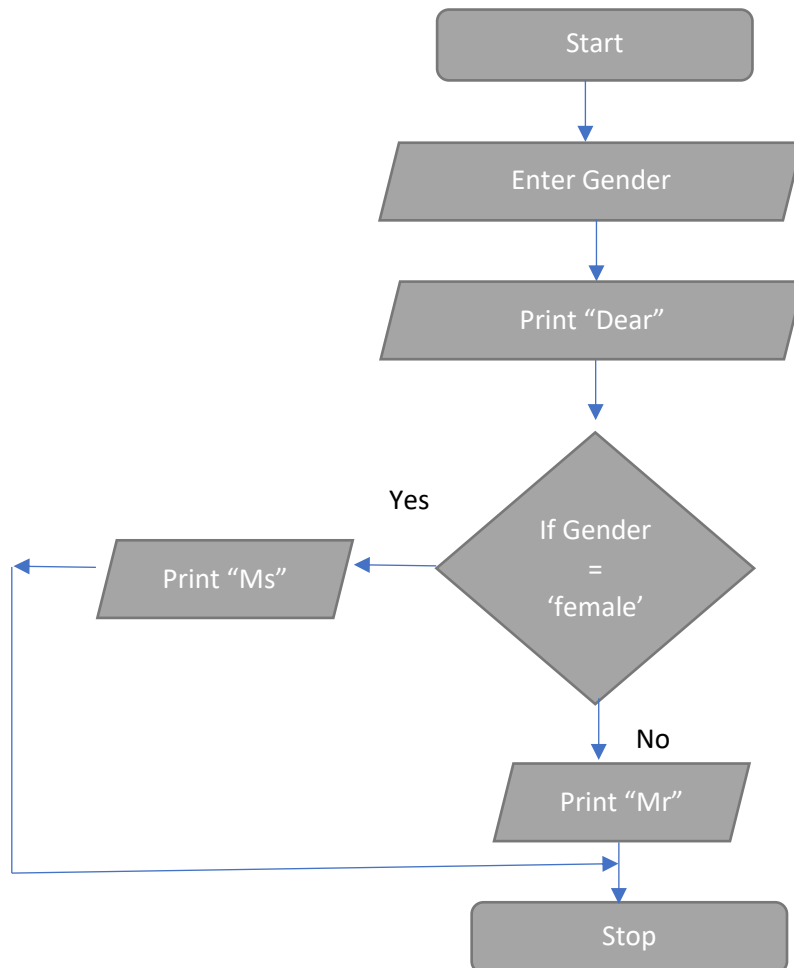


- What is the value of X should be in order to achieve 100% statement coverage in just one test case?

Ans: Value of X larger than 25, such that when it multiplies by 2, the value of Y is large than 50.

So, Only Test Case:  $X > 25$  (True Condition) that covers 100%; prints Y and stops the program.

## Exercise 2:

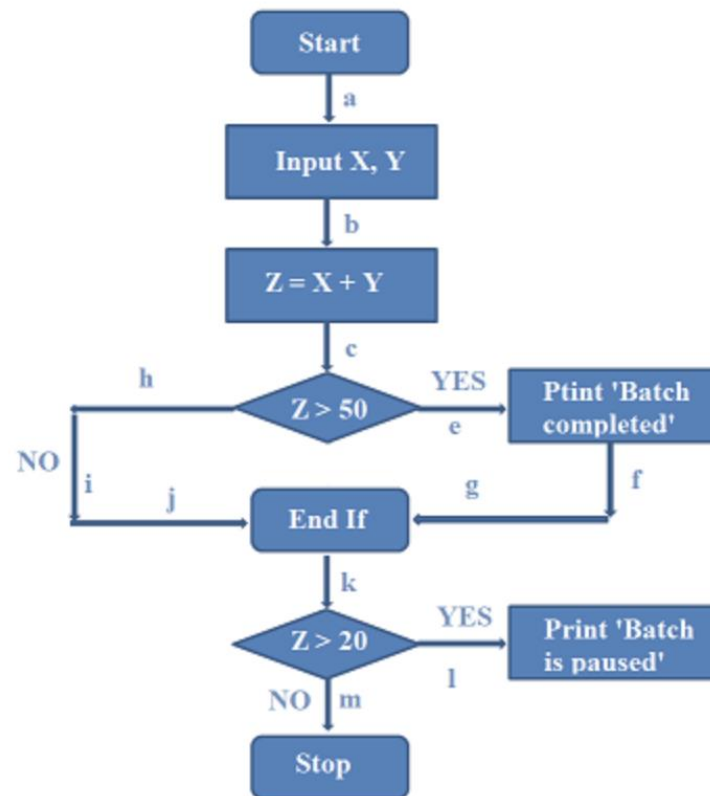


- How many tests are required to achieve 100% decision coverage?

Ans: To achieve 100% decision coverage for the above segment of code we need 2 test cases, such as:

- Test case 1: Gender = female. This case executes that the 'Print "Ms"' statement (True Statement).
- Test case 2: Gender = male. This case executes that the 'Print = "Mr"' statement (False Statement).

## Exercise 3:



Pseudo code for the diagram:

1. Begin
2. Read X, Y
3.  $Z = X + Y$
4. If-then  $Z > 50$
5. Print ("Batch completed")
6. End if
7. If  $Z > 20$
8. Print ("Batch is paused")
9. Else
10. Stop

Ans: To achieve 100% path coverage for the above segment of code we need 2 test cases, such as:

- Test case 1: a,b,c,h,i,j,k,m (False, False).
- Test case 2: a,b,c,e,f,g,k,m (True, False).
- Test case 3: a,b,c,h,i,j,k,l (False, True).
- Test case 4: a,b,c,e,f,g,k,l (True, True).