# **TESTING DOCUMENT**

## 1. Unit testing:

### **1.1. What is unit testing?**

A unit can be a function, a class, a package, or a subsystem. So, the term unit testing refers to the practice of testing such small units of your code.

### **1.2. Unit testing in Java**

The most popular testing framework in Java is JUnit. As this guide is focused to JUnit, another popular testing framework in Java is TestNG.

## 2. JUnit introduction:

JUnit is an open source testing framework which is used to write and run repeatable automated tests, so that we can be ensured that our code works as expected.

**JUnit provides:**

* Assertions for testing expected results.
* Test features for sharing common test data.
* Test suites for easily organizing and running tests.
* Graphical and textual test runners.

**JUnit is used to test:**

* An entire object
* part of an object – a method or some interacting methods
* Interaction between several objects

# **3. TEST CLASSES:**

**3.1 - Registration Test Class:**

package org.sda.model;

import static org.junit.Assert.\*;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import org.sda.model.Registration;

public class RegistrationTest {

//create Registration instance

Registration reg = new Registration("Indu", "Yekkala", "234", "edwadf", 234, "India","Indu12@gmail.com", "234", "234");

@Before

public void setUp() throws Exception {

}

@After

public void tearDown() throws Exception {

}

@Test

public void testGetFirstName() {

String expected = "Indu";

String actual = reg.getFirstName();

assertEquals(expected,actual);

//fail("Not yet implemented");

}

@Test

public void testGetLastName() {

String expected = "Yekkala";

String actual = reg.getLastName();

assertTrue(expected.equals(actual));

//fail("Not yet implemented");

}

@Test

public void testGetPhonenumber() {

String expected = "234";

String actual = reg.getPhonenumber();

assertEquals(expected,actual);

//fail("Not yet implemented");

}

@Test

public void testGetAddress() {

String expected = "edwadf";

String actual = reg.getAddress();

assertTrue(expected.equals(actual));

//fail("Not yet implemented");

}

@Test

public void testGetZipcode() {

int expected = -1;

int actual = reg.getZipcode();

boolean value=expected<0;

if(value==false)

assertFalse(value);

else

assertEquals(expected,actual);

fail("Not yet implemented");

}

@Test

public void testGetEmail() {

String expected = "Indu12@gmail.com";

String actual = reg.getEmail();

assertTrue(expected.equals(actual));

//fail("Not yet implemented");

}

@Test

public void testGetPwd1() {

String expected = "234";

String actual = reg.getPwd1();

assertEquals(expected,actual);

//fail("Not yet implemented");

}

@Test

public void testGetPwd2() {

String expected = "234";

String actual = reg.getPwd2();

assertTrue (expected.equals(actual));

//fail("Not yet implemented");

}

@Test

public void testGetCountry() {

String expected = "India";

String actual = reg.getCountry();

assertEquals (expected, actual);

//fail("Not yet implemented");

}

}

**3.2 - Login Test Class:**

package org.sda.model;

import static org.junit.Assert.\*;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

public class LoginTest {

//create Registration instance

Registration reg = new Registration ("Indu12@gmail.com", "234");

@Before

public void setUp() throws Exception {

}

@After

public void tearDown() throws Exception {

}

@Test

public void testGetEmail() {

String expected = "Indu12@gmail.com";

String actual = reg.getEmail();

assertEquals(expected, actual);

assertNotEquals(expected, actual);

//fail("Not yet implemented");

}

@Test

public void testGetPwd1() {

String expected = "234";

String actual = reg.getPwd1();

assertTrue(expected.equals(actual));

assertNotEquals(expected, actual);

//fail("Not yet implemented");

}

}