

PFA day 5 test case,

Process	Steps To Execute	Expected Results
Edit Users	1. Navigate to Systems section, shipping Option 2. Choose Users 3. Edit a new details and Save	Validate the presence of the modified Users

Here User tab is not inside Shipping section so I have used my test cases like as below

System--- Users—edit user details----- save --- Verify

Main Method :

```
package com.ibm.test;

import static org.testng.Assert.assertTrue;

import java.io.File;
import java.io.IOException;
import java.util.Date;
import java.util.HashMap;
import java.util.concurrent.TimeUnit;

import org.apache.commons.io.FileUtils;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.Assert;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.BeforeSuite;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Test;
import com.ibm.pages.AdminPage;
import com.ibm.pages.LoginPage;
import com.ibm.pages.MarketingPage;
import com.ibm.pages.ReturnSatusPage;
import com.ibm.pages.UserPage;
import com.ibm.utilities.ExcelUtil;
import com.ibm.utilities.PropertiesFileHandler;

public class BaseTest {

    private static final String Userconfrimpasword = null;
    // initializing the driver, wait , profilehandler and HashMap
```

```

private static final String Userpassword = null;

WebDriver driver;
WebDriverWait wait;
PropertiesFileHandler propFileHandler;
HashMap<String, String> data;

// beforeSuite method is used to declare the statement that are going to
execute
// before the running of testcase

@BeforeSuite

public void preSetForTest() throws IOException {

    String file = "./TestData/data.properties";
    propFileHandler = new PropertiesFileHandler();
    data = propFileHandler.getPropertiesAsMap(file);

}

// beforeMethod is used to declare the statement that are going to run at the
// starting of all test cases
@BeforeMethod

public void Initialization() {

    System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
    System.setProperty("webdriver.gecko.driver", "./driver/geckodriver.exe");
    driver = new ChromeDriver();// opening the chrome browser
    //driver=new FirefoxDriver();
    wait = new WebDriverWait(driver, 60);// giving explicit wait
    driver.manage().window().maximize();// maximizing the window
    driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS); //
declaring implicit wait

}

// afterMethod is used to run once all the test cases are run

@AfterMethod
public void closeBrowser() {
    driver.quit();// close the browser
}

@Test
public void testcase5() throws InterruptedException, IOException {

    String url = data.get("url");
    String username = data.get("user");
    String password = data.get("password");

```

```

        String updatepassword= data.get("Userpassword");
        String confirmpassword= data.get("Userconfirmpasword");
        String expectedsavemsg= data.get("savemsg");
        // open the url https://atozgroceries.com/admin
        driver.get(url);

        // calling login page method to enter the user name, password and click on log
        // in button
        LoginPage login = new LoginPage(driver, wait);
        login.enterEmailAddress(username);
        login.enterPassword(password);
        login.clickOnLogin();

//Calling return status page to click on system tab
        ReturnSatusPage rstatus = new ReturnSatusPage(driver, wait);

        rstatus.clickOnSystem();

// Calling user page on main method to perform the operation on user page
        UserPage userpg = new UserPage(driver, wait);
        userpg.clickOnUser();
        userpg.clickonAction();
        userpg.clickOnEdit();
        String Upass=userpg.updatePassword(confirmpassword);
        String Confpass=userpg.updateConfirmPassword(confirmpassword);

// in order to click on save I am calling save method that is declare in admin page
        AdminPage admin = new AdminPage(driver, wait);
        admin.clickOnSave();
        Thread.sleep(10000);
// Assertion to verify the text that comes after saving the text on user page
        String actualsavemsg = userpg.getTextToVerifySaveMsg();
        System.out.println(actualsavemsg);
        System.out.println(expectedsavemsg);

        Assert.assertTrue(actualsavemsg.contains(expectedsavemsg), "Assertion
on saveing user data");

        // code to take screen shot
        TakesScreenshot ts = (TakesScreenshot) driver;
        File file = ts.getScreenshotAs(OutputType.FILE);
        Date date = new Date();
        String currentDate = date.toString().replace(":", "-");
        FileUtils.copyFile(file, new File("./screenshots/Error_" +
currentDate + ".png"));

    }

```

```
}
```

User page code :

```
package com.ibm.pages;

import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class UserPage{

    WebDriverWait wait;
    WebDriver driver;
    JavascriptExecutor js;

    @FindBy(xpath="//*[contains(text(),' Users')]")
    WebElement userEle;

    @FindBy(xpath="//*[@id=\"dataTableExample2\"]/tbody/tr[3]/td[5]/div/button")
    WebElement actionEle;

    @FindBy(xpath="//*[@id=\"dataTableExample2\"]/tbody/tr[3]/td[5]/div/ul/li[1]/a")
    WebElement editEle;

    @FindBy(xpath="//input[@name='pword']")
    WebElement passwordEle;
    @FindBy(xpath="//input[@name='cpword']")
    WebElement confirmPassEle;

    @FindBy(xpath="//*[@id=\"page-wrapper\"]/div/form/div[1]/button")
    WebElement saveEle;

    @FindBy(xpath="//*[contains(text(),' Success: You have successfully updated user!')]")
    WebElement savemsg;

    //constructor
    public UserPage(WebDriver driver, WebDriverWait wait) {
        PageFactory.initElements(driver, this);
        this.driver = driver;
        this.wait = wait;
    }
}
```

```

        js = (JavascriptExecutor) driver;
    }

    // method to click on User tab
    public void clickOnUser() {

        userEle.click();

    }

    // method to click on Action on user page
    public void clickonAction() {
        actionEle.click();

    }

    // method to click on Edit on user page
    public void clickOnEdit() {

        editEle.click();

    }

    public String updatePassword(String Userpassword) {
        passwordEle.sendKeys(Userpassword);
        return(Userpassword);

    }

    public String updateConfirmPassword(String Userconfirmpassword) {
        confirmPassEle.sendKeys(Userconfirmpassword);
        return Userconfirmpassword;

    }

    public void clickonSave() {

        saveEle.click();

    }

    // method to verify Assertion
    public String getTextToverifySaveMsg() {

        wait.until(ExpectedConditions.presenceOfAllElementsLocatedBy(By.xpath("//*[contains(text(),' Success: You have successfully updated user!')]")));

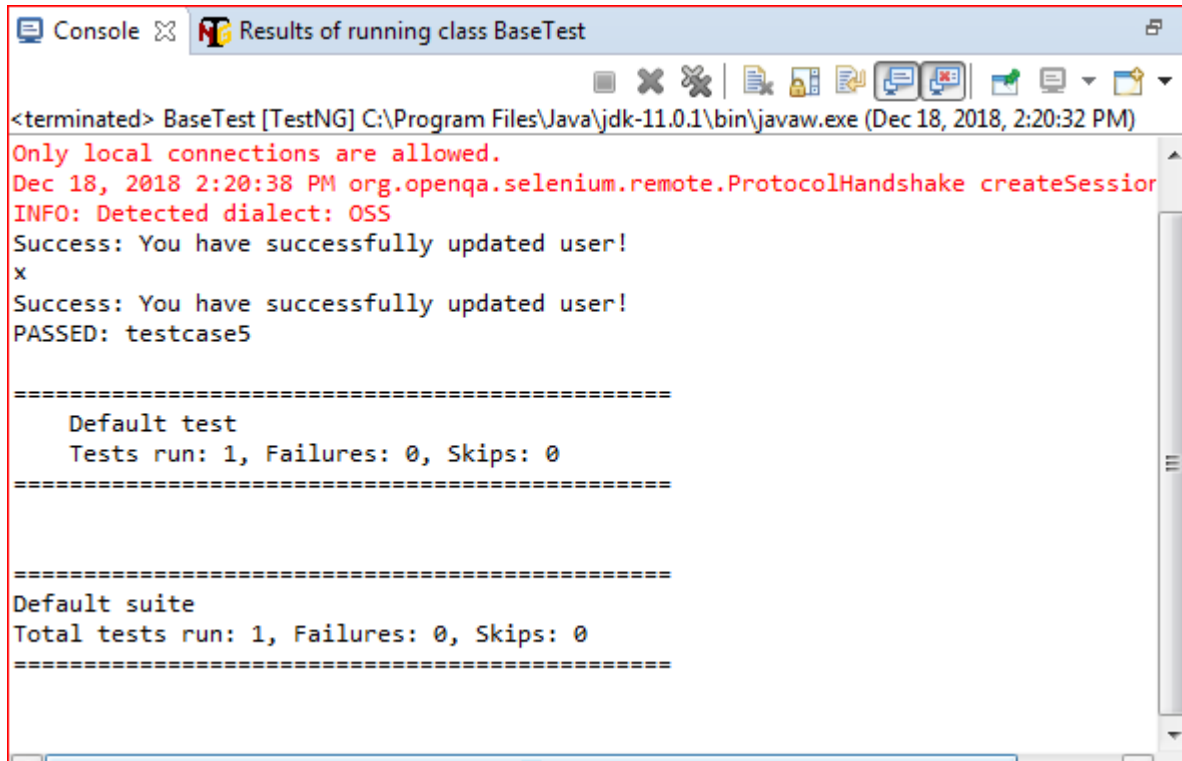
        String msg= savemsg.getText();
        return msg;

    }

}

```

Screen Shot from console

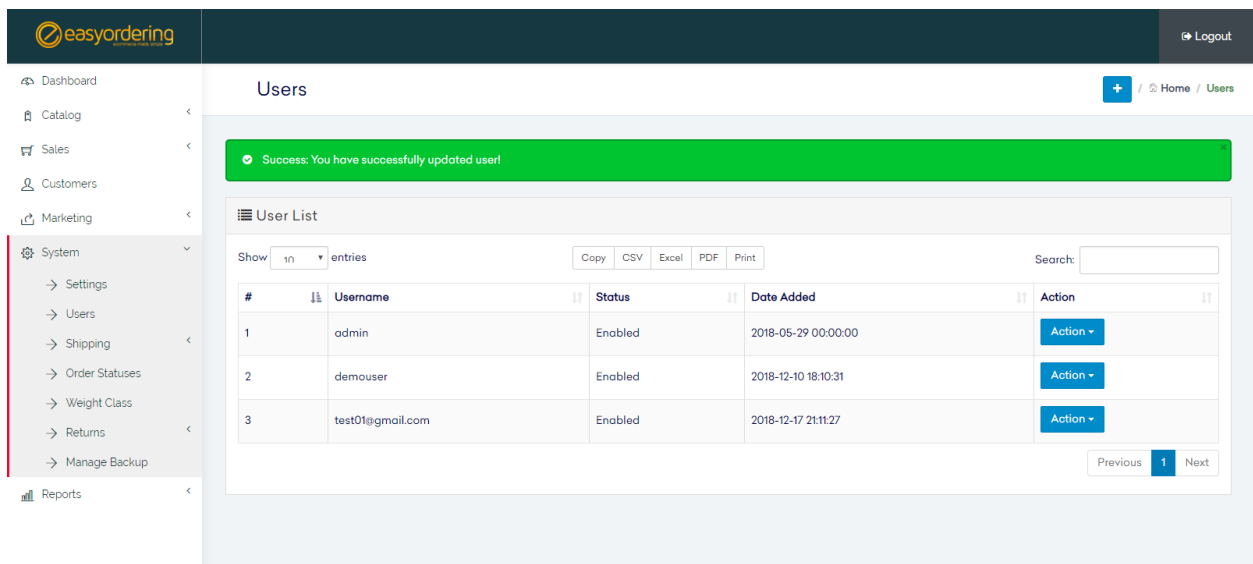


The screenshot shows a Java console window titled "Console" and "Results of running class BaseTest". The output text is as follows:

```
<terminated> BaseTest [TestNG] C:\Program Files\Java\jdk-11.0.1\bin\javaw.exe (Dec 18, 2018, 2:20:32 PM)
Only local connections are allowed.
Dec 18, 2018 2:20:38 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS
Success: You have successfully updated user!
x
Success: You have successfully updated user!
PASSED: testcase5

=====
      Default test
      Tests run: 1, Failures: 0, Skips: 0
=====

=====
Default suite
Total tests run: 1, Failures: 0, Skips: 0
=====
```



The screenshot shows the "easyordering" web application interface. The left sidebar contains a navigation menu with items: Dashboard, Catalog, Sales, Customers, Marketing, System (expanded), and Reports. The "System" menu is expanded, showing sub-items: Settings, Users, Shipping, Order Statuses, Weight Class, Returns, and Manage Backup. The main content area is titled "Users" and displays a success message: "Success: You have successfully updated user!". Below this is a "User List" table with columns: #, Username, Status, Date Added, and Action. The table contains three rows of user data. At the bottom right of the table, there are "Previous", "1", and "Next" pagination controls.

#	Username	Status	Date Added	Action
1	admin	Enabled	2018-05-29 00:00:00	Action -
2	demouser	Enabled	2018-12-10 18:10:31	Action -
3	test01@gmail.com	Enabled	2018-12-17 21:11:27	Action -