Kemble Pillay

A227853

Part 2

1a)

UserStory 1

- As a User I want to be able to login So that I will be able to purchase goods online

  Acceptance criteria

  *Username needs to be a valid email address
  *password must be at least 6 characters' long
  *password must contain alphanumeric Characters
  *password must contain a capital letter

  Scenario

  Scenario 1: Login with valid credentials

  Given that the user has a registered account with amazon
  When the user logs in with valid credentials
  Then the user is redirected to their homepage

  Scenario 2: Login with invalid credentials

  Given that the user has a registered account with amazon
  When the user logs in with invalid credentials
  Then the user is prompt with an invalid login message

UserStory 2

- As a user I want to be able to Create my user profile so that I can be able to manage my personal Information

Acceptance criteria

*Name field should not contain special characters

*Surname field should not contain special characters

*ID number should be 13 characters for South African citizens

Scenario

Scenario 1: User creates profile with valid information

Given that the user is on the create profile page

When the user enters their correct personal information

Then the users profile will be created

Scenario 2: User creates profile with invalid information

Given that the user is on the create profile page

When the user enters incorrect personal information

Then the user will be prompt with an invalid information message

UserStory 3

As a user I want to search for products so that I will be able to find the products that I am interested in

Acceptance criteria

*valid product name

*Search field categorizes products

*search field cannot be empty

Scenario

Scenario 1: User searches for product with valid product name

Given that the user is on the search page

When the user enters the product they would like to search for

And clicks the search button

Then a list of products that match the keyword searched

1b) Test Cases

TCID	Objective	Prerequisite	Input data	Description steps	Expected results	Actual results
Test Scenario: Login						
TC01	log into the system successfully	AUT	1. Username: kemble@gmail.com 2. Password: kem1234	1. Enter Username as mentioned 2. Enter Password as mentioned 3. Click 'OK' button	Profile of the customer is displayed	
TC02	Error Message	AUT	1. Username: 123 2. Password: kem1234	1. Enter Username as mentioned 2. Enter Password as mentioned 3. Click 'OK' button	Error message displayed: Incorrect username. Username must be a valid email address	
TC03	Error Message	AUT	1. Username: Kemble@gmail.com 2. Password: kem	1. Enter Username as mentioned 2. Enter Password as mentioned 3. Click 'OK' button	Error message: incorrect Password. password must be at least 6 characters	
TC04	Error Message	AUT	1. Username: Kemble 2. Password: kem	1. Enter Username as mentioned 2. Enter Password as mentioned 3. Click 'OK' button	Error message: incorrect user name and Password.	

TCID	Objective	Prerequisite	Input data	Description steps	Expected results	Actual results
Test Scenario: Create user profile						
TC05	Create user profile successfully	AUT-Profile page	Name: Kemble Surname: Pillay Phone number:07955612 Address: 344 westville avenue	1.Enter name as mentioned 2. Enter surname as mentioned	**The user profile is successfully created**	

				3. Enter phone number as mentioned 4. Enter address as mentioned 5.Click create profile		
TC06	Create user profile unsuccessfully	AUT-Profile page	Name: Kemble Phone number:07955612 Address: 344 westville avenue	1.Enter name as mentioned 3. Enter phone number as mentioned 4. Enter address as mentioned 5.Click create profile	**Error message:The user profile is not created. Surname is mandatory**	

TCID	Objective	Prerequisite	Input data	Description steps	Expected results	Actual results
Test Scenario: Search for product						
TC07	Search for valid product	AUT-Search page	Product Name: shoes	1.Enter product name as mentioned 2.Click search	**List of products that match the keyword is displayed**	
TC08	Search for invalid product	AUT-Search page	Product Name: dgdsgdsa	1.Enter product name as mentioned 2.Click search	**Error match: No product matches**	

1C)

```
package test;

import static org.junit.Assert.*;

import org.junit.Test;

import junit.framework.Assert;

public class testLogin {

	@Test

		public static void main(String[] args)
```

```
{
 TestLogin();
}
// arguments are passed using the text field below this editor
public static void TestLogin()
{

  Login login= new Login();

  String Username = "Kemble@gmail.com";
  String Password = "kemble12";

  login.TestLogin (Username, Password);

  boolean result = login.TestLogin (Username, Password);

  Assert.assertEquals("True", result);

 }
}
```

```
public static void TestLogin2()
    {

      Login login2= new Login();

      String Username = "Kemble ";
      String Password = "kemble12";

      Login2.TestLogin (Username, Password);

      boolean result = login2.TestLogin (Username, Password);

      Assert.assertEquals("false", result);

    }
public static void TestLogin3()
    {

      Login login3= new Login();

      String Username = "Kemble@gmail.com ";
      String Password = "kem";

      Login.TestLogin (Username, Password);
```

```
            boolean result = login3.TestLogin (Username, Password);

            Assert.assertEquals("false", result);

          }
public static void TestLogin4()
          {

            Login login4= new Login();

            String Username = "Kemble ";
            String Password = "kemb";

            login.TestLogin (Username, Password);

            boolean result = login4.TestLogin (Username, Password);

            Assert.assertEquals("False", result);

          }
```

1D)Implementation

```
public class Login {
       private String  usr;
       Private String pass
        public boolean TestLogin (String usr, String pass)
         {

           try
           {
             if (usr.length() >= 4 && usr.length() <= 15 && pass.length() >= 6 &&
pass.length() <= 10 && usr.contains('@'))
             {
               return true;
             }
             else
             {
               return false;
             }
           }

           catch (Exception ex)
           {
             return false;
           }
         }

}
```

1C) Create user profile Unit tests

```
	package test;

import static org.junit.Assert.*;

import org.junit.Test;

import junit.framework.Assert;

public class testProfile {

	@Test
		public static void main(String[] args)
	  {
	   TestProfile ();
	  }
	  // arguments are passed using the text field below this editor
	  public static void TestProfile()
	  {

	    Profile profile= new Profile();

	    String name = "Kemble ";
	    String surname = "pillay";
	    String phone = "0795561234";
	    String id = "pillay";


	    Profile.TestProfile (name, surname,phone,id);

	    string result = Profile.TestProfile (name, surname,phone,id);

	    Assert.assertEquals("Profile Created", result);

	  }
public static void TestProfile2()
	  {

	    Profile profile2= new Profile();


	    String surname = "pillay";
	    String phone = "0795561234";
	    String id = "pillay";


	    Profile.TestProfile (surname,phone,id);

	    string result = Profile.TestProfile (surname,phone,id);

	    Assert.assertEquals("Profile not Created", result);

	  }
```

```
}
```

1D)Implementation Create profile

```
public class Profile {
	private String  name;
	Private String surname;
	private String  phone;
	Private String id;

	public String TestProfile(String name, String surname,String phone,String
	id)
	  {

	    try
	    {

	        Return ”Profile Created”;
	      }

	    }

	    catch (Exception ex)
	    {
	      return “Profile not created”;
	    }
	  }

}
```

1C)Search unit test

```
package test;

import static org.junit.Assert.*;

import org.junit.Test;

import junit.framework.Assert;

public class testSearch{

	@Test
		public static void main(String[] args)
	  {
	   TestSearch();
	  }
	  // arguments are passed using the text field below this editor
	  public static void TestSearch()
	  {
```

```
        Search search= new Search();

        String keyword = "shoes";


       search.TestSearch (keyword);

        boolean result = Search.TestSearch (keyword);

        Assert.assertEquals("True", result);

      }


    public static void TestSearch()
      {

        Search search= new Search();

        String keyword = "jshdjsah";


       search.TestSearch (keyword);

        boolean result = Search.TestSearch (keyword);

        Assert.assertEquals("false", result);

      }


    }
```

1D)Implementation for search

```
public class Search {
       private String  keyword;

       public String TestSearch(String keyword)
         {

           try
           {

               Return true;
             }

           }

           catch (Exception ex)
           {
             return false”;
```

```
            }
        }

}
```