

CA OOSE | Part B

Sérgio Vinício da Silva Oliveira
13 Dec 2023

The link also allows access to this content
<https://github.com/sergio-oliveira-br/OOSE>

Note

First, I would like to mention that Nickolas Franco gave up this course. For this reason, the development of this was carried out only by me.

1 / 17

1. Development of the class in Java	1
2. Full test of the class	9
3. Details of the methodology followed.....	16


Extra

4. Codes	17
----------------	----

1 / 17

1. Development of the class in Java.

The classes chosen for development are related to the Login attribute, executed by the garden centre cashier.

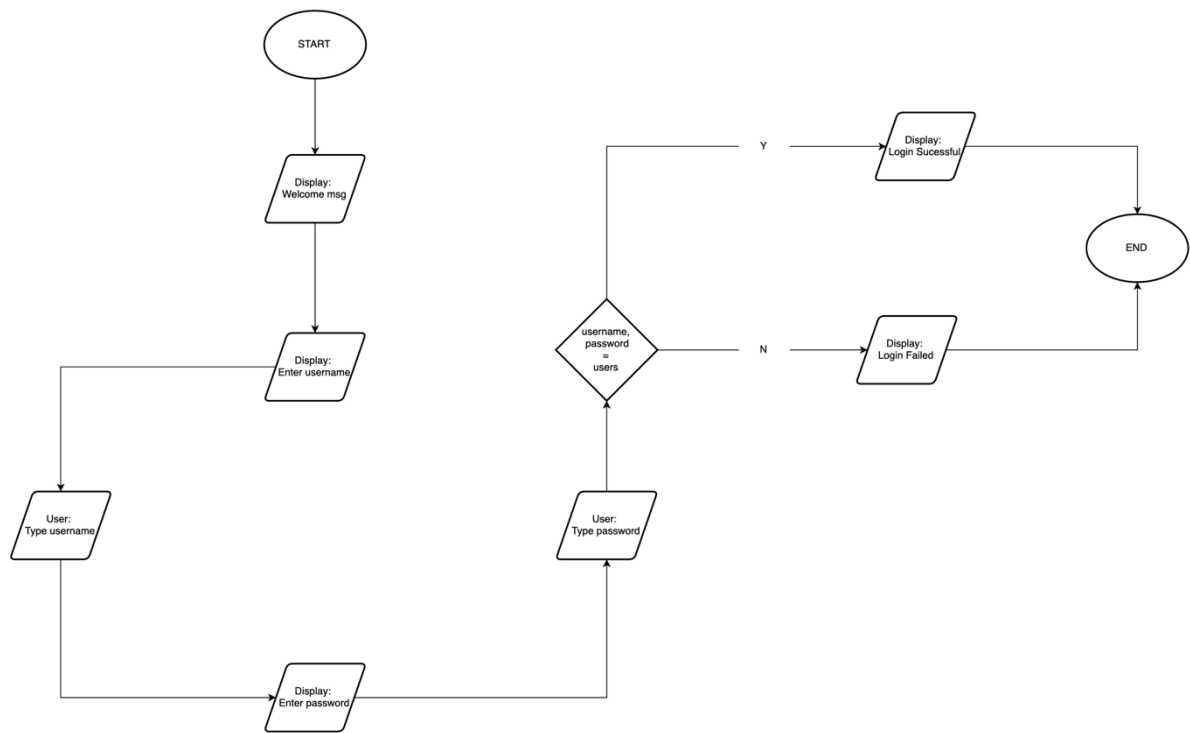
 Cashier
employee_id
login () create_sales () search_item () end_sale () refund ()

As a main flow, we have the user carrying his username and password in the system registration. In possession of these details, the system would allow the user to access the sales system.

When contingency mode is enabled, an alternate flow is enabled when an administrator key is present.

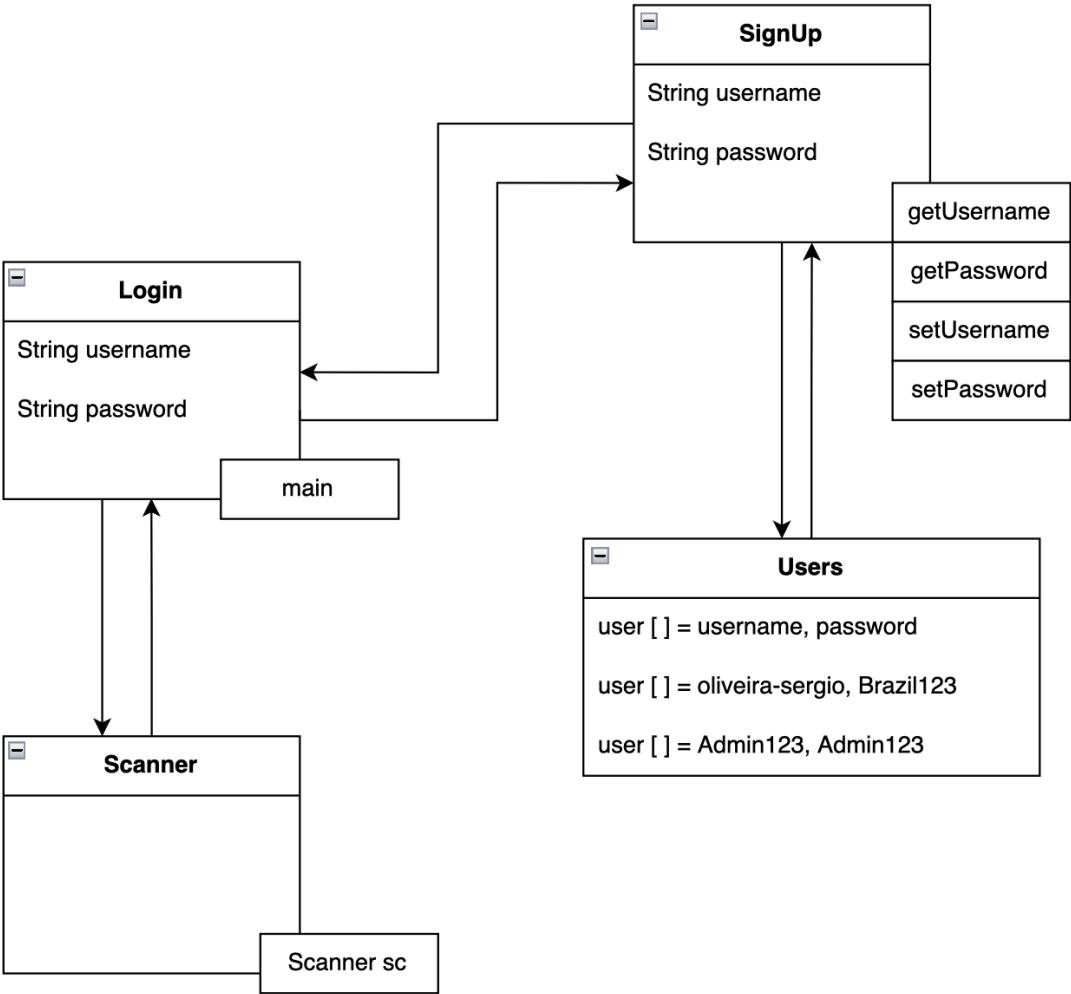
In the system's exception flow, any attempt not registered by the users is denied access.

The flowchart below can be used to represent the reasoning behind the program in question.

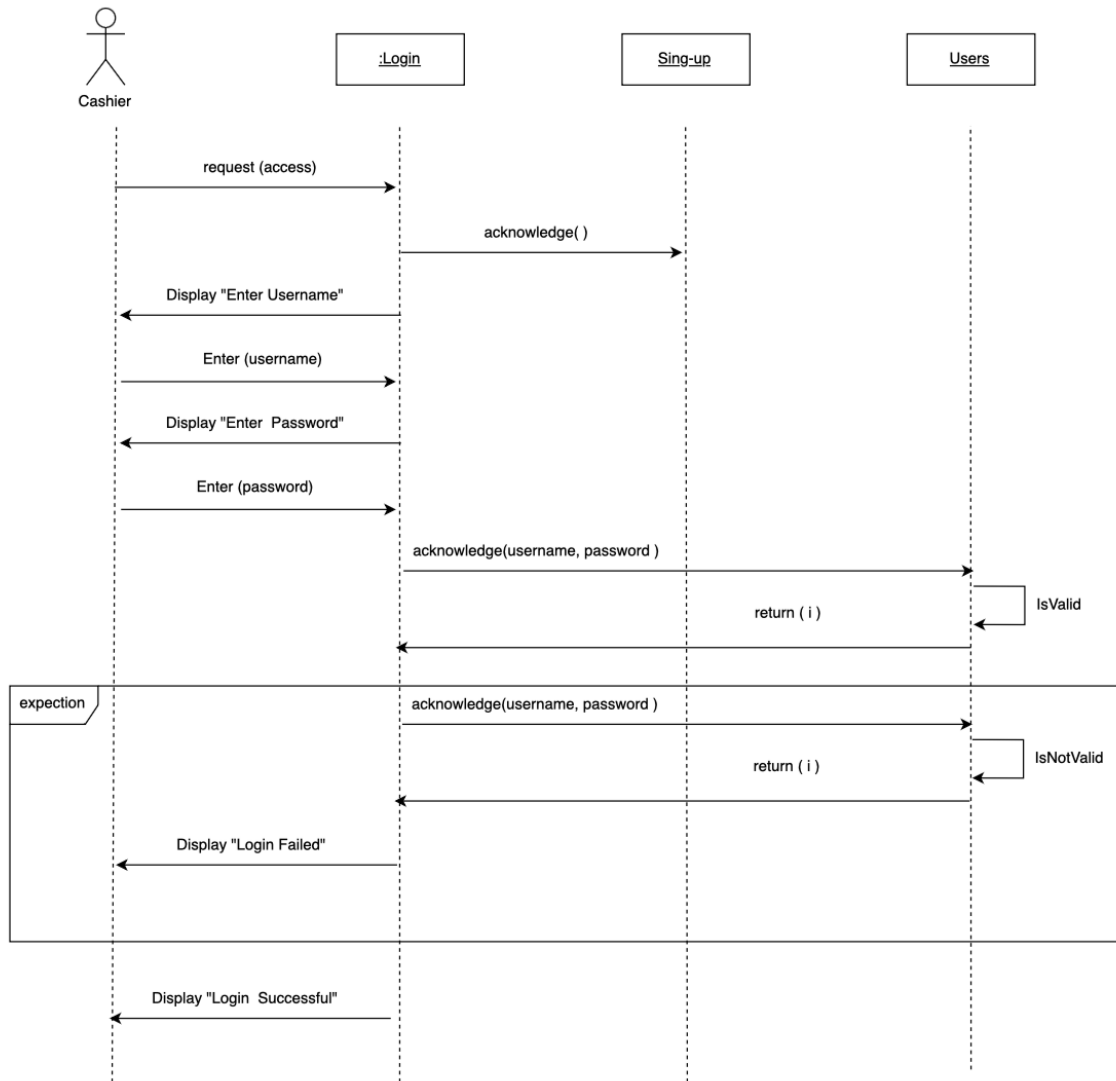


The class diagram, description of input, main process and output are part of the programming that has been performed.

Login ()		
Input	Process	Output
String username	SignedUp [] = username, password	Display: Login Successful
String password		Display: Login Failed
Row 3		



Communication Diagram

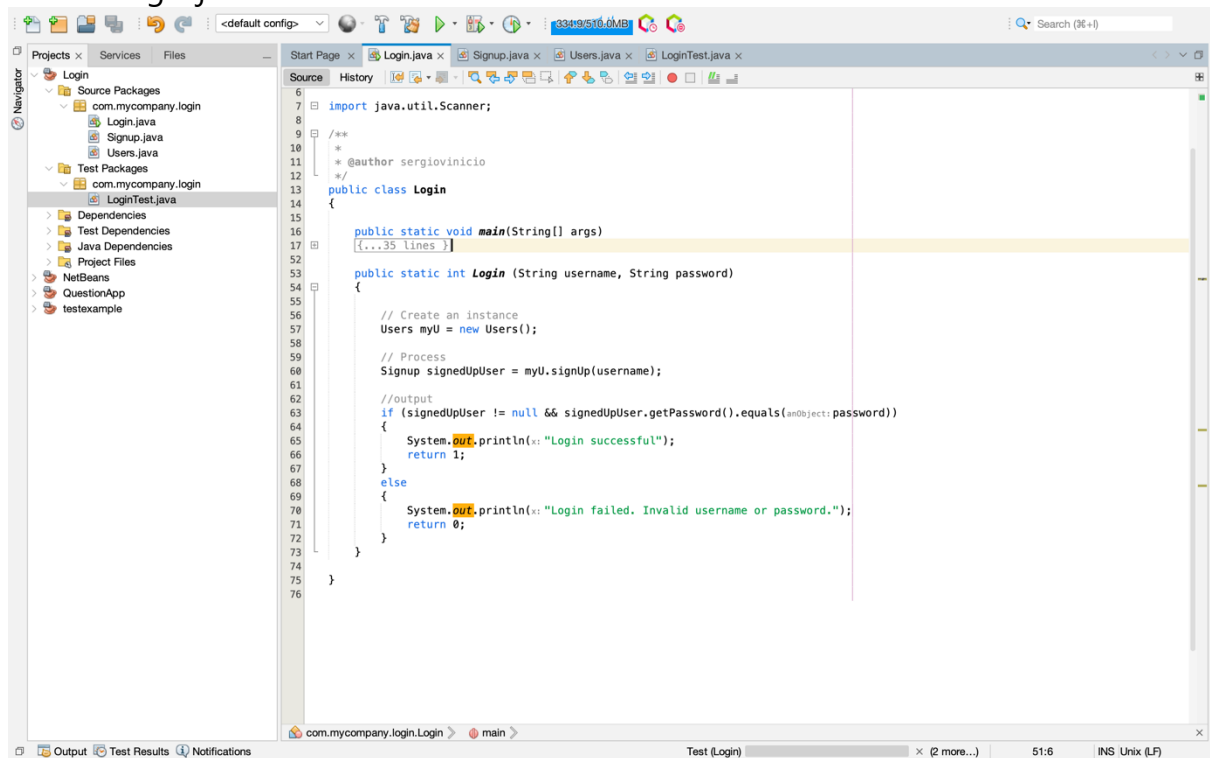


Verify system functionality by examining evidence.

Keep in mind that the link holds all the evidence found in this, including the codes created.

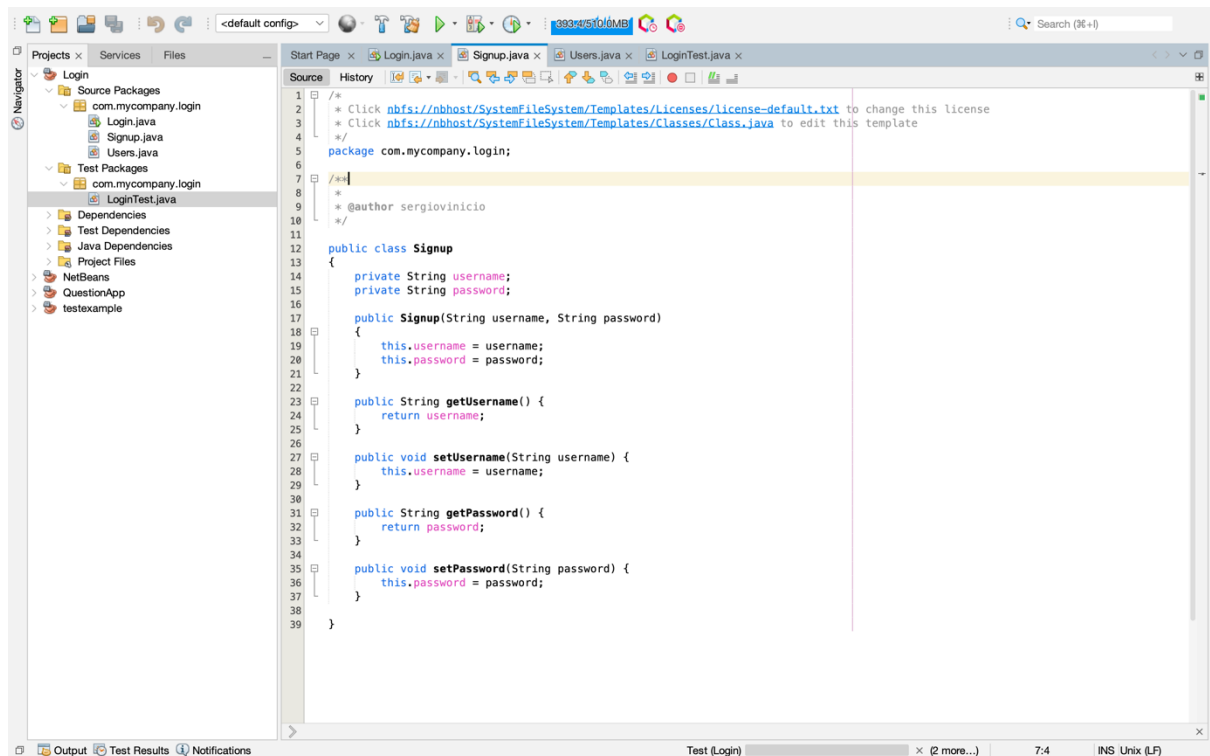
<https://github.com/sergio-oliveira-br/OOSE>

Class – Login.java

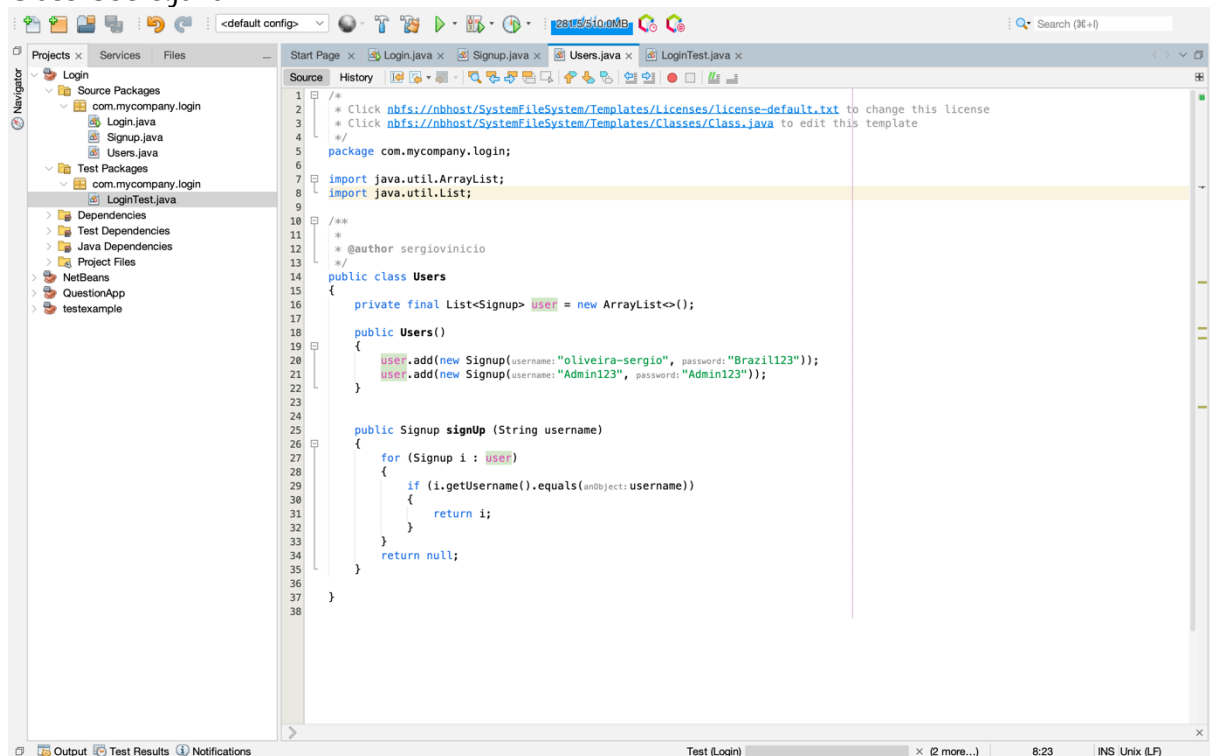


```
6  import java.util.Scanner;
7
8
9
10 /**
11  * @author sergiovinicio
12  */
13 public class Login
14 {
15
16     public static void main(String[] args)
17     {...35 lines ...}
18
19     public static int Login (String username, String password)
20     {
21         // Create an instance
22         Users myU = new Users();
23
24         // Process
25         Signup signedUpUser = myU.signup(username);
26
27         //output
28         if (signedUpUser != null && signedUpUser.getPassword().equals(password))
29         {
30             System.out.println("Login successful");
31             return 1;
32         }
33         else
34         {
35             System.out.println("Login failed. Invalid username or password.");
36             return 0;
37         }
38     }
39 }
```

Class Signup.java

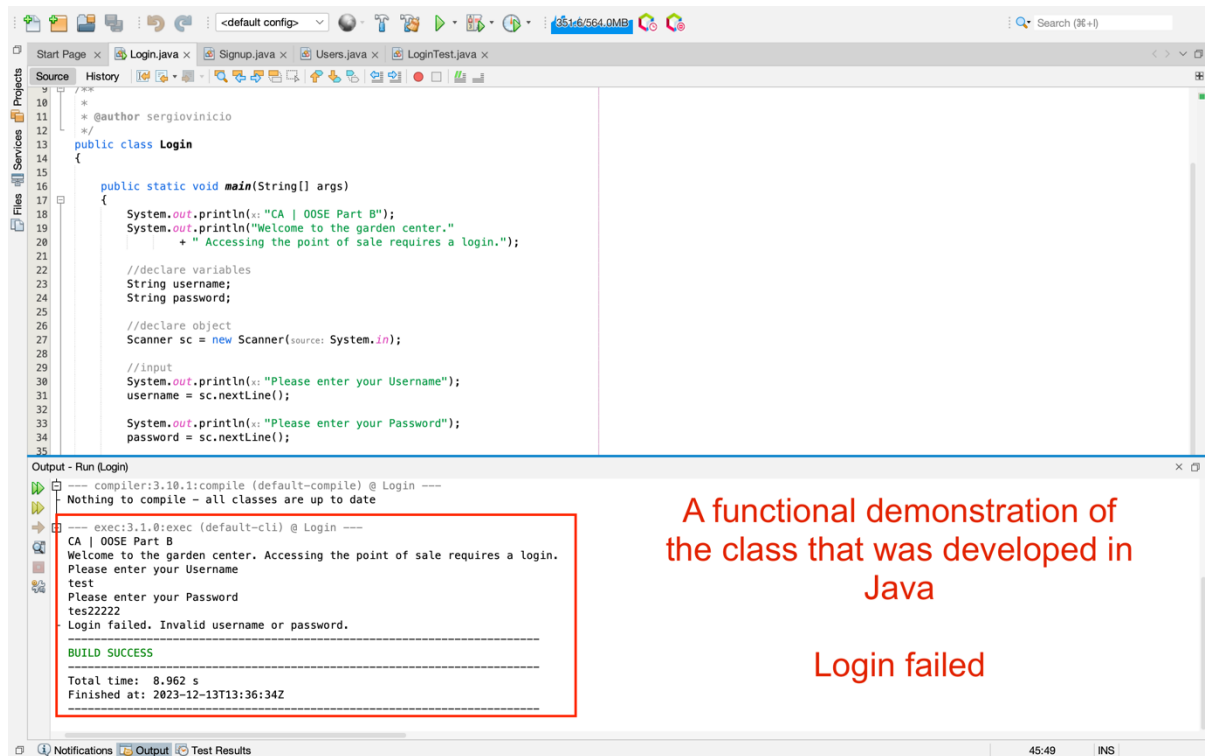


Class Users.java

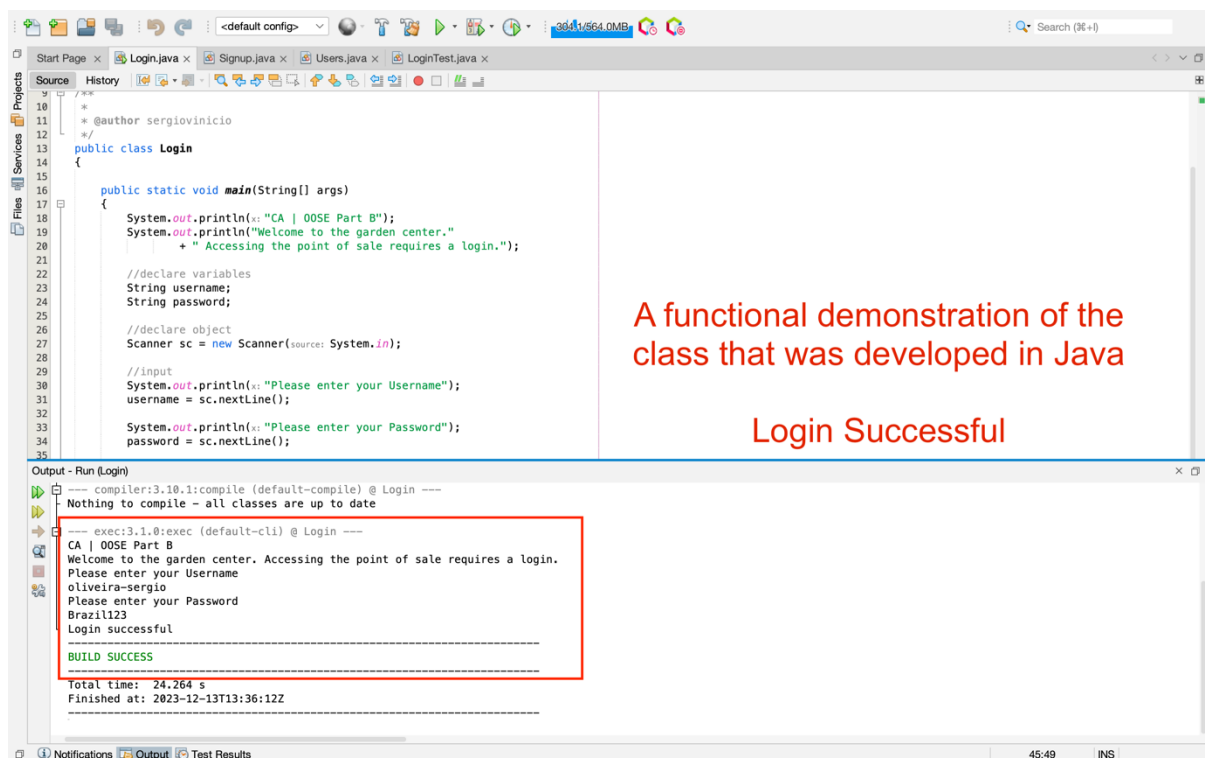


Login.java

Program executed with incorrect credentials.



Login.java
Program executed with right credentials.



2. Full test of the class.

The classes chosen for development are related to the Login attribute, executed by the garden centre cashier.

The contract below contains the necessary details.

Contract: Login	
Name	Login
Actor	Cashier
Description	Access to the system is being requested by the user
Trigger	The sales process needs to be started
Responsibilities	Enable the sales process
Precondition	System on / System updated
Postcondition	Login Successful
Exception	Login failed. Invalid username or password
Type	System

The creation of the login method requires each user to have a username and password to ensure their roles are only accessed.

This project will not address the process of creating users.

The proposal here outlines the system's capabilities, testing only the individual components of the program. In other words, the approach is to verify the login functionality, which will determine whether or not access to the system is allowed.

The policy adopted addresses testing all possible combinations, including as guidelines correct and incorrect entries, being user input fundamental. Moreover, two users have been created to verify the extension of registered users.

To put it simply, the tests were:

Name:		testLoginBlank
Input	username:	" "
	password:	" "
Output	ExpResult:	0
	Result:	0
Status:		Passed
Comments:	To validate null entries	

Name:		testLoginWrongPassword
Input	username:	oliveira-sergio
	password:	BraziL123
Output	ExpResult:	0
	Result:	0
Status:		Passed
Comments:	To validate incorrect entries.Note that the 'L' is written in capital letters, but it's incorrect.	

Name:		testLoginAllRightSergio
Input	username:	oliveira-sergio
	password:	Brazil123
Output	ExpResult:	1
	Result:	1
Status:		Passed
Comments:	To validate correct entries	

Name:		testLoginAllRightAdmin
Input	username:	Admin123
	password:	Admin123
Output	ExpResult:	1
	Result:	1
Status:		Passed
Comments:	To assess the length of the list that was created.	

Name:		testLoginAllRightAdminButExpResultWrong
Input	username:	Admin123
	password:	Admin123
Output	ExpResult:	0
	Result:	1
Status:		Failed
Comments:	To confirm the accuracy of the tests carried out. ExpResult should be 1.	

The evidence of the tests performed can be found below.

Note that one of the results did not pass the test. This was due to the test validating the accuracy of the test.

JUnit Test – Test Results

The screenshot displays an IDE interface with a Java source file and its corresponding JUnit test results.

Source File: Users.java

```
9
10
11 /**
12  * @author sergiovinicio
13  */
14 public class Users
15 {
16     private List<Signup> user = new ArrayList<>();
17
18     public Users()
19     {
20         user.add(new Signup(username: "oliveira-sergio", password: "Brazil123"));
21         user.add(new Signup(username: "Admin123", password: "Admin123"));
22     }
23
24     public Signup signUp (String username)
25     {
26         for (Signup i : user)
27         {
28             if (i.getUsername().equals(username))
29             {
30                 return i;
31             }
32         }
33     }
34 }
```

Test Results

com.mycompany.Login:jar:1.0-SNAPSHOT (Unit) x

Tests passed: 80.00 %

4 tests passed, 1 test failed (0.021 s)

- com.mycompany.login.LoginTest Failed
 - testLoginAllRightSergio passed (0.006 s)
 - testLoginAllRightAdmin passed (0.0 s)
 - testLoginWrongPassword passed (0.001 s)
 - testLoginBlank passed (0.001 s)

Test Log:

- Login -> Correct User and Password
Login successful
- Login -> Correct User and Password
Login successful
- Login -> Correct User and Wrong Password
Login failed. Invalid username or password.
- Login -> Correct User and Password, but expectedResult = 0
Login successful
- Login -> All Blank
Login failed. Invalid username or password.

Junit Test – Test Results

The screenshot displays an IDE interface with a project named 'Login'. The 'Source' view shows the 'Users.java' file, which contains a `Users` class with a `signup` method. The `signup` method iterates over a list of `Signup` objects and returns the first one whose username matches the input. The `Signup` class is not shown but is referenced in the code.

```
9  // ...
10 /**
11  *
12  * @author sergiovinicio
13  */
14 public class Users
15 {
16     private List<Signup> user = new ArrayList<>();
17
18     public Users()
19     {
20         user.add(new Signup(username: "oliveira-sergio", password: "Brazil123"));
21         user.add(new Signup(username: "Admin123", password: "Admin123"));
22     }
23
24     public Signup signup (String username)
25     {
26         for (Signup i : user)
27         {
28             if (i.getUsername().equals(username))
29             {
30                 return i;
31             }
32         }
33     }
34 }
```

The 'Test Results' view shows the execution of the `com.mycompany.login.LoginTest` class. The overall status is '4 tests passed, 1 test failed. (0.021 s)'. The failed test is `testLoginAllRightAdminButExpResultWrong`, which failed because the expected result was `<0>` but the actual result was `<1>`.

Test Name	Status	Duration
testLoginAllRightSergio	passed	0.006 s
testLoginAllRightAdmin	passed	0.0 s
testLoginWrongPassword	passed	0.001 s
testLoginAllRightAdminButExpResultWrong	Failed: expected: <0> but was: <1>	0.001 s
testLoginBlank	passed	0.001 s

The 'Output' view shows the following messages:

- Login -> Correct User and Password
Login successful
- Login -> Correct User and Passord
Login successful
- Login -> Correct User and Wrong Passord
Login failed. Invalid username or password.
- Login -> Correct User and Passord, but expResult = 0
Login successful
- Login -> All Blank
Login failed. Invalid username or password.

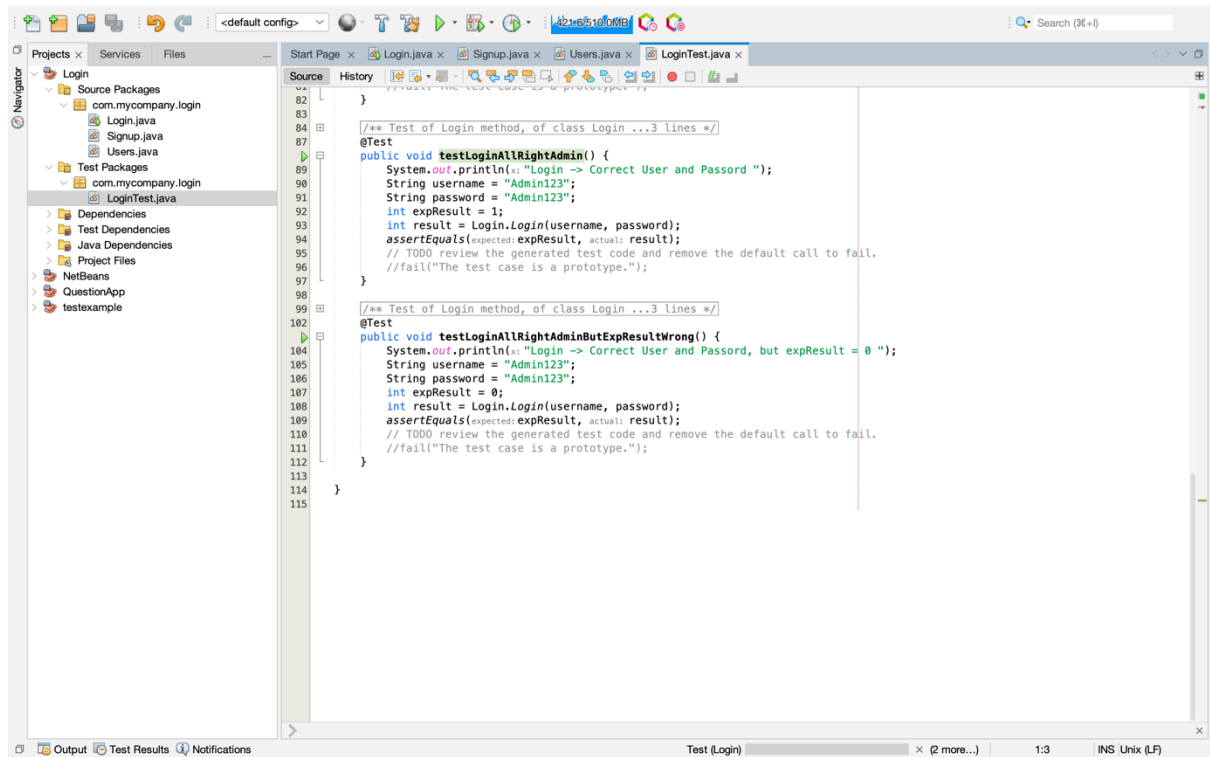
LoginTest.java

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/UnitTests/JUnit5TestClass.java to edit this template
4  */
5  package com.mycompany.login;
6
7  import org.junit.jupiter.api.AfterEach;
8  import org.junit.jupiter.api.AfterAll;
9  import org.junit.jupiter.api.BeforeEach;
10 import org.junit.jupiter.api.BeforeAll;
11 import org.junit.jupiter.api.Test;
12 import static org.junit.jupiter.api.Assertions.*;
13
14 /**
15  *
16  * @author sergioinicio
17  */
18 public class LoginTest {
19
20     public LoginTest() { ...2 Lines }
21
22     @BeforeAll
23     public static void setUpClass() { ...2 Lines }
24
25     @AfterAll
26     public static void tearDownClass() { ...2 Lines }
27
28     @BeforeEach
29     public void setUp() { ...2 Lines }
30
31     @AfterEach
32     public void tearDown() { ...2 Lines }
33
34     /** Test of Login method, of class Login ...3 Lines */
35     @Test
36     public void testLoginBlank() {
37         System.out.println("Login -> All Blank");
38         String username = "";
39         String password = "";
40         int expectedResult = 0;
41         int result = Login.Login(username, password);
42         assertEquals(expected: expectedResult, actual: result);
43         // TODO review the generated test code and remove the default call to fail.
44         //fail("The test case is a prototype.");
45     }
46 }
```

LoginTest.java

```
35 /** Test of Login method, of class Login ...3 Lines */
36 @Test
37 public void testLoginBlank() {
38     System.out.println("Login -> All Blank");
39     String username = "";
40     String password = "";
41     int expectedResult = 0;
42     int result = Login.Login(username, password);
43     assertEquals(expected: expectedResult, actual: result);
44     // TODO review the generated test code and remove the default call to fail.
45     //fail("The test case is a prototype.");
46 }
47
48 /** Test of Login method, of class Login ...3 Lines */
49 @Test
50 public void testLoginAllRightSergio() {
51     System.out.println("Login -> Correct User and Password");
52     String username = "oliveira-sergio";
53     String password = "Brazil123";
54     int expectedResult = 1;
55     int result = Login.Login(username, password);
56     assertEquals(expected: expectedResult, actual: result);
57     // TODO review the generated test code and remove the default call to fail.
58     //fail("The test case is a prototype.");
59 }
60
61 /** Test of Login method, of class Login ...3 Lines */
62 @Test
63 public void testLoginWrongPassword() {
64     System.out.println("Login -> Correct User and Wrong Passord ");
65     String username = "oliveira-sergio";
66     String password = "Brazil123";
67     int expectedResult = 0;
68     int result = Login.Login(username, password);
69     assertEquals(expected: expectedResult, actual: result);
70     // TODO review the generated test code and remove the default call to fail.
71     //fail("The test case is a prototype.");
72 }
73
74 /** Test of Login method, of class Login ...3 Lines */
75 @Test
76 public void testLoginAllRightAdmin() {
77     System.out.println("Login -> Correct User and Passord ");
78     String username = "Admin123";
79     String password = "Admin123";
80 }
```

LoginTest.java



3. Details of the methodology followed.

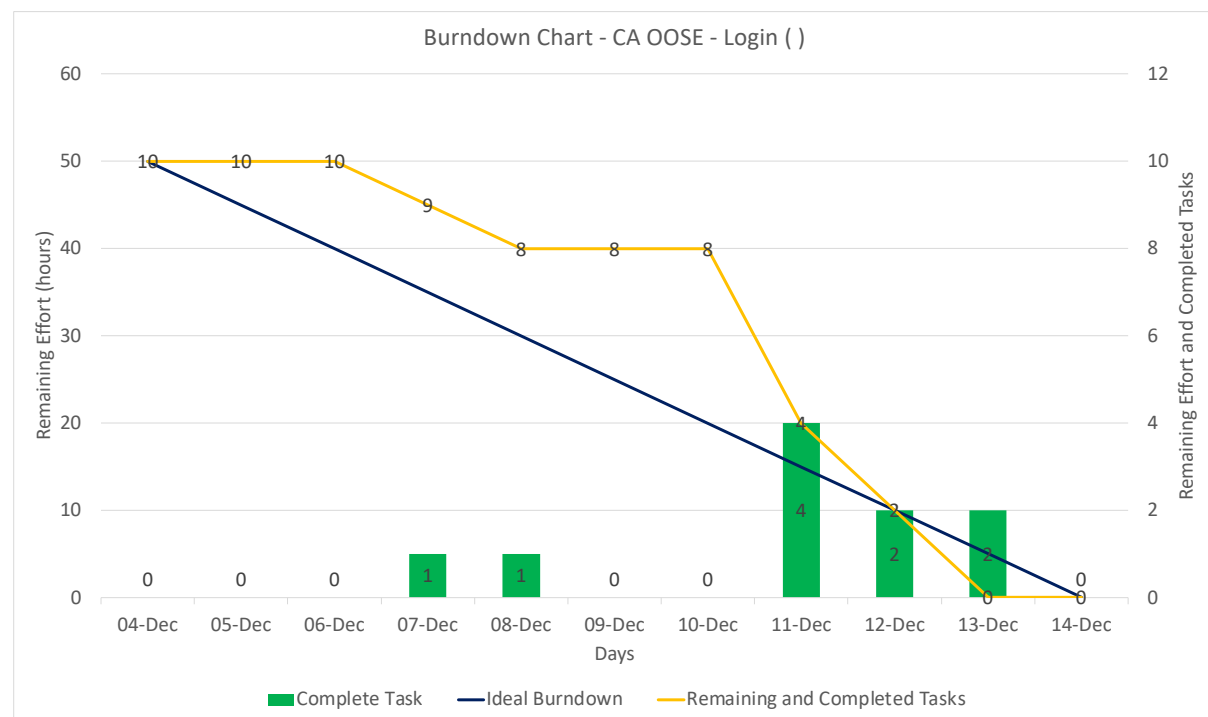
The methodology used in the tests followed the principles of Test Oriented Development hereinafter TDD.

The reason for this choice is due to the rapid test cycle, coding and revaluation. JUnit was the tool chosen to test the frameworks and is justified by the fact that the test is focused only on the classes and methods at hand. Moreover, for have the ability to be fast.

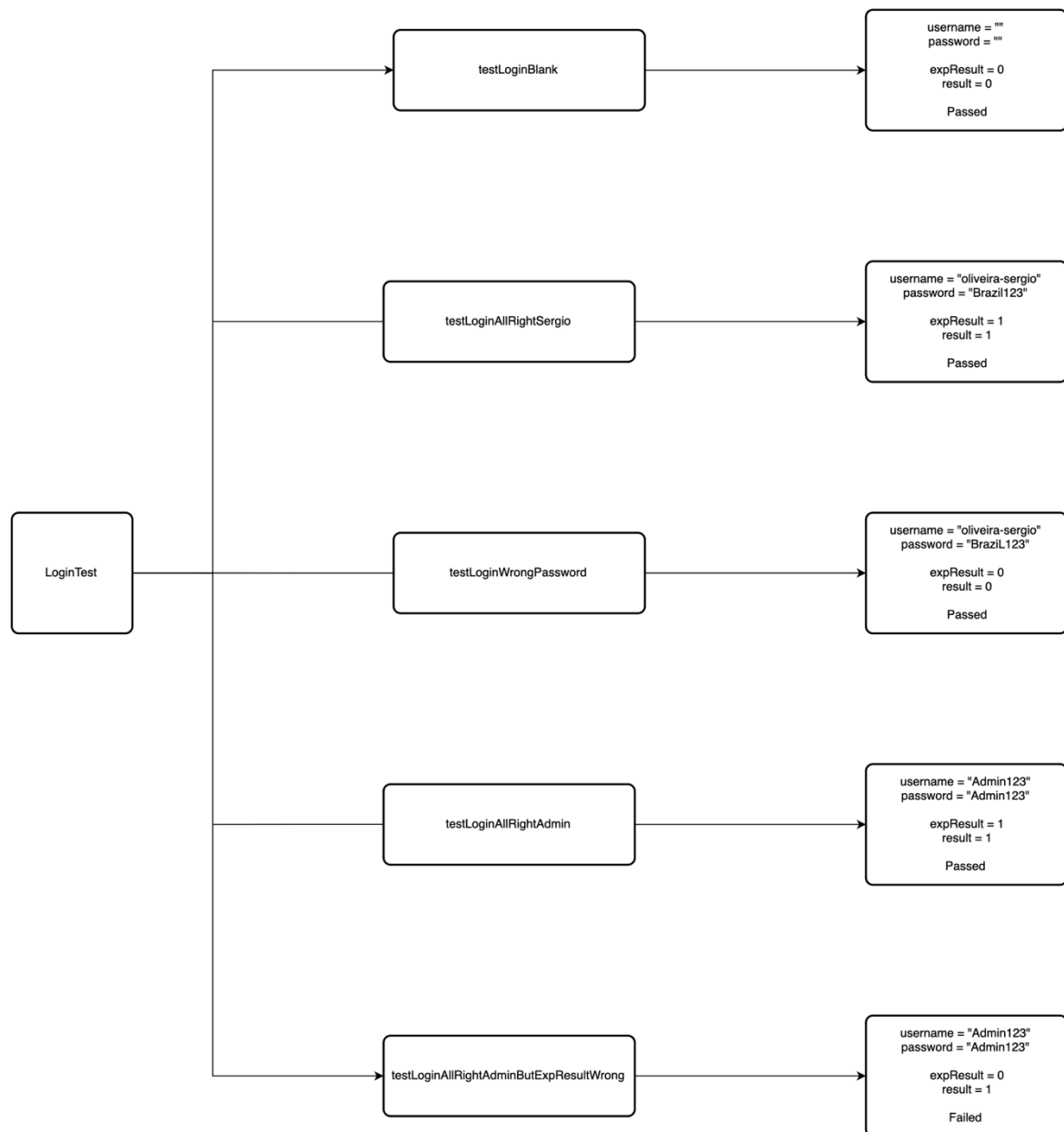
Based on the choices that I have made and the test results, we can conclude that the login method is free from faults. And it was guaranteed thanks to the TDD applied in JUnit.

The burndown chart below illustrates the way I worked during this stage.

Activities like functional class development, test development, and reports were considered for the mapped tests.



User Stories



Keep in mind that the link holds all the evidence found in this, including the codes created.

<https://github.com/sergio-oliveira-br/OOSE>

CODE

Main Class - Login

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
 default.txt to change this license
 */

package com.mycompany.login;

import java.util.Scanner;

/**
 *
 * @author sergiovinicio
 */
public class Login
{

    public static void main(String[] args)
    {
        System.out.println("CA | OOSE Part B");
        System.out.println("Welcome to the garden center."
            + " Accessing the point of sale requires a login.");

        //declare variables
        String username;
        String password;

        //declare object
        Scanner sc = new Scanner(System.in);

        //input
        System.out.println("Please enter your Username");
        username = sc.nextLine();

        System.out.println("Please enter your Password");
        password = sc.nextLine();
    }
}
```

```
// Create an instance of the Users class
Users myU = new Users();

// Process
Signup signedUpUser = myU.signUp(username);

//output
if (signedUpUser != null &&
signedUpUser.getPassword().equals(password))
{
    System.out.println("Login successful");
}
else
{
    System.out.println("Login failed. Invalid username or
password.");
}
}
```

Login created to test

```
public static int Login (String username, String password)
{

    // Create an instance
    Users myU = new Users();

    // Process
    Signup signedUpUser = myU.signUp(username);

    //output
    if (signedUpUser != null &&
signedUpUser.getPassword().equals(password))
    {
        System.out.println("Login successful");
        return 1;
    }
    else
    {
        System.out.println("Login failed. Invalid username or
password.");
        return 0;
    }
}
```

Class Users

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to
edit this template
 */
package com.mycompany.login;

import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author sergiovinicio
 */
public class Users
{
    private final List<Signup> user = new ArrayList<>();

    public Users()
    {
        user.add(new Signup("oliveira-sergio", "Brazil123"));
        user.add(new Signup("Admin123", "Admin123"));
    }

    public Signup signUp (String username)
    {
        for (Signup i : user)
        {
            if (i.getUsername().equals(username))
            {
                return i;
            }
        }
        return null;
    }
}
```

Class SingUp

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
 default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to
 edit this template
 */
package com.mycompany.login;

/**
 *
 * @author sergiovinicio
 */
public class Signup
{
    private String username;
    private String password;

    public Signup(String username, String password)
    {
        this.username = username;
        this.password = password;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }
}
```

Class Login Test – By Junit Test

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click
nbfs://nbhost/SystemFileSystem/Templates/UnitTests/JUnit5TestClass.java
to edit this template
 */
package com.mycompany.login;

import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.AfterAll;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

/**
 *
 * @author sergiovinicio
 */
public class LoginTest {

    public LoginTest() {
    }

    @BeforeAll
    public static void setUpClass() {
    }

    @AfterAll
    public static void tearDownClass() {
    }

    @BeforeEach
    public void setUp() {
    }

    @AfterEach
    public void tearDown() {
    }

    /**
```

```

    * Test of Login method, of class Login.
    */
@Test
public void testLoginBlank() {
    System.out.println("Login -> All Blank");
    String username = "";
    String password = "";
    int expectedResult = 0;
    int result = Login.Login(username, password);
    assertEquals(expectedResult, result);
    // TODO review the generated test code and remove the default
call to fail.
    //fail("The test case is a prototype.");
}

/**
 * Test of Login method, of class Login.
 */
@Test
public void testLoginAllRightSergio() {
    System.out.println("Login -> Correct User and Password");
    String username = "oliveira-sergio";
    String password = "Brazil123";
    int expectedResult = 1;
    int result = Login.Login(username, password);
    assertEquals(expectedResult, result);
    // TODO review the generated test code and remove the default
call to fail.
    //fail("The test case is a prototype.");
}

/**
 * Test of Login method, of class Login.
 */
@Test
public void testLoginWrongPassword() {
    System.out.println("Login -> Correct User and Wrong Passord ");
    String username = "oliveira-sergio";
    String password = "Brazil123";
    int expectedResult = 0;
    int result = Login.Login(username, password);
    assertEquals(expectedResult, result);
    // TODO review the generated test code and remove the default
call to fail.

```



```

        //fail("The test case is a prototype.");
    }

    /**
     * Test of Login method, of class Login.
     */
    @Test
    public void testLoginAllRightAdmin() {
        System.out.println("Login -> Correct User and Passord ");
        String username = "Admin123";
        String password = "Admin123";
        int expResult = 1;
        int result = Login.Login(username, password);
        assertEquals(expResult, result);
        // TODO review the generated test code and remove the default
call to fail.
        //fail("The test case is a prototype.");
    }

    /**
     * Test of Login method, of class Login.
     */
    @Test
    public void testLoginAllRightAdminButExpResultWrong() {
        System.out.println("Login -> Correct User and Passord, but
expResult = 0 ");
        String username = "Admin123";
        String password = "Admin123";
        int expResult = 0;
        int result = Login.Login(username, password);
        assertEquals(expResult, result);
        // TODO review the generated test code and remove the default
call to fail.
        //fail("The test case is a prototype.");
    }
}

```