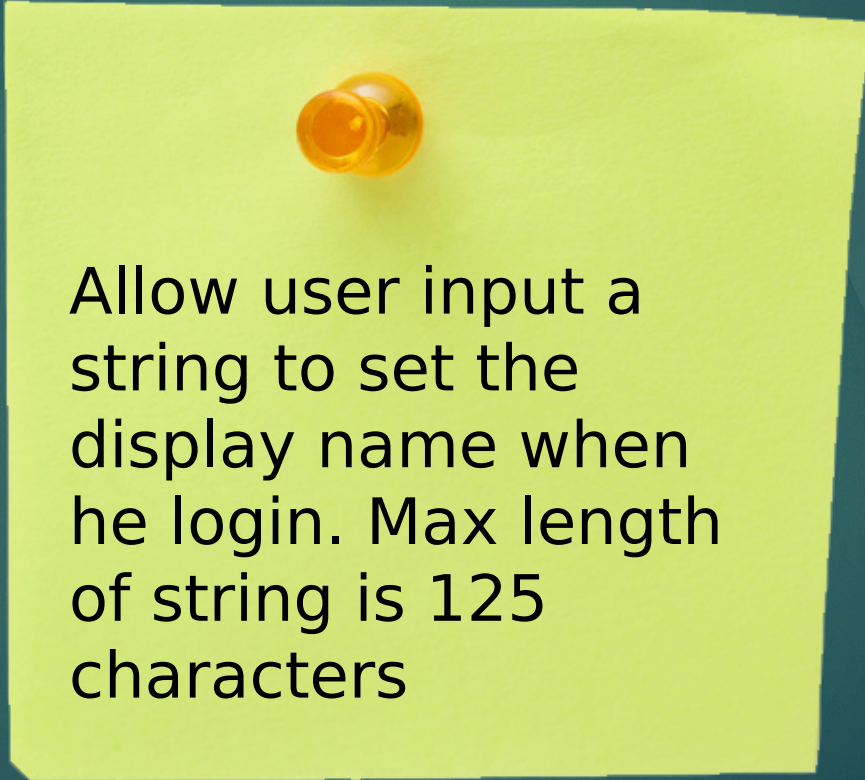


How to do TDD?

- ▶ We have a user story, use 3 step rule to make your idea (baby requirement)



Allow user input a string to set the display name when he login. Max length of string is 125 characters

How to do TDD?

- ▶ String can contain special characters ?
- ▶ What kind of special characters (!@#\$%^&...) ?
- ▶ String can contain space ?
- ▶ String can contain number ?
- ▶ Min length of string ?
- ▶ String can not contain select, delete, update because of SQL injection
- ▶ String can be duplicate ?

Note : Don't think the requirement is very simple therefore we don't need to confirm with Product Owner. Confirmation is a way to prevent bugs

TDD by examples

- ▶ Example 1: write a method that reverse last 2 characters of string.
- ▶ If null -> return null, if empty -> return empty, if length of string equal 1 -> return itself
- ▶ Ex: "A" -> "A", "" -> "", null -> null, "AB" -> "BA", "RAIN" -> "RANI"

TDD by examples

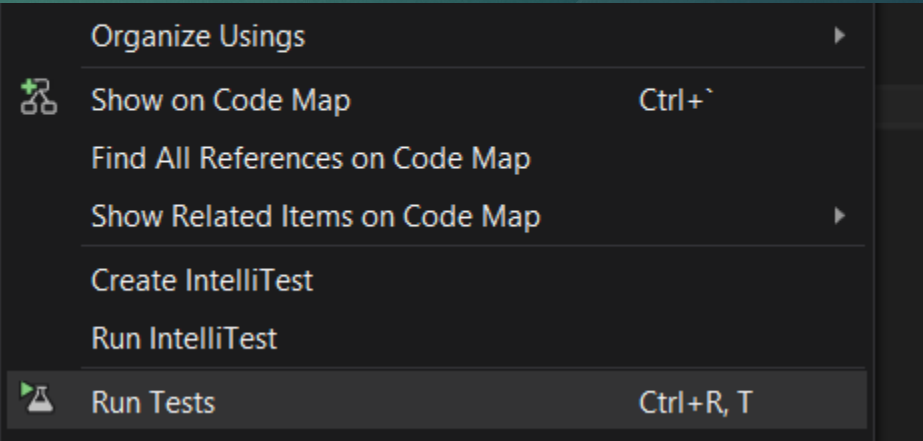
- ▶ Create first test case

```
[TestClass]
0 references
public class StringHelperTest
{
    [TestMethod]
    0 references
    public void ReserseSuccess_WhenInputHas2Characters()
    {
        StringHelper helper = new StringHelper();
        Assert.AreEqual(helper.SwapLast2Chars("BA"), "AB");
    }
}
```

TDD by examples

- ▶ Can not compile because StringHelper class is not created

```
[TestMethod]
0 references
public void ReserseSuccess_W
{
    StringHelper helper = ne
    Assert.AreEqual(helper.S
}
}
```



100 %

Error List

Entire Solution 2 Errors 0 Warnings 0 Messages Build + IntelliSense

Code	Description	Project
CS0246	The type or namespace name 'StringHelper' could not be found (are you missing StringLibTest a using directive or an assembly reference?)	
CS0246	The type or namespace name 'StringHelper' could not be found (are you missing StringLibTest a using directive or an assembly reference?)	

TDD by examples

- ▶ After StringHelper class is created, run **ALL TEST CASE** to see they(or one of them) fail

```
[TestClass]
0 references
public class StringHelperTest
{
    [TestMethod]
    ❌ | 0 references
    public void ReserseSuccess_WhenInputHas2Characters()
    {
        StringHelper helper = new StringHelper();
        Assert.AreEqual(helper.SwapLast2Chars("BA"), "AB");
    }
}
```

Streaming Video: Improving quality with unit tests ar ▾
Run All | Run... ▾ | Playlist: All Tests ▾

Failed Tests (1)
❌ ReserseSuccess_WhenInputHas2Characters 128 ms

TDD by examples

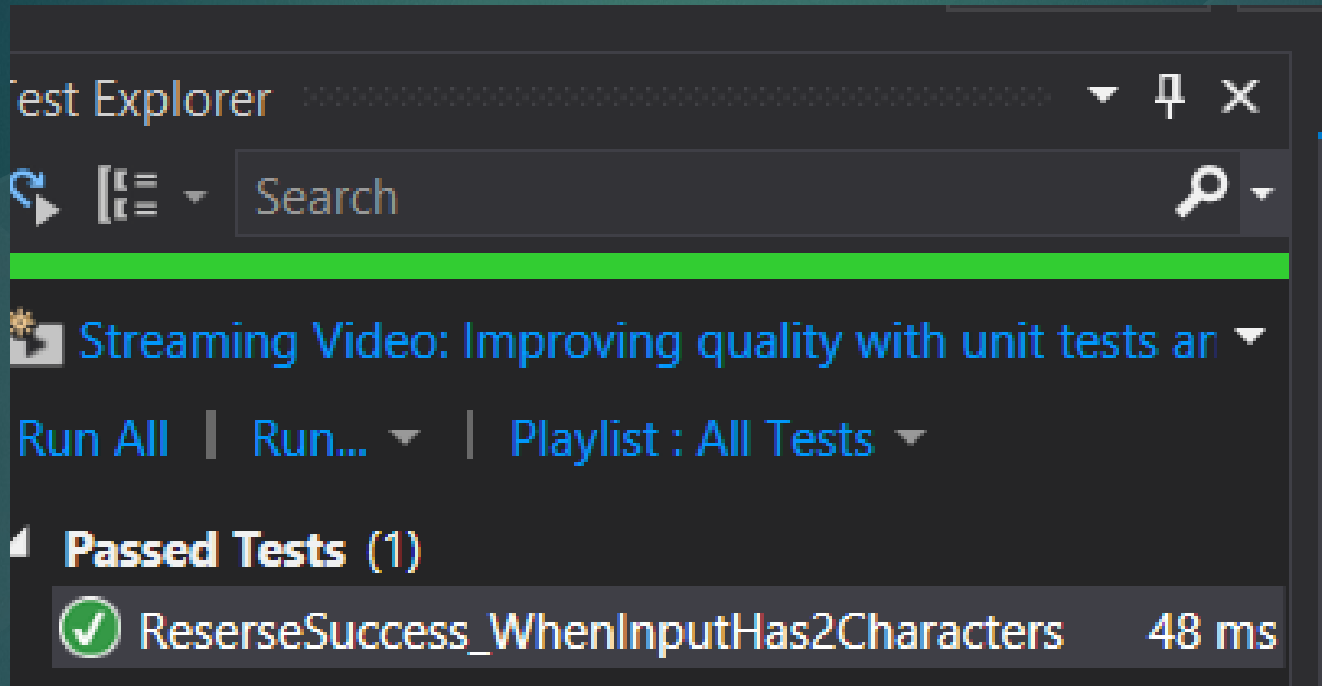
- ▶ Make a **litte change** to pass this test cases

```
4 references
public class StringHelper
{
    2 references | 1/1 passing
    public string SwapLast2Chars(string input)
    {
        string first = input[0].ToString();
        string second = input[1].ToString();
        return (second + first).ToString();
    }
}
```

Question : Who has a better implementation ?

TDD by examples

- ▶ Run ALL TEST CASES to see they pass



TDD by examples

- ▶ We have just finished a baby step of TDD cycle. Repeat it again.

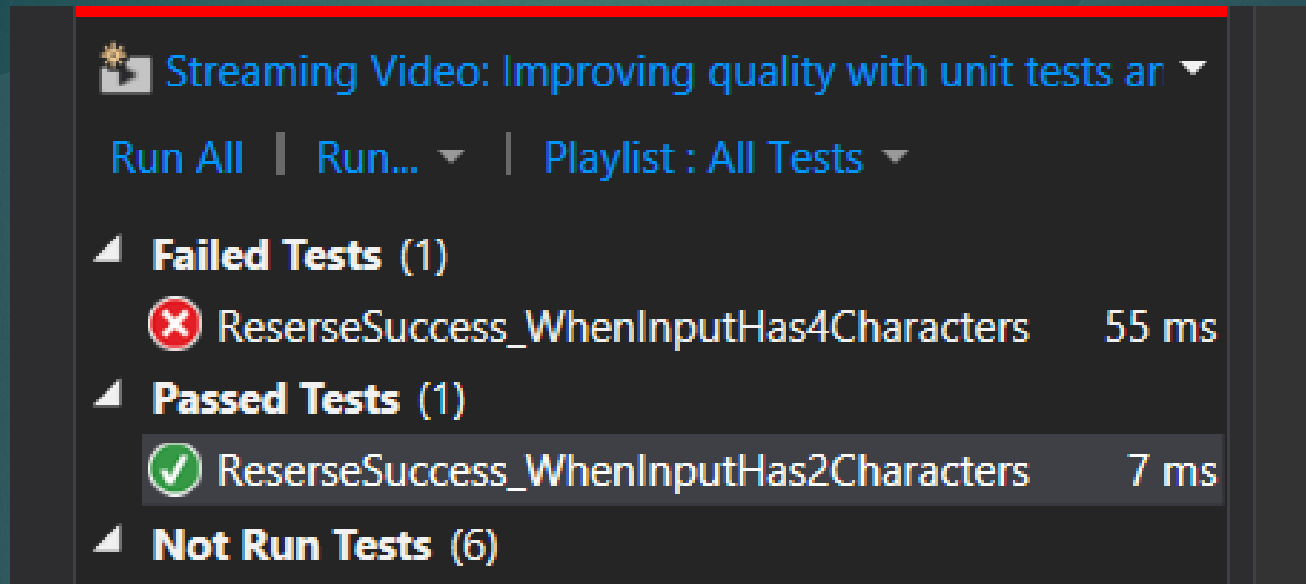
```
[TestClass]
0 references
public class StringHelperTest
{
    [TestMethod]
    ✓ | 0 references
    public void ReserseSuccess_WhenInputHas2Characters()
    {
        StringHelper helper = new StringHelper();
        Assert.AreEqual(helper.SwapLast2Chars("BA"), "AB");
    }
}
```

```
[TestMethod]
0 references
public void ReserseSuccess_WhenInputHas4Characters()
{
    StringHelper helper = new StringHelper();
    Assert.AreEqual(helper.SwapLast2Chars("ABCD"), "ABDC");
}
```

```
}
```

TDD by examples

- ▶ After StringHelper class is created, run **ALL TEST CASE** to see they(or one of them) fail



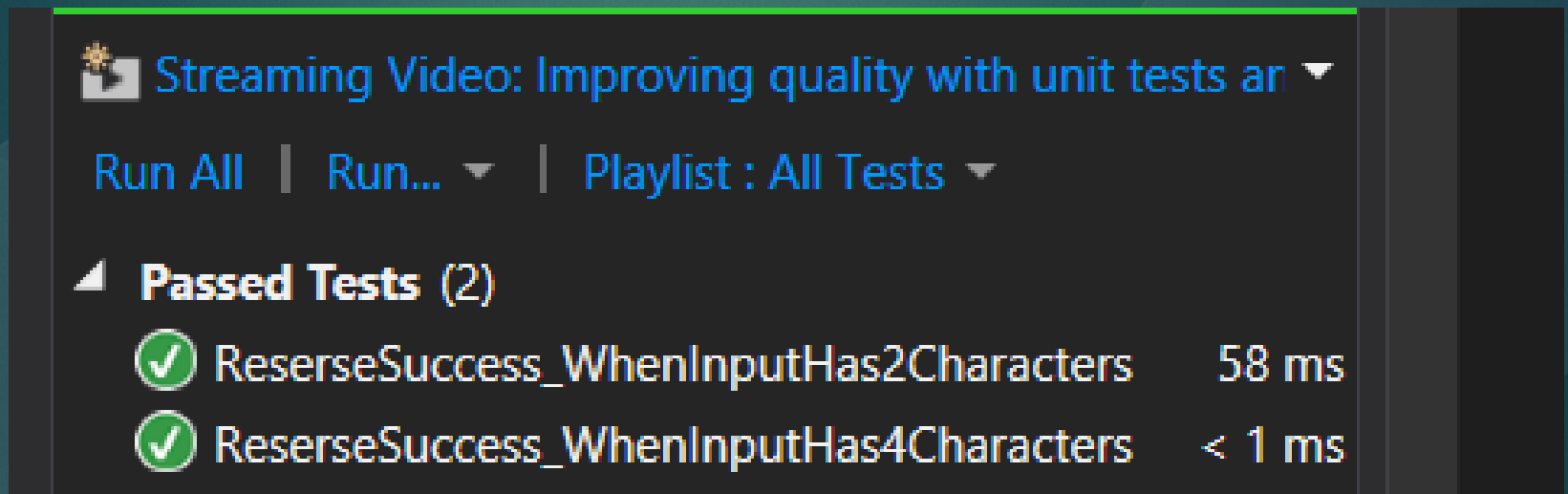
TDD by examples

- ▶ Refactor code to pass this test case

```
3 references | 2/2 passing
public string SwapLast2Chars(string input)
{
    //string first = input[0].ToString();
    //string second = input[1].ToString();
    //return (second + first).ToString();
    int length = input.Length;
    string stringMinus2LastChars = input.Substring(0, length - 2);
    string secondLastChar = input[length - 2].ToString();
    string lastChar = input[length - 1].ToString();
    return stringMinus2LastChars + lastChar + secondLastChar;
}
```

TDD by examples

- ▶ Run all test cases to see they pass



TDD by examples

- ▶ Refactor test code if necessary

```
[TestClass]
0 references
public class StringHelperTest
{
    StringHelper helper = new StringHelper();

    [TestMethod]
    ✓ | 0 references
    public void ReserseSuccess_WhenInputHas2Characters()
    {
        //StringHelper helper = new StringHelper();
        Assert.AreEqual(helper.SwapLast2Chars("BA"), "AB");
    }

    [TestMethod]
    ✓ | 0 references
    public void ReserseSuccess_WhenInputHas4Characters()
    {
        //StringHelper helper = new StringHelper();
        Assert.AreEqual(helper.SwapLast2Chars("ABCD"), "ABDC");
    }
}
```

TDD by examples

- ▶ Add new test case

```
StringHelper helper = new StringHelper();
```

```
[TestMethod]
```

```
✓ | 0 references
```

```
public void ReserseSuccess_WhenInputHas2Characters()
```

```
{
```

```
    Assert.AreEqual(helper.SwapLast2Chars("BA"), "AB");
```

```
}
```

```
[TestMethod]
```

```
✓ | 0 references
```

```
public void ReserseSuccess_WhenInputHas4Characters()
```

```
{
```

```
    Assert.AreEqual(helper.SwapLast2Chars("ABCD"), "ABDC");
```

```
}
```

```
[TestMethod]
```

```
0 references
```

```
public void ReserseSuccess_WhenInputHas1Characters()
```

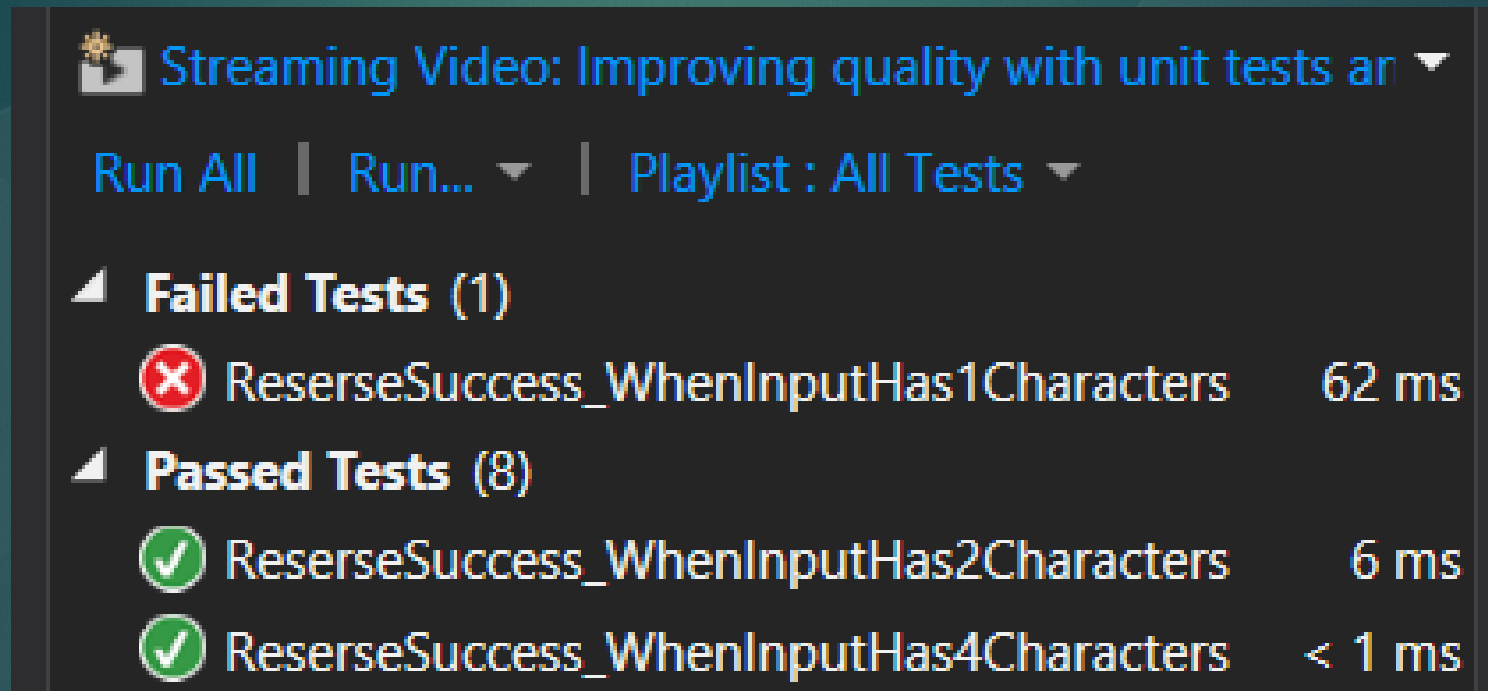
```
{
```

```
    Assert.AreEqual(helper.SwapLast2Chars("A"), "A");
```

```
}
```

TDD by examples

- ▶ After StringHelper class is created, run **ALL TEST CASE** to see they(or one of them) fail



The screenshot shows a test runner interface with a dark background. At the top, there is a gear icon and the text "Streaming Video: Improving quality with unit tests an" followed by a dropdown arrow. Below this, there are three buttons: "Run All", "Run..." with a dropdown arrow, and "Playlist : All Tests" with a dropdown arrow. The main area is divided into two sections: "Failed Tests (1)" and "Passed Tests (8)". Under "Failed Tests", there is one entry: "ReserseSuccess_WhenInputHas1Characters" with a red 'X' icon and a duration of "62 ms". Under "Passed Tests", there are two entries: "ReserseSuccess_WhenInputHas2Characters" with a green checkmark icon and a duration of "6 ms", and "ReserseSuccess_WhenInputHas4Characters" with a green checkmark icon and a duration of "< 1 ms".

Streaming Video: Improving quality with unit tests an ▼		
Run All	Run... ▼	Playlist : All Tests ▼
▲	Failed Tests (1)	
⊗	ReserseSuccess_WhenInputHas1Characters	62 ms
▲	Passed Tests (8)	
✔	ReserseSuccess_WhenInputHas2Characters	6 ms
✔	ReserseSuccess_WhenInputHas4Characters	< 1 ms

TDD by examples

- ▶ Refractor code to pass the test case

```
4 references | ❌ 2/3 passing
public string SwapLast2Chars(string input)
{
    int length = input.Length;
    if (length < 2)
    {
        return input;
    }
    string stringMinus2LastChars = input.Substring(0, length - 2);
    string secondLastChar = input[length - 2].ToString();
    string lastChar = input[length - 1].ToString();
    return stringMinus2LastChars + lastChar + secondLastChar;
}
```


TDD by examples

- ▶ Run all test cases to see they pass

▶ Passed Tests (9)

✓ ReserseSuccess_WhenInputHas1Characters	< 1 ms
✓ ReserseSuccess_WhenInputHas2Characters	7 ms
✓ ReserseSuccess_WhenInputHas4Characters	< 1 ms

TDD by examples

- ▶ Add new test case, this test case will be pass without any changes

```
[TestMethod]
✓ | 0 references
public void ReserseSuccess_WhenInputEmpty()
{
    Assert.AreEqual(helper.SwapLast2Chars(""), "");
}
```

▲ Passed Tests (10)

✓ ReserseSuccess_WhenInputEmpty	< 1 ms
✓ ReserseSuccess_WhenInputHas1Characters	< 1 ms
✓ ReserseSuccess_WhenInputHas2Characters	8 ms
✓ ReserseSuccess_WhenInputHas4Characters	< 1 ms

TDD by examples

- ▶ Add new test case, this test case will be fail

```
[TestMethod]
❌ | 0 references
public void ReserseSuccess_WhenInputIsNull()
{
    Assert.AreEqual(helper.SwapLast2Chars(null), null);
}
```

Failed Tests (1)

❌ ReserseSuccess_WhenInputIsNull	69 ms
----------------------------------	-------

TDD by examples

- ▶ Refactor code

```
6 references | ✖ 4/5 passing
public string SwapLast2Chars(string input)
{
    if (input == null)
    {
        return null;
    }
    int length = input.Length;
    if (length < 2)
    {
        return input;
    }
    string stringMinus2LastChars = input.Substring(0, length - 2);
    string secondLastChar = input[length - 2].ToString();
    string lastChar = input[length - 1].ToString();
    return stringMinus2LastChars + lastChar + secondLastChar;
}
```

TDD by examples

- ▶ Run all test cases, if they pass and you don't have any test case. The TDD cycle is completed. Refactor code if necessary

▲ Passed Tests (5)

✓ ReserseSuccess_WhenInputEmpty	< 1 ms
✓ ReserseSuccess_WhenInputHas1Characters	< 1 ms
✓ ReserseSuccess_WhenInputHas2Characters	19 ms
✓ ReserseSuccess_WhenInputHas4Characters	< 1 ms
✓ ReserseSuccess_WhenInputIsNull	< 1 ms