**MOQ**

**Task 1: Create Class Library Project –** CustomerCommLib

**Step 1: Create the Project**

In Visual Studio:

Create a new **Class Library (.NET Framework)** project.

Name it **CustomerCommLib**.

**Step 2: Create and Define the** IMailSender **Interface**

Create a new file called IMailSender.cs and define:

namespace CustomerCommLib

{

public interface IMailSender

{

bool SendMail(string toAddress, string message);

}

}

**Step 3: Implement the Interface in** MailSender

Rename Class1.cs to MailSender.cs and replace content with:

using System.Net;

using System.Net.Mail;

namespace CustomerCommLib

{

public class MailSender : IMailSender

{

public bool SendMail(string toAddress, string message)

{

MailMessage mail = new MailMessage();

SmtpClient smtpServer = new SmtpClient("smtp.gmail.com");

mail.From = new MailAddress("your\_email\_address@gmail.com");

mail.To.Add(toAddress);

mail.Subject = "Test Mail";

mail.Body = message;

smtpServer.Port = 587;

smtpServer.Credentials = new NetworkCredential("username", "password");

smtpServer.EnableSsl = true;

smtpServer.Send(mail);

return true;

}

}

}

**Step 4: Create the** CustomerComm **Class**

Create a new file CustomerComm.cs:

namespace CustomerCommLib

{

public class CustomerComm

{

private IMailSender \_mailSender;

public CustomerComm(IMailSender mailSender)

{

\_mailSender = mailSender;

}

public bool SendMailToCustomer()

{

// Hardcoded for demo purposes

return \_mailSender.SendMail("cust123@abc.com", "Some Message");

}

}

}

**Step 5: Build the Project**

Go to **Build > Build Solution**.

Ensure CustomerCommLib.dll is built successfully.

### ****Output****

DLL created successfully.

No compilation errors.

**Task 2: Create Unit Test Project –** CustomerComm.Tests

### ****Step 1: Create Test Project****

Create another **Class Library (.NET Framework)** project called **CustomerComm.Tests**.

Add **NuGet Packages**:

NUnit

NUnit3TestAdapter

Moq

Right-click project > Manage NuGet Packages > Browse > Install.

**Step 2: Add Reference to Main Library**

Add reference to CustomerCommLib.dll from this test project.

**Step 3: Create the Test Class**

Create CustomerCommTests.cs with the following code:

using Moq;

using NUnit.Framework;

using CustomerCommLib;

namespace CustomerComm.Tests

{

[TestFixture]

public class CustomerCommTests

{

private Mock<IMailSender> \_mockMailSender;

private CustomerComm \_customerComm;

[OneTimeSetUp]

public void Init()

{

\_mockMailSender = new Mock<IMailSender>();

\_mockMailSender.Setup(m => m.SendMail(It.IsAny<string>(), It.IsAny<string>()))

.Returns(true);

\_customerComm = new CustomerComm(\_mockMailSender.Object);

}

[TestCase]

public void SendMailToCustomer\_ShouldReturnTrue\_WhenMailIsSentSuccessfully()

{

// Act

var result = \_customerComm.SendMailToCustomer();

// Assert

Assert.IsTrue(result);

}

}

}

**Step 4: Build the Solution**

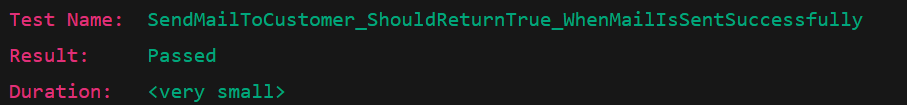
Rebuild both projects.

**Step 5: Run the Test**

Open **Test Explorer**: Test > Windows > Test Explorer.

Run the test case.

OUTPUT



**Summary**

| **Step** | **Description** | **Status** |
| --- | --- | --- |
| Task 1.1 | Created Class Library CustomerCommLib | ✅ |
| Task 1.2 | Defined IMailSender interface | ✅ |
| Task 1.3 | Implemented MailSender class | ✅ |
| Task 1.4 | Created CustomerComm with DI | ✅ |
| Task 2.1 | Created Unit Test project CustomerComm.Tests | ✅ |
| Task 2.2 | Added Moq and NUnit | ✅ |
| Task 2.3 | Wrote and passed test case using mock | ✅ |