

CustomerCommLib - Unit Testing Report

1. MailSender.cs

```
using System.Net;
using System.Net.Mail;

namespace CustomerCommLib
{
    public interface IMailSender
    {
        bool SendMail(string toAddress, string message);
    }

    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@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;
        }
    }
}
```

2. CustomerComm.cs

```
namespace CustomerCommLib
{
    public class CustomerComm
    {
        private readonly IMailSender _mailSender;

        public CustomerComm(IMailSender mailSender)
        {
            _mailSender = mailSender;
        }

        public bool SendMailToCustomer()
        {
            return _mailSender.SendMail("cust123@abc.com", "Some Message");
        }
    }
}
```

3. CustomerCommTests.cs

```
using NUnit.Framework;
using Moq;
using CustomerCommLib;

namespace CustomerCommLib.Tests
{
    [TestFixture]
    public class CustomerCommTests
    {
        [Test]
        public void SendMailToCustomer_ShouldReturnTrue_WhenMailIsSentSuccessfully()
        {
            var mockMailSender = new Mock();
            mockMailSender
                .Setup(m => m.SendMail(It.IsAny<string>(), It.IsAny<string>()))
                .Returns(true);

            var customerComm = new CustomerComm(mockMailSender.Object);
        }
    }
}
```

```

        var result = customerComm.SendMailToCustomer();

        Assert.That(result, Is.True);
    }
}
}

```

4. Test Output / Screenshot

