

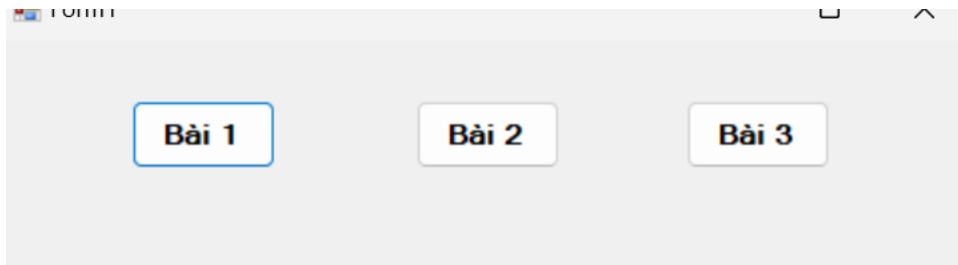
Name: Phan Phạm Trọng Phúc, Ngô Thanh Sang, Lưu Gia Huy

ID: 21520402 – 21522543 – 21520916

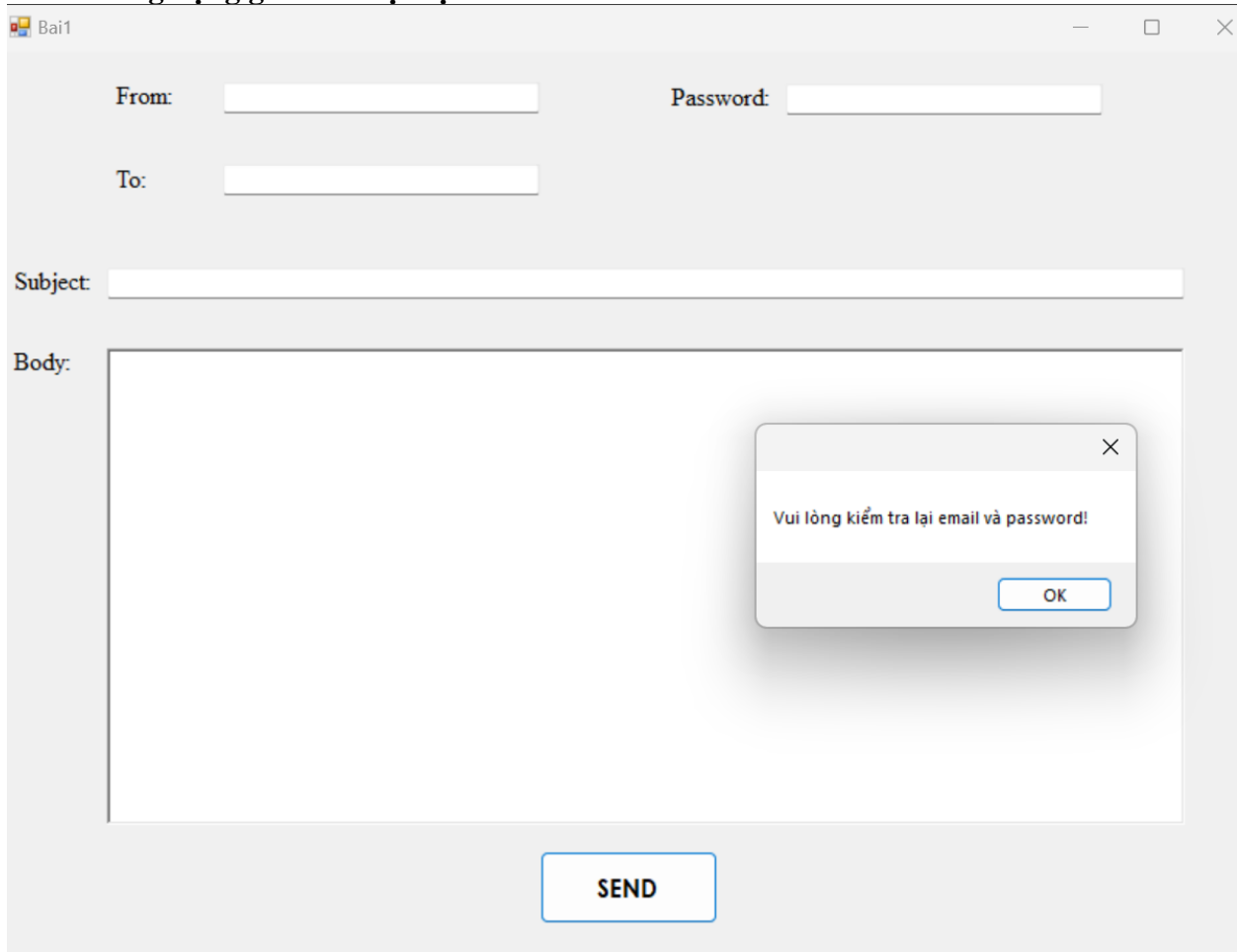
Class: NT216.N21.ANTN

Lập trình mạng căn bản LAB 5'S REPORT

Tạo DashBoard để di chuyển giữa các bài tập:



Câu 1: Ứng dụng gửi mail nội bộ:

A screenshot of a Windows application window titled 'Bai1'. The window contains a form for sending an email. It has input fields for 'From:', 'Password:', and 'To:'. Below these is a 'Subject:' label followed by a text box. The 'Body:' label is followed by a large text area. At the bottom center is a 'SEND' button. A modal dialog box is open over the body text area, displaying the message 'Vui lòng kiểm tra lại email và password!' (Please check your email and password again!) and an 'OK' button.

Kiểm tra dữ liệu trước khi gửi

Code:

```
1 reference
private void btnSend_Click(object sender, EventArgs e)
{
    if(!IsValidEmail(tbSender.Text) || !IsValidEmail(tbReceiver.Text) || tbPassword.Text.Length < 8)
    {
        MessageBox.Show("Vui lòng kiểm tra lại email và password!");
    }
    else
    {
        using (SmtpClient smtpClient = new SmtpClient("192.168.210.112"))
        {
            string mailfrom = tbSender.Text.ToString().Trim();
            string mailto = tbReceiver.Text.ToString().Trim();
            string password = tbPassword.Text.ToString().Trim();
            var basicCredential = new NetworkCredential(mailfrom, password);
            using (MailMessage message = new MailMessage())
            {
                MailAddress fromAddress = new MailAddress(mailfrom);
                smtpClient.UseDefaultCredentials = false;
                smtpClient.Credentials = basicCredential;
                message.From = fromAddress;
                message.Subject = tbSubject.Text.ToString().Trim();
                // Set IsBodyHtml to true means you can send HTML email.
                message.IsBodyHtml = true;
                message.Body = rtbBody.Text.ToString();
                message.To.Add(mailto);
                try
                {
                    smtpClient.Send(message);
                }
                catch (Exception ex)
                {
                    MessageBox.Show(ex.ToString());
                }
            }
        }
    }
}
```

- Sử dụng SmtpClient để kết nối tới máy chủ SMTP với địa chỉ IP là "192.168.210.112". –
- NetworkCredential được tạo để chứa thông tin đăng nhập của người gửi, bao gồm địa chỉ email và mật khẩu.
- MailMessage để chứa nội dung của email. Địa chỉ email của người gửi được lưu trữ trong MailAddress và được gán cho thuộc tính From của đối tượng MailMessage.
- Thuộc tính IsBodyHtml của đối tượng MailMessage được thiết lập thành true, cho phép gửi email với định dạng HTML. Nội dung của email được lấy từ rtbBody.Text và được gán cho thuộc tính Body của đối tượng MailMessage. Địa chỉ email của người nhận được thêm vào danh sách To của đối tượng MailMessage.
- Sử dụng phương thức Send() của đối tượng SmtpClient để gửi email. Nếu có lỗi xảy ra trong quá trình gửi email, một hộp thoại thông báo sẽ được hiển thị để thông báo cho người dùng.

Demo: (Đã report giảng viên trên lớp)

Câu 2: Ứng dụng đọc mail nội bộ:

Email:

Password:

LOGIN

Subject	Total	Recent
	From	

Vui lòng kiểm tra email và password!

OK

Kiểm tra các dữ liệu trước khi gửi

Code:

```

84 1 reference
85 public static bool IsValidEmail(string email)
86 {
87     if (string.IsNullOrEmpty(email))
88         return false;
89
90     try
91     {
92         // Normalize the domain
93         email = Regex.Replace(email, @"(@)(.+)$", DomainMapper,
94                               RegexOptions.None, TimeSpan.FromMilliseconds(200));
95
96         // Examines the domain part of the email and normalizes it.
97         string DomainMapper(Match match)
98         {
99             // Use IdnMapping class to convert Unicode domain names.
100             var idn = new IdnMapping();
101
102             // Pull out and process domain name (throws ArgumentException on invalid)
103             string domainName = idn.GetAscii(match.Groups[2].Value);
104
105             return match.Groups[1].Value + domainName;
106         }
107     }
108     catch (RegexMatchTimeoutException e)
109     {
110         return false;
111     }
112     catch (ArgumentException e)
113     {
114         return false;
115     }
116
117     try
118     {
119         return Regex.IsMatch(email,
120                             @"^[^@\s]+@([^\s]+\.[^\s]+)$",
121                             RegexOptions.IgnoreCase, TimeSpan.FromMilliseconds(250));
122     }
123     catch (RegexMatchTimeoutException)
124     {
125         return false;
126     }
127 }
128
129

```

Kiểm tra xem có phải là email hợp lệ

Bai2

Email:

Password:

LOGIN

Subject	Total	Recent	Date
<div><div>Vui lòng kiểm tra email và password!</div><div>OK</div></div>			

```

1 reference
private void btnSend_Click(object sender, EventArgs e)
{
    if (!IsValidEmail(tbEmail.Text) || tbPassword.Text.Length < 8)
    {
        MessageBox.Show("Vui lòng kiểm tra email và password!");
    }
    else
    {
        try
        {
            using (var client = new ImapClient())
            {
                client.ServerCertificateValidationCallback = (s, c, h, err) => true;
                // Connect to the IMAP server
                client.Connect("localhost", 993, true);
                // Authenticate with the server
                client.Authenticate(tbEmail.Text, tbPassword.Text);

                // Open the inbox folder
                var inbox = client.Inbox;
                inbox.Open(FolderAccess.ReadOnly);

                // Clear existing items in the ListView
                listView1.Items.Clear();

                lbTotal.Text = inbox.Count.ToString();
                lbRecent.Text = inbox.Recent.ToString();

                for (int i = 0; i < inbox.Count; i++)
                {
                    var message = inbox.GetMessage(i);

                    ListViewItem item = new ListViewItem();
                    item.Text = message.Subject;
                    item.SubItems.Add(new ListViewItem.ListViewSubItem() { Text = message.From.ToString() });
                    item.SubItems.Add(new ListViewItem.ListViewSubItem() { Text = message.Date.Date.ToString() });
                    listView1.Items.Add(item);
                }

                // Disconnect from the server
                client.Disconnect(true);
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}

```

- Tạo đối tượng ImapClient
- Phương thức ServerCertificateValidationCallback được thiết lập để bỏ qua việc xác thực chứng chỉ SSL/TLS của máy chủ. Sau đó, phương thức Connect() được gọi với các tham số là tên máy chủ (localhost), cổng (993) và sử dụng SSL (true) để thiết lập kết nối đến máy chủ IMAP.
- Phương thức Authenticate() được gọi để xác thực tài khoản email của người dùng với máy chủ IMAP. Giá trị của hai biến tbEmail.Text và tbPassword.Text được truyền vào để đăng nhập tài khoản email.
- Sau khi xác thực thành công sử dụng phương thức Open() để mở thư mục "Inbox" của tài khoản email hiện tại với quyền truy cập chỉ đọc (FolderAccess.ReadOnly). Sau khi thư mục được mở, đếm số lượng email trong thư mục "Inbox" và gán giá trị này vào một Label. Đồng thời, số lượng email mới nhất (Recent) cũng được gán giá trị vào một Label khác.
- Vòng lặp for được sử dụng để lặp qua từng email trong thư mục "Inbox". Đối tượng Message được lấy từ phương thức GetMessage() và lưu trữ vào biến message.

- ListViewItem được tạo và các thuộc tính của nó được gán giá trị từ các thuộc tính của Message, bao gồm tiêu đề (Subject), người gửi (From) và ngày gửi (Date). Cuối cùng, ListViewItem được thêm vào ListView.
- Phương thức Disconnect() được gọi để ngắt kết nối với máy chủ IMAP. Tham số true được truyền vào để cho phép gửi lệnh "LOGOUT" đến máy chủ IMAP trước khi ngắt kết nối.

Demo: (Đã report giảng viên trên lớp)

Câu 3: Ứng dụng gửi mail:

The screenshot shows a Windows-style application window titled "Bai3". It has a light gray background. At the top, there are two input fields: "From:" and "Password:". Below them is a "To:" input field. Further down is a "Subject:" input field. The main area is a large text box for the "Body:". At the bottom center is a blue "SEND" button. A modal dialog box is open in the center of the body area, with a close button (X) in the top right corner. The dialog contains the text "Vui lòng kiểm tra lại email và password!" and an "OK" button at the bottom right.

Kiểm tra dữ liệu trước khi gửi

From: 21520402@gm.uit.edu.vn Password: *****

To: trongphucphan7@gmail.com

Subject: Xin chào đây là nhóm 8

Body:

The SMTP server requires a secure connection or the client was not authenticated. The server response was: 5.7.0 Authentication Required. Learn more at

OK

SEND

Nếu login fail thì sẽ hiển thị

From: 21520402@gm.uit.edu.vn Password: *****

To: trongphucphan7@gmail.com

Subject: Xin chào đây là Nhóm 8

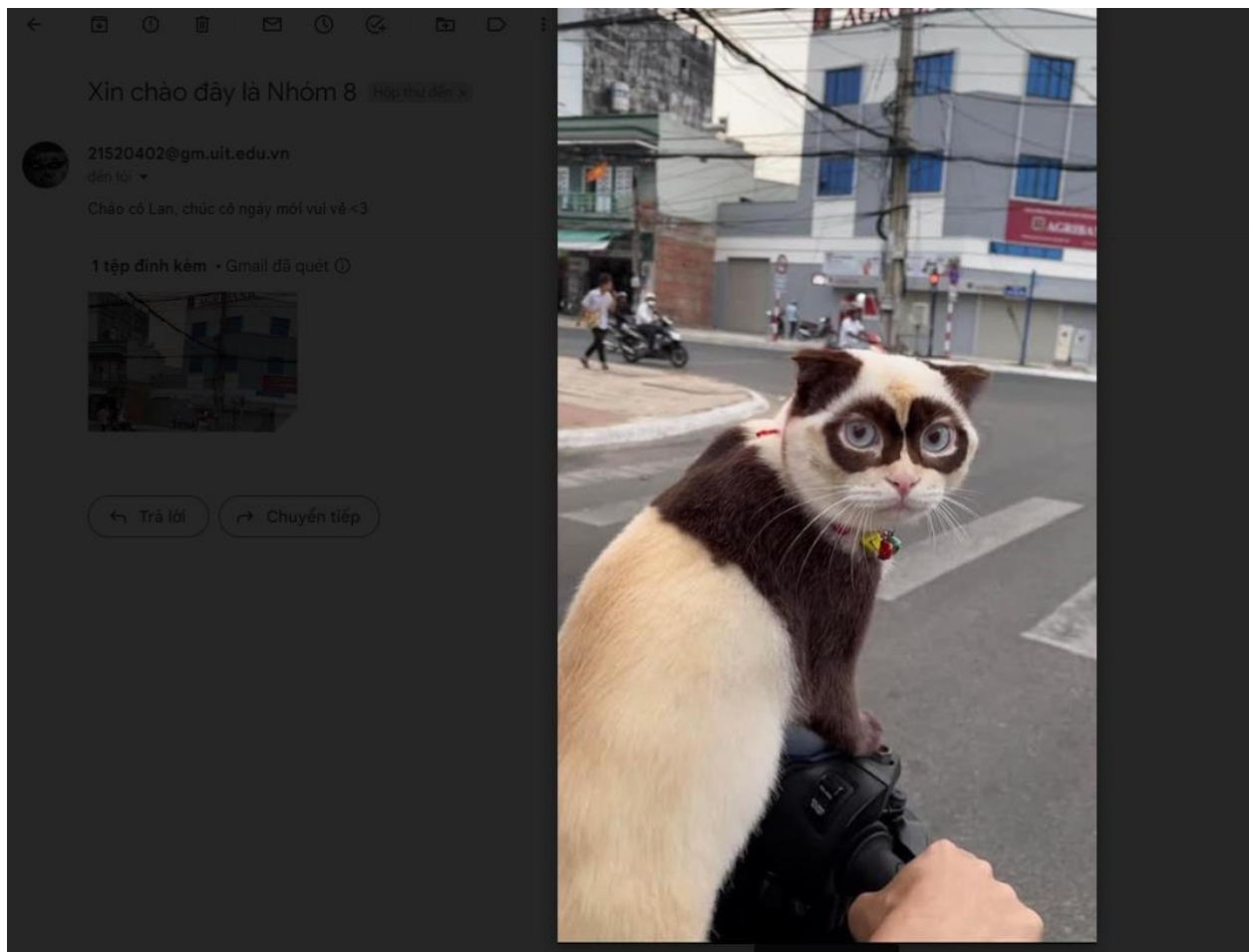
Body: Chào cô Lan, chúc cô ngày mới vui vẻ <3

Email sent successfully.

OK

SEND

9:36 CH
07/06/2023



Code:


```

if (!IsValidEmail(tbSender.Text) || !IsValidEmail(tbReceiver.Text) || tbPassword.Text.Length < 8)
{
    MessageBox.Show("Vui lòng kiểm tra lại email và password!");
}
else
{
    try
    {
        string mailFrom = tbSender.Text.Trim();
        string mailTo = tbReceiver.Text.Trim();
        string password = tbPassword.Text;

        // Set up SmtpClient
        SmtpClient smtpServer = new SmtpClient();
        smtpServer.Port = 587;
        smtpServer.Host = "smtp.gmail.com";
        smtpServer.EnableSsl = true;
        smtpServer.DeliveryMethod = SmtpDeliveryMethod.Network;
        smtpServer.UseDefaultCredentials = false;
        smtpServer.Credentials = new NetworkCredential(mailFrom, tbPassword.Text);

        // Set up mailmessage
        MailMessage mail = new MailMessage();
        mail.From = new MailAddress(mailFrom);
        mail.To.Add(mailTo);
        mail.Subject = tbSubject.Text;
        mail.Body = rtbBody.Text;
        mail.IsBodyHtml = true;

        // Choose file to attach
        OpenFileDialog openFileDialog = new OpenFileDialog();
        if (openFileDialog.ShowDialog() == DialogResult.OK)
        {
            string filePath = openFileDialog.FileName;

            // Create attachment
            Attachment attachment = new Attachment(filePath);
            mail.Attachments.Add(attachment);
        }

        smtpServer.Send(mail);
        MessageBox.Show("Email sent successfully.");
    }
    catch (Exception err)
    {
        MessageBox.Show(err.Message);
    }
}

```

- Sử dụng phương thức SmtpClient để gửi mail.
- Thiết lập các thuộc tính của đối tượng SmtpClient để chỉ định máy chủ SMTP, số cổng và mã hóa SSL, thiết lập thông tin đăng nhập được sử dụng cho xác thực với máy chủ SMTP.
- Thiết lập các thuộc tính From, To, Subject và Body của đối tượng MailMessage.
- Thiết lập thuộc tính IsBodyHtml thành true nếu nội dung email chứa HTML.