

LAB 8: NETWORKING



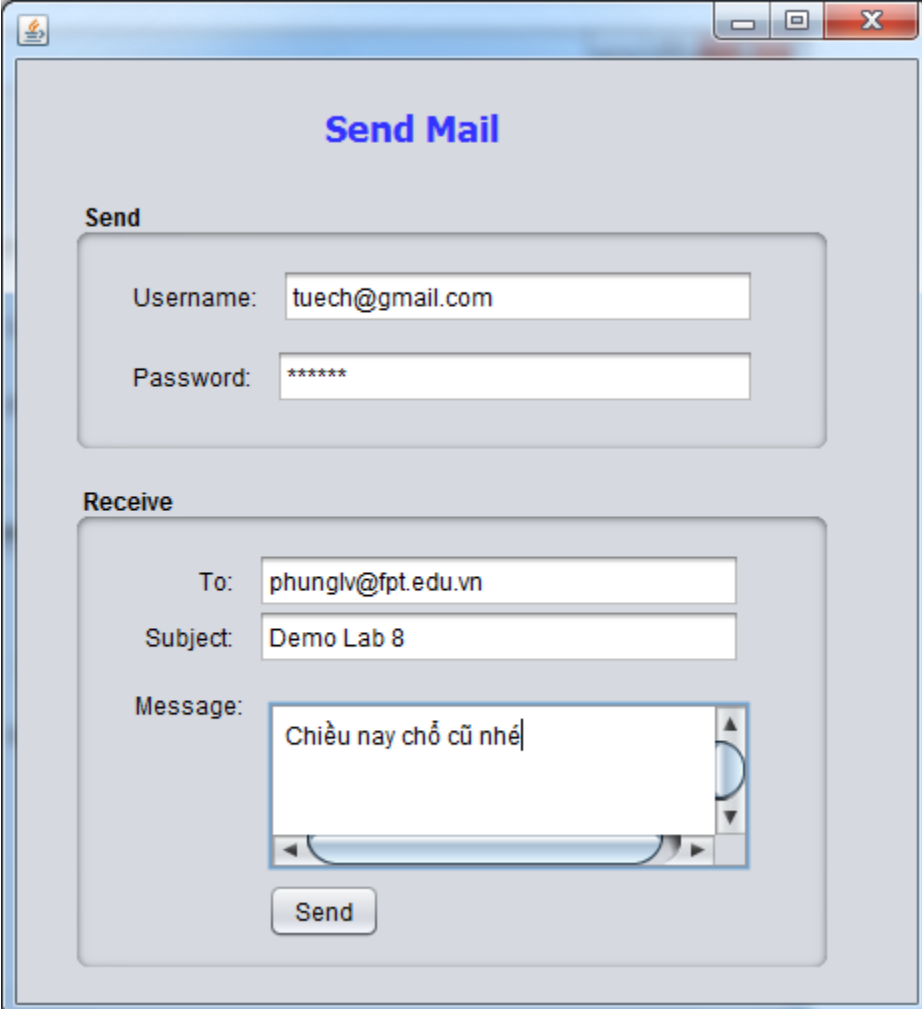
MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Viết chương trình gửi email
- ✓ Gửi email có kèm theo file
- ✓ Chức năng gửi email CC..., BCC...

PHẦN I

Bài 1: Thiết kế ứng dụng như hình và thực hiện các chức năng theo yêu cầu sau (5 điểm)



Send Mail

Send

Username: tuech@gmail.com

Password: *****

Receive

To: phunglv@fpt.edu.vn

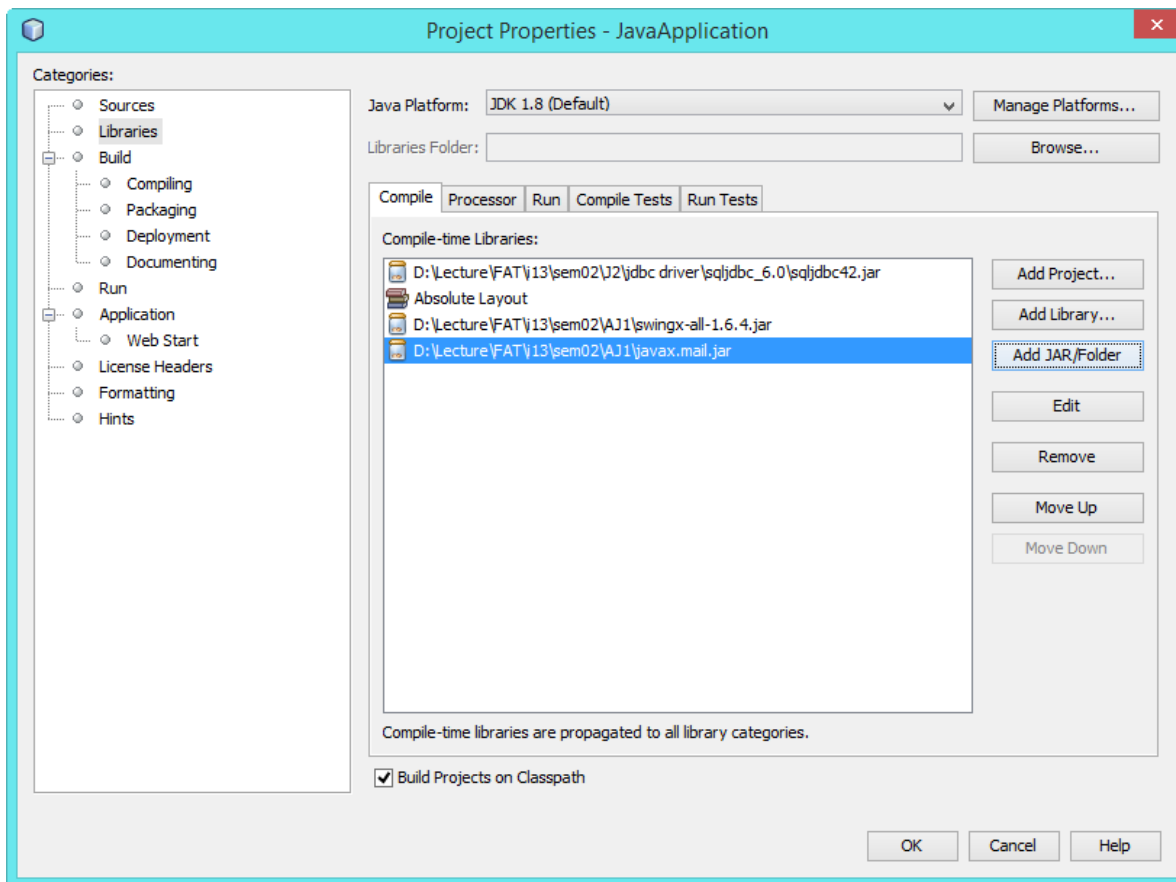
Subject: Demo Lab 8

Message: Chiều nay chỗ cũ nhé

Send

Chuẩn bị:

- Tải [JavaMail API](#)
- Thêm JavaMail API (javax.mail.jar) vào project: Chuột phải Project -> chọn **Properties** -> chọn **Libraries** tại Categories -> chọn **Add JAR/Folder**



- Tài khoản gmail đã kích hợp tính năng POP: Sau khi đăng nhập vào tài khoản gmail -> chọn Setting -> chọn Forwarding and POP/IMAP



Cài đặt:

Bước 1: Tạo đối tượng Properties và chỉ định thông tin host, port

```
Properties p = new Properties();
p.put("mail.smtp.auth", "true");
p.put("mail.smtp.starttls.enable", "true");
p.put("mail.smtp.host", "smtp.gmail.com");
p.put("mail.smtp.port", 587);
```

Bước 2: Tạo đối tượng Session (phiên làm việc)

```
Session s = Session.getInstance(p,
    new javax.mail.Authenticator() {
        protected PasswordAuthentication getPasswordAuthentication() {
            return new PasswordAuthentication(accountName, accountPassword);
        }
    });
```

Trong đó **accountName** và **accountPassword** là tài khoản gmail.

Bước 3: Tạo đối tượng message

```
Message msg = new MimeMessage(s);
msg.setFrom(new InternetAddress(from));
msg.setRecipients(Message.RecipientType.TO, InternetAddress.parse(to));
msg.setSubject(subject);
```

```
msg.setText(body) ;
```

Bước 4: Gửi mail

```
Transport.send(msg) ;
```

Xử lý sự kiện cho nút “Send”

- Các thông tin như Username, Password, To, Subject và Message là bắt buộc nhập.
- Khi người dùng chọn nút lệnh “Send”, thực hiện gửi mail đến địa chỉ được chỉ định ở **To**

Lưu ý nếu chương trình phát sinh lỗi “**javax.mail.AuthenticationFailedException**” trong quá trình chạy thì truy cập link <https://www.google.com/settings/security/lesssecureapps> và đăng nhập vào tài khoản gmail được dùng để thiết lập và chọn **Turn on** như hình bên dưới.

Tham khảo code nút “Send”

```

201 private void btnSendActionPerformed(java.awt.event.ActionEvent evt) {
202     try {
203         // TODO add your handling code here:
204         Properties p = new Properties();
205         p.put("mail.smtp.auth", "true");
206         p.put("mail.smtp.starttls.enable", "true");
207         p.put("mail.smtp.host", "smtp.gmail.com");
208         p.put("mail.smtp.port", 587);
209         //-----
210         String accountName = txtUser.getText();
211         String accountPassword = txtPass.getText();
212         Session s = Session.getInstance(p,
213             new javax.mail.Authenticator() {
214                 protected PasswordAuthentication getPasswordAuthentication() {
215                     return new PasswordAuthentication(accountName, accountPassword);
216                 }
217             });
218         //-----
219         String from = txtUser.getText();
220         String to = txtTo.getText();
221         String subject = txtSubject.getText();
222         String body = txtMessage.getText();
223         Message msg = new MimeMessage(s);
224         msg.setFrom(new InternetAddress(from));
225         msg.setRecipients(Message.RecipientType.TO, InternetAddress.parse(to));
226         msg.setSubject(subject);
227         msg.setText(body);
228         //-----
229         Transport.send(msg);
230         JOptionPane.showMessageDialog(null, "Mail was sent successfully.", "Message",
231             javax.swing.JOptionPane.INFORMATION_MESSAGE);
232     } catch (MessagingException ex) {
233         Logger.getLogger(SendMail.class.getName()).log(Level.SEVERE, null, ex);
234     }
235 }
236

```

PHẦN I

Bài 2: Phát triển thêm các chức năng (5 điểm)

- **Attach file:** cho gửi kèm theo file
- **CC...:** cho phép cùng lúc nội dung email được gửi đến nhiều người.