

Internet Technology Lab Report – Assignment 6

TITLE

Name – Sourav Dutta

Roll – 001610501076

Class – BCSE 4th year

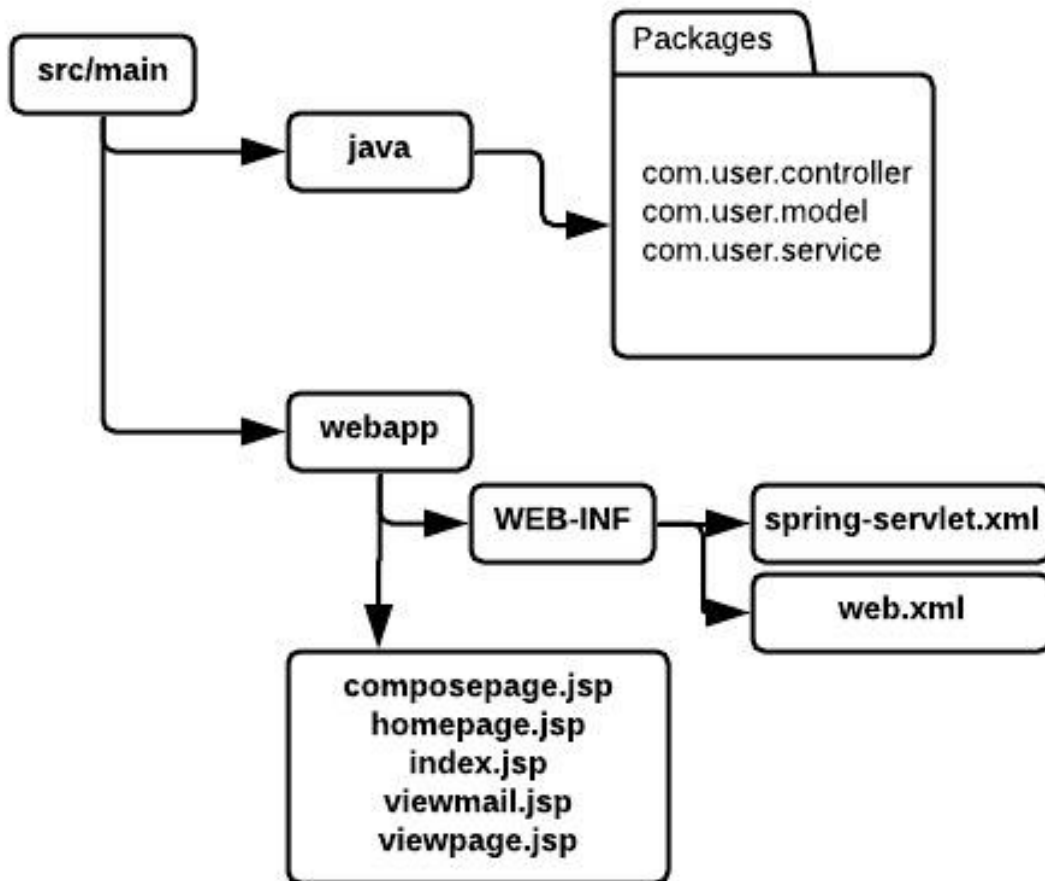
Group – A3

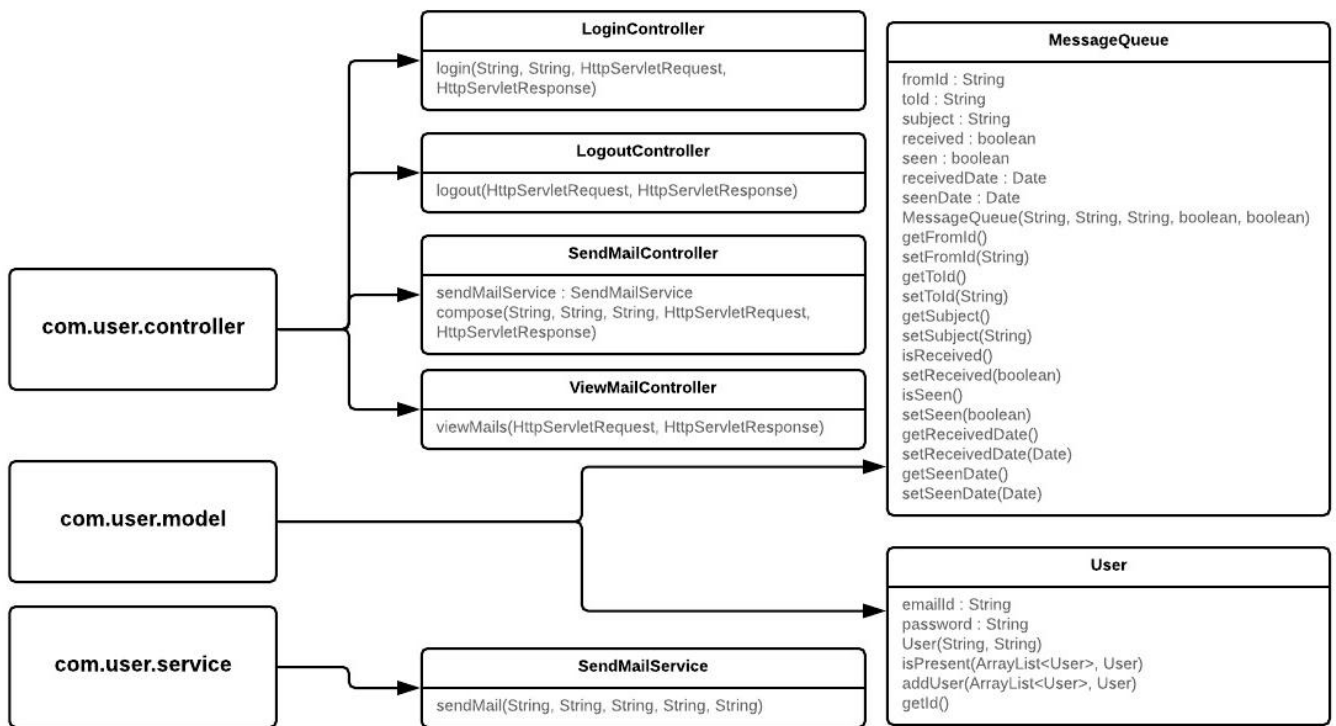
Assignment Numbers – 6

Design a Mail server and client program that implements SMTP and POP3 protocols for sending and receiving emails. An acknowledgement is sent to the sender whenever the recipient views the email.

Recipient should find email notifications whenever (s)he logs in. Two possible use cases are shown above. For the details of POP3 and SMTP protocols you can refer to RFC 1939 (<http://tools.ietf.org/html/rfc1939>) and RFC 821 (<http://tools.ietf.org/html/rfc821>), respectively

CLASS DIAGRAMS





CODE SNIPPET

Deployment Descriptor [web.xml] :

```

<!DOCTYPE web-app PUBLIC
"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd" >

<web-app>
  <display-name>Archetype Created Web Application</display-name>

  <servlet>
    <servlet-name>spring</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>spring</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>
</web-app>
  
```

spring-servlet.xml :

```

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:ctx="http://www.springframework.org/schema/context"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:mvc="http://www.springframework.org/schema/mvc"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
    http://www.springframework.org/schema/mvc
    http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd"
  >
  </beans>
  
```

```
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.0.xsd">
```

```
<ctx:annotation-config></ctx:annotation-config>
<ctx:component-scan base-package="com.user"></ctx:component-scan>
```

```
</beans>
```

pom.xml :

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-
v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.sourav</groupId>
  <artifactId>ass6_mail</artifactId>
  <packaging>war</packaging>
  <version>0.0.1-SNAPSHOT</version>
  <name>ass6_mail Maven Webapp</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-webmvc</artifactId>
      <version>5.0.5.RELEASE</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>8.0.11</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.apache.httpcomponents/httpclient -->
    <dependency>
      <groupId>org.apache.httpcomponents</groupId>
      <artifactId>httpclient</artifactId>
      <version>4.3.3</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/com.sun.mail/javax.mail -->
    <dependency>
      <groupId>com.sun.mail</groupId>
      <artifactId>javax.mail</artifactId>
      <version>1.6.0</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/javax.servlet/servlet-api -->
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>servlet-api</artifactId>
      <version>3.0-alpha-1</version>
```

```

        <scope>provided</scope>
    </dependency>
</dependencies>
</build>
    <finalName>ass6_mail</finalName>
</build>
</project>

```

index.jsp :

```

<html>
<body>
<h2> Welcome to the Email Application </h2>

<form action = "Login" >
    Email Id <input type="text" name="text"><br><br>
    Password <input type="password" name="pwd"><br><br>
    <input type="submit">
</form>

</body>
</html>

```

homepage.jsp :

```

<%@page import="com.user.model.MessageQueue"%>
<%@ page import="java.util.ArrayList" %>
<%@ page import="com.user.model.User" %>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

Welcome to your mailbox <%= request.getAttribute("currentuser") %>

<br> <br> <br>

<form action = "Logout">
    <div align="right">
        <input type="submit" value="LOGOUT">
    </div>
</form>

<form action = "composepage.jsp">
    <input type="submit" value="COMPOSE" >
    <input type="hidden" name="currentuser" value=
<%=request.getAttribute("currentuser") %> >
    <input type="hidden" name="password" value= <%=request.getAttribute("password") %>
>
</form>
<br>
<form action = "viewmails">
    <input type="submit" value="VIEW MAILS" >
    <input type="hidden" name="currentuser" value=
<%=request.getAttribute("currentuser") %> >
    <input type="hidden" name="password" value= <%=request.getAttribute("password") %>
>

```

```

</form>

<%
    if(session.getAttribute("messagequeue")!=null) {
        System.out.println("got message queue");
        ArrayList<MessageQueue> received = new ArrayList<MessageQueue>();
        ArrayList<MessageQueue> mqs =
        (ArrayList<MessageQueue>)session.getAttribute("messagequeue");
        ArrayList<MessageQueue> newmqs = new ArrayList<MessageQueue>();
        for(MessageQueue mq : mqs) {
            System.out.println("yo "+mq.getFromId());
            if(mq.getFromId().equals(request.getAttribute("currentuser"))) {
                if(mq.isSeen())
                    received.add(mq);
                else if(mq.isReceived()) {
                    received.add(mq);
                    newmqs.add(mq);
                }
            }
            else newmqs.add(mq);
        }
        session.setAttribute("messagequeue", newmqs);
        if(!received.isEmpty()) {
            System.out.println("sending notification"); %>
            <h4>Notification Received</h4>
        }
        for(MessageQueue mq : received) {
            if(mq.isSeen()) { %>
                <br> <%= mq.getToId() %> has seen your mail "<%=
mq.getSubject()%>"<br>
            } else { %>
                <br> <%= mq.getToId() %> has received your mail "<%=
mq.getSubject()%>"<br>
            }
        }
    }
} %>

</body>
</html>

```

composepage.jsp :

```

<html>
<body>
<h2> Welcome to the Email Application </h2>
<br>
Compose a new mail
<br> <br>
<form action="compose" method="post" >
    <h3>To</h3>
    <input type="text" placeholder="Receiver's gmail id" name="toId">
    <br> <br> <br>
    <h3>Subject</h3>
    <input type="text" name="subject">

    <h3>Body</h3>
    <textarea rows="7" cols="100" name="body"></textarea>
    <br>
    <input type="submit" value="SEND" >

```

```

        <input type="hidden" name="currentuser" value=
<%=request.getParameter("currentuser") %> >
        <input type="hidden" name="password" value= <%=request.getParameter("password") %>
    >
</form>

</body>
</html>

```

viewpage.jsp :

```

<%@page import="javax.mail.Message"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

```

```

Welcome to your mailbox <%= request.getAttribute("currentuser") %>

```

```

<br>

```

```

<h2> Messages : </h2>

```

```

<form action="viewmail.jsp">
<table>
<tr>
<td>From</td>
<td>Subject</td>
<td>Select</td>
</tr>

```

```

<% Message[] messages = (Message[])request.getAttribute("messages");
    for(int i=messages.length-1;i>=0;i--) {
        %>
        <tr>
        <td> <%=messages[i].getFrom()[0] %> </td>
        <td> <%=messages[i].getSubject() %> </td>
        <td> <input type="radio" name="select" value=<%= String.valueOf(i) %> > </td>
        </tr>
        <% }
    %>

```

```

</table>
        <input type="submit" value="view" >
        <input type="hidden" name="currentuser" value=<%= request.getAttribute("currentuser")
%> >
        <input type="hidden" name="password" value=<%= request.getAttribute("password") %> >
</form>

</body>
</html>

```

viewmail.jsp :

```

<%@page import="java.util.ArrayList"%>
<%@page import="javax.mail.Message"%>
<%@page import="com.user.model.*, java.util.Date"%>

```

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

Welcome to your mailbox <%= request.getParameter("currentuser") %>

<br>
<% Message[] messages = (Message[])request.getSession().getAttribute("messages");
    // System.out.println("you selected" + request.getAttribute("select"));
    System.out.println("you selected" + request.getParameter("select"));
    int i = Integer.parseInt((String)request.getParameter("select"));
    Message ms = messages[i];
    if(request.getSession().getAttribute("messagequeue") != null) {
        System.out.println("message queue found");
        System.out.println("requirement " + ms.getFrom()[0] + " currentuser");
        ArrayList<MessageQueue> mqs =
(
ArrayList<MessageQueue>)request.getSession().getAttribute("messagequeue");
        for(MessageQueue mq : mqs) {
            System.out.println(mq.getFromId() + ":" + ms.getFrom()[0]);
            if(mq.getFromId().equals(ms.getFrom()[0].toString()) &&
mq.getToId().equals((String)request.getParameter("currentuser")) &&
mq.getSubject().equals(ms.getSubject())) {
                if(!mq.isSeen()) {
                    mq.setSeenDate(new Date());
                }
                mq.setSeen(true);
                System.out.println("setting true seen");
            }
        }
        request.getSession().setAttribute("messagequeue",mqs);
    }
    else System.out.println("no messagequeue found");
%>
<h2>From</h2>
<%=ms.getFrom()[0] %>
<br> <br>
<h2> Subject</h2>
<%=ms.getSubject() %>
<br>
<h2> Body </h2>
<%=ms.getContent().toString() %>

</body>
</html>

```

LoginController.java :

```

package com.user.controller;

import java.util.ArrayList;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.springframework.stereotype.Controller;

```



```

import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;

import com.user.model.User;

@Controller
public class LoginController {

    @RequestMapping("login")
    public ModelAndView login(@RequestParam("text") String emailId,
        @RequestParam("pwd") String password,
        HttpServletRequest req, HttpServletResponse res ) {
        ModelAndView mv = new ModelAndView();
        mv.setViewName("homepage.jsp");
        User user = new User(emailId, password);
        ArrayList<User> users = null;
        if(req.getSession().getAttribute("users")!=null)
            users = (ArrayList<User>)req.getSession().getAttribute("users");
        else
            users = new ArrayList<User>();
        users = User.addUser(users, user);
        mv.addObject("currentuser",emailId);
        mv.addObject("password", password);
        return mv;
    }
}

```

LogoutController.java :

```

package com.user.controller;

import java.util.ArrayList;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;

import com.user.model.User;

@Controller
public class LoginController {

    @RequestMapping("login")
    public ModelAndView login(@RequestParam("text") String emailId,
        @RequestParam("pwd") String password,
        HttpServletRequest req, HttpServletResponse res ) {
        ModelAndView mv = new ModelAndView();
        mv.setViewName("homepage.jsp");
        User user = new User(emailId, password);
        ArrayList<User> users = null;
        if(req.getSession().getAttribute("users")!=null)
            users = (ArrayList<User>)req.getSession().getAttribute("users");
        else
            users = new ArrayList<User>();
        users = User.addUser(users, user);
    }
}

```

```

        mv.addObject("currentuser",emailId);
        mv.addObject("password", password);
        return mv;
    }
}

```

SendMailController.java :

```
package com.user.controller;
```

```
import java.util.ArrayList;
```

```
import java.util.Date;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestParam;
```

```
import org.springframework.web.servlet.ModelAndView;
```

```
import com.user.model.MessageQueue;
```

```
import com.user.service.SendMailService;
```

```
@Controller
```

```
public class SendMailController {
```

```
    @Autowired
```

```
    SendMailService sendMailService;
```

```
    @RequestMapping("compose")
```

```
    public ModelAndView compose(@RequestParam("toId") String toId,
```

```
    @RequestParam("subject") String subject,
```

```
    @RequestParam("body") String body,HttpServletRequest req,
```

```
    HttpServletResponse res) {
```

```
        MessageQueue mq = new MessageQueue((String)req.getParameter("currentuser"),
        toId, subject, true, false);
```

```
        mq.setReceivedDate(new Date());
```

```
        ArrayList<MessageQueue> mp = new ArrayList<MessageQueue>();
```

```
        if(req.getSession().getAttribute("messagequeue") != null)
```

```
            mp =
```

```
            (ArrayList<MessageQueue>)req.getSession().getAttribute("messagequeue");
```

```
        else System.out.println("no messagequeue found");
```

```
        mp.add(mq);
```

```
        req.getSession().setAttribute("messagequeue", mp);
```

```
        System.out.println(toId);
```

```
        System.out.println(subject);
```

```
        System.out.println(body);
```

```
        sendMailService.sendMail((String)req.getParameter("currentuser"),
        (String)req.getParameter("password"), toId, subject, body);
```

```
        ModelAndView mv = new ModelAndView();
```

```
        mv.addObject("currentuser", (String)req.getParameter("currentuser") );
```

```
        mv.addObject("password", (String)req.getParameter("password") );
```

```
        mv.setViewName("homepage.jsp");
```

```
        return mv;
```

```
    }
```

```
}
```

ViewMailController.java :

```
package com.user.controller;

import java.io.IOException;
import java.util.Properties;

import javax.mail.Authenticator;
import javax.mail.Folder;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.NoSuchProviderException;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Store;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;

@Controller
public class ViewMailController {

    @RequestMapping("viewmails")
    public ModelAndView viewMails(HttpServletRequest req, HttpServletResponse res) {
        ModelAndView mav = new ModelAndView();
        mav.setViewName("viewpage.jsp");
        mav.addObject("currentuser", (String)req.getParameter("currentuser"));
        mav.addObject("password", (String)req.getParameter("password"));
        final String user = (String)req.getParameter("currentuser");
        final String pwd = (String)req.getParameter("password");

        try {
            // create properties field
            Properties properties = new Properties();
            properties.put("mail.pop3.host", "pop.gmail.com");
            properties.put("mail.pop3.port", "995");
            properties.put("mail.pop3.starttls.enable", "true");
            properties.put("mail.pop3.socketFactory.class",
"javax.net.ssl.SSLSocketFactory" );
            //Session emailSession = Session.getDefaultInstance(properties);
            Session emailSession = Session.getDefaultInstance(properties , new
Authenticator() {

                @Override
                protected PasswordAuthentication
getPasswordAuthentication() {
                    return new PasswordAuthentication( user , pwd);
                }
            });
            // create the POP3 store object and connect with the pop server
            Store store = emailSession.getStore("pop3s");

            store.connect("pop.gmail.com", user, pwd);

            // create the folder object and open it
            Folder emailFolder = store.getFolder("INBOX");
            emailFolder.open(Folder.READ_WRITE);

            // retrieve the messages from the folder in an array and print it
```

```

        Message[] messages = emailFolder.getMessages();
        System.out.println("messages.length---" + messages.length);
        int c = 0;
        for (int i = messages.length-1; i >= 0; i--) {
            Message message = messages[i];
            System.out.println("-----");
            System.out.println("Email Number " + (c + 1));
            System.out.println("Subject: " + message.getSubject());
            System.out.println("From: " + message.getFrom()[0]);
            System.out.println("Body: " +
message.getContent().toString());
            c++;
        }
        mav.addObject("messages", messages);
        req.getSession().setAttribute("messages", messages);
    } catch (NoSuchProviderException e) {
        e.printStackTrace();
    } catch (MessagingException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    return mav;
}
}

```

SendMailService.java :

```

package com.user.service;

import java.util.Date;
import java.util.Properties;

import javax.mail.Authenticator;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.AddressException;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

import org.springframework.stereotype.Service;

@Service
public class SendMailService {

    public void sendMail(String fromEmail, String fromPassword, String toEmail, String
subject, String body) {

        try {
            Properties props = new Properties();
            props.put("mail.smtp.host", "smtp.gmail.com"); //SMTP Host
            props.put("mail.smtp.socketFactory.port", "465"); //SSL Port
            props.put("mail.smtp.socketFactory.class",
"javax.net.ssl.SSLSocketFactory"); //SSL Factory Class
            props.put("mail.smtp.auth", "true"); //Enabling SMTP Authentication
            props.put("mail.smtp.port", "465"); //SMTP Port

            final String from = fromEmail;

```

```

        final String pwd = fromPassword;
        //System.out.println("check "+from + " " + pwd);
        //create Authenticator object to pass in Session.getInstance argument
        Authenticator auth = new Authenticator() {
            //override the getPasswordAuthentication method
            protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication(from, pwd);
            }
        };
        Session session = Session.getInstance(props, auth);

        MimeMessage msg = new MimeMessage(session);
        //set message headers
        msg.addHeader("Content-type", "text/HTML; charset=UTF-8");
        msg.addHeader("format", "flowed");
        msg.addHeader("Content-Transfer-Encoding", "8bit");

        msg.setFrom(fromEmail);
        msg.setReplyTo(InternetAddress.parse(toEmail, false));
        msg.setSubject(subject, "UTF-8");
        msg.setText(body, "UTF-8");
        msg.setSentDate(new Date());
        msg.setRecipients(Message.RecipientType.TO,
InternetAddress.parse(toEmail, false));
        System.out.println("Message is ready");
        session.setDebug(true);
        Transport.send(msg);

        System.out.println("EMail Sent Successfully!!");
    } catch (AddressException e) {
        e.printStackTrace();
    } catch (MessagingException e) {
        e.printStackTrace();
    }
}

}
}

```

MessageQueue.java :

```

package com.user.model;

import java.util.Date;

public class MessageQueue {

    String fromId;
    String toId;
    String subject;
    boolean received;
    boolean seen;
    Date receivedDate;
    Date seenDate;

    public MessageQueue(String fromId, String toId, String subject, boolean received,
boolean seen) {
        super();
        this.fromId = fromId;
        this.toId = toId;
        this.received = received;
        this.subject = subject;
    }
}

```

```

        this.seen = seen;
    }
    public String getFromId() {
        return fromId;
    }
    public void setFromId(String fromId) {
        this.fromId = fromId;
    }
    public String getToId() {
        return toId;
    }
    public void setToId(String toId) {
        this.toId = toId;
    }
    public String getSubject() {
        return subject;
    }
    public void setSubject(String subject) {
        this.subject = subject;
    }
    public boolean isReceived() {
        return received;
    }
    public void setReceived(boolean received) {
        this.received = received;
    }
    public boolean isSeen() {
        return seen;
    }
    public void setSeen(boolean seen) {
        this.seen = seen;
    }
    public Date getReceivedDate() {
        return receivedDate;
    }
    public void setReceivedDate(Date receivedDate) {
        this.receivedDate = receivedDate;
    }
    public Date getSeenDate() {
        return seenDate;
    }
    public void setSeenDate(Date seenDate) {
        this.seenDate = seenDate;
    }
}

```

User.java :

```

package com.user.model;

import java.util.ArrayList;

public class User {
    String emailId;
    String password;

    public User(String email,String pwd) {
        this.emailId = email;
        this.password = pwd;
    }

    public static boolean isPresent(ArrayList<User> users, User user) {

```

```

        if(users.isEmpty() || users == null)
            return false;
        for(User usr : users) {
            if(usr.emailId.equals(user.emailId) &&
usr.password.equals(user.password) )
                return true;
        }
        return false;
    }

    public static ArrayList<User> addUser(ArrayList<User> users, User user) {
        if(isPresent(users, user))
            return users;
        else if(users != null) {
            users.add(user);
            return users;
        }
        else {
            users = new ArrayList<User>();
            users.add(user);
            return users;
        }
    }

    public String getId() {
        return emailId;
    }
}

```

OUTPUT

The first screenshot shows the login page of the email application. The browser address bar displays `http://localhost:8080/ass6_mail/`. The page title is "Welcome to the Email Application". It features two input fields: "Email Id" with the value `sduttabhs@gmail.com` and "Pass Word" with masked characters. A "Submit Query" button is located below the password field.

The second screenshot shows the mailbox view after successful login. The browser address bar displays `http://localhost:8080/ass6_mail/register.html?username=sduttabhs@gmail.com&password=happybdays...`. The page title is "Welcome to your mailbox sduttabhs@gmail.com". It contains three buttons: "COMPOSE", "VIEW MAILS", and "LOGOUT".

The third screenshot shows the email composition page. The browser address bar displays `http://localhost:8080/ass6_mail/composepage.jsp?currentuser=sduttabhs@gmail.com&password=happybdays...`. The page title is "Welcome to the Email Application". It includes a "Compose a new mail" section with the following fields: "To" (containing `nplelab111@gmail.com`), "Subject" (containing `hello1`), and "Body" (containing `sample body1`). A "SEND" button is positioned at the bottom left of the body text area.

