```java

import java.awt.Color;
import java.awt.Desktop;
import java.awt.Dimension;
import java.io.*;
import java.util.ArrayList;
import java.util.Date;
import java.util.Iterator;
import java.util.Properties;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.activation.DataHandler;
import javax.mail.Address;
import javax.mail.BodyPart;
import javax.mail.Flags;
import javax.mail.Flags.Flag;
import javax.mail.Folder;
import javax.mail.MessagingException;
import javax.mail.Multipart;
import javax.mail.NoSuchProviderException;
import javax.mail.Session;
import javax.mail.Store;
import javax.mail.URLName;
import javax.mail.internet.MimeBodyPart;
import javax.mail.internet.MimeMessage;
import javax.swing.*;

//Fenetre principale
public class Accueil extends javax.swing.JFrame {


    public Accueil() {
        initComponents();
         this.setSize(new Dimension(1200,650));
         this.setLocationByPlatform(true);
         this.setLocationRelativeTo(null);
    }


    /* Cette connexion consiste en la connexion au serveur de messagerie,
     * recuperration des messages et affichages de leurs entetes*/

    public void connect(boolean auth){

         // Afficher la boite de dialogue pour entrer les paramaitres de connexions.
        if(auth)
        {
             dialog = new  Authentification(this,true);
        }


        // Construire une URL de connexion à partir des paramètres de connexion de la boite de  dialogue.
        StringBuffer connectionUrl = new StringBuffer();
        if(dialog.getServeurSMTP().getText().equals("smtp.gmail.com"))
        {connectionUrl.append("pop3s" + "://");}
        else
        {connectionUrl.append("pop3" + "://");}
        connectionUrl.append(dialog.getNomUser().getText() + ":");
        connectionUrl.append(dialog.getMotPasse() + "@");
        connectionUrl.append(dialog.getServeurPOP().getText() + "/");
       // this.getProfil().setText(dialog.getNomUser().getText());

        // Etablir une session JavaMail et se connecter au serveur.
        Store store = null;
        try {
            // configurer l'objet "Properties" à partir de la boite de dialog.
            props=new Properties();
            if(dialog.getServeurSMTP().getText().equals("smtp.gmail.com"))
            {
            props.setProperty("mail.smtp.starttls.enable", "true");
            props.setProperty("mail.smtp.socketFactory.port", dialog.getPortSMTP().getText());
            props.setProperty("mail.smtp.socketFactory.class","javax.net.ssl.SSLSocketFactory");
            }

            props.setProperty("mail.smtp.host",dialog.getServeurSMTP().getText());
            props.setProperty("mail.smtp.port",dialog.getPortSMTP().getText());
            props.setProperty("mail.smtp.password",dialog.getMotPasse());
            props.setProperty("mail.smtp.auth", "true");

            // Etablir la session javamail à partir de l'objet props et SMTPAuthenticator
            session = Session.getDefaultInstance(props,new
SMTPAuthenticator(dialog.getNomUser().getText(),dialog.getMotPasse(),true));

            // Se connecter au serveur e-mail.
            URLName urln = new URLName(connectionUrl.toString());
            store = session.getStore(urln);
            store.connect();

        } catch (Exception ex) {
            // afficher un message d'erreur.

            Erreur("connexion impossible.", true);
        }


        try {
            //Recuperer le dossier des messages reçus du serveur
            folder = store.getFolder("INBOX");

            // Ouvrir le dossier "INBOX" en mode lecture ecriture.
            folder.open(Folder.READ_WRITE);

            // Recuperer la liste des messages à partir du dossier.
             messages = folder.getMessages();

             // Affichage des messeges reçus et envoyés
             TaskMessagesReçus task=new TaskMessagesReçus(this);
             task.execute();
             afficherMessagesEnvoyés(construireEnvoyes());

            }
        catch (Exception e) {
            // afficher un message d'erreur.
            Erreur("Impossible De Telecharger Les Messages.", true);
        }




    }


    // retourne un tableau de messages à partir du dossier contenant les mesg envoyés stockés
    public javax.mail.Message[] construireEnvoyes() throws FileNotFoundException, MessagingException, IOException
    {File listFiles[] = null;
     listFiles=listFiles(getfichierElementsEnvoyes());
     javax.mail.Message msgs[]=new javax.mail.Message[listFiles.length];

      for(int i=0;i<listFiles.length;i++)
      {
        InputStream source = new FileInputStream(listFiles[i]);
        MimeMessage message = new MimeMessage(session, source);
        msgs[i]=message;
      }
       return msgs;

    }

    //retourne les fichiers contenus dans un repertoire donné en parametre
 public File[] listFiles(String directoryPath){
File[] files = null;
File directoryToScan = new File(directoryPath);
files = directoryToScan.listFiles();
return files;
}


    //Affichage des messages reçus
    public void afficherMessagesReçus(javax.mail.Message msgs[]) throws Exception
    {
        String[] listInfo=new String[3];
        listeMessageReçus=new ArrayList<Message>(); // listMessage est une liste de panneau
                                                     //où chaque panneau va contenir une en-tete de message

            int j=0;
            for(int i = msgs.length - 1; i >= 0; i--){


                // Recuperer l'adresse de l'emetteur
                Address[] senders = msgs[i].getFrom();
                if (senders != null || senders.length > 0) {
                        listInfo[0]=senders[0].toString();
                    } else {
                        listInfo[0]= "[none]";
                    }
                // recuperer l'objet du message
                String subject = msgs[i].getSubject();
                if (subject != null && subject.length() > 0) {
                        listInfo[1]=subject.toString();
                    } else {
                        listInfo[1]= "[none]";
                    }
                //Recupere la date
                 Date date = msgs[i].getSentDate();
                 if (date != null) {
                         listInfo[2]=date.toString();
                    } else {
                        listInfo[2]= "[none]";
                    }
                //Création du JPannel qui va contenir l'entete du message
                Message msg=new Message(listInfo,i);

                Dimension taille=msg.getPreferredSize();
                msg.setBounds(0, j*(taille.height+1),taille.width, taille.height);
                msg.addMouseListener(new java.awt.event.MouseAdapter() {

                    public void mouseReleased(java.awt.event.MouseEvent evt){
                        try {
                            messageMouseReleased1(evt);
                        } catch (Exception ex) {
                            ex.printStackTrace();
                        }
                    }
                 });

                //inserer le pannel du message dans le panneau des messeges reçus

                boiteMessage1.add(msg);
                taille.setSize(taille.width,(j+1)*(taille.height+2));
                boiteMessage1.setPreferredSize(taille);
                listeMessageReçus.add(msg);
                boiteMessage1.validate();
                 j++;
            }
    }

    public void afficherMessagesEnvoyés(javax.mail.Message msgs[]) throws Exception
    {
        String[] listInfo=new String[3];
        listeMessageEnvoyés=new ArrayList<Message>();

            int j=0;
            for(int i = msgs.length - 1; i >= 0; i--){



                Address[] senders = msgs[i].getFrom();
                // Recuperer l'adresse de l'emetteur
                if (senders != null || senders.length > 0) {
                        listInfo[0]=senders[0].toString();
                    } else {
                        listInfo[0]= "[none]";
                    }
                // recuperer l'objet du message
```

```
                    String subject = msgs[i].getSubject();
                    if (subject != null && subject.length() > 0) {
                            listInfo[1]=subject.toString();
                        } else {
                            listInfo[1]= "[none]";
                        }
                    //Recupere la date
                     Date date = msgs[i].getSentDate();
                     if (date != null) {
                             listInfo[2]=date.toString();
                        } else {
                            listInfo[2]= "[none]";
                        }
                    Message msg=new Message(listInfo,i);

                   Dimension  taille=msg.getPreferredSize();
                    msg.setBounds(0, j*(taille.height+1),taille.width, taille.height);
                    msg.addMouseListener(new java.awt.event.MouseAdapter() {
                        public void mouseReleased(java.awt.event.MouseEvent evt){
                            try {
                                messageMouseReleased2(evt);
                            } catch (Exception ex) {
                                ex.printStackTrace();
                            }
                        }
                     });


                    boiteMessage2.add(msg);
                    taille.setSize(taille.width,(j+1)*(taille.height+2));
                    boiteMessage2.setPreferredSize(taille);
                    listeMessageEnvoyés.add(msg);
                    boiteMessage2.validate();
                    j++;
                }
        }

        // crée le nom du dossier des fichiers contenant les msg envoyés et le renvoie
        public String getfichierElementsEnvoyes() {
            fichierElementsEnvoyes="C:/"+dialog.getNomUser().getText()+"_"+dialog.getServeurSMTP().getText();
            File file=new File(fichierElementsEnvoyes);
            if(!file.exists())
            {
                file.mkdir();

            }
            return fichierElementsEnvoyes;
        }
         public String getCarnet() throws IOException {

            carnet="C:/"+dialog.getNomUser().getText()+".txt";
            File f = new File(carnet);

                try {
                   fW = new BufferedWriter(new FileWriter(f));
                   fR = null;
        }
        catch (IOException e)     {


        }


            return carnet;
        }




        // afficher un message d'erreur et quitter si necessaire.
        private void Erreur(String message, boolean exit) {
            JOptionPane.showMessageDialog(this, message, "Erreur",
                    JOptionPane.ERROR_MESSAGE);
            if (exit)
                System.exit(0);
        }

        //Affichege du contenu d'un message reçu au click de l'utilisateur
        private void messageMouseReleased1(java.awt.event.MouseEvent evt)
             {
                  Message msgSelectionné=(Message)evt.getSource();
           try{

            msgSelectionné.setBackground(Color.gray);

            // recuperer le numero du message selectionnÃ©
            int num=msgSelectionné.getId();

             //Ouvrir la boite de dialogue permettant d'afficher le msg
            LireMessage dialog= new LireMessage(this,true,messages[num]);


            //Afficher le message
            dialog.getemetteur().setText(messages[num].getFrom()[0].toString());
            dialog.getdate().setText(messages[num].getSentDate().toString());
            dialog.getobjet().setText(messages[num].getSubject().toString());
            //Afficher les destinataires
            int j=0;
            do
            {
                dialog.getdests().setText(messages[num].getAllRecipients()[j].toString()+"|");
            j++;
            }while (messages[num].getAllRecipients().length<j);

            //rÃ©cuperer le contenu du message
            Object content = messages[ num].getContent();
            //Le contenu est un multipart
            if (content instanceof Multipart) {
                Multipart multipart = (Multipart) content;
                //parcourir tous les bodypart du messages
                for (int x = 0; x < multipart.getCount(); x++) {
                      BodyPart bodyPart = multipart.getBodyPart(x);
                      String disposition = bodyPart.getDisposition();
                      if (disposition != null && (disposition.equals(BodyPart.ATTACHMENT)))
                      {
                          //si le bodypart est une piÃ¨ce jointe on affiche le nom du fichier
                          dialog.getSave().setEnabled(true);
                           DataHandler handler = bodyPart.getDataHandler();


                      } //

                          if (disposition == null) { // Body uniquement

          // Vérifier si de type plain
           String contentType=bodyPart.getContentType();
          if ((contentType.length() >= 10) &&

              (contentType.toLowerCase().substring(

               0, 10).equals("text/plain"))) {

              InputStream is= bodyPart.getInputStream();



             String  msg_body=(new StringUtils()).inputStreamToString(is);
            dialog.setjTextArea1(msg_body);

          }


                          }
                    }

            } else {
                dialog.setjTextArea1( content.toString());
            }

                dialog.setVisible(true);
                msgSelectionné.setBackground(Color.white);

                        Flags flags=messages[num].getFlags();
                     Flags.Flag[] flag=flags.getSystemFlags();
                    for (int h = 0; h < flag.length; h++) {
                    }
           }catch(Exception e){

              Erreur("Echec Lors Du Telechargement Du Message",false);
              msgSelectionné.setBackground(Color.white);
           }

       }

        //Affichege du contenu d'un message reçu au click de l'utilisateur
         private void messageMouseReleased2(java.awt.event.MouseEvent evt)

             {
                 Message msgSelectionné=(Message)evt.getSource();
                 try{

            msgSelectionné.setBackground(Color.gray);

            // recuperer le numero du message selectionné
            int num=msgSelectionné.getId();


            javax.mail.Message messages2[]=construireEnvoyes();

            //Ouvrir la boite de dialogue permettant d'afficher le msg
            LireMessage dialog= new LireMessage(this,true,messages2[num]);

            dialog.getemetteur().setText(messages2[num].getFrom()[0].toString());
            dialog.getdate().setText(messages2[num].getSentDate().toString());
            dialog.getobjet().setText(messages2[num].getSubject().toString());
            int j=0;
            do
            {
                dialog.getdests().setText(messages2[num].getAllRecipients()[j].toString()+"|");
            j++;
            }while (messages2[num].getAllRecipients().length<j);

            //récuperer le contenu du message
            Object content = messages2[ num].getContent();
            //Le contenu est un multipart
           if (content instanceof Multipart) {
                Multipart multipart = (Multipart) content;
                //parcourir tous les bodypart du messages
                for (int x = 0; x < multipart.getCount(); x++) {
                      BodyPart bodyPart = multipart.getBodyPart(x);
                      String disposition = bodyPart.getDisposition();
                      if (disposition != null && (disposition.equals(BodyPart.ATTACHMENT)))
                      {
                          //si le bodypart est une piÃ¨ce jointe on affiche le nom du fichier
                          dialog.getSave().setEnabled(true);
                           DataHandler handler = bodyPart.getDataHandler();


                      } //

                          if (disposition == null) { // Body uniquement

          // Vérifier si de type plain
           String contentType=bodyPart.getContentType();
          if ((contentType.length() >= 10) &&

              (contentType.toLowerCase().substring(

                0, 10).equals("text/plain"))) {

              InputStream is= bodyPart.getInputStream();
             String  msg_body=(new StringUtils()).inputStreamToString(is);
            dialog.setjTextArea1(msg_body);

          }


                          }
                }

            } else {
                dialog.setjTextArea1( content.toString());
            }

                dialog.setVisible(true);
                msgSelectionné.setBackground(Color.white);

        }catch(Exception e){
            Erreur("Echec Lors Du Telechargement Du Message",false);
            msgSelectionné.setBackground(Color.white);
        }
        }

        @SuppressWarnings("unchecked")
        // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents
        private void initComponents() {

            jPanel2 = new javax.swing.JPanel();
            jLabel1 = new javax.swing.JLabel();
            Répondre = new javax.swing.JButton();
            Transferer = new javax.swing.JButton();
            Supprimer = new javax.swing.JButton();
            jButton2 = new javax.swing.JButton();
            Déconnexion = new javax.swing.JButton();
            jPanel1 = new javax.swing.JPanel();
            jPanel3 = new javax.swing.JPanel();
            jPanel5 = new javax.swing.JPanel();
            jButton1 = new javax.swing.JButton();
            Nouveau = new javax.swing.JButton();
            Carnet = new javax.swing.JButton();
            dateChooserPanel1 = new datechooser.beans.DateChooserPanel();
            jTabbedPane1 = new javax.swing.JTabbedPane();
            Recu = new javax.swing.JPanel();
            jScrollPane2 = new javax.swing.JScrollPane();
            boiteMessage1 = new javax.swing.JPanel();
            jPanel4 = new javax.swing.JPanel();
            from = new javax.swing.JLabel();
            object = new javax.swing.JLabel();
            date = new javax.swing.JLabel();
            piece = new javax.swing.JLabel();
            Envoye = new javax.swing.JPanel();
            jScrollPane1 = new javax.swing.JScrollPane();
            boiteMessage2 = new javax.swing.JPanel();
            jPanel6 = new javax.swing.JPanel();
            from1 = new javax.swing.JLabel();
            object1 = new javax.swing.JLabel();
            date1 = new javax.swing.JLabel();
            piece1 = new javax.swing.JLabel();

            setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
            setTitle("UMail Accueil");
            setBackground(new java.awt.Color(255, 255, 255));
            setMinimumSize(new java.awt.Dimension(300, 157));
            setName("accueil"); // NOI18N
            setResizable(false);

            jPanel2.setBackground(new java.awt.Color(255, 255, 255));
            jPanel2.setLayout(new java.awt.GridLayout(1, 0));

            jLabel1.setForeground(new java.awt.Color(255, 255, 255));
            jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/logoUmail.png"))); // NOI18N
            jPanel2.add(jLabel1);

            Répondre.setBackground(new java.awt.Color(0, 102, 102));
            Répondre.setFont(new java.awt.Font("Garamond", 1, 18));
            Répondre.setForeground(new java.awt.Color(255, 255, 255));
            Répondre.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Répondre.png"))); // NOI18N
            Répondre.setText("Répondre");
            Répondre.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent evt) {
                    RépondreActionPerformed(evt);
                }
            });
            jPanel2.add(Répondre);

            Transferer.setBackground(new java.awt.Color(0, 102, 102));
            Transferer.setFont(new java.awt.Font("Garamond", 1, 18));
            Transferer.setForeground(new java.awt.Color(255, 255, 255));
            Transferer.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Transférer.png"))); // NOI18N
            Transferer.setText("Transférer");
            Transferer.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent evt) {
                    TransfererActionPerformed(evt);
                }
            });
            jPanel2.add(Transferer);

            Supprimer.setBackground(new java.awt.Color(0, 102, 102));
            Supprimer.setFont(new java.awt.Font("Garamond", 1, 18));
            Supprimer.setForeground(new java.awt.Color(255, 255, 255));
            Supprimer.setIcon(new javax.swing.ImageIcon(getClass().getResource("/corbeille.png"))); // NOI18N
            Supprimer.setText("Supprimer");
            Supprimer.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent evt) {
                    SupprimerActionPerformed(evt);
                }
            });
            jPanel2.add(Supprimer);

            jButton2.setBackground(new java.awt.Color(0, 102, 102));
            jButton2.setFont(new java.awt.Font("Garamond", 1, 18)); // NOI18N
            jButton2.setForeground(new java.awt.Color(255, 255, 255));
            jButton2.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/aide-point-interrogation-bouclier-icone-5530-48.png"))); // NOI18N
            jButton2.setText("Aide");
            jButton2.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent evt) {
                    jButton2ActionPerformed(evt);
                }
            });
            jPanel2.add(jButton2);

            Déconnexion.setBackground(new java.awt.Color(0, 102, 102));
            Déconnexion.setFont(new java.awt.Font("Garamond", 1, 18));
            Déconnexion.setForeground(new java.awt.Color(255, 255, 255));
            Déconnexion.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Exit.png"))); // NOI18N
            Déconnexion.setText("Déconnexion");
            Déconnexion.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent evt) {
                    DéconnexionActionPerformed(evt);
                }
            });
            jPanel2.add(Déconnexion);

            getContentPane().add(jPanel2, java.awt.BorderLayout.PAGE_START);

            jPanel1.setBackground(new java.awt.Color(255, 255, 255));
            jPanel1.setLayout(new java.awt.BorderLayout());

            jPanel3.setBackground(new java.awt.Color(255, 255, 255));

            jPanel5.setLayout(new java.awt.GridLayout(3, 1));

            jButton1.setBackground(new java.awt.Color(0, 102, 102));
            jButton1.setFont(new java.awt.Font("Garamond", 1, 18));
            jButton1.setForeground(new java.awt.Color(255, 255, 255));
            jButton1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/dossier-mises-a-jour-icone-6204-32.png"))); // NOI18N
            jButton1.setText("Rafraîchir");
            jButton1.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent evt) {
                    jButton1ActionPerformed(evt);
                }
            });
            jPanel5.add(jButton1);

            Nouveau.setBackground(new java.awt.Color(0, 102, 102));
            Nouveau.setFont(new java.awt.Font("Garamond", 1, 18));
            Nouveau.setForeground(new java.awt.Color(255, 255, 255));
            Nouveau.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Mail.png"))); // NOI18N
            Nouveau.setText("Nouveau");
            Nouveau.setPreferredSize(new java.awt.Dimension(60, 60));
            Nouveau.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent evt) {
                    NouveauActionPerformed(evt);
                }
            });
            jPanel5.add(Nouveau);

            Carnet.setBackground(new java.awt.Color(0, 102, 102));
            Carnet.setFont(new java.awt.Font("Garamond", 1, 18));
            Carnet.setForeground(new java.awt.Color(255, 255, 255));
            Carnet.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Carnet.png"))); // NOI18N
            Carnet.setText("Carnet");
            Carnet.addActionListener(new java.awt.event.ActionListener() {
                public void actionPerformed(java.awt.event.ActionEvent evt) {
                    CarnetActionPerformed(evt);
                }
            });
            jPanel5.add(Carnet);

            dateChooserPanel1.setCurrentView(new datechooser.view.appearance.AppearancesList("Umail",
                new datechooser.view.appearance.ViewAppearance("umail",
                    new datechooser.view.appearance.swing.SwingCellAppearance(new java.awt.Font("Tahoma", java.awt.Font.PLAIN, 11),
                        new java.awt.Color(0, 0, 0),
                        new java.awt.Color(0, 0, 255),
                        false,
                        true,
                        new datechooser.view.appearance.swing.ButtonPainter()),
                    new datechooser.view.appearance.swing.SwingCellAppearance(new java.awt.Font("Tahoma", java.awt.Font.PLAIN, 11),
                        new java.awt.Color(0, 0, 0),
                        new java.awt.Color(0, 0, 255),
                        true,
                        true,
                        new datechooser.view.appearance.swing.ButtonPainter()),
                    new datechooser.view.appearance.swing.SwingCellAppearance(new java.awt.Font("Tahoma", java.awt.Font.PLAIN, 11),
                        new java.awt.Color(0, 0, 255),
                        new java.awt.Color(0, 0, 255),
                        false,
                        true,
                        new datechooser.view.appearance.swing.ButtonPainter()),
                    new datechooser.view.appearance.swing.SwingCellAppearance(new java.awt.Font("Tahoma", java.awt.Font.PLAIN, 11),
                        new java.awt.Color(128, 128, 128),
                        new java.awt.Color(0, 0, 255),
                        false,
                        true,
                        new datechooser.view.appearance.swing.LabelPainter()),
                    new datechooser.view.appearance.swing.SwingCellAppearance(new java.awt.Font("Tahoma", java.awt.Font.PLAIN, 11),
                        new java.awt.Color(0, 0, 0),
                        new java.awt.Color(0, 0, 255),
                        false,
                        true,
                        new datechooser.view.appearance.swing.LabelPainter()),
                    new datechooser.view.appearance.swing.SwingCellAppearance(new java.awt.Font("Tahoma", java.awt.Font.PLAIN, 11),
                        new java.awt.Color(0, 0, 0),
                        new java.awt.Color(255, 0, 0),
                        false,
                        false,
                        new datechooser.view.appearance.swing.ButtonPainter()),
                    (datechooser.view.BackRenderer)null,
                    false,
                    true),
                new datechooser.view.appearance.ViewAppearance("Umail",
                    new datechooser.view.appearance.custom.CustomCellAppearance(new java.awt.Color(25
                        new java.awt.Color(0, 51, 51),
                        (javax.swing.border.Border)null,
                        new java.awt.Font("Arial", java.awt.Font.BOLD, 12),
                        new java.awt.Color(0, 0, 153),
                        0.0f),
                    new datechooser.view.appearance.custom.CustomCellAppearance(new java.awt.Color(0, 102, 102),
                        new java.awt.Color(141, 134, 42),
                        (javax.swing.border.Border)null,
                        new java.awt.Font("Serif", java.awt.Font.BOLD, 14),
                        new java.awt.Color(0, 0, 102),
                        0.4f),
                    new datechooser.view.appearance.custom.CustomCellAppearance(new java.awt.Color(44, 153, 143),
                        new java.awt.Color(51, 51, 51),
                        (javax.swing.border.Border)null,
                        new java.awt.Font("Arial", java.awt.Font.BOLD, 12),
                        new java.awt.Color(0, 0, 153),
                        0.5f),
                    new datechooser.view.appearance.custom.CustomCellAppearance(new java.awt.Color(255, 255, 255),
                        new java.awt.Color(0, 102, 102),
                        (javax.swing.border.Border)null,
                        new java.awt.Font("Serif", java.awt.Font.BOLD + java.awt.Font.ITALIC, 10),
                        new java.awt.Color(0, 0, 255),
                        0.0f),
                    new datechooser.view.appearance.custom.CustomCellAppearance(new java.awt.Color(0, 0, 0),
                        new java.awt.Color(255, 255, 255),
                        (javax.swing.border.Border)null,
                        new java.awt.Font("Serif", java.awt.Font.BOLD, 12),
                        new java.awt.Color(0, 0, 255),
                        0.4f),
                    new datechooser.view.appearance.custom.CustomCellAppearance(new java.awt.Color(255, 255, 255),
                        new java.awt.Color(255, 0, 0),
                        (javax.swing.border.Border)null,
                        new java.awt.Font("Serif", java.awt.Font.PLAIN, 12),
                        new java.awt.Color(255, 0, 0),
                        0.0f),
                    new datechooser.view.BackRenderer( 2,
                        "/C:/Users/Nahla/Documents/UMail_fella/src/test_2.png"),
                    true,
                    true),
                new datechooser.view.appearance.ViewAppearance("custom",
                    new datechooser.view.appearance.swing.SwingCellAppearance(new java.awt.Font("Tahoma", java.awt.Font.PLAIN, 11),
                        new java.awt.Color(0, 0, 0),
                        new java.awt.Color(0, 0, 255),
                        false,
                        true,
                        new datechooser.view.appearance.swing.ButtonPainter()),
                    new datechooser.view.appearance.swing.SwingCellAppearance(new java.awt.Font("Tahoma", java.awt.Font.PLAIN, 11),
                        new java.awt.Color(0, 0, 0),
                        new java.awt.Color(0, 0, 255),
                        true,
                        true,
                        new datechooser.view.appearance.swing.ButtonPainter()),
                    new datechooser.view.appearance.swing.SwingCellAppearance(new java.awt.Font("Tahoma", java.awt.Font.PLAIN, 11),
                        new java.awt.Color(0, 0, 255),
                        new java.awt.Color(0, 0, 255),
                        false,
                        true,
                        new datechooser.view.appearance.swing.ButtonPainter()),
                    new datechooser.view.appearance.swing.SwingCellAppearance(new java.awt.Font("Tahoma", java.awt.Font.PLAIN, 11),
                        new java.awt.Color(128, 128, 128),
                        new java.awt.Color(0, 0, 255),
                        false,
                        true,
                        new datechooser.view.appearance.swing.LabelPainter()),
                    new datechooser.view.appearance.swing.SwingCellAppearance(new java.awt.Font("Tahoma", java.awt.Font.PLAIN, 11),
                        new java.awt.Color(0, 0, 0),
                        new java.awt.Color(0, 0, 255),
                        false,
                        true,
                        new datechooser.view.appearance.swing.LabelPainter()),
                    new datechooser.view.appearance.swing.SwingCellAppearance(new java.awt.Font("Tahoma", java.awt.Font.PLAIN, 11),
                        new java.awt.Color(0, 0, 0),
                        new java.awt.Color(255, 0, 0),
                        false,
                        false,
                        new datechooser.view.appearance.swing.ButtonPainter()),
                    (datechooser.view.BackRenderer)null,
                    false,
                    true)));
        dateChooserPanel1.setCalendarBackground(new java.awt.Color(255, 255, 255));
        dateChooserPanel1.setLocked(true);

        javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
        jPanel3.setLayout(jPanel3Layout);
        jPanel3Layout.setHorizontalGroup(
            jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel5, javax.swing.GroupLayout.DEFAULT_SIZE, 195, Short.MAX_VALUE)
            .addComponent(dateChooserPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, 195, Short.MAX_VALUE)
        );
        jPanel3Layout.setVerticalGroup(
            jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel3Layout.createSequentialGroup()
                .addGap(44, 44, 44)
                .addComponent(jPanel5, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(dateChooserPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, 313, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(263, Short.MAX_VALUE))
        );

        jPanel1.add(jPanel3, java.awt.BorderLayout.LINE_START);

        jTabbedPane1.setBackground(new java.awt.Color(234, 224, 190));
        jTabbedPane1.setFont(new java.awt.Font("Garamond", 1, 14));

        Recu.setLayout(new java.awt.BorderLayout());

        jScrollPane2.setHorizontalScrollBarPolicy(javax.swing.ScrollPaneConstants.HORIZONTAL_SCROLLBA

        boiteMessage1.setBackground(new java.awt.Color(255, 255, 255));

        javax.swing.GroupLayout boiteMessage1Layout = new javax.swing.GroupLayout(boiteMessage1);
        boiteMessage1.setLayout(boiteMessage1Layout);
        boiteMessage1Layout.setHorizontalGroup(
            boiteMessage1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 1793, Short.MAX_VALUE)
        );
        boiteMessage1Layout.setVerticalGroup(
            boiteMessage1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 762, Short.MAX_VALUE)
        );

        jScrollPane2.setViewportView(boiteMessage1);

        Recu.add(jScrollPane2, java.awt.BorderLayout.CENTER);

        jPanel4.setBackground(new java.awt.Color(234, 224, 190));
        jPanel4.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 102, 102)));
        jPanel4.setPreferredSize(new java.awt.Dimension(1773, 45));

        from.setFont(new java.awt.Font("Tahoma", 1, 11));
        from.setText("Emetteur ");

        object.setFont(new java.awt.Font("Tahoma", 1, 11));
        object.setText("Objet");

        date.setFont(new java.awt.Font("Tahoma", 1, 11));
        date.setText("Date");

        javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);
        jPanel4.setLayout(jPanel4Layout);
        jPanel4Layout.setHorizontalGroup(
            jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel4Layout.createSequentialGroup()
                .addGap(40, 40, 40)
                .addComponent(piece, javax.swing.GroupLayout.PREFERRED_SIZE, 17, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(from, javax.swing.GroupLayout.PREFERRED_SIZE, 311, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(object, javax.swing.GroupLayout.PREFERRED_SIZE, 266, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(date, javax.swing.GroupLayout.DEFAULT_SIZE, 1077, Short.MAX_VALUE))
        );
        jPanel4Layout.setVerticalGroup(
            jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(object, javax.swing.GroupLayout.PREFERRED_SIZE, 27, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(date, javax.swing.GroupLayout.PREFERRED_SIZE, 43, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(from, javax.swing.GroupLayout.PREFERRED_SIZE, 38, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(jPanel4Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(piece, javax.swing.GroupLayout.DEFAULT_SIZE, 21, Short.MAX_VALUE)
                .addContainerGap())
        );

        Recu.add(jPanel4, java.awt.BorderLayout.PAGE_START);

        jTabbedPane1.addTab("Messages Reçus", Recu);

        Envoye.setLayout(new java.awt.BorderLayout());

        jScrollPane1.setHorizontalScrollBarPolicy(javax.swing.ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);

        boiteMessage2.setBackground(new java.awt.Color(255, 255, 255));

        javax.swing.GroupLayout boiteMessage2Layout = new javax.swing.GroupLayout(boiteMessage2);
        boiteMessage2.setLayout(boiteMessage2Layout);
        boiteMessage2Layout.setHorizontalGroup(
            boiteMessage2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 1790, Short.MAX_VALUE)
        );
        boiteMessage2Layout.setVerticalGroup(
            boiteMessage2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 762, Short.MAX_VALUE)
        );

        jScrollPane1.setViewportView(boiteMessage2);

        Envoye.add(jScrollPane1, java.awt.BorderLayout.CENTER);

        jPanel6.setBackground(new java.awt.Color(234, 224, 190));
        jPanel6.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 102, 102)));
        jPanel6.setPreferredSize(new java.awt.Dimension(1773, 45));

        from1.setFont(new java.awt.Font("Tahoma", 1, 11));
        from1.setText("Emetteur ");

        object1.setFont(new java.awt.Font("Tahoma", 1, 11));
        object1.setText("Objet");

        date1.setFont(new java.awt.Font("Tahoma", 1, 11));
        date1.setText("Date");

        javax.swing.GroupLayout jPanel6Layout = new javax.swing.GroupLayout(jPanel6);
        jPanel6.setLayout(jPanel6Layout);
        jPanel6Layout.setHorizontalGroup(
            jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel6Layout.createSequentialGroup()
                .addGap(59, 59, 59)
                .addComponent(piece1, javax.swing.GroupLayout.PREFERRED_SIZE, 17, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(from1, javax.swing.GroupLayout.PREFERRED_SIZE, 311, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(object1, javax.swing.GroupLayout.PREFERRED_SIZE, 266, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(date1, javax.swing.GroupLayout.DEFAULT_SIZE, 1058, Short.MAX_VALUE))
        );
        jPanel6Layout.setVerticalGroup(
            jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(object1, javax.swing.GroupLayout.PREFERRED_SIZE, 27, javax.swing.Group
                .addComponent(date1, javax.swing.GroupLayout.PREFERRED_SIZE, 43, javax.swing.GroupLa
                .addComponent(from1, javax.swing.GroupLayout.PREFERRED_SIZE, 38, javax.swing.GroupLa
            .addGroup(jPanel6Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(piece1, javax.swing.GroupLayout.DEFAULT_SIZE, 21, Short.MAX_VALUE)
                .addContainerGap())
        );

        Envoye.add(jPanel6, java.awt.BorderLayout.PAGE_START);

        jTabbedPane1.addTab("Messages Envoyés ", Envoye);

        jPanel1.add(jTabbedPane1, java.awt.BorderLayout.CENTER);

        getContentPane().add(jPanel1, java.awt.BorderLayout.CENTER);

        pack();
        }// </editor-fold>//GEN-END:initComponents

        private void DéconnexionActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_DéconnexionActionPerformed
        try{
            folder.close(true);
            System.exit(0);
        }catch(Exception e){
            Erreur("Folder not closed!!",false);
        }

    // TODO add your handling code here:}//GEN-LAST:event_DéconnexionActionPerformed
        }

            private void NouveauActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_NouveauActionPerformed

            editeur = new SendEmail(this, true);// TODO add your handling code here:}//GEN-LAST:event_NouveauActionPerformed
            }

            //transferer un message
            private void TransfererActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_TransfererActionPerformed
            if(jTabbedPane1.getSelectedComponent().equals(Recu)) {
            int i=0;boolean selected =false;
            while(i<listeMessageReçus.size() && !selected)
                {
                  if(listeMessageReçus.get(i).getSelectionner().isSelected()){
                      selected=true;
                  }
                  i++;
               }
            if(selected==false){Erreur("Aucun Message Selectionné",false);}
            else{

                int j= listeMessageReçus.get(i-1).getId();
                editeur= new SendEmail(this,true);
                 Object content;

                    try {
                        content = messages[j].getContent();
                         if (content instanceof Multipart) {
                Multipart multipart = (Multipart) content;
                //parcourir tous les bodypart du messages
                for (int x = 0; x < multipart.getCount(); x++) {
                      BodyPart bodyPart = multipart.getBodyPart(x);
                      String disposition = bodyPart.getDisposition();
                      if (disposition != null && (disposition.equals(BodyPart.ATTACHMENT)))
                      {

                      }

                          if (disposition == null) { // Body uniquement

          // Vérifier si de type plain
           String contentType=bodyPart.getContentType();
          if ((contentType.length() >= 10) &&

               (contentType.toLowerCase().substring(

                0, 10).equals("text/plain"))) {

               InputStream is= bodyPart.getInputStream();



              String  msg_body=(new StringUtils()).inputStreamToString(is);
            editeur.getHTMLeditor().setHTMLContent(msg_body);

          }
                             }
                 }

            } else {
                editeur.getHTMLeditor().setHTMLContent( content.toString());
            }


                 }



                     catch (IOException ex) {

                    } catch (MessagingException ex) {

                    }



            // TODO add your handling code here:}//GEN-LAST:event_TransfererActionPerformed
    }}}

      // supprime le messeges selectionné par l'utilisateur
    private void SupprimerActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SupprimerActionPerformed
         //si l'onglet des messages reçus est actif
        if(jTabbedPane1.getSelectedComponent().equals(Recu)) {
               boolean select=false;
               for(int i=0;i<listeMessageReçus.size();i++){
                  if(listeMessageReçus.get(i).getSelectionner().isSelected()){
                      supprimerMessageReçu(i);select=true;
                  }
               }
               // si aucun message n'est selectionné afficher un msg d'erreur
               if(select==false){Erreur("Aucun Message Selectionné",false);}
           }else { // si l'onglet des messages envoyés est actif
           if(jTabbedPane1.getSelectedComponent().equals(Envoye)){
               boolean select2=false;
               for(int i=0;i<listeMessageEnvoyés.size();i++){
                  if(listeMessageEnvoyés.get(i).getSelectionner().isSelected()){
                      supprimerMessageEnvoyé(i);select2=true;
                  }
               }
               // si aucun message n'est selectionné afficher un msg d'erreur
               if(select2==false){Erreur("Aucun Message Selectionné",false);}
               }
    }

    }//GEN-LAST:event_SupprimerActionPerformed
      //pour supprimer un message reçu
    public void supprimerMessageReçu(int i){

        try{
        //attribuer au message selectioné un flag  "Flags.Flag.DELETED"
        messages[messages.length-i-1].setFlag(Flags.Flag.DELETED,true);
        //Suprimer le messages du panneau des messages reçus
        listeMessageReçus.remove(i);
        //réafficher le panneau
        Iterator<Message> it =listeMessageReçus.iterator();
        boiteMessage1.removeAll();
        int j=0;
        while(it.hasNext()){
              Message msg=it.next();
              Dimension taille=msg.getPreferredSize();
              msg.setBounds(0, j*(taille.height+1),taille.width, taille.height);
              boiteMessage1.add(msg);
              j++;
        }
        boiteMessage1.setVisible(false);
        boiteMessage1.setVisible(true);
        boiteMessage1.validate();

        }
        catch (Exception e){
            Erreur("Echec De La Suppression",false);
        }

        }
      //pour supprimer un messages envoyé
    public void supprimerMessageEnvoyé(int i){
        //recuperer un tableau de fichier à partir du dossier des messages envoyés
         File[] listFiles=listFiles(fichierElementsEnvoyes);
         //supprimer le fichier du messages selectionné
         listFiles[i].delete();
         //supprimer le messages du panneau des msg envoyés
        listeMessageEnvoyés.remove(i);
        // réafficher le panneau
        Iterator<Message> it =listeMessageEnvoyés.iterator();
        boiteMessage2.removeAll();
        int j=0;
        while(it.hasNext()){
             Message msg=it.next();
            Dimension taille=msg.getPreferredSize();
            msg.setBounds(0, j*(taille.height+1),taille.width, taille.height);
            boiteMessage2.add(msg);
            j++;
        }
        boiteMessage2.setVisible(false);
        boiteMessage2.setVisible(true);
        boiteMessage2.validate();

        }

    private void CarnetActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_CarnetActionPerformed
    // TODO add your handling code here:
        Carnet_View carnet;

                carnet = new Carnet_View(this,true);
                carnet.setVisible(true);



    }//GEN-LAST:event_CarnetActionPerformed

    private void RépondreActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_RépondreActionPerformed

        if(jTabbedPane1.getSelectedComponent().equals(Recu)) {
            int i=0;boolean selected =false;
            while(i<listeMessageReçus.size() && !selected)
                {
                  if(listeMessageReçus.get(i).getSelectionner().isSelected()){
                      selected=true;
                  }
                  i++;
               }
            if(selected==false){Erreur("Aucun Message Selectionné",false);}
            else{

                editeur= new SendEmail(this, true);
                int j = listeMessageReçus.get(i-1).getId();
                    try {
                        editeur.setdestinataire(messages[j].getFrom()[0].toString());
                    } catch (MessagingException ex) {
                        Logger.getLogger(Accueil.class.getName()).log(Level.SEVERE, null, ex);
                    }
                editeur.setVisible(true);
            }
        }
    }//GEN-LAST:event_RépondreActionPerformed

        private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jBut
            try{
            folder.close(true);

        }catch(Exception e){
            Erreur("Folder not closed!!",false);
        }

            this.connect(false);
            this.setVisible(true);
        }//GEN-LAST:event_jButton1ActionPerformed

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton2ActionPerformed
    // TODO add your handling code here:
        try {

            File chmFile = new File("Aide.chm");
            if (chmFile.exists()) {

                if (Desktop.isDesktopSupported()) {
                    Desktop.getDesktop().open(chmFile);
                } else {
                }

            } else {
            }


          } catch (Exception ex) {
          }

    }//GEN-LAST:event_jButton2ActionPerformed



        public Session getSession() {
            return session;
        }

        public Authentification getDialog() {
            return dialog;
        }




        public static void main(String args[])  {
            /* Set the Nimbus look and feel */
            //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
            /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
             * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
             */
            try {
                for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                    if ("Nimbus".equals(info.getName())) {
                        javax.swing.UIManager.setLookAndFeel(info.getClassName());
                        break;
                    }
                }
            } catch (ClassNotFoundException ex) {
                java.util.logging.Logger.getLogger(Accueil.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
            } catch (InstantiationException ex) {
                java.util.logging.Logger.getLogger(Accueil.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
            } catch (IllegalAccessException ex) {
                java.util.logging.Logger.getLogger(Accueil.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
            } catch (javax.swing.UnsupportedLookAndFeelException ex) {
                java.util.logging.Logger.getLogger(Accueil.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
            }
            //</editor-fold>
                  try {
                UIManager.setLookAndFeel("com.jtattoo.plaf.acryl.AcrylLookAndFeel");

            } catch (ClassNotFoundException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            } catch (InstantiationException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            } catch (IllegalAccessException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            } catch (UnsupportedLookAndFeelException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

            /* Create and display the form */
            java.awt.EventQueue.invokeLater(new Runnable() {

                  public void run() {
                        Accueil client = new Accueil();
                        client.connect(true);
                        client.setVisible(true);



                }
            });

        }

        public javax.mail.Message[] getMessages() {
            return messages;
        }











        private javax.mail.Message[] messages;
        private Folder folder;
```

```java
    private Session session;
    private Properties props;
    private Authentification dialog;
    private SendEmail editeur;
    private String fichierElementsEnvoyes;
    private String fichierMessagesReçus;
    private String carnet;
    private RoundButton rButton;
    private ArrayList<Message> listeMessageReçus;
    private ArrayList<Message> listeMessageEnvoyés;
    private   BufferedWriter fW;
    private   BufferedReader fR;
    ImageIcon iconPhoto = new ImageIcon("piece-jointe-16.png");
     java.awt.Image uneImage= iconPhoto.getImage();

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton Carnet;
    private javax.swing.JButton Déconnexion;
    private javax.swing.JPanel Envoye;
    private javax.swing.JButton Nouveau;
    private javax.swing.JPanel Recu;
    private javax.swing.JButton Répondre;
    private javax.swing.JButton Supprimer;
    private javax.swing.JButton Transferer;
    private javax.swing.JPanel boiteMessage1;
    private javax.swing.JPanel boiteMessage2;
    private javax.swing.JLabel date;
    private javax.swing.JLabel date1;
    private datechooser.beans.DateChooserPanel dateChooserPanel1;
    private javax.swing.JLabel from;
    private javax.swing.JLabel from1;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JPanel jPanel4;
    private javax.swing.JPanel jPanel5;
    private javax.swing.JPanel jPanel6;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JTabbedPane jTabbedPane1;
    private javax.swing.JLabel object;
    private javax.swing.JLabel object1;
    private javax.swing.JLabel piece;
    private javax.swing.JLabel piece1;
    // End of variables declaration//GEN-END:variables


}
```