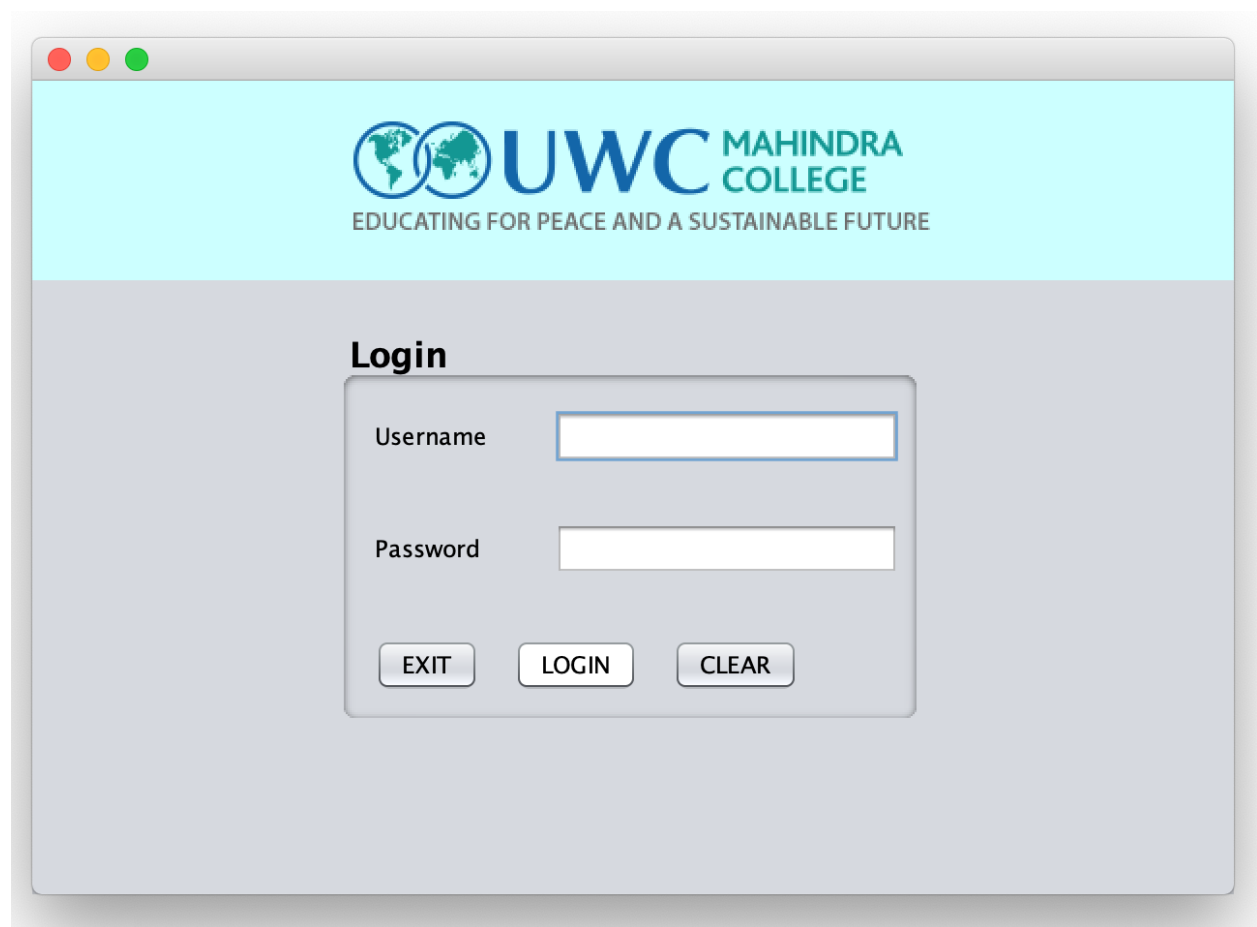


Criterion C

Techniques Used	Logical Reasoning
Use of a Relational Database Management System (RDBMS), in this case SQLite in order to handle data on the databases in the back-end.	Use of an RDBMS instead of text files was required because of the features of the RDBMS and the included security of the data in SQLite.
Use of Java Swing in Netbeans in order to make text fields to input data, password fields, buttons to execute functions, tables to show the records of the database and lastly pictures and panels.	This was done to make sure that it was easy for the user to carry out desired functions and to make it more user friendly.
Use of Object Oriented Programming with the use of constructors, objects and connection of different classes and extending classes.	OOP was used in order to avoid complication in making the code and OOP helped by making the code shorter than it would be without OOP.
Use of Encapsulation in terms of making classes private, public and also writing methods in order to call them back in another class	One way that encapsulation helped was when one variable was called multiple times in different classes but it did not create an interruption. Moreover, one method for connection of the project on Netbeans to SQLite has been called multiple times across all the classes.
Using various java libraries such as javax mail, itext and mysql connector.	To establish connection with SQLite database, to send emails using the SMTP server on gmail and lastly to generate pdf reports outside the program itself
Use of DB Browser for SQLite	DB Browser for SQLite was a separate application that had to be downloaded and used in order to manage the databases in SQLite. Setting parameters, datatypes and etc were done on DB Browser for SQLite.
Use of try-catch blocks	This was to make sure that parts of the program for example the check for login credentials could handle errors.

Techniques Used	Logical Reasoning
Use “finally” try-catch blocks in order to close queries which were establishing connections to the database	This try-catch block is used to stop the previous query from accessing the database in order for the next query to be able to access the database.

Login Screen



The screenshot shows a web application window with a light blue header and a grey body. The header contains the UWC Mahindra College logo, which consists of two overlapping globes and the text "UWC MAHINDRA COLLEGE" and "EDUCATING FOR PEACE AND A SUSTAINABLE FUTURE". The main content area is titled "Login" and contains a form with two input fields: "Username" and "Password". Below the input fields are three buttons: "EXIT", "LOGIN", and "CLEAR".



UWC MAHINDRA COLLEGE
EDUCATING FOR PEACE AND A SUSTAINABLE FUTURE

Login

Username

Password

```

private void cmd_loginActionPerformed(java.awt.event.ActionEvent evt) {
    String sql ="select * from UserDetails where Username=? and Password=?";
    try{
        pst=conn.prepareStatement(sql);
        pst.setString(1,txt_username.getText());
        pst.setString(2,txt_password.getText());
        rs=pst.executeQuery();

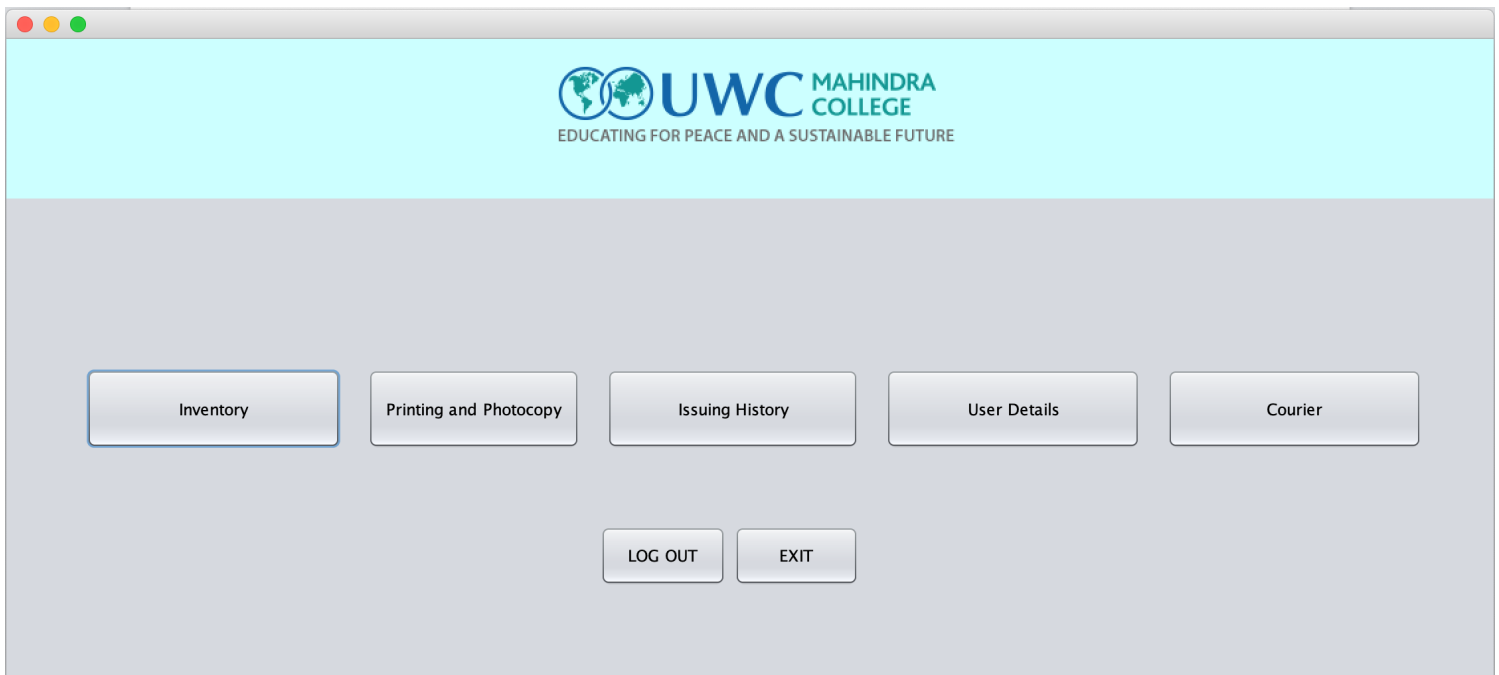
        if(rs.next()){
            JOptionPane.showMessageDialog(null, "Welcome to Reprographics!");
            close();
            User_Details s =new User_Details();
            s.setVisible(false);
            Reprographicsdata a =new Reprographicsdata();
            a.setVisible(true);
        }

        else{
            JOptionPane.showMessageDialog(null, "Access Denied");
        }
    } catch(Exception e){
        JOptionPane.showMessageDialog(null, e);
    } finally{
        try{
            rs.close();
            pst.close();
        } catch(Exception e){
        }
    }
}
}

```

This method matches the input credentials in the connected database table which is called User Details. If the credentials input are correct then the user is shown a JOptionPane with a message saying, ‘Welcome to Reprographics’ to confirm that the user is logged in. If the credentials are not correct then the JOptionPane shows the message, ‘Access Denied’.

Reprographics Main Menu



```

private void Printing_Photocopy_menuActionPerformed(java.awt.event.ActionEvent evt) {
    Printing_and_Photocopy a =new Printing_and_Photocopy();
    a.setVisible(true);
    close();
}

private void inventory_menuActionPerformed(java.awt.event.ActionEvent evt) {
    Inventory a =new Inventory();
    a.setVisible(true);
    close();
}

private void issuing_history_menuActionPerformed(java.awt.event.ActionEvent evt) {
    Issuing_History a =new Issuing_History();
    a.setVisible(true);
    close();
}

private void User_detailsActionPerformed(java.awt.event.ActionEvent evt) {
    User_Details a =new User_Details();
    a.setVisible(true);
    close();
}

private void cmd_exitActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}


private void cmd_logoutActionPerformed(java.awt.event.ActionEvent evt) {
    Loginpage a =new Loginpage();
    a.setVisible(true);
    close();// TODO add your handling code here:
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    Courier a =new Courier();
    a.setVisible(true);
    close();// TODO add your handling code here:
}

```

The code above is for each of the buttons on the menu screen which leads to different screens. Each of this button has the links set with the respective screens. Other than the buttons used to access the screens, there is a button which exits the program and another button to log out which takes the user back to the login page.

Inventory


UWC MAHINDRA COLLEGE
 EDUCATING FOR PEACE AND A SUSTAINABLE FUTURE

Search

File Chooser

Inventory
 Product ID
 Item Name
 Type of Item
 Units in Inventory

ID_Number	Item_Name	Type_of_Item	Units_In_Inventory
1	Pentel White Bo...	White Board Ma...	69
2	Pilot V5 Techp...	Gel Pen Blue	10
3	Blue Chart Paper	Project Paper	100
4	Classmate Spir...	Notebook	69
5	Pelikan Stapler	Stapler	10
6	Deli Dust Free ...	Eraser	15

```

private void cmd_updateActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        String value1 = txt_product_id.getText();
        String value2 = txt_item_name.getText();
        String value3 = txt_type_of_item.getText();
        String value4 = txt_units_in_inventory.getText();
        String sql= "update Inventory set ID_Number='"+value1+"' ,Item_Name='"+value2+"' ,Type_of_Item='"+value3+"' ,Units_In_Inventory='"+value4+"' where ID_Num
        pst=conn.prepareStatement(sql);
        pst.execute();
        JOptionPane.showMessageDialog(null, "Updated");
    }
    catch(Exception e){
        JOptionPane.showMessageDialog(null, e);
    }
    Update_table();
}

```

This is the code used for the Update button. After the user edits a record on the JTextfields next to the JTable, they should press this button in order to save the changes made to the record on the database on SQLite. **This same code has been used on other classes such as in Printing and Photocopy, Issuing History, User Details and Courier except they are connected to seperate databases on SQLite.**

```

private void Table_inventoryMouseClicked(java.awt.event.MouseEvent evt) {
    try{
        int row = Table_inventory.getSelectedRow();
        String Table_click= (Table_inventory.getModel().getValueAt(row, 0).toString());
        String sql ="select * from Inventory where ID_Number='"+Table_click+"'";
        pst=conn.prepareStatement(sql);
        rs=pst.executeQuery();
        if(rs.next()){
            String add1=rs.getString("ID_Number");
            txt_product_id.setText(add1);
            String add2=rs.getString("Item_Name");
            txt_item_name.setText(add2);
            String add3=rs.getString("Type_of_Item");
            txt_type_of_item.setText(add3);
            String add4=rs.getString("Units_In_Inventory");
            txt_units_in_inventory.setText(add4);
        }
    }
    catch(Exception e){
        JOptionPane.showMessageDialog(null, e);
    }
    finally{
        try{
            rs.close();
            pst.close();
        }
        catch(Exception e){
        }
    }
}

```

This is the code for the records of the database to be displayed on the table to show on the JTextFields every time the user clicks a particular record.

```

private void backActionPerformed(java.awt.event.ActionEvent evt) {
    Reprographicsdata a =new Reprographicsdata();
    a.setVisible(true);
    close();// TODO add your handling code here:
}


private void cmd_saveActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        String sql= "Insert into Inventory (ID_Number,Item_Name,Type_of_Item,Units_In_Inventory) values (?,?,?,?)";
        pst=conn.prepareStatement(sql);
        pst.setString(1, txt_product_id.getText());
        pst.setString(2, txt_item_name.getText());
        pst.setString(3, txt_type_of_item.getText());
        pst.setString(4, txt_units_in_inventory.getText());
        pst.execute();
        JOptionPane.showMessageDialog(null, "Saved");
    }
    catch(Exception e){
        JOptionPane.showMessageDialog(null, e);
    }
    Update_table();
}

private void cmd_deleteActionPerformed(java.awt.event.ActionEvent evt) {
    String sql= "delete from Inventory where ID_Number=?";
    try{
        pst=conn.prepareStatement(sql);
        pst.setString(1, txt_product_id.getText());
        pst.execute();
        JOptionPane.showMessageDialog(null, "Deleted");
    }
    catch(Exception e){
        JOptionPane.showMessageDialog(null, e);
    }
    Update_table();
}

```

The code above is for the user to save a new record input in the JTextFields and for the user to delete a whole record after selecting it on the JTable.

Printing and Photocopy



UWC MAHINDRA COLLEGE
EDUCATING FOR PEACE AND A SUSTAINABLE FUTURE

Search

File Chooser

Printing and Photocopy

ID

Name

Work Type

Quantity

Paper Type

ID	Name	Work_Type	Quantity	Paper_Type
C20007	Abdullah R...	Photocopy	3	A4 Paper
C20006	Numair Islam	Photocopy	13	A4 Paper

```

private void Table_PnPMouseClicked(java.awt.event.MouseEvent evt) {
try{
    int row = Table_PnP.getSelectedRow();
    String Table_click= (Table_PnP.getModel().getValueAt(row, 0).toString());
    String sql ="select * from Printing_and_Photocopy where ID='"+Table_click+"'";
    pst=conn.prepareStatement(sql);
    rs=pst.executeQuery();
    if(rs.next()){
        String add1=rs.getString("ID");
        txt_id.setText(add1);
        String add2=rs.getString("Name");
        txt_name.setText(add2);
        String add3=rs.getString("Work_Type");
        txt_work_type.setText(add3);
        String add4=rs.getString("Quantity");
        txt_quantity.setText(add4);
        String add5=rs.getString("Paper_Type");
        txt_choice_of_paper.setText(add5);
    }
}
catch(Exception e){
    JOptionPane.showMessageDialog(null, e);
}
finally{
    try{
        rs.close();
        pst.close();
    }
    catch(Exception e){
    }
}
} // TODO add your handling code here:
}

```

The code above is the code to display the records of the database on the JTextfields right beside the JTable.

```

private void txt_searchKeyReleased(java.awt.event.KeyEvent evt) {
try{
    String sql= "select * from Printing_and_Photocopy where Name=?";
    pst=conn.prepareStatement(sql);
    pst.setString(1, txt_search.getText());
    rs=pst.executeQuery();
    if(rs.next()){
        String add1=rs.getString("ID");
        txt_id.setText(add1);
        String add2=rs.getString("Name");
        txt_name.setText(add2);
        String add3=rs.getString("Work_Type");
        txt_work_type.setText(add3);
        String add4=rs.getString("Quantity");
        txt_quantity.setText(add4);
        String add5=rs.getString("Paper_Type");
        txt_choice_of_paper.setText(add5);
    }
}
catch(Exception e){
    JOptionPane.showMessageDialog(null, e);
}
finally{
    try{
        rs.close();
        pst.close();
    }
    catch(Exception e){
    }
}
}

try{
    String sql= "select * from Printing_and_Photocopy where ID=?";
    pst=conn.prepareStatement(sql);
    pst.setString(1, txt_search.getText());
    rs=pst.executeQuery();
    if(rs.next()){
        String add1=rs.getString("ID");
        txt_id.setText(add1);
        String add2=rs.getString("Name");
        txt_name.setText(add2);
        String add3=rs.getString("Work_Type");
        txt_work_type.setText(add3);
        String add4=rs.getString("Quantity");
        txt_quantity.setText(add4);
        String add5=rs.getString("Paper_Type");
        txt_choice_of_paper.setText(add5);
    }
}
catch(Exception e){
    JOptionPane.showMessageDialog(null, e);
}
finally{
    try{
        rs.close();
        pst.close();
    }
    catch(Exception e){
    }
}
}

```

```

try{
    String sql= "select * from Printing_and_Photocopy where Work_Type=?";
    pst=conn.prepareStatement(sql);
    pst.setString(1, txt_search.getText());
    rs=pst.executeQuery();
    if(rs.next()){
        String add1=rs.getString("ID");
        txt_id.setText(add1);
        String add2=rs.getString("Name");
        txt_name.setText(add2);
        String add3=rs.getString("Work_Type");
        txt_work_type.setText(add3);
        String add4=rs.getString("Quantity");
        txt_quantity.setText(add4);
        String add5=rs.getString("Paper_Type");
        txt_choice_of_paper.setText(add5);
    }
}
catch(Exception e){
    JOptionPane.showMessageDialog(null, e);
}finally{
    try{
        rs.close();
        pst.close();
    }
    catch(Exception e){
    }
}

try{
    String sql= "select * from Printing_and_Photocopy where Paper_Type=?";
    pst=conn.prepareStatement(sql);
    pst.setString(1, txt_search.getText());
    rs=pst.executeQuery();
    if(rs.next()){
        String add1=rs.getString("ID");
        txt_id.setText(add1);
        String add2=rs.getString("Name");
        txt_name.setText(add2);
        String add3=rs.getString("Work_Type");
        txt_work_type.setText(add3);
        String add4=rs.getString("Quantity");
        txt_quantity.setText(add4);
        String add5=rs.getString("Paper_Type");
        txt_choice_of_paper.setText(add5);
    }
}
catch(Exception e){
    JOptionPane.showMessageDialog(null, e);
}finally{
    try{
        rs.close();
        pst.close();
    }
    catch(Exception e){
    }
}
}

```

The code above is for the search bar. Whenever any data from any of the columns of the records are typed into the search bar, the records matching with the information on the database appears on all of the JTextFields. **This code has also been used in all the other classes in order to make search functional, but the different classes are connected to separate databases.**

Issuing History

UWC MAHINDRA COLLEGE
EDUCATING FOR PEACE AND A SUSTAINABLE FUTURE

Issuing History

Search

File Chooser

Send Email

ID	Name	Issued_Item	Quantity_Iss...	Email
C20007	Raib Mukt...	Gel Pen Blue	3	raibm@m...

This is the screen for the Issuing History

```
private void cmd_updateActionPerformed(java.awt.event.ActionEvent evt) {  
try{  
    String value1 = txt_id.getText();  
    String value2 = txt_name.getText();  
    String value3 = txt_issueditem.getText();  
    String value4 = txt_quantityissued.getText();  
    String value5 = txt_email.getText();  
    String sql= "update Issuing_History set ID='"+value1+"' ,Name='"+value2+"' ,Issued_Item='"+value3+"' ,Quantity_Issued='"+value4+"' ,Email='"+value5+"' wh  
    pst=conn.prepareStatement(sql);  
    pst.execute();  
    JOptionPane.showMessageDialog(null, "Updated");  
}  
catch(Exception e){  
    JOptionPane.showMessageDialog(null, e);  
}  
Update_table();  
}
```

This is the code for the system to update the information on the database with the information input into the JTextFields by the user. **This code has been reused in the different classes including Printing and Photocopy, User Details, Inventory and Courier.**

User Details



Id	Username	Password	admin_or_not	Name
2	raqib	raqib	admin	Raqib
3	reva	reva	standard	Reva
4	simon	simon	standard	Simon
5	hila	hila	standard	Hila

This is the user interface for an authorized user to alter the details and credentials of other users.

```
private void cmd_printActionPerformed(java.awt.event.ActionEvent evt) {  
    MessageFormat header= new MessageFormat("User Details Report");  
    MessageFormat footer= new MessageFormat("Page{0,number,integer}");  
    try{  
        Table_UserDetails.print(JTable.PrintMode.NORMAL, header, footer);  
    }  
    catch(java.awt.print.PrinterException e){  
        System.err.format("Cannot print", e.getMessage());  
    }  
}
```

This is the code used to print a selected record on the database for example, if a user wants to print their credentials. **The same code is used in the other classes in order to enable the user to print records.**

Courier

Recipient Name	Origin	Email	Date of Arrival
raaqib	Dhaka	raaqibm@muwci.net	25 Oct
Reva	Dwarka, Delhi	revas@muwci.net	30 oct
miguel	argentina	miguela@muwci.net	30 oct
Kuba	Prague	jakubl@muwci.net	30 oct
Aryaman	Chandigarh	aryamans@muwci.net	30 oct
Sneh	Pune	snehd@muwci.net	30 oct
Victor	France	victorh@muwci.net	30 oct
Bhavesha	Allahabad	bhavesha@muwci.net	31 oct
Farzana	Dhaka	farzanat@muwci.net	2 Nov
Gayatri Sharma	Agra	gayatris@muwci.net	14 Nov

This is the user interface for the Courier page which is used to record the data of the couriers coming in and then automatically send emails to the person's email.

```
private void cmd_emailActionPerformed(java.awt.event.ActionEvent evt) {  
    String To= txt_sendmail.getText();  
  
    Properties props= new Properties();  
    props.put("mail.smtp.host", "smtp.gmail.com");  
    props.put("mail.smtp.socketFactory.port", "465");  
    props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");  
    props.put("mail.smtp.auth", "true");  
    props.put("mail.smtp.port", "465");  
  
    Session session =Session.getDefaultInstance(props,  
        new javax.mail.Authenticator() {  
            protected PasswordAuthentication getPasswordAuthentication(){  
                return new PasswordAuthentication("raaqibm@muwci.net", "Playstationvital1");  
            }  
        });  
    try{  
        Message message=new MimeMessage(session);  
        message.setFrom(new InternetAddress("raaqibm@muwci.net"));  
        message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(To));  
        message.setSubject("Courier");  
        message.setText("Please collect your Courier from Reprographics");  
        Transport.send(message);  
        JOptionPane.showMessageDialog(null, "Message Sent");  
    }catch(Exception e){  
        JOptionPane.showMessageDialog(null, e);  
    }  
}
```

This is the code used to send automated emails to the email address of the people who had couriers come into the Reprographics department.