

République Tunisienne Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Tunis El Manar
Faculté des Sciences Mathématiques, Physiques et Naturelles de Tunis



RAPPORT DE PROJET :

"Réalisation d'une application de gestion des absences."

Réalisé par

Barkallah Meryem
Akkeri Ameni

Enseignant

Zammali Saloua

2ème année licence fondamentale en informatique

Année Universitaire 2019-2020



Remerciements

Tout d'abord au terme de ce travail nous tenons à remercier **Mme Zammali Saloua** qui nous a formé et accompagné tout au long de cette expérience avec beaucoup de patience et de pédagogie.

Ce projet nous a permis d'acquérir beaucoup des compétences telles que l'utilisation des technique paradigmes orientés objet, la modélisation des fonctionnalités avec des cas d'utilisations, la représentation des données du système avec des diagrammes, la maîtrise de la programmation avec le langage java et l'utilisation de système de gestion de la base de données MySQL pour la définition et la manipulation des données.

Sommaire

- I Introduction
- II Modélisation du système
- III Réalisation du système
- IV Conclusion
- V Bibliographie

LISTE DES FIGURES

Figure 1 : Vue Globale.....	2
Figure 2 : Diagramme des acteurs.....	2
Figure 3 : Diagramme descriptif des acteurs.....	3
Figure 4 : Diagramme de contexte statique.....	3
Figure 5 : Diagramme de cas d'utilisation.....	4
Figure 6 : Conception BD.....	4
Figure 7 : Register.java.....	6
Figure 8 : Consulter les absences(enseignants).....	7
Figure 9 : Ajouter les absences.....	8
Figure 10 : Consulter les absences(étudiants).....	9
Figure 11 : Consulter les absences(Responsables Administratifs).....	10
Figure 12 : Supprimer une absence (Responsables Administratifs).....	11
Figure 13 : Imprimer les absences.....	12
Figure 14 : La liste des absences.....	12
Figure 15 : Générer un graphe.....	13
Figure 16 : Graphe des absences d'un étudiant.....	13
Figure 17 : Mails d'alertes.....	14

I.Introduction

• Description du l'application :

L'objectif de cette application est d'offrir aux utilisateurs (Responsables, Enseignants et Etudiants) un outil pour gérer des informations (La gestion des absences) .

• Les besoins fonctionnels :

L'application de gestion des absences offrira les services suivants :

- Mode authentification

Une authentification est nécessaire pour que l'utilisateur puisse accéder aux différents Dashboard en tant que :

- Un responsable administratif
- Un enseignant
- Un étudiant

- Mode absence :

- Le responsable administratif peut consulter, imprimer et supprimer (en cas de justification par l'étudiant) des absences
- L'enseignant doit ajouter les étudiants absents en classe.
- L'étudiant peut consulter les absences.

- Mode contact :

- Le responsable administratif peut envoyer des mails d'alertes aux étudiants absentéistes.

- Autres options :

- Le responsable administratif peut generer le graphe de taux d'absentéisme.



II.Modélisation de système :

- **Vue Globale :**

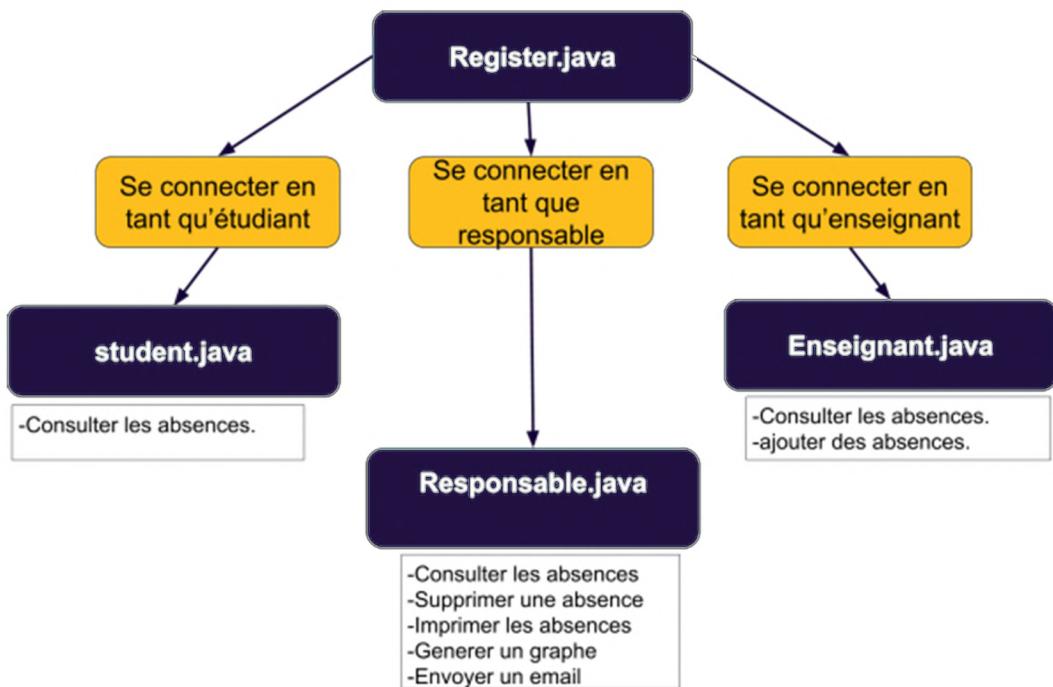


Figure 1 : Vue Globale

- **Analyse du projet :**

- Diagramme des acteurs :

Dans ce projet on a 3 acteurs qui interagissent avec le système de gestion des absences.

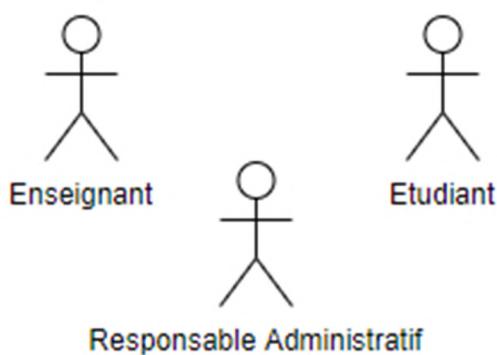


Figure 2 : Diagramme des acteurs

- Diagramme descriptif des acteurs

Une explication des différentes tâches des 3 acteurs.

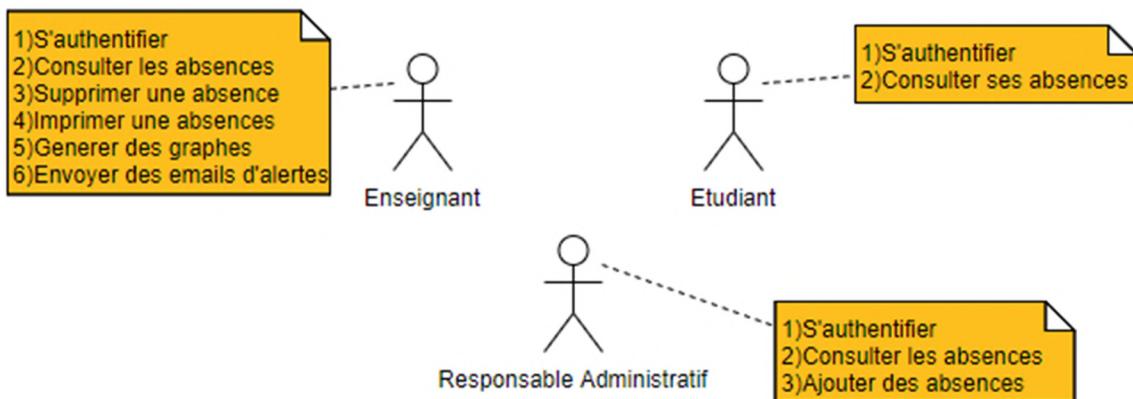


Figure 3 : Diagramme descriptif des acteurs

- Diagramme de contexte statique :

Un diagramme qui décrit le type des relations entre le système et les acteurs.

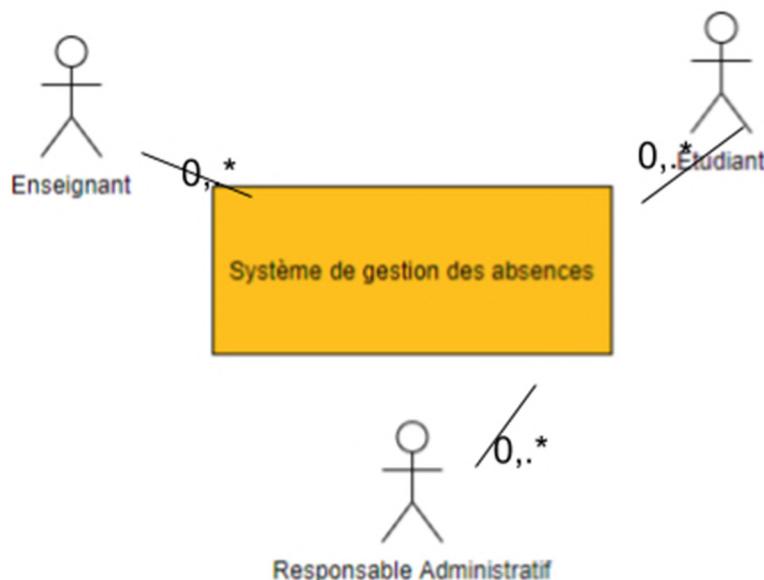


Figure 4 : Diagramme de contexte statique

- Diagramme de cas d'utilisation :

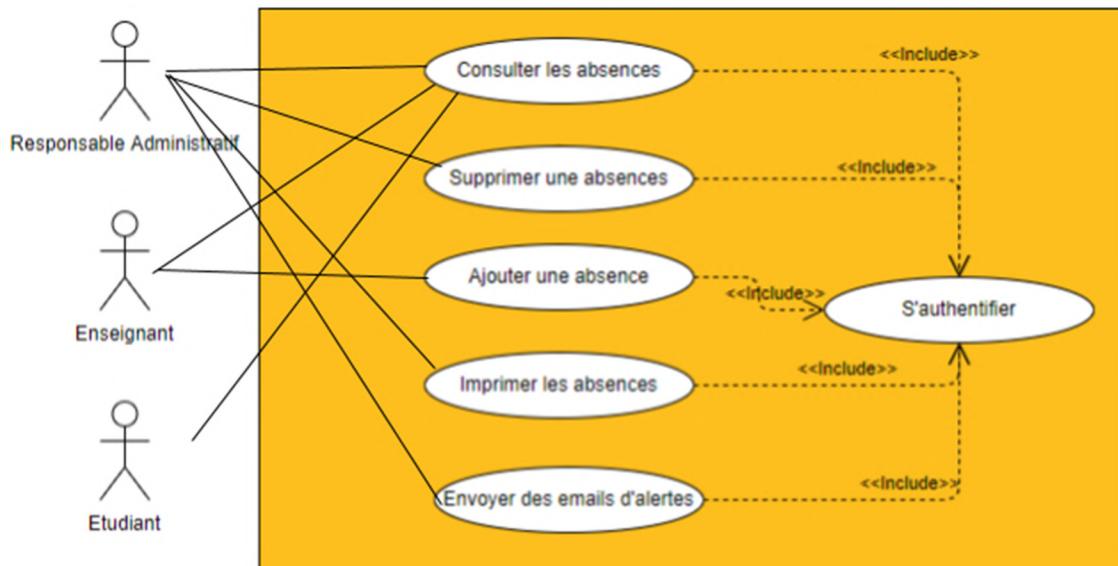


Figure 5 : Diagramme de cas d'utilisation

III. Réalisation de système :

- Partie base des données : (MySQL)

- Conception BD :

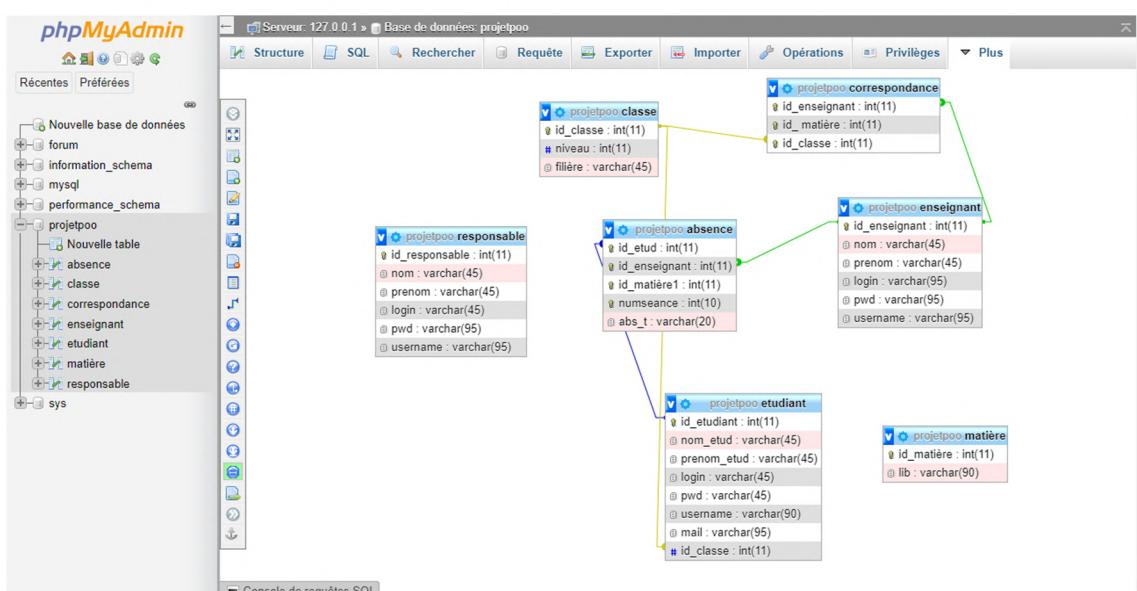
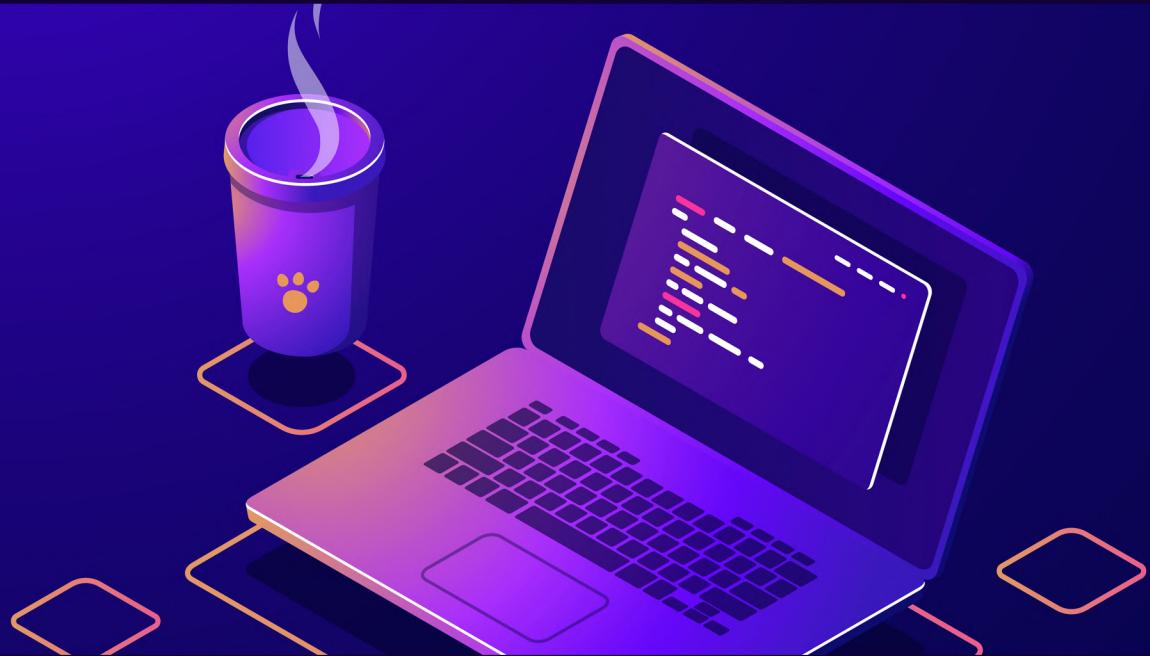


Figure 6 : Conception BD



• Partie code : (JAVA)

- La connexion entre la BD et le code:(JDBC)

Java Database Connectivity (JDBC) est une interface de programmation d'application (API) pour le langage de programmation Java, qui définit comment un client peut accéder à une base de données.

Code{

```
package login;

import javax.swing.*;

public class Connexion {
    Connection conn=null;

    public static Connection ConnexionDB() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn=DriverManager.getConnection("jdbc:mysql://localhost/ProjetPoo?autoReconnect=true&useSSL=false","root","root");
            return conn;
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e);
            return null;
        }
    }
}
```

• Les interfaces :

• Register.java :

Cette interface permet aux utilisateurs (Responsables administratifs, Enseignants et Etudiants) de s'authentifier pour accéder les dashboards.

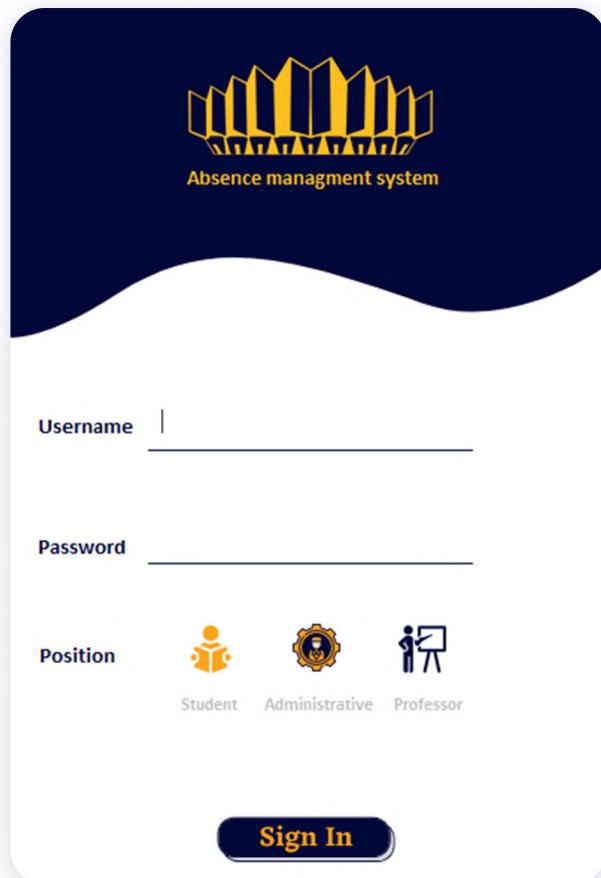


Figure 7 : Register.java

Code{

```

if (responsable.isSelected()) {
    try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/projetPoo?autoReconnect=true&useSSL=false","root","root");
        Statement psmt=conn.createStatement();
        String query="select * from responsable where username='"+name.getText()+"' and pwd ='"+pass.getText().toString()+"'";
        ResultSet rs=psmt.executeQuery(query);
        if(rs.next())
            JOptionPane.showMessageDialog(null, "you are connected ...");
        dispose();
        Responsable frame = new Responsable();
        frame.setVisible(true);
        conn.close();
    }catch (Exception ex){ System.out.println("oupsss try again !!");}
}
}

```

- Enseignant.java :

Cette interface permet aux enseignants de :

-Consulter les absences :

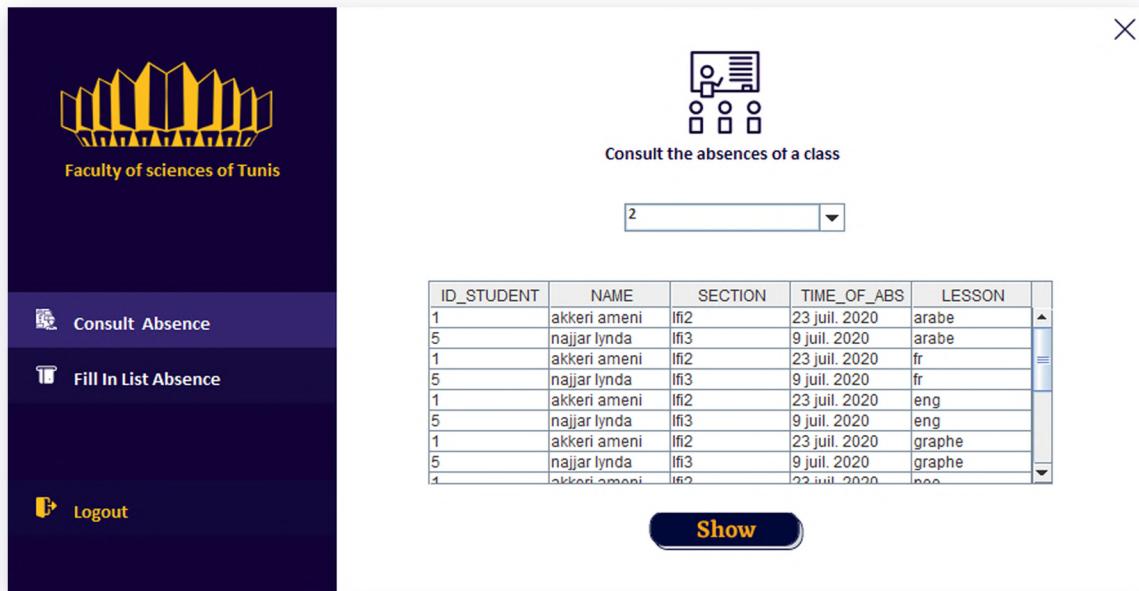


Figure 8 : Consulter les absences(enseignants)

Code{

```
try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/projetPoo?autoReconnect=true&useSSL=false","root","root");
    Statement psmt=conn.createStatement();
    String query="select id_etudiant AS ID_STUDENT , CONCAT( nom_etud , ' ' , prenom_etud ) AS NAME , CONCAT( filière,niveau) AS SECTION , "
        + " abs_t as TIME_OF_ABS , lib as LESSON from absence , etudiant,enseignant, matière , classe where etudiant.id_classe= classe.id_classe "
        + " and absence.id_etud= etudiant.id_etud and absence.id_enseignant= enseignant.id_enseignant and "
        + " absence.id_enseignant='"+combo.getSelectedItem().toString()+"' ";
    ResultSet rs=psmt.executeQuery(query);
    table.setModel(DbUtils.resultSetToTableModel(rs));
}

} catch (SQLException e1) {
    // TODO Auto-generated catch block
    e1.printStackTrace();
} catch (ClassNotFoundException e1) {
    // TODO Auto-generated catch block
    e1.printStackTrace();
}
```

-Ajouter une absence

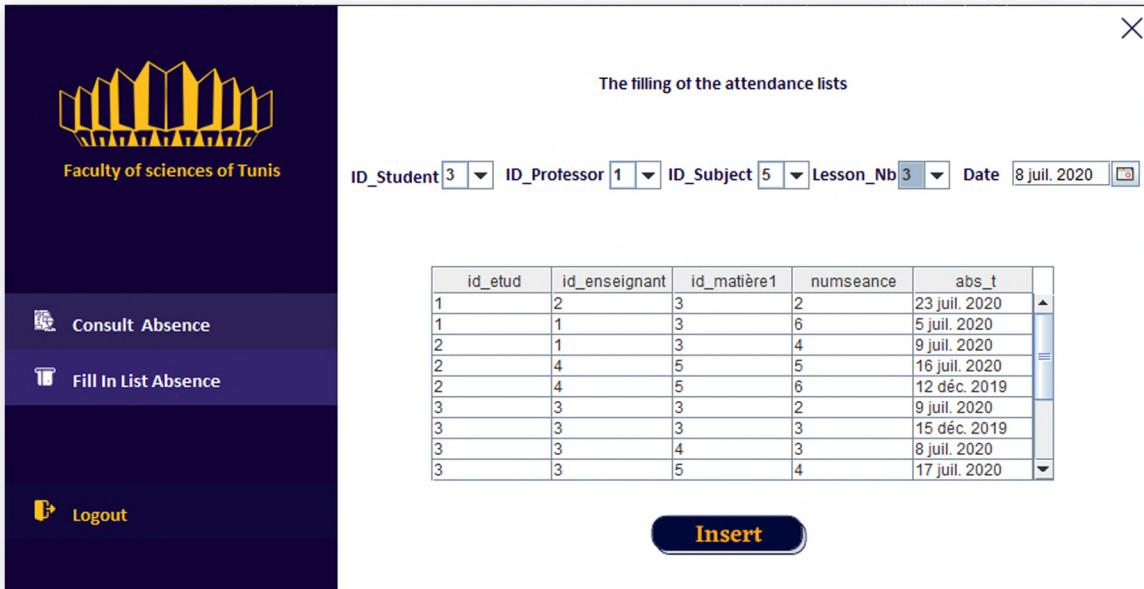


Figure 9 : Ajouter les absences

-Cette interface lui permet aussi de classer les absences selon son ID

Code{

```
public void mouseClicked(MouseEvent e) {

    String id_etudiant= combo1.getSelectedItem().toString();
    Date date = dateChooser.getDate();
    String date1 = DateFormat.getDateInstance().format(date);

    String numseance= combo4.getSelectedItem().toString();
    String id_matiere = combo2.getSelectedItem().toString();
    String id_enseignant = combo3.getSelectedItem().toString();

    String sql= " insert into absence ( id_etud , id_enseignant ,id_matiere1 , numseance , abs_t ) values ( ? ,? ,? , ? , ? )";

    try {
        if (!id_etudiant.equals("0") && !date1.equals("") && !numseance.equals("0") && !id_matiere.equals("0") && !id_enseignant.equals("0") ) {
            prepared=cnx.prepareStatement(sql);
            prepared.setString(1, id_etudiant);
            prepared.setString(5, date1);
            prepared.setString(4,numseance );
            prepared.setString(2, id_matiere);
            prepared.setString(3, id_enseignant);

            prepared.execute();
            JOptionPane.showMessageDialog(null, "absence inserted ");
            combo1.setSelectedItem("0");
            combo2.setSelectedItem("0");
            combo3.setSelectedItem("0");
            combo4.setSelectedItem("0");
        }
    } catch (Exception e1) {
        e1.printStackTrace();
    }
}
```

• student.java :

Cette interface permet aux étudiants de consulter ses absences selon son ID.

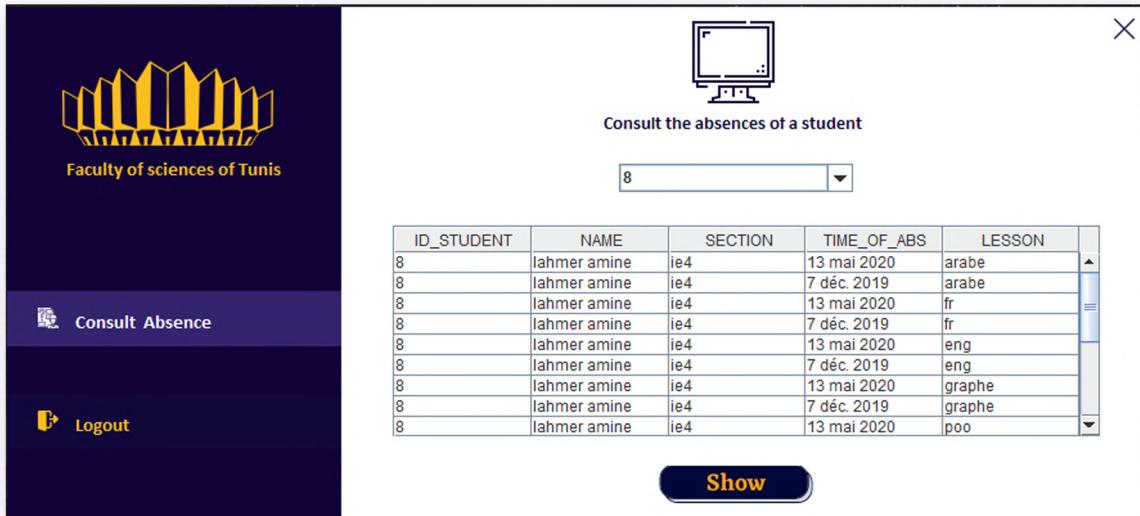


Figure 10 : Consulter les absences(étudiants)

Code{

```

public void mouseClicked(MouseEvent e) {
    try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/projetPoo?autoReconnect=true&useSSL=false","root","root");
        Statement psmt=conn.createStatement();
        String query="select id_etudiant AS ID_STUDENT , CONCAT (nom_etud , ' ' , prenom_etud ) AS NAME , CONCAT( filière,niveau) AS SECTION , "
        + "abs_t as TIME_OF_ABS , lib as LESSON from absence , etudiant, matière , classe where etudiant.id_classe= classe.id_classe and "
        + " absence.id_etud= etudiant.id_etudiant and absence.id_etud='"+combo.getSelectedItem().toString()+"' ";
        ResultSet rs=psmt.executeQuery(query);
        table.setModel(DbUtils.resultSetToTableModel(rs));
    } catch (SQLException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    } catch (ClassNotFoundException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
}

```

- Responsable.java :

Cette interface permet aux responsables administratifs de :

-Consulter les absences

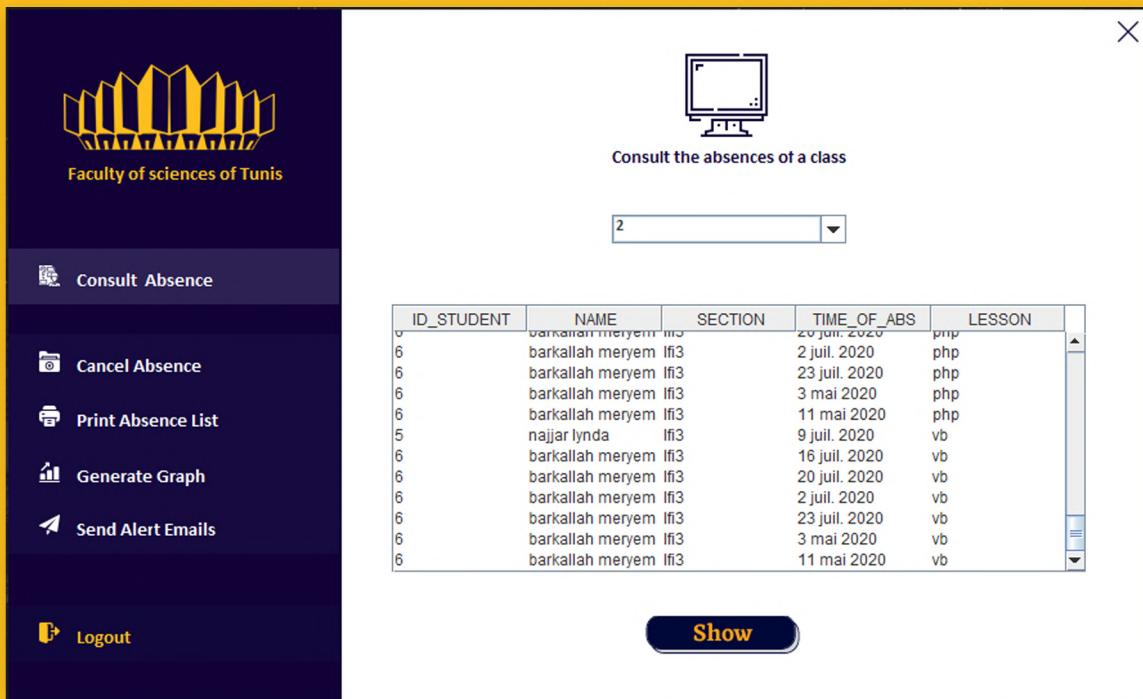


Figure 11 : Consulter les absences(Responsables Administratifs)

Code{

```
public void actionPerformed(ActionEvent e) {
}
});
check.addMouseListener(new MouseAdapter() {
@Override
public void mouseClicked(MouseEvent e) {
try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/projetPoo?autoReconnect=true&useSSL=false","root","root");
    Statement psmt=conn.createStatement();
    String query="select id_etudiant AS ID_STUDENT , CONCAT (nom_etud , ' ' , prenom_etud ) AS NAME , CONCAT( filière,niveau ) AS SECTION , "
    + "abs_ as TIME_OF_ABS , lib as LESSON from absence , etudiant,enseignant , matière , classe where absence.id_etud= etudiant.id_etudiant "
    +" and absence.id_enseignant= enseignant.id_enseignant and classe.id_classe = etudiant.id_classe and"
    +" [ ] etudiant.id_classe='"+combo.getSelectedItem().toString()+"' ";
    ResultSet rs=psmt.executeQuery(query);
    table.setModel(DButils.resultSetToTableModel(rs));
}
catch (SQLException e1) {
    // TODO Auto-generated catch block
    e1.printStackTrace();
}
catch (ClassNotFoundException e1) {
    // TODO Auto-generated catch block
    e1.printStackTrace();
}
}
});
```

Cette interface permet aux responsables administratifs de :

-Supprimer une absence :

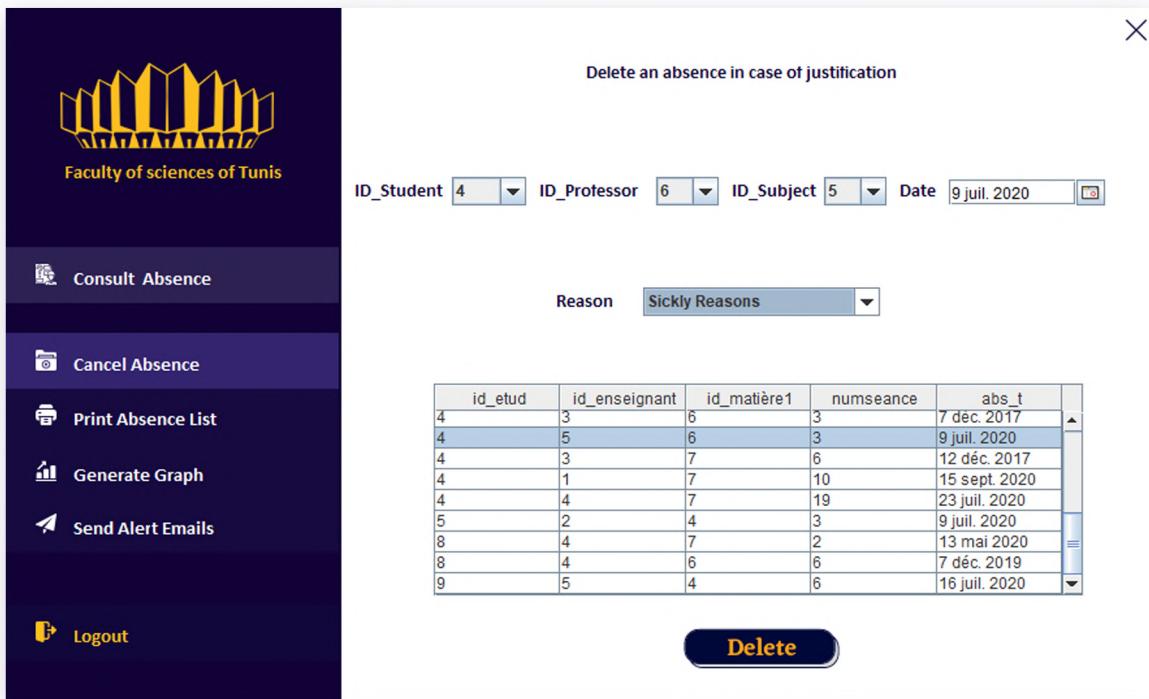


Figure 12 : Supprimer une absence (Responsables Administratifs)

Code{

```

public void actionPerformed(ActionEvent e) {
    int ligne1 = table_2.getSelectedRow(); // ligne=-1 au debut
    if (ligne1== -1)
        JOptionPane.showMessageDialog(null, "Fill in the empty fields");

    else {
        int op = JOptionPane.showConfirmDialog(null, "Are you sure you want to delete this file?", "Delete a file", JOptionPane.YES_NO_OPTION);

        String id1 =table_2.getModel().getValueAt(ligne1, 0).toString();

        try {
            if ( op==0) {
                Class.forName("com.mysql.jdbc.Driver");
                Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/projetPoo?autoReconnect=true&useSSL=false","root","root");
                Statement statement = conn.createStatement();
                statement.executeUpdate("DELETE FROM absence WHERE id_etud=" +id1 + "");
                JOptionPane.showMessageDialog(null, "absence deleted");
            }
            catch(ClassNotFoundException e1) {
                // TODO Auto-generated catch block
                e1.printStackTrace();
            } catch (SQLException e1) {
                // TODO Auto-generated catch block
                e1.printStackTrace();
            }
        }
    }
}

```

-Imprimer les absences :

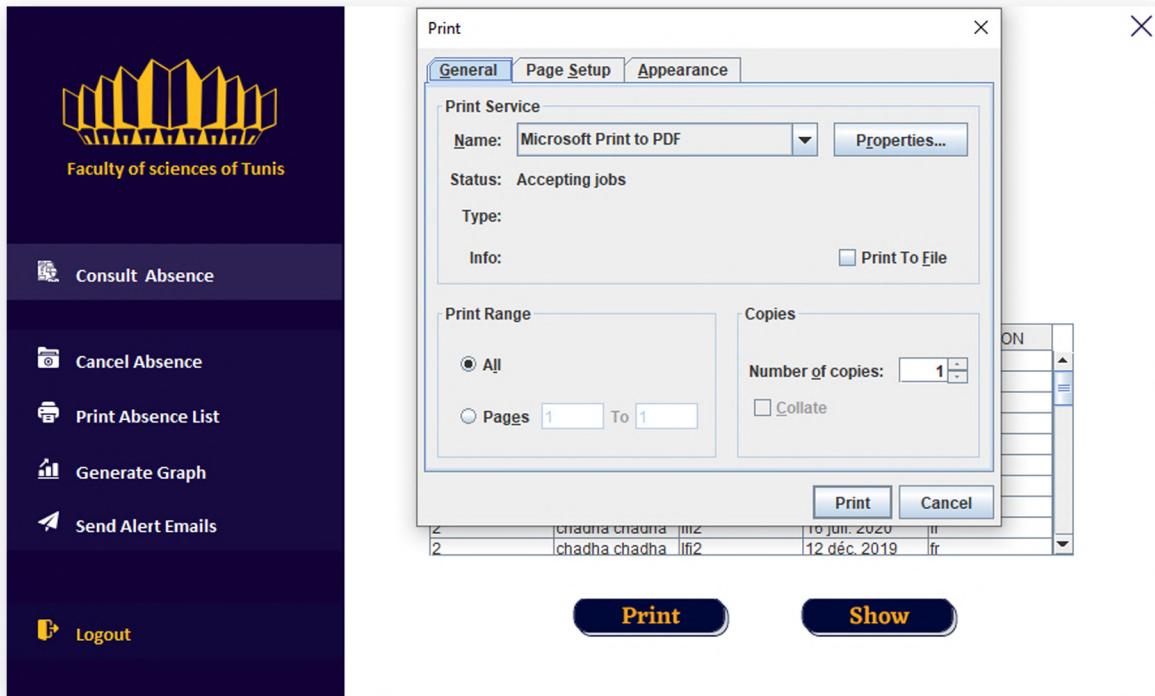


Figure 13 : Imprimer les absences

Code{

```
public void actionPerformed(ActionEvent e) {
    MessageFormat header = new MessageFormat("absence list");
    MessageFormat footer = new MessageFormat("Page{0,number,integer}");

    try {
        table_1.print(PrintMode.NORMAL, header, footer);
    }catch (java.awt.print.PrinterException e3) {
        System.err.format("erreur impression", e3.getMessage());
    }
});
```

ID_STUDENT	NAME	SECTION	TIME_OF_ABS
1	akkeri ameni	lfi2	23 juil. 2020
1	akkeri ameni	lfi2	5 juil. 2020
2	chadha chadha	lfi2	9 juil. 2020
2	chadha chadha	lfi2	16 juil. 2020
2	chadha chadha	lfi2	12 déc. 2019
1	akkeri ameni	lfi2	23 juil. 2020
1	akkeri ameni	lfi2	5 juil. 2020
2	chadha chadha	lfi2	9 juil. 2020
2	chadha chadha	lfi2	16 juil. 2020
2	chadha chadha	lfi2	12 déc. 2019
1	akkeri ameni	lfi2	23 juil. 2020
1	akkeri ameni	lfi2	5 juil. 2020
2	chadha chadha	lfi2	9 juil. 2020
2	chadha chadha	lfi2	16 juil. 2020
2	chadha chadha	lfi2	12 déc. 2019

Figure 14 : La liste des absences

-Générer un graphe

ID_STUDENT	NAME	SECTION	TIME_OF_ABS	LESSON
3	rejeb rayen	lf13	10 déc. 2017	reseaux
3	rejeb rayen	lf13	10 déc. 2017	eng
3	rejeb rayen	lf13	10 déc. 2017	graphe
3	rejeb rayen	lf13	10 déc. 2017	php
3	rejeb rayen	lf13	10 déc. 2017	poo
3	rejeb rayen	lf13	10 déc. 2017	vb
3	rejeb rayen	lf13	10 déc. 2017	arabe
3	rejeb rayen	lf13	10 déc. 2017	python
3	rejeb rayen	lf13	10 déc. 2017	fr
4	ben yaala aziz	lf13	12 déc. 2017	python

Figure 15 : Générer un graphe



Figure 16 : Graphe des absences d'un étudiant

Code{

```

public void actionPerformed(ActionEvent e) {
    try {

        String query ="select abs_t,id_classe From absence , etudiant where absence.id_etud = etudiant.id_etudiant " ;

        prepared=cnx.prepareStatement(query);
        JDBCCategoryDataset dataset=new JDBCCategoryDataset( cnx,query);
        resultat=prepared.executeQuery();

        JFreeChart chart=ChartFactory.createLineChart("absenteeism rate graph","Date"," Class",dataset,PlotOrientation.VERTICAL,false,true,true);

        BarRenderer renderer=null;
        CategoryPlot plot=null;
        renderer=new BarRenderer();
        ChartFrame frame=new ChartFrame("absenteeism rate graph",chart);
        frame.setVisible(true);
        frame.setSize(800,700);
    } catch (SQLException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
}
}

```

- Envoyer des de mails d'alertes aux étudiants absentéistes.

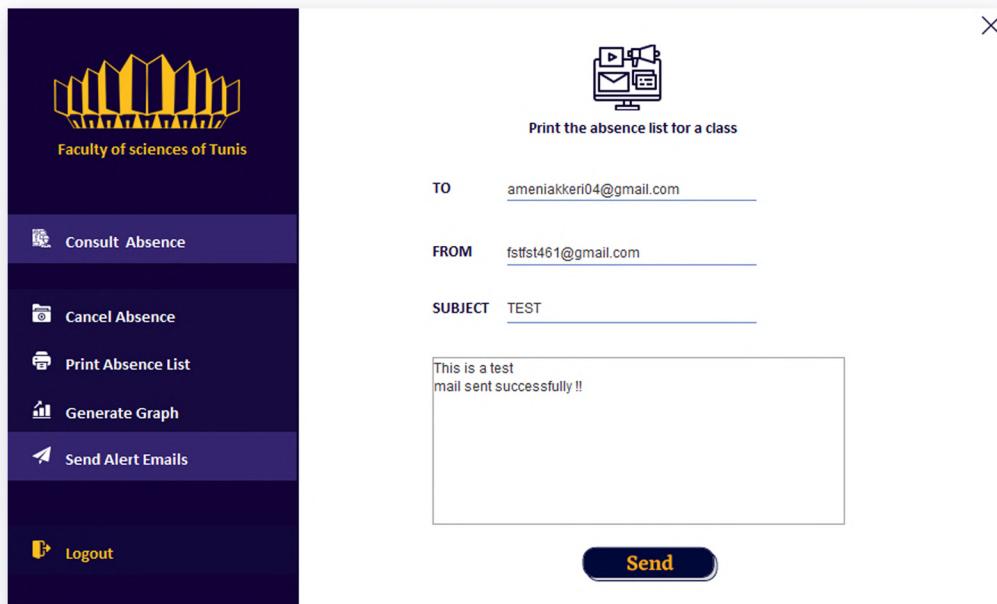


Figure 17 : Mails d'alertes

Code{

```

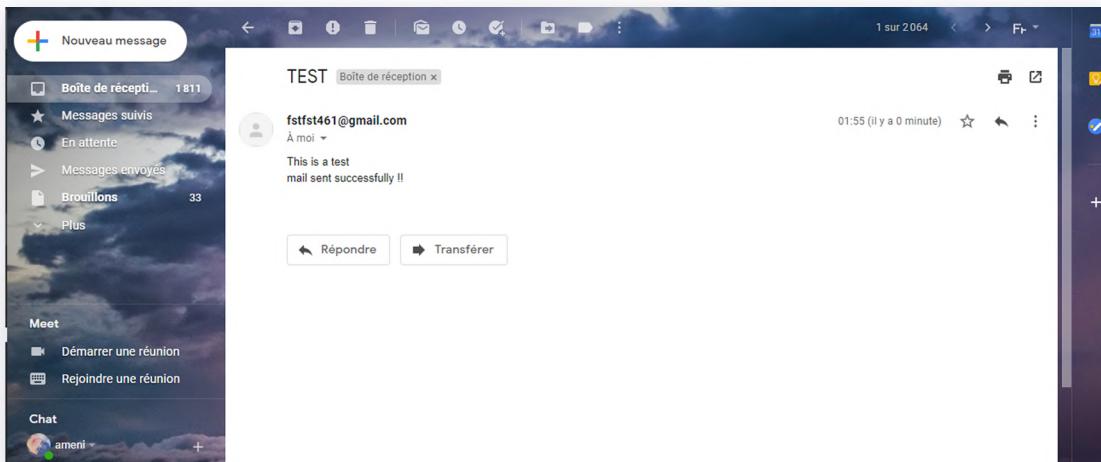
public void mouseClicked(MouseEvent e) {
    String ToEmail = txtToEmail.getText();
    String FromEmail = txtFromEmail.getText(); //fstfst461@gmail.com
    String FromEmailPassword = "adminadminfst"; //Your email Password from you want to send email
    String Subjects = txtSubject.getText();

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

    Session session = Session.getDefaultInstance(properties,new javax.mail.Authenticator() {
        protected PasswordAuthentication getPasswordAuthentication(){
            return new PasswordAuthentication(FromEmail, FromEmailPassword);
        }
    });
    JOptionPane.showMessageDialog(null, " send successfully...");

    try{
        MimeMessage message = new MimeMessage(session);
        message.setFrom(new InternetAddress(FromEmail));
        message.addRecipient(Message.RecipientType.TO, new InternetAddress(ToEmail));
        message.setSubject(Subjects);
        message.setText(txtpnWriteHereYour.getText());
        Transport.send(message);
    }catch(Exception ex){
        System.out.println(""+ex);
    }
}

```



IV. Conclusion

Au terme de ce projet, nous tenons à souligner que sa réalisation était très bénéfique pour nous car c'était une bonne occasion pour consolider mes connaissances dans le domaine de conception et la réalisation des applications en JAVA.

V. Bibliographie : (Outils & sources)

- Eclipse IDE .
- Wamp server (BD: (MySQL)).
- MySQL workbench (Conception BD).
- <https://coolors.co/> (Palette des couleurs).
- <https://online.visual-paradigm.com/> (Conception cas d'utilisation)
- <https://www.flaticon.com/> (les icons).
- <https://stackoverflow.com/> (les erreurs).
- activation-1.1.1.jar
- itextpdf-5.1.0.jar (Impression)
- com.mysql.jdbc-5.1.5.jar (JDBC)
- javax.mail.jar (Email)
- Jcalendar-1.4.jar (Calendrier)
- JfreeChart-1.0.1.9.jar (Graphe)
- rs2xml.jar