**JAVA SWING BASED**

**MALWARE AND IT’S COUNTERFEIT**

**SQL CONNECTIVITY USING JDBC**

*A*

*Report*

*Submitted in partial fulfillment of the Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING IN**

**INFORMATION TECHNOLOGY**

**By**

### R. Sai Sathvik <1602-20-737-035>

**Under the guidance of Ms B. Leelavathy**



### Department of Information Technology Vasavi College of Engineering (Autonomous) (Affiliated to Osmania University) Ibrahimbagh, Hyderabad-31

**2021-2022**

BONAFIDE CERTIFICATE

This is to certify that this project report titled

***‘Malware and it’s counterfeit’***

is a project work of **R. Sai Sathvik** bearing roll no. 1602-20-737-35 who carried out this

project under my supervision in the IV semester for the academic year 2021- 2022

Signature Signature

External Examiner Internal Examiner

# ABSTRACT

These days individuals turned to store everything in computer systems in their lives. We usually download many software which may or may not harm our systems. The software that gains that control over our systems or does unwanted activities in our system is known as Malware. This project Malware and its counterfeit is a database application designed using sql and java which helps users to know what the counterfeits are available in order to protect the information and what are the different effects observed when the software is corrupted by a malware.

**REQUIREMENT ANALYSIS:**

**LIST OF TABLES:**

* Malware
* Cause
* Counterfeit
* UserLogin
* Registration

**LIST OF ATTRIBUTES WITH DOMAIN TYPES:**

**Malware**

Malid varchar2(8)

Malname varchar2(15)

Attackstyle varchar2(100)

**Cause**

Malid varchar2(8)

Effect varchar2(50)

**Counterfeit:**

malid varchar2(8)

measure varchar2(20)

**UserLogin:**

username varchar2(15)

password varchar2(16)

**Registration:**  
 username varchar2(15)

name varchar2(20)

email varchar2(30)

password varchar2(20)

phonenumber number(10)

# AIM AND PRIORITY OF THE PROJECT

To create a **Java GUI-based** desktop application that helps user to acknowledge the malware that the software they have been effected or could be effected with and required counterfeit measure to prevent of attack and protect the software. It takes values like Malware ID, Malware etc through forms which are then updated in the database using JDBC connectivity.

# ARCHITECTURE AND TECHNOLOGY

### Software used:

Java Eclipse, Oracle 11g Database, Java SE version 8, SQL\*Plus.

### Java SWING:

**Java SWING** is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) - an API for providing a graphical user interface (GUI) for Java programs.

Swing was developed to provide a more sophisticated set of GUI components than the earlier AWT. Swing provides a look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

### SQL:

Structure Query Language(SQL) is a database query language used for storing and managing data in **Relational** DBMS. SQL was the first commercial language introduced for E.F Codd's Relational model of database. Today almost all RDBMS (MySql, Oracle, Infomix, Sybase, MS Access) use **SQL** as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

**DESIGN ANALYSIS**

**Entity Relationship Diagram**

**Diagram

Description automatically generated**

### DDL Operations:

**Malware**

CREATE TABLE malware

(malid VARCHAR2(8),

malName VARCHAR(10),

attackStyle VARCHAR2(100));

Text

Description automatically generated

A picture containing text

Description automatically generated

**Cause**

CREATE TABLE cause

(malid VARCHAR2(8),

effect VARCHAR2(50));

**Text

Description automatically generated**

**A picture containing graphical user interface

Description automatically generated**

**Counterfeit**

CREATE TABLE counterfeit

(malid VARCHAR(8),

measure VARCHAR(20));

Text

Description automatically generated

Graphical user interface

Description automatically generated with low confidence

**UserLogin**

CREATE TABLE userLogin

(username VARCHAR2(15),

password VARCHAR2(16));

Text

Description automatically generated

Text

Description automatically generated

**Registration**

CREATE TABLE registration

(username VARCHAR2(15),

name VARCHAR2(20),

email VARCHAR2(30),

password VARCHAR2(20),

phonenumber NUMBER(10));

**Text

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

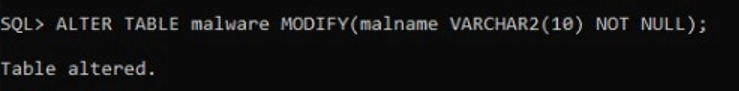
**KEY CONSTRAINTS:**

ALTER TABLE malware ADD PRIMARY KEY(malid);

**Text

Description automatically generated**

ALTER TABLE malware MODIFY(malname VARCHAR2(10) NOT NULL);

****

**Graphical user interface

Description automatically generated with medium confidence**

ALTER TABLE cause ADD FOREIGN KEY(malid) REFERENCES malware;

**Text

Description automatically generated**

ALTER TABLE malware MODIFY(malname VARCHAR(10) NOT NULL);

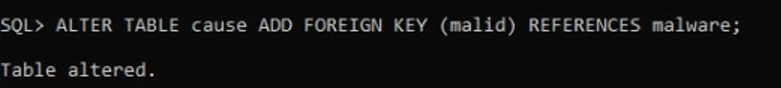
**Text

Description automatically generated**

**Graphical user interface

Description automatically generated with medium confidence**

ALTER TABLE cause ADD FOREIGN KEY(malid) REFERENCES malware;

****

ALTER TABLE cause MODIFY effect VARCHAR2(50) NOT NULL;

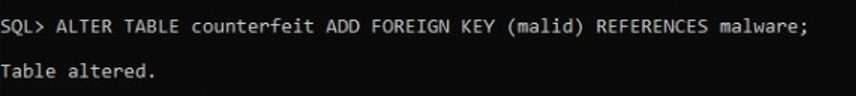
**Text

Description automatically generated**

**Text

Description automatically generated with medium confidence**

ALTER TABLE counterfeit ADD FOREIGN KEY(malid) REFERENCES malware;

****

**Graphical user interface

Description automatically generated with low confidence**

ALTER TABLE registration ADD PRIMARY KEY(username);

**Text

Description automatically generated**

**Graphical user interface

Description automatically generated with low confidence**

ALTER TABLE registration MODIFY password VARCHAR(16) NOT NULL;

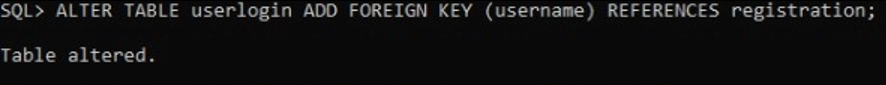
**Text

Description automatically generated**

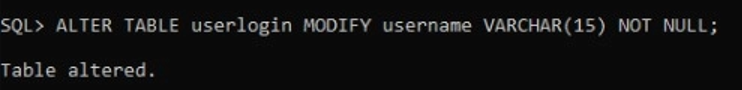
**Graphical user interface

Description automatically generated**

ALTER TABLE userlogin ADD FOREIGN KEY(username) REFERENCES registration;

****

ALTER TABLE userlogin MODIFY username VARCHAR(15) NOT NULL;

****

ALTER TABLE userlogin MODIFY password VARCHAR(16) NOT NULL;

**Text

Description automatically generated**

**Text

Description automatically generated with low confidence**

**MODIFICATION OF DB:  
 DML OPERATIONS:**

**Malware**

INSERT INTO malware VALUES(‘&malid’,’&malname’,’&attackstyle’);

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Cause**

INSERT INTO cause VALUES(‘&malid’,’&effect’);

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Counterfeit**

INSERT INTO counterfeit VALUES(‘&malid’,’&measure’);

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**UserLogin:**

INSERT INTO userlogin VALUES(‘username’,’password’);

**Text

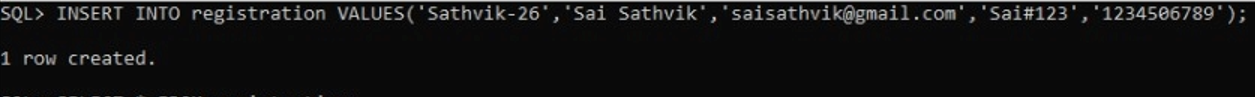
Description automatically generated**

**Graphical user interface, text

Description automatically generated with medium confidence**

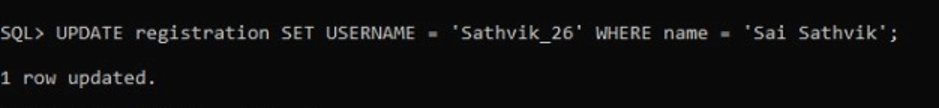
**Registration:**

INSERT INTO registration VALUES(‘username’,’name’,’email’,’password’,’phonenumber’);

****

**Graphical user interface, text

Description automatically generated**

****

**Graphical user interface, application

Description automatically generated**

**FRONT END**

**Implementation:**

**Java Swing tutorial** is a part of Java Foundation Classes (JFC) that isused to create window- based applications*.* It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java. Unlike AWT, Java Swing provides platform-independent and lightweight components. The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc. Swing provides a [look and feel](https://en.wikipedia.org/wiki/Look_and_feel) that emulates the look and feel of several platforms, and also supports a [pluggable look and feel](https://en.wikipedia.org/wiki/Pluggable_look_and_feel) that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT.

**Code:**

//Home.java

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.util.regex.\*;

public class Home{

JFrame frame;

JLabel lbluname,lblpass,lblor,lblmal,lblreg;

JButton btnlogin,btnreg,btnmal;

JTextField tfuname;

JPasswordField tfpass;

JMenuBar menubar;

JMenu about,lkfl;

JMenuItem miabtproject,miabtstudent,miMotif,miNimbus,miCross;

JPanel p1,p2,p3,p4,p5,p6;

DBAccess db;

public Home(){

super();

}

public Home(DBAccess db){

this.db = db;

initializeComponents();

registerListeners();

addComponentsToFrame();

frame.setBackground(Color.lightGray);

frame.setLayout(new GridLayout(6,1));

frame.setSize(450,400);

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);

}

public void initializeComponents(){

menubar = new JMenuBar();

about = new JMenu("About");

miabtproject = new JMenuItem("About Project");

miabtstudent = new JMenuItem("About Student");

lkfl = new JMenu("LAF");

miMotif = new JMenuItem("Motif");

miNimbus = new JMenuItem("Nimbus");

miCross = new JMenuItem("Cross Platform");

frame = new JFrame("Malware counterfeit");

lblmal = new JLabel("Malware Counterfeit");

lbluname = new JLabel("Username: ");

tfuname = new JTextField(12);

lblpass = new JLabel("Password: ");

tfpass = new JPasswordField(12);

btnlogin = new JButton("Login");

lblreg = new JLabel("Not registered? ");

btnreg = new JButton("Register");

lblor = new JLabel("or");

btnmal = new JButton("Malware Counterfeit");

p1 = new JPanel();

p2 = new JPanel();

p3 = new JPanel();

p4 = new JPanel();

p5 = new JPanel();

p6 = new JPanel();

}

public void registerListeners(){

btnlogin.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent ae){

String inpass = db.getPass(tfuname.getText());

if(inpass.equals(String.valueOf(tfpass.getPassword()))){

JOptionPane.showMessageDialog(frame, "Login Successfull.","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

frame.dispose();

db.enterLogin(tfuname.getText(),String.valueOf(tfpass.getPassword()));

new Operations(tfuname.getText(),db);

}

else if(inpass.equals(null)){

JOptionPane.showMessageDialog(frame, "You are not registerd.\nPlease register to login.","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

else{

JOptionPane.showMessageDialog(frame,"Incorrect Password.\nEnter again.","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

tfpass.setText(null);

}

}

});

btnmal.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent ae){

frame.dispose();

new MalwareDetails(db);

}

});

btnreg.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent ae){

frame.dispose();

new Registration(db);

}

});

miabtproject.addActionListener (new ActionListener () {

public void actionPerformed(ActionEvent e) {

JOptionPane.showMessageDialog(frame, "Project Name: Malware Counterfeit\nDetails: This Project describes about the differenct types \nof software malwares present and their effects caused by \ntheir their presence amd ways to counterfeit it.","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

});

miabtstudent.addActionListener (new ActionListener () {

public void actionPerformed(ActionEvent e) {

JOptionPane.showMessageDialog(frame, "Name of Student: R. Sai Sathvik\nRoll number: 1602-20-737-035\nMain purpose of project: To implement a 1-Tier Architecture java application","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

});

miMotif.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent evt)

{

try{

UIManager.setLookAndFeel("com.sun.java.swing.plaf.motif.MotifLookAndFeel");

SwingUtilities.updateComponentTreeUI(frame);

}

catch(ClassNotFoundException|InstantiationException|IllegalAccessException|UnsupportedLookAndFeelException ex)

{}

}

});

miNimbus.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent evt)

{

try{

UIManager.setLookAndFeel("javax.swing.plaf.nimbus.NimbusLookAndFeel");

SwingUtilities.updateComponentTreeUI(frame);

}

catch(ClassNotFoundException|InstantiationException|IllegalAccessException|UnsupportedLookAndFeelException ex)

{}

}

});

miCross.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent evt)

{

try{

UIManager.setLookAndFeel(UIManager.getCrossPlatformLookAndFeelClassName());

SwingUtilities.updateComponentTreeUI(frame);

}

catch(ClassNotFoundException|InstantiationException|IllegalAccessException|UnsupportedLookAndFeelException ex)

{}

}

});

frame.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

if(db.closeConnection()){

JOptionPane.showMessageDialog(frame, "You are Exiting and \n Connection is closed successfully.","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);;

}

System.exit(0);

}

});

}

public void addComponentsToFrame(){

about.add(miabtproject);

about.add(miabtstudent);

lkfl.add(miCross);

lkfl.add(miMotif);

lkfl.add(miNimbus);

menubar.add(about);

menubar.add(lkfl);

frame.setJMenuBar(menubar);

p1.add(lblmal);

p2.add(lbluname);

p2.add(tfuname);

p3.add(lblpass);

p3.add(tfpass);

p4.add(btnlogin);

p5.add(lblreg);

p5.add(btnreg);

p6.add(lblor);

p6.add(btnmal);

frame.add(p1);

frame.add(p2);

frame.add(p3);

frame.add(p4);

frame.add(p5);

frame.add(p6);

}

public static void main(String[] args){

DBAccess db;

db = new DBAccess();

new Home(db);

}

}

//Registration.java

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.util.regex.\*;

import java.util.\*;

public class Registration{

JFrame frame;

JLabel lblname,lbluname,lblemail,lblphno,lblpass,lblpass1;

JTextField tfname,tfuname,tfemail,tfphno,tfpass,tfpass1;

JButton btnsubmit;

JMenuBar menubar;

JMenu Navi,about,lkfl;

JMenuItem miHome,miabtproject,miabtstudent,miMotif,miNimbus,miCross;

JPanel p1,p2,p3,p4,p5,p6,p7;

DBAccess db;

public Registration(DBAccess db){

this.db = db;

initializeComponents();

registerListeners();

addComponentsToFrame();

frame.setBackground(Color.lightGray);

frame.setLayout(new GridLayout(7,1));

frame.setSize(450,400);

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);

}

public void initializeComponents(){

menubar = new JMenuBar();

Navi = new JMenu("Navigations");

miHome = new JMenuItem("Home");

about = new JMenu("About");

miabtproject = new JMenuItem("About Project");

miabtstudent = new JMenuItem("About Student");

lkfl = new JMenu("LAF");

miMotif = new JMenuItem("Motif");

miNimbus = new JMenuItem("Nimbus");

miCross = new JMenuItem("Cross Platform");

frame = new JFrame("Registration");

lblname = new JLabel("Name: ");

tfname = new JTextField(10);

lbluname = new JLabel("User name: ");

tfuname = new JTextField(10);

lblemail = new JLabel("Email: ");

tfemail = new JTextField(10);

lblphno = new JLabel("Phone number: ");

tfphno = new JTextField(10);

lblpass = new JLabel("Password: ");

tfpass = new JTextField(10);

lblpass1 = new JLabel("Conform Password: ");

tfpass1 = new JTextField(10);

btnsubmit = new JButton("Submit");

p1 = new JPanel();

p2 = new JPanel();

p3 = new JPanel();

p4 = new JPanel();

p5 = new JPanel();

p6 = new JPanel();

p7 = new JPanel();

}

public void registerListeners(){

tfpass.addFocusListener(new FocusAdapter(){

public void focusLost(FocusEvent e)

{

if(tfpass.getText().length()<8)

{

JOptionPane.showMessageDialog(frame, "Password should have at least 8 characters","WARNING", JOptionPane.WARNING\_MESSAGE);

}

}

});

btnsubmit.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

try{

long n = Long.parseLong(tfphno.getText());

if(tfuname.getText().equals("")||tfname.getText().equals(null)||tfemail.getText().equals(null)||tfpass.getText().equals(null))

JOptionPane.showMessageDialog(frame, "Enter all feilds.","WARNING", JOptionPane.WARNING\_MESSAGE);

else if(tfphno.getText().length() != 10)

JOptionPane.showMessageDialog(frame, "Phone number is of 10 digits.","WARNING", JOptionPane.WARNING\_MESSAGE);

else if(!(tfpass.getText().equals(tfpass1.getText()))){

JOptionPane.showMessageDialog(frame, "Re-enter the conformation password.","WARNING", JOptionPane.WARNING\_MESSAGE);

tfpass.setText(null);

tfpass1.setText(null);

}

else{

db.getRegistered(tfuname.getText(),tfname.getText(),tfemail.getText(),tfpass.getText(),n);

JOptionPane.showMessageDialog(frame, "You have been registered","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

frame.dispose();

new Home(db);

}

}

catch(NumberFormatException ae){

JOptionPane.showMessageDialog(frame, "Unauthorized error.\nCrash !!!\nPhone number is of 10 digits.","WARNING", JOptionPane.WARNING\_MESSAGE);

}

}

});

miHome.addActionListener(new ActionListener () {

public void actionPerformed(ActionEvent e) {

frame.dispose();

new Home(db);

}

});

miabtproject.addActionListener (new ActionListener () {

public void actionPerformed(ActionEvent e) {

JOptionPane.showMessageDialog(frame, "Project Name: Malware Counterfeit\nDetails: This Project describes about the differenct types \nof software malwares present and their effects caused by \ntheir their presence amd ways to counterfeit it.","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

});

miabtstudent.addActionListener (new ActionListener () {

public void actionPerformed(ActionEvent e) {

JOptionPane.showMessageDialog(frame, "Name of Student: R. Sai Sathvik\nRoll number: 1602-20-737-035\nMain purpose of project: To implement a 1-Tier Architecture java application","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

});

miMotif.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent evt)

{

try{

UIManager.setLookAndFeel("com.sun.java.swing.plaf.motif.MotifLookAndFeel");

SwingUtilities.updateComponentTreeUI(frame);

}

catch(ClassNotFoundException|InstantiationException|IllegalAccessException|UnsupportedLookAndFeelException ex)

{}

}

});

miNimbus.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent evt)

{

try{

UIManager.setLookAndFeel("javax.swing.plaf.nimbus.NimbusLookAndFeel");

SwingUtilities.updateComponentTreeUI(frame);

}

catch(ClassNotFoundException|InstantiationException|IllegalAccessException|UnsupportedLookAndFeelException ex)

{}

}

});

miCross.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent evt)

{

try{

UIManager.setLookAndFeel(UIManager.getCrossPlatformLookAndFeelClassName());

SwingUtilities.updateComponentTreeUI(frame);

}

catch(ClassNotFoundException|InstantiationException|IllegalAccessException|UnsupportedLookAndFeelException ex)

{}

}

});

frame.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

if(db.closeConnection()){

JOptionPane.showMessageDialog(frame, "You are Exiting and \n Connection is closed successfully.","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);;

}

frame.dispose();

}

});

}

public void addComponentsToFrame(){

Navi.add(miHome);

about.add(miabtproject);

about.add(miabtstudent);

lkfl.add(miCross);

lkfl.add(miMotif);

lkfl.add(miNimbus);

menubar.add(Navi);

menubar.add(about);

menubar.add(lkfl);

frame.setJMenuBar(menubar);

p1.add(lblname);

p1.add(tfname);

p2.add(lbluname);

p2.add(tfuname);

p3.add(lblemail);

p3.add(tfemail);

p4.add(lblphno);

p4.add(tfphno);

p5.add(lblpass);

p5.add(tfpass);

p6.add(lblpass1);

p6.add(tfpass1);

p7.add(btnsubmit);

frame.add(p1);

frame.add(p2);

frame.add(p3);

frame.add(p4);

frame.add(p5);

frame.add(p6);

frame.add(p7);

}

}

**//**MalwareDetails.java

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.util.regex.\*;

import java.util.\*;

public class MalwareDetails{

JFrame frame;

JMenuBar menubar;

JMenu Navi,about,lkfl;

JMenuItem miHome,miabtproject,miabtstudent,miMotif,miNimbus,miCross;

JLabel lblMal,lblcode,lblstyle,lbleffect,lblcounter;

JComboBox<String> jcMal;

JTextField tfcode;

JTextArea tastyle,taeffect,tacounter;

JPanel p1,p2,p3,p4;

DBAccess db;

public MalwareDetails(){

super();

}

public MalwareDetails(DBAccess db){

this.db = db;

initializeComponents();

registerListeners();

addComponentsToFrame();

frame.setBackground(Color.lightGray);

frame.setLayout(new GridLayout(6,1));

frame.setSize(450,400);

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);

}

public void initializeComponents(){

frame = new JFrame("Malware Details");

menubar = new JMenuBar();

Navi = new JMenu("Navigations");

miHome = new JMenuItem("Home");

about = new JMenu("About");

miabtproject = new JMenuItem("About Project");

miabtstudent = new JMenuItem("About Student");

lkfl = new JMenu("LAF");

miMotif = new JMenuItem("Motif");

miNimbus = new JMenuItem("Nimbus");

miCross = new JMenuItem("Cross Platform");

lblMal = new JLabel("Malware: ");

jcMal = new JComboBox<>(db.getMalwareNames());

lblcode = new JLabel("Mal\_id: ");

tfcode = new JTextField(8);

lblstyle = new JLabel("Style of Attack: ");

tastyle = new JTextArea(2,25);

tastyle.setEditable(false);

lbleffect = new JLabel("Effect: ");

taeffect = new JTextArea(2,25);

taeffect.setEditable(false);

lblcounter = new JLabel("Counterfeit: ");

tacounter = new JTextArea(1,25);

tacounter.setEditable(false);

p1 = new JPanel();

p2 = new JPanel();

p3 = new JPanel();

p4 = new JPanel();

}

public void registerListeners(){

jcMal.addActionListener (new ActionListener () {

public void actionPerformed(ActionEvent e) {

String str = jcMal.getItemAt(jcMal.getSelectedIndex());

Vector v1 = db.getOfMalware(str);

Vector v2 = db.getOfCause((String)v1.get(0));

Vector v3 = db.getOfCounterfeit((String)v1.get(0));

String s1 = new String();

String s2 = new String();

tfcode.setText((String)v1.get(0));

tastyle.setText((String)v1.get(2));

for(Object s:v2)

s1 = s1+(String)s+"\n";

taeffect.setText(s1);

for(Object s:v3)

s2 = s2+(String)s+"\n";

tacounter.setText(s2);

}

});

miHome.addActionListener (new ActionListener () {

public void actionPerformed(ActionEvent e) {

frame.dispose();

new Home(db);

}

});

miabtproject.addActionListener (new ActionListener () {

public void actionPerformed(ActionEvent e) {

JOptionPane.showMessageDialog(frame, "Project Name: Malware Counterfeit\nDetails: This Project describes about the differenct types \nof software malwares present and their effects caused by \ntheir their presence amd ways to counterfeit it.","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

});

miabtstudent.addActionListener (new ActionListener () {

public void actionPerformed(ActionEvent e) {

JOptionPane.showMessageDialog(frame, "Name of Student: R. Sai Sathvik\nRoll number: 1602-20-737-035\nMain purpose of project: To implement a 1-Tier Architecture java application","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

});

miMotif.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent evt)

{

try{

UIManager.setLookAndFeel("com.sun.java.swing.plaf.motif.MotifLookAndFeel");

SwingUtilities.updateComponentTreeUI(frame);

}

catch(ClassNotFoundException|InstantiationException|IllegalAccessException|UnsupportedLookAndFeelException ex)

{}

}

});

miNimbus.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent evt)

{

try{

UIManager.setLookAndFeel("javax.swing.plaf.nimbus.NimbusLookAndFeel");

SwingUtilities.updateComponentTreeUI(frame);

}

catch(ClassNotFoundException|InstantiationException|IllegalAccessException|UnsupportedLookAndFeelException ex)

{}

}

});

miCross.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent evt)

{

try{

UIManager.setLookAndFeel(UIManager.getCrossPlatformLookAndFeelClassName());

SwingUtilities.updateComponentTreeUI(frame);

}

catch(ClassNotFoundException|InstantiationException|IllegalAccessException|UnsupportedLookAndFeelException ex)

{}

}

});

frame.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

if(db.closeConnection()){

JOptionPane.showMessageDialog(frame, "You are Exiting and \n Connection is closed successfully.","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);;

}

System.exit(0);

}

});

}

public void addComponentsToFrame(){

Navi.add(miHome);

about.add(miabtproject);

about.add(miabtstudent);

lkfl.add(miCross);

lkfl.add(miMotif);

lkfl.add(miNimbus);

menubar.add(Navi);

menubar.add(about);

menubar.add(lkfl);

frame.setJMenuBar(menubar);

tfcode.setEditable(false);

tacounter.setEditable(false);

taeffect.setEditable(false);

tastyle.setEditable(false);

p1.add(lblMal);

p1.add(jcMal);

p1.add(lblcode);

p1.add(tfcode);

p2.add(lblstyle);

p2.add(tastyle);

p3.add(lbleffect);

p3.add(taeffect);

p4.add(lblcounter);

p4.add(tacounter);

frame.add(p1);

frame.add(p2);

frame.add(p3);

frame.add(p4);

}

}

**//**Operations.java

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.util.regex.\*;

import java.util.\*;

import java.sql.\*;

public class Operations{

JFrame frame1,frame2,frame3;

JMenuBar menubar;

JMenu Navi,user,about,lkfl;

JMenuItem miHome,miinsert,miupdate,midelete,milogedit,milogins,miabtproject,miabtstudent,miMotif,miNimbus,miCross;

JLabel lblhead,lblmal,lblmal1,lblcode,lblstyle,lblalleffects,lbleffect,lblallmeasure,lblcounter;

JTextField tfmal,tfcode;

JTextArea tastyle,taeffect,tacounter;

JComboBox<String> jcmal,jceffect,jcmeasure;

JButton btn;

JPanel p1,p2,p3,p4,p5,p6,p7,p8,peffect,pmeasure;

String logname;

DBAccess db;

public Operations(){

super();

}

public Operations(String uname,DBAccess db){

logname = uname;

this.db = db;

initializeComponents();

registerListeners();

addComponentsToFrame();

frame1.setBackground(Color.lightGray);

frame1.setLayout(new GridLayout(10,1));

frame1.setSize(450,400);

frame1.setVisible(true);

frame1.setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);

}

public void initializeComponents(){

frame1 = new JFrame("Home->User insertions");

frame2 = new JFrame("Home->User updates");

frame3 = new JFrame("Home->User deletions");

menubar = new JMenuBar();

Navi = new JMenu("Navigations");

miinsert = new JMenuItem("insert");

miupdate = new JMenuItem("update");

midelete = new JMenuItem("delete");

miHome = new JMenuItem("Log out");

user = new JMenu("User");

milogedit = new JMenuItem("User details");

milogins = new JMenuItem("User history");

about = new JMenu("About");

miabtproject = new JMenuItem("About Project");

miabtstudent = new JMenuItem("About Student");

lkfl = new JMenu("LAF");

miMotif = new JMenuItem("Motif");

miNimbus = new JMenuItem("Nimbus");

miCross = new JMenuItem("Cross Platform");

lblhead = new JLabel("Insertion");

lblmal = new JLabel("Malware name: ");

lblmal1 = new JLabel("Malware name: ");

tfmal = new JTextField(10);

jcmal = new JComboBox<>(db.getMalwareNames());

lblcode = new JLabel("Mal\_id");

tfcode = new JTextField(5);

lblstyle = new JLabel("Style of attack");

tastyle = new JTextArea(2,25);

lblalleffects = new JLabel("All effects: ");

jceffect = new JComboBox<>(db.getEffects());

lbleffect = new JLabel("Effect: ");

taeffect = new JTextArea(2,25);

lblallmeasure = new JLabel("All Measures: ");

jcmeasure = new JComboBox<>(db.getMeasure());

lblcounter = new JLabel("Counterfeit: ");

tacounter = new JTextArea(1,25);

btn= new JButton("Submit");

p1 = new JPanel();

p2 = new JPanel();

p3 = new JPanel();

p4 = new JPanel();

peffect = new JPanel();

p5 = new JPanel();

pmeasure = new JPanel();

p6 = new JPanel();

p7 = new JPanel();

p8 = new JPanel();

}

public void registerListeners(){

jceffect.addActionListener(new ActionListener (){

public void actionPerformed(ActionEvent e){

String str = taeffect.getText()+jceffect.getSelectedItem()+"\n";

taeffect.setText(str);

}

});

jcmeasure.addActionListener(new ActionListener (){

public void actionPerformed(ActionEvent e){

String str =tacounter.getText()+jcmeasure.getSelectedItem()+"\n";

tacounter.setText(str);

}

});

btn.addActionListener(new ActionListener (){

public void actionPerformed(ActionEvent e){

if(btn.getText().equals("Submit")){

StringTokenizer strt = new StringTokenizer(taeffect.getText(),"\n");

Vector<String> str1 = new Vector<String>();

while(strt.hasMoreTokens())

str1.add(strt.nextToken());

strt = new StringTokenizer(tacounter.getText(),"\n");

Vector<String> str2 = new Vector<String>();

while(strt.hasMoreTokens())

str2.add(strt.nextToken());

JOptionPane.showMessageDialog(frame1, "Malware name: "+tfmal.getText()+"\nMalware code: "+tfcode.getText()+"\nStyle of attack: "+tastyle.getText()+"\nEffects: "+str1+"\nCounterfeit: "+str2,"INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

db.insertion(tfmal.getText(),tfcode.getText(),tastyle.getText(),str1,str2);

JOptionPane.showMessageDialog(frame1, "Insertion successfull.","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

if(btn.getText().equals("Modify")){

StringTokenizer strt = new StringTokenizer(taeffect.getText(),"\n");

Vector<String> str1 = new Vector<String>();

while(strt.hasMoreTokens())

str1.add(strt.nextToken());

strt = new StringTokenizer(tacounter.getText(),"\n");

Vector<String> str2 = new Vector<String>();

while(strt.hasMoreTokens())

str2.add(strt.nextToken());

JOptionPane.showMessageDialog(frame1, "Malware name: "+tfmal.getText()+"\nMalware code: "+tfcode.getText()+"\nStyle of attack: "+tastyle.getText()+"\nEffects: "+str1+"\nCounterfeit: "+str2,"INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

db.updation(tfmal.getText(),tfcode.getText(),tastyle.getText(),str1,str2);

JOptionPane.showMessageDialog(frame2, "Updated successfull.","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

if(btn.getText().equals("Delete")){

StringTokenizer strt = new StringTokenizer(taeffect.getText(),"\n");

Vector<String> str1 = new Vector<String>();

while(strt.hasMoreTokens())

str1.add(strt.nextToken());

strt = new StringTokenizer(tacounter.getText(),"\n");

Vector<String> str2 = new Vector<String>();

while(strt.hasMoreTokens())

str2.add(strt.nextToken());

JOptionPane.showMessageDialog(frame1, "Malware name: "+tfmal.getText()+"\nMalware code: "+tfcode.getText()+"\nStyle of attack: "+tastyle.getText()+"\nEffects: "+str1+"\nCounterfeit: "+str2,"INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

db.deletion((String)jcmal.getItemAt(jcmal.getSelectedIndex()));

JOptionPane.showMessageDialog(frame3, "Deletion successfull.","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

}

});

jcmal.addActionListener (new ActionListener () {

public void actionPerformed(ActionEvent e) {

String str = (String)jcmal.getItemAt(jcmal.getSelectedIndex());

Vector v1 = db.getOfMalware(str);

Vector v2 = db.getOfCause((String)v1.get(0));

Vector v3 = db.getOfCounterfeit((String)v1.get(0));

String s1 = new String();

String s2 = new String();

tfcode.setText((String)v1.get(0));

tastyle.setText((String)v1.get(2));

for(Object s:v2)

s1 = s1+(String)s+"\n";

taeffect.setText(s1);

for(Object s:v3)

s2 = s2+(String)s+"\n";

tacounter.setText(s2);

}

});

miHome.addActionListener (new ActionListener () {

public void actionPerformed(ActionEvent e) {

if(lblhead.getText().equals("Insertion")){

frame1.dispose();

}

if(lblhead.getText().equals("Updation")){

frame2.dispose();

}

if(lblhead.getText().equals("Deletion")){

frame3.dispose();

}

new Home(db);

}

});

miinsert.addActionListener (new ActionListener () {

public void actionPerformed(ActionEvent e) {

if(lblhead.getText().equals("Updation")){

frame2.dispose();

}

if(lblhead.getText().equals("Deletion")){

frame3.dispose();

}

lblhead.setText("Insertion");

btn.setText("Submit");

frame1.setJMenuBar(menubar);

frame1.add(p1);

frame1.add(p2);

frame1.add(p8);

frame1.add(p4);

frame1.add(peffect);

frame1.add(p5);

frame1.add(pmeasure);

frame1.add(p6);

frame1.add(p7);

frame1.setBackground(Color.lightGray);

frame1.setLayout(new GridLayout(10,1));

frame1.setSize(450,400);

frame1.setVisible(true);

frame1.setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);

}

});

miupdate.addActionListener (new ActionListener () {

public void actionPerformed(ActionEvent e) {

if(lblhead.getText().equals("Insertion")){

frame1.dispose();

}

if(lblhead.getText().equals("Deletion")){

frame3.dispose();

}

lblhead.setText("Updation");

btn.setText("Modify");

frame2.setJMenuBar(menubar);

frame2.add(p1);

frame2.add(p3);

frame2.add(p8);

frame2.add(p4);

frame2.add(peffect);

frame2.add(p5);

frame2.add(pmeasure);

frame2.add(p6);

frame2.add(p7);

frame2.setBackground(Color.lightGray);

frame2.setLayout(new GridLayout(10,1));

frame2.setSize(450,400);

frame2.setVisible(true);

frame2.setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);

}

});

midelete.addActionListener (new ActionListener () {

public void actionPerformed(ActionEvent e) {

if(lblhead.getText().equals("Insertion")){

frame1.dispose();

}

if(lblhead.getText().equals("Updation")){

frame2.dispose();

}

lblhead.setText("Deletion");

btn.setText("Delete");

frame3.setJMenuBar(menubar);

frame3.add(p1);

frame3.add(p3);

frame3.add(p8);

frame3.add(p4);

frame3.add(p5);

frame3.add(p6);

frame3.add(p7);

frame3.setBackground(Color.lightGray);

frame3.setLayout(new GridLayout(7,1));

frame3.setSize(450,400);

frame3.setVisible(true);

frame3.setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);

}

});

milogedit.addActionListener (new ActionListener () {

public void actionPerformed(ActionEvent e) {

Vector<String> v = db.getOfRegister(logname);

if(lblhead.getText().equals("Insertion")){

JOptionPane.showMessageDialog(frame1, "Username: "+v.get(0)+"\nName: "+v.get(1)+"\nEmail: "+v.get(2)+"\nPassword: "+v.get(3)+"\nPhone number: "+v.get(4),"INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

if(lblhead.getText().equals("Updation")){

JOptionPane.showMessageDialog(frame2, "Username: "+v.get(0)+"\nName: "+v.get(1)+"\nEmail: "+v.get(2)+"\nPassword: "+v.get(3)+"\nPhone number: "+v.get(4),"INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

if(lblhead.getText().equals("Deletion")){

JOptionPane.showMessageDialog(frame3, "Username: "+v.get(0)+"\nName: "+v.get(1)+"\nEmail: "+v.get(2)+"\nPassword: "+v.get(3)+"\nPhone number: "+v.get(4),"INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

}

});

milogins.addActionListener (new ActionListener () {

public void actionPerformed(ActionEvent e) {

Vector<String> v = db.getHistory(logname);

String str = new String("User name Name Date\n");

for(int i=0;i<v.size();i++){

str = str+v.get(i);

if(i%3 == 2)

str = str+"\n";

else

str = str+" ";

}

if(lblhead.getText().equals("Insertion")){

JOptionPane.showMessageDialog(frame1,str,"INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

if(lblhead.getText().equals("Updation")){

JOptionPane.showMessageDialog(frame2,str,"INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

if(lblhead.getText().equals("Deletion")){

JOptionPane.showMessageDialog(frame3,str,"INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

}

});

miabtproject.addActionListener (new ActionListener () {

public void actionPerformed(ActionEvent e) {

if(lblhead.getText().equals("Insertion")){

JOptionPane.showMessageDialog(frame1, "Project Name: Malware Counterfeit\nDetails: This Project describes about the differenct types \nof software malwares present and their effects caused by \ntheir their presence amd ways to counterfeit it.","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

if(lblhead.getText().equals("Updation")){

JOptionPane.showMessageDialog(frame2, "Project Name: Malware Counterfeit\nDetails: This Project describes about the differenct types \nof software malwares present and their effects caused by \ntheir their presence amd ways to counterfeit it.","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

if(lblhead.getText().equals("Deletion")){

JOptionPane.showMessageDialog(frame3, "Project Name: Malware Counterfeit\nDetails: This Project describes about the differenct types \nof software malwares present and their effects caused by \ntheir their presence amd ways to counterfeit it.","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

}

});

miabtstudent.addActionListener (new ActionListener () {

public void actionPerformed(ActionEvent e) {

if(lblhead.getText().equals("Insertion")){

JOptionPane.showMessageDialog(frame1, "Name of Student: R. Sai Sathvik\nRoll number: 1602-20-737-035\nMain purpose of project: To implement a 1-Tier Architecture java application","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

if(lblhead.getText().equals("Updation")){

JOptionPane.showMessageDialog(frame2, "Name of Student: R. Sai Sathvik\nRoll number: 1602-20-737-035\nMain purpose of project: To implement a 1-Tier Architecture java application","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

if(lblhead.getText().equals("Deletion")){

JOptionPane.showMessageDialog(frame3, "Name of Student: R. Sai Sathvik\nRoll number: 1602-20-737-035\nMain purpose of project: To implement a 1-Tier Architecture java application","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);

}

}

});

miMotif.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent evt)

{

try{

UIManager.setLookAndFeel("com.sun.java.swing.plaf.motif.MotifLookAndFeel");

SwingUtilities.updateComponentTreeUI(frame1);

SwingUtilities.updateComponentTreeUI(frame2);

SwingUtilities.updateComponentTreeUI(frame3);

}

catch(ClassNotFoundException|InstantiationException|IllegalAccessException|UnsupportedLookAndFeelException ex)

{}

}

});

miNimbus.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent evt)

{

try{

UIManager.setLookAndFeel("javax.swing.plaf.nimbus.NimbusLookAndFeel");

SwingUtilities.updateComponentTreeUI(frame1);

SwingUtilities.updateComponentTreeUI(frame2);

SwingUtilities.updateComponentTreeUI(frame3);

}

catch(ClassNotFoundException|InstantiationException|IllegalAccessException|UnsupportedLookAndFeelException ex)

{}

}

});

miCross.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent evt)

{

try{

UIManager.setLookAndFeel(UIManager.getCrossPlatformLookAndFeelClassName());

SwingUtilities.updateComponentTreeUI(frame1);

SwingUtilities.updateComponentTreeUI(frame2);

SwingUtilities.updateComponentTreeUI(frame3);

}

catch(ClassNotFoundException|InstantiationException|IllegalAccessException|UnsupportedLookAndFeelException ex)

{}

}

});

frame1.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

if(db.closeConnection()){

JOptionPane.showMessageDialog(frame1, "You are Exiting and \n Connection is closed successfully.","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);;

}

System.exit(0);

}

});

frame2.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

if(db.closeConnection()){

JOptionPane.showMessageDialog(frame2, "You are Exiting and \n Connection is closed successfully.","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);;

}

System.exit(0);

}

});

frame3.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

if(db.closeConnection()){

JOptionPane.showMessageDialog(frame3, "You are Exiting and \n Connection is closed successfully.","INFORMATION", JOptionPane.INFORMATION\_MESSAGE);;

}

System.exit(0);

}

});

}

public void addComponentsToFrame(){

Navi.add(miinsert);

Navi.add(miupdate);

Navi.add(midelete);

Navi.addSeparator();

Navi.add(miHome);

user.add(milogedit);

user.add(milogins);

about.add(miabtproject);

about.add(miabtstudent);

lkfl.add(miCross);

lkfl.add(miMotif);

lkfl.add(miNimbus);

menubar.add(Navi);

menubar.add(user);

menubar.add(about);

menubar.add(lkfl);

frame1.setJMenuBar(menubar);

p1.add(lblhead);

p2.add(lblmal);

p2.add(tfmal);

p8.add(lblcode);

p8.add(tfcode);

p3.add(lblmal1);

p3.add(jcmal);

p4.add(lblstyle);

p4.add(tastyle);

peffect.add(lblalleffects);

peffect.add(jceffect);

p5.add(lbleffect);

p5.add(taeffect);

pmeasure.add(lblallmeasure);

pmeasure.add(jcmeasure);

p6.add(lblcounter);

p6.add(tacounter);

p7.add(btn);

frame1.add(p1);

frame1.add(p2);

frame1.add(p8);

frame1.add(p4);

frame1.add(peffect);

frame1.add(p5);

frame1.add(pmeasure);

frame1.add(p6);

frame1.add(p7);

}

}

//DBAcess.java

import java.sql.\*;

import java.util.\*;

Set classpath=C:\Program Files\Java\jdk-16.0.1\lib\ojdbc6-11.2.0.4.jar;.;

public class DBAccess{

String pass;

static Connection con;

Statement stmt;

ResultSet rs,rsUpdate;

Savepoint savePoint,savePoint1;

int flag;

public DBAccess(){

try{

Class.forName("oracle.jdbc.OracleDriver");

con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","sathvik","vasavi");

con.setAutoCommit(false);

stmt=con.createStatement();

savePoint = con.setSavepoint("lastSave");

}

catch(ClassNotFoundException ex){

System.out.println(ex);

}

catch(SQLException ex){

System.out.println(ex);

}

}

public String getPass(String uname){

try{

rs=stmt.executeQuery("select password from registration where username = '"+uname+"'");

rs.next();

pass = rs.getString(1);

return pass;

}

catch(SQLException ex){

System.out.println(ex);

return null;

}

}

public void enterLogin(String uname,String pass){

try{

flag = stmt.executeUpdate("insert into userlogin values ('"+uname+"','"+pass+"',sysdate)");

}

catch(SQLException ex){

System.out.println(ex);

}

}

public Vector<String> getHistory(String uname){

try{

rs = stmt.executeQuery("select \* from userlogin where username = '"+uname+"'");

Vector<String> v = new Vector<>();

while(rs.next()){

v.add(rs.getString(1));

v.add(rs.getString(2));

v.add(rs.getString(3));

}

return v;

}

catch(SQLException ex){

System.out.println(ex);

return null;

}

}

public Vector<String> getMalwareNames(){//alla names

try{

rs = stmt.executeQuery("select malname from malware");

Vector<String> v = new Vector<>();

while(rs.next())

v.add(rs.getString(1));

return v;

}

catch(SQLException ex){

System.out.println(ex);

return null;

}

}

public Vector<String> getEffects(){

try{

rs = stmt.executeQuery("select effect from cause");

Vector<String> v = new Vector<>();

while(rs.next())

v.add(rs.getString(1));

return v;

}

catch(SQLException ex){

System.out.println(ex);

return null;

}

}

public Vector<String> getMeasure(){

try{

rs = stmt.executeQuery("select measure from counterfeit");

Vector<String> v = new Vector<>();

while(rs.next())

v.add(rs.getString(1));

return v;

}

catch(SQLException ex){

System.out.println(ex);

return null;

}

}

public Vector<String> getOfMalware(String malname){

try{

rs = stmt.executeQuery("select \* from malware where malname = '"+malname+"'");

Vector<String> v = new Vector<>();

while(rs.next()){

v.add(rs.getString(1));

v.add(rs.getString(2));

v.add(rs.getString(3));

}

return v;

}

catch(SQLException ex){

System.out.println(ex);

return null;

}

}

public Vector<String> getOfCause(String malid){

try{

rs = stmt.executeQuery("select effect from cause where malid = '"+malid+"'");

Vector<String> v = new Vector<>();

while(rs.next())

v.add(rs.getString(1));

return v;

}

catch(SQLException ex){

System.out.println(ex);

return null;

}

}

public Vector<String> getOfCounterfeit(String malid){

try{

rs = stmt.executeQuery("select measure from counterfeit where malid = '"+malid+"'");

Vector<String> v = new Vector<>();

while(rs.next())

v.add(rs.getString(1));

return v;

}

catch(SQLException ex){

System.out.println(ex);

return null;

}

}

public Vector<String> getOfRegister(String uname){

try{

rs = stmt.executeQuery("select \* from registration where username = '"+uname+"'");

Vector<String> v = new Vector<>();

while(rs.next()){

v.add(rs.getString(1));

v.add(rs.getString(2));

v.add(rs.getString(3));

v.add(rs.getString(4));

v.add(rs.getString(5));

}

return v;

}

catch(SQLException ex){

System.out.println(ex);

return null;

}

}

public boolean insertion(String malname,String malid,String style,Vector<String> effect,Vector<String> counter){

try{

savePoint = con.setSavepoint("lastSave");

stmt.executeUpdate("insert into malware values ('"+malid+"','"+malname+"','"+style+"')");

for(String str : effect)

stmt.executeUpdate("insert into cause values ('"+malid+"','"+(String)str+"')");

for(String str : counter)

stmt.executeUpdate("insert into counterfeit values ('"+malid+"','"+(String)str+"')");

return true;

}

catch(SQLIntegrityConstraintViolationException ae){

System.out.println("already exists");

return false;

}

catch(SQLException ex){

try{

con.rollback(savePoint);

}

catch(SQLException ex1){}

System.out.println(ex);

return false;

}

}

public boolean deletion(String malname){

try{

savePoint1 = con.setSavepoint("lastSave1");

rs = stmt.executeQuery("select malid from malware where malname = '"+malname+"'");

rs.next();

String malid = rs.getString(1);

stmt.executeUpdate("delete from cause where malid = '"+malid+"'");

stmt.executeUpdate("delete from counterfeit where malid = '"+malid+"'");

stmt.executeUpdate("delete from malware where malid = '"+malid+"'");

return true;

}

catch(SQLException ex){

try{

con.rollback(savePoint1);

}

catch(SQLException ex1){}

System.out.println(ex);

return false;

}

}

public boolean updation(String malname,String malid,String style,Vector<String> effect,Vector<String> counter){

try{

if(deletion(malname)){

if(insertion(malname,malid,style,effect,counter)){}

else{

con.rollback(savePoint1);

throw new SQLException();

}

}

else{

con.rollback(savePoint1);

throw new SQLException();

}

return true;

}

catch(SQLException ex){

try{

con.rollback(savePoint1);

}

catch(SQLException ex1){System.out.println(ex);}

System.out.println(ex);

return false;

}

}

public boolean getRegistered(String uname,String name,String email,String pass,Long phno){

try{

stmt.executeUpdate("insert into registration values ('"+uname+"','"+name+"','"+email+"','"+pass+"',"+phno+")");

return true;

}

catch(SQLException ex){

return false;

}

}

public boolean closeConnection(){

try{

con.commit();

if(!con.isClosed())

con.close();

return true;

}

catch(SQLException e){

System.out.println(e);

return false;

} }

}

**GitHub Links and Folder Structure**

<https://github.com/Sathvik8500/Malware-and-it-s-Counterfeit.git>

**Folder Structure:**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Graphical user interface, application

Description automatically generated**

**Testing:**

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, Word

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface

Description automatically generated

**Results:**  I have successfully completed my mini project “***Malware and it’s counterfeit”.***

## DISCUSSION AND FUTURE WORK

This project is useful to detect the malware and it’s counterfeit by intaking malware Id, style of attack, e.t.c. Future scope would be to make the UI more appealing by using graphics. One more feature would be that it can be helpful to build an application where in a network of computers we could detect if any of them has malware attack over it or any malware trying to infiltrate the software and give suggestions or prevent its infiltration into software and protect the network from malware attacks.

**REFERENCES**

<https://www.kaspersky.com/resource-center/threats/types-of-malware>

<https://www.upguard.com/blog/types-of-malware#toc-2>

<https://www.esecurityplanet.com/threats/malware-types/>