LOGIN LOGOUT APPLICATION

Submitted by:

Organization Profile:

EME GROUP was established in 2006 by dedicated team of faculty members who are contributing their knowledge in Research, Industry and Teaching. EME Group is a fastest growing group has founded various Institutes under its umbrella and consists:

• ENGINEERING MADE EASY: It provides coaching for Engineering Entrance Exams like: IES / GATE / PSU

• EME TECHNOLOGIES: It provides 6 Weeks / 6 Months Industrial Training to B-Tech / Diploma / MCA / BCA students

EME Group is software Development & Training Centre, managed by a team of highly qualified software & hardware professionals. They provide trusted and expert training for a few IT companies to their utmost satisfaction. Also provide coaching for exams such as GATE, IES and PSU, full study material and regular updates.

The institute’s objectives is to empower the future computer Professionals by providing them decent work atmosphere, individual attention, creating confidence in them by encouraging them take-up the Project on their own, right from selection of topic until its implementation, under the supervision and guidance of experienced and expert faculty.

EME-Overview

The “EMETECHNOLOGIES”, Mohali was formally known as “ZOOM IT”, Mohali in 7 Phase Industrial Area. EMETECHNOLOGIES are Offshore Outsourcing Consultants with a leading edge technology focus on delivering the best and most cost-effective solutions to their clients in various areas of web development services and solutions.

The team at EMETECHNOLOGIES consists of over 30 highly skilled professionals associated with Information Technology. EMETECHNOLOGIES delivers total solutions for software development and maintenance needs, serving companies from the smallest of start-ups to the largest of the Global 2000. We specialize in offshore software development and web applications.

At EMETECHNOLOGIES a talented group of designers and interface engineers are masters at effectively conveying a consistent corporate message and brand while concentrating on ensuring a pleasant and useful user experience. They help in effectively market the company by utilizing their skills in web strategy, creative interface design, corporate branding and logo design, online marketing

Project Objective

The main objective of the project is to create a LOGIN LOGOUT APPLICATION software which is divided into two major parts.

First one is LOGIN LOGOUT FORM, Second part is, the application of database and SQL server. While forming the resume, the user has an option to register the values in a database. And the companies can choose candidates with apt qualifications. Here the company representatives are asked to fill in information on basis of which the candidates can be chosen. After choosing the candidates, these are represented in a jTable.

The program is small sized and written in java using eclipse software and finally it is wrapped and made an application using an installer. With the help of installer, anyone can download and install this software.

“LOGIN LOGOUT FORM” is the perfect solution if the user wants to differentiate between human and machine.

Tools used in the Project:

* Java(eclipse)
* JDBC Connector
* Installer

Java:

The whole project code is created using eclipse in Java. Java is a computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to bytecode (class file) that can run on any Java virtual machine (JVM) regardless of computer architecture. Java is, as of 2014, one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers.[10][11] Java was originally developed by James Gosling at Sun Microsystems (which has since merged into Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

Eclipse:

Eclipse is an integrated development environment (IDE) for developing primarily with Java, but also with other languages, in particular PHP, C/C++, and HTML5. It is also an application platform framework for Java desktop applications and others. The eclipse IDE is written in Java and can run on Windows, OS X, Linux, Solaris and other platforms supporting a compatible JVM.

The eclipse Platform allows applications to be developed from a set of modular software components called modules. Applications based on the eclipse Platform (including the eclipse IDE itself) can be extended by third party developers. The eclipse Team actively support the product and seek future suggestions from the wider community.

JDBC Connector:

Java database connectivity (JDBC) is the JavaSoft specification of a standard application programming interface (API) that allows Java programs to access database management systems. The JDBC API consists of a set of interfaces and classes written in the Java programming language.

Using these standard interfaces and classes, programmers can write applications that connect to databases, send queries written in structured query language (SQL), and process the results.

The JDBC API is consistent with the style of the core Java interfaces and classes, such as java.lang and java.awt.

Installer

Installer is a program that is used to create a software that can be used to install any java project. Installer is a software that can be downloaded on any computer, irrespective of the computer platform. The project can be run after being installed by the installer. Any file to be used along with the project, can be attached to the installer while creating it. JDK is also attached to the installer in case any computer does not have the required jdk.

About the Technology: JAVA

Java is a programming language originally developed by James Gosling at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to bytecode (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere". Java is currently one of the most popular programming languages in use, and is widely used from application software to web applications.

The original and reference implementation Java compilers, virtual machines, and class libraries were developed by Sun from 1995. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java, GNU Classpath, and Dalvik.

Getting Started With Java

When you create a new computer program, you go through a multistepprocess. The process involves three important tools:

Compiler: A compiler translates your code into computer-friendly(human-unfriendly) instructions.

Virtual machine: A virtual machine steps through the computer-friendly instructions.

Application programming interface: An application programming interface contains useful prewritten code.

Java Programming Toolset

To write Java programs, you need these tools:

* You need a Java compiler.
* You need a Java virtual machine.
* You need the Java API.
* You need the Java API documentation.
* You also need some less exotic tools:
* You need an editor to compose your Java programs.

When you come right down to it, a computer program is a big bunch of text. So to write a computer program, you need an editor — a tool for creating text documents. An editor is a lot like Microsoft Word, or like any other word processing program. The big difference is that an editor adds no formatting to your text — no bold, no italic, no distinctions among fonts. Computer programs have no formatting whatsoever. They have nothing except plain old letters, numbers, and other familiar keyboard characters.

You need a way to issue commands.

You need a way to say things like “compile this program” and “run the Java virtual machine.”Every computer provides ways of issuing commands. (You can double click icons or type verbose commands in a Run dialog box.) But when you use your computer’s facilities, you jump from one window to another. You open one window to read Java documentation, another window to edit a Java program, and a third window to start up the Java compiler. The process can be very tedious.

In the best of all possible worlds, you do all your program editing, documentation reading, and command issuing through one nice interface. This interface is called an integrated development environment (IDE).

A typical IDE divides your screen’s work area into several panes — one pane for editing programs, another pane for listing the names of programs, a third pane for issuing commands, and other panes to help you compose and test programs. You can arrange the panes for quick access. Better yet, if you change the information in one pane, the IDE automatically updates the information in all the other panes.

Java Development Kit

The Java Development Kit (JDK) is a Sun Microsystems product aimed at Java developers. Since the introduction of Java, it has been by far the most widely used Java SDK. On 17 November 2006, Sun announced that it would be released under the GNU General Public License (GPL), thus making it free software. This happened in large part on 8 May 2007; Sun contributed the source code to the Open JDK.



JDK contents:

The JDK has as its primary components a collection of programming tools, including:

* java – the loader for Java applications. This tool is an interpreter and can interpret the class files generated by the javac compiler. Now a single launcher is used for both development and deployment. The old deployment launcher, jre, no longer comes with Sun JDK.
* javac – the compiler, which converts source code into Java bytecode
* jar – the archiver, which packages related class libraries into a single JAR file. This tool also helps manage JAR files.
* javadoc – the documentation generator, which automatically generates documentation from source code comments
* jdb – the debugger
* jps – the process status tool, which displays process information for current Java processes
* javap – the class file disassembler
* appletviewer – this tool can be used to run and debug Java applets without a web browser.
* javah – the C header and stub generator, used to write native methods
* javaws – the Java Web Start launcher for JNLP applications
* extcheck – a utility which can detect JAR-file conflicts.
* apt – the annotation-processing tool
* jhat – (experimental) Java heap analysis tool
* jstack – (experimental) utility which prints Java stack traces of Java threads
* jstat – (experimental) Java Virtual Machine statistics monitoring tool
* jstatd – (experimental) jstat daemon
* jinfo – (experimental) This utility gets configuration information from a running Java process or crash dump.
* jmap – (experimental) This utility outputs the memory map for Java and can print shared object memory maps or heap memory details of a given process or core dump.
* idlj – the IDL-to-Java compiler. This utility generates Java bindings from a given IDL file.
* policytool – the policy creation and management tool, which can determine policy for a Java runtime, specifying which permissions are available for code from various sources
* VisualVM – visual tool integrating several commandline JDK tools and lightweight performance and memory profiling capabilities
* wsimport – generates portable JAX-WS artifacts for invoking a web service.
* jrunscript – Java command-line script shell.

The JDK also comes with a complete Java Runtime Environment, usually called a private runtime. It consists of a Java Virtual Machine and all of the class libraries present in the production environment, as well as additional libraries only useful to developers, such as the internationalization libraries and the IDL libraries.

Copies of the JDK also include a wide selection of example programs demonstrating the use of almost all portions of the Java API.

Java (Software Platform)

An edition of the Java platform is the name for a bundle of related programs, or platform, from Sun which allow for developing and running programs written in the Java programming language. The platform is not specific to any one processor or operating system, but rather an execution engine (called a virtualmachine) and a compiler with a set of libraries that are implemented for various hardware and operating systems so that Java programs can run identically on all of them.

* Java Card: refers to a technology that allows small Java-based applications (applets) to be run securely on smart cards and similar small memory footprint devices.
* Java ME (Micro Edition): Specifies several different sets of libraries (known as profiles) for devices which are sufficiently limited that supplying the full set of Java libraries would take up unacceptably large amounts of storage.
* Java SE (Standard Edition): For general purpose use on desktop PCs, servers and similar devices.
* Java EE (Enterprise Edition): Java SE plus various APIs useful for multi-tier client–server enterprise applications.

As of September 2009, the current version of the Java Platform is specified as either 1.6.0 or 6 (both refer to the same version). Version 6 is the product version, while 1.6.0 is the developer version.

The Java Platform consists of several programs, each of which provides a distinct portion of its overall capabilities. For example, the Java compiler, which converts Java source code into Java bytecode (an intermediate language for the Java Virtual Machine (JVM)), is provided as part of the Java Development Kit (JDK). The Java Runtime Environment (JRE), complementing the JVM with a just-in-time (JIT) compiler, converts intermediate bytecode into native machine code on the fly. Also supplied are extensive libraries, pre-compiled in which are several other components, some available only in certain editions.

The essential components in the platform are the Java language compiler, the libraries, and the runtime environment in which Java intermediate byte code "executes" according to the rules laid out in the virtual machine specification.

### Java Virtual Machine

The heart of the Java Platform is the concept of a "virtual machine" that executes Java byte code programs. This byte code is the same no matter what hardware or operating system the program is running under. There is a JIT compiler within the Java Virtual Machine, or JVM. The JIT compiler translates the Java byte code into native processor instructions at run-time and caches the native code in memory during execution.

Java Byte code:

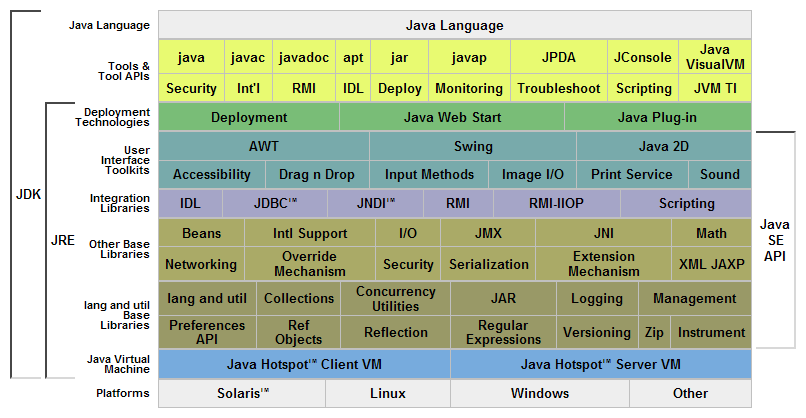
The use of bytecode as an intermediate language permits Java programs to run on any platform that has a virtual machine available. The use of a JIT compiler means that Java applications, after a short delay during loading and once they have "warmed up" by being all or mostly JIT-compiled, tend to run about as fast as native programs. Since JRE version 1.2, Sun's JVM implementation has included a just-in-time compiler instead of an interpreter.

Although Java programs are platform independent, the code of the Java Virtual Machine (JVM) that execute these programs is not; every supported operating platform has its own JVM.

### Class libraries

In most modern operating systems, a large body of reusable code is provided to simplify the programmer's job. This code is typically provided as a set of dynamically loadable librariesthat applications can call at runtime. Because the Java Platform is not dependent on any specific operating system, applications cannot rely on any of the pre-existing OS libraries. Instead, the Java Platform provides a comprehensive set of its own standard class libraries containing much of the same reusable functions commonly found in modern operating systems.

The Java class libraries serve three purposes within the Java Platform. First, like other standard code libraries, the Java libraries provide the programmer a well-known set of functions to perform common tasks, such as maintaining lists of items or performing complex string parsing. Second, the class libraries provide an abstract interface to tasks that would normally depend heavily on the hardware and operating system. Tasks such as network access and file access are often heavily intertwined with the distinctive implementations of each platform. The Java java.net and java.io libraries implement an abstraction layer in native OS code, then provide a standard interface for the Java applications to perform those tasks. Finally, when some underlying platform does not support all of the features a Java application expects, the class libraries work to gracefully handle the absent components, either by emulation to provide a substitute, or at least by providing a consistent way to check for the presence of a specific feature. The Android OS is using Java class libraries which are open source for anyone to use.



JAVA Platform diagram from Sun

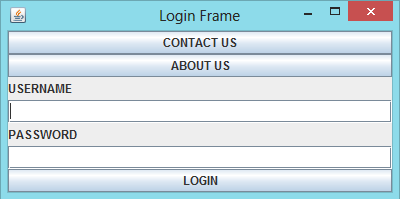
Execution environment

Programs intended to run on a JVM must be compiled into a standardized portable binary format, which typically comes in the form of .class files. A program may consist of many classes in different files. For easier distribution of large programs, multiple class files may be packaged together in a .jar file (short for Java archive).

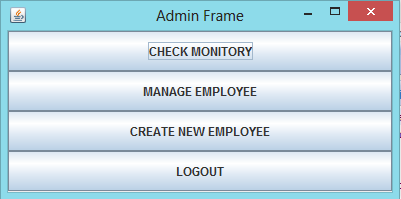
The JVM runtime executes .class or .jar files, emulating the JVM instruction set by interpreting it, or using a just-in-time compiler (JIT) such as Sun's HotSpot. JIT compiling, not interpreting, is used in most JVMs today to achieve greater speed. Ahead-of-time compilers that enable the developer to precompile class files into native code for a particular platforms also exist.Like most virtual machines, the Java Virtual Machine has a stack-based architecture akin to a microcontroller/microprocessor. However, the JVM also has low-level support for Java-like classes and methods, which amounts to a highly idiosyncratic memory model and capability-based architecture.The JVM, which is the instance of the 'JRE' (Java Runtime Environment), comes into action when a Java program is executed. When execution is complete, this instance is garbage collected. JIT is the part of the JVM that is used to speed up the execution time. JIT compiles parts of the byte code that have similar functionality at the same time, and hence reduces the amount of time needed for compilation.

CAPTCHA REGISTRATION FORM APP

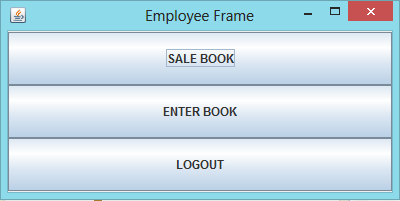
1. MAIN PAGE:



1. ADMIN FRAME



1. EMPLOYEE FRAME



JAVA CODE FOR LOGIN LOGOUT APP

1. JAVA CODE FOR MAIN FORM:

package abc1;

import Dbconnect.\*;

import java.awt.Dimension;

import java.awt.GridLayout;

import java.awt.Insets;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.\*;

import javax.swing.text.JTextComponent;

import java.sql.\*;

public class Mainframe extends Dbconn {

public Mainframe(){

//FRAME AND PANEL

final JFrame f=new JFrame("Login Frame");

JPanel p=new JPanel(new GridLayout(0,1));

//TEXT FIELD

final JTextComponent t1=new JTextField();

t1.setBounds(20,150,20,40);

//PASSWORD FIELD

final JPasswordField ps1=new JPasswordField();

ps1.setEchoChar('\*');

//BUTTON

JButton b1=new JButton("CONTACT US");

b1.setBounds(20,130,80,80);

JButton b2=new JButton("ABOUT US");

b1.setBounds(50,130,80,80);

JButton b3=new JButton("LOGIN");

b3.setBounds(20,30,80,80);

//LABELS

JLabel L1=new JLabel("USERNAME");

L1.setBounds(20,50,80,80);

JLabel L2=new JLabel("PASSWORD");

L1.setBounds(20,30,80,80);

//Contact and About button

p.add(b1);

p.add(b2);

//Labels

p.add(L1);

p.add(t1);//username

p.add(L2);

p.add(ps1);//password

//ACTION-LISTNER

b3.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent ae) {

final String Um= t1.getText();

@SuppressWarnings("deprecation")

final String ps= ps1.getText();

try{

String sql= "select USERNAME,PASSWORD from admin";

PreparedStatement s= con.prepareStatement(sql);

ResultSet r=s.executeQuery(sql);

while(r.next()){

String u= r.getString("USERNAME");

String p=r.getString("PASSWORD");

if (Um.equals(u)&&ps.equals(p)) {

f.dispose();

JOptionPane.showMessageDialog(null,"Username and Password exist");

new Admin();

}

else {

String query="select USERNAME,PASSWORD from employee";

PreparedStatement q = con.prepareStatement(query);

ResultSet qr=q.executeQuery(query);

while(qr.next()){

String uq=qr.getString("USERNAME");

String pq=qr.getString("PASSWORD");

if (Um.equals(uq)&&ps.equals(pq)) {

f.dispose();

JOptionPane.showMessageDialog(null,"Username and Password exist");

new Employee();

}

else{

JOptionPane.showMessageDialog(null, "Please Check Username and Password ");

}

}

}

}}

catch(Exception e)

{

System.out.println(e);

}}});

p.add(b3);

Insets insets = p.getInsets();

Dimension size = b1.getPreferredSize();

b1.setBounds(25 + insets.left, 5 + insets.top,size.width, size.height);

size = b2.getPreferredSize();

b2.setBounds(55 + insets.left, 40 + insets.top,size.width, size.height);

size = b3.getPreferredSize();

b3.setBounds(150 + insets.left, 15 + insets.top,size.width + 50, size.height + 20);

f.getContentPane().add(p);

f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

f.setSize(400,200);

f.setVisible(true);

}

public static void main(String[]args){

new Mainframe();

}

}

1. JAVA CODE FOR ADMIN FORM:

package abc1;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

public class Admin {

public Admin(){

final JFrame f=new JFrame("Admin Frame");

JPanel p=new JPanel(new GridLayout(0,1));

JButton b114=new JButton("LOGOUT");

JButton b111=new JButton("CHECK MONITORY");

JButton b112=new JButton("MANAGE EMPLOYEE");

JButton b113=new JButton("CREATE NEW EMPLOYEE");

b114.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent ae) {

f.dispose();

new Mainframe();

}

});f.setVisible(true);

p.add(b111);

p.add(b112);

p.add(b113);

p.add(b114);

f.getContentPane().add(p);

f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

f.setSize(400,200);

f.setVisible(true);

}

}

1. JAVA CODE FOR EMPLOYEE FORM:

package abc1;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

public class Employee {

public Employee(){

final JFrame f=new JFrame("Employee Frame");

JPanel p=new JPanel(new GridLayout(0,1));

JButton b13=new JButton("LOGOUT");

JButton b11=new JButton("SALE BOOK");

JButton b12=new JButton("ENTER BOOK");

f.setDefaultCloseOperation(JFrame.HIDE\_ON\_CLOSE);

b13.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent ae) {

f.dispose();

new Mainframe();

}

});

p.add(b11);

p.add(b12);

p.add(b13);

f.getContentPane().add(p);

f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

f.setSize(400,200);

f.setVisible(true);

}

}

1. JAVA CODE FOR DATABASE CONNECTION:

package connect;

import java.sql.Connection;

import java.sql.DriverManager;

public class Dbconn {

public Connection con;

public Dbconn(){

try{

Class.forName("com.mysql.jdbc.Driver");

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/cap","root","");

System.out.println("connectoin created");

}

catch(Exception e){

System.out.println(e.getMessage());

}

}

}

Bibliography

Internet:

• http://wikipedia.org/

• http://www.google.co.in/

• http://www.roseindia.net/

• http://www.mkyong.com

• http://forums.techarena.in

• http://www.w3schools.com

• http://websequencediagrams.com

• http://www.freejavaguide.com/

• http://www.buildingjavaprograms.com

• http://www.kodejava.org

• http://learnjava.awardspace.com

• http://www.javaworld.com

• www.javabeginner.com/

• www.webdeveloper.com

• www.slideshare.net

• www.webdesigners-directory.com/webdesignskill/JAVA

Books:

• Complete Reference for java(j2ee) By Herbert Schildt

• Thinking In Java By Bruce Eckel

• Java in a nutshell By David Flanagan

• Java Programming Language By Ken Arnold, James Gosling, David Holmes

• Head First Java By Kathy Sierra, Bert Bates

• Programming with Java By E Balagurusamy

• Beginning JSP Web Development By Casey Kochmer

• More Servlets and Java-Server Pages By Marty Hall

• Core Web Programming By Marty Hall and Larry Brown