THEMIS: A Mutually Verifiable Billing System

for the Cloud Computing Environment

**Modules**

* Cloud Notary Authority (CNA)

**Module Description**

**Cloud Notary Authority (CNA)**

Cloud Notary Authority acts as a THEMIS in our cloud billing transaction. He is an authority to generate the billing transaction for the cloud service. The CNA provides a mutually verifiable integrity mechanism that combats the malicious behavior of users or the CSP. The process, which involves a generation of mutually verifiable binding information among all the involved entities on the basis of a one-way hash chain (One time key), is computationally efficient for a user and the CSP. If user wants billing for the service then it sends the contract of the user and contract of CSP to the CNA. In CNA it checks both the contract; if it is found as identical then it generates the bill as binding information and sends the confirmation message to the user and the CSP. If it is not identical then it receives the log details from the monitor. If forgery found at user side it sends the penalty to the user. If it found at CSP side it cancels the payment to the CSP. CNA provide the billing transaction which can be verifiable and also forgery resistive in cloud environment.

**MODULE DIAGRAM:**

**Cloud Notary Authority (CNA)**

**CNA**

CSP Details

User Details

Bill Generation

**USER**

**CSP**

**Local Repository**

Contract

BILL Transaction

BILL Transaction

**IMPLEMENTAION**

**Cloud Notary Authority (CNA)**

public class receiveSLA extends HttpServlet {

String saveFile,uploadedfile;

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public receiveSLA() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

ServletContext context=getServletContext();

String status=request.getParameter("name");

try

{

String contentType = request.getContentType();

System.out.println("content type "+contentType);

if ((contentType != null) && (contentType.indexOf("multipart/form-data") >= 0)) {

DataInputStream in = new DataInputStream(request.getInputStream());

//we are taking the length of Content type data

int formDataLength = request.getContentLength();

byte dataBytes[] = new byte[formDataLength];

int byteRead = 0;

int totalBytesRead = 0;

//this loop converting the uploaded file into byte code

while (totalBytesRead < formDataLength)

{

byteRead = in.read(dataBytes, totalBytesRead, formDataLength);

totalBytesRead += byteRead;

}

int fileSize=dataBytes.length;

String file = new String(dataBytes);

//for saving the file name

// System.out.println("file is nnnnnnnnnnn="+file+"xxxxxxxxxxxxxxx");

saveFile= file.substring(file.indexOf("filename=\"") + 10);

// System.out.println("save file is "+file);

saveFile = saveFile.substring(0, saveFile.indexOf("\n"));

saveFile = saveFile.substring(saveFile.lastIndexOf("\\") + 1, saveFile.indexOf("\""));

int lastIndex = contentType.lastIndexOf("=");

String boundary = contentType.substring(lastIndex + 1, contentType.length());

int pos;

//extracting the index of file

pos = file.indexOf("filename=\"");

pos = file.indexOf("\n", pos) + 1;

pos = file.indexOf("\n", pos) + 1;

pos = file.indexOf("\n", pos) + 1;

int boundaryLocation = file.indexOf(boundary, pos) - 4;

int startPos = ((file.substring(0, pos)).getBytes()).length;

int endPos = ((file.substring(0, boundaryLocation)).getBytes()).length;

// creating a new file with the same name and writing the content in new file

String path = request.getRealPath("");

System.out.println("path="+path);

System.out.println("saved name is"+saveFile);

// FileOutputStream fileOut = new FileOutputStream("D:\\VIVEK\\ITJCC04\\ITJCC04\\PUBLIC CLOUD\\build\\web\\"+saveFile);

uploadedfile =context.getInitParameter("uploadedpath");

FileOutputStream fileOut = new FileOutputStream(uploadedfile+"\\"+saveFile);

fileOut.write(dataBytes, startPos, (endPos - startPos));

fileOut.flush();

fileOut.close();

HttpSession hs=request.getSession();

hs.setAttribute("filename", saveFile);

String[] s=saveFile.split("-");

String name=s[0];

System.out.println(s[1]+"hnhnhnhnh");

String path1="E:/THEMIS/ITJCC05-Themis/WebContent/SLA"+"/"+saveFile;

int p=s[1].indexOf(".");

String sw=s[1].substring(0,p);

System.out.println(s[1]+"gggggg"+name+"jjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjj"+path1+"ioio"+sw);

try

{

DBClass1 db= new DBClass1();

System.out.println("db con is "+db);

java.sql.Connection cont=db.connect();

System.out.println("hghghg"+cont);

/\*java.sql.Connection con=Dbcontion.getinstance().connect();

System.out.println(con+":::::::::::::::::::::::::::::::::::::::");\*/

Statement st= cont.createStatement();

int j=st.executeUpdate("insert into receivesla values('"+name+"','"+path1+"','"+0+"','"+sw+"','"+0+"')");

System.out.println("insert into receivesla values('"+name+"','"+path1+"','"+0+"','"+sw+"')");

}

catch(Exception e)

{

e.printStackTrace();

}

response.sendRedirect("http://localhost:8080/ITJCC05/Bill1.jsp");

/\*RequestDispatcher rdp=request.getRequestDispatcher();

dp.forward(request, response);\*/

}

else

{

System.out.println("file not uploaded");

}

String ContentType;

byte[] b;

}

package db;

import java.sql.Connection;

import java.sql.DriverManager;

public class DBClass1 implements Interface\_DBClass

{

Connection con=null;

String url = "jdbc:mysql://192.168.0.9:3306/";

String dbName = "itjcc05-themis";

String driver = "com.mysql.jdbc.Driver";

String userName = "root";

String password = "root";

public Connection connect()

{

try

{

System.out.println("hi");

Class.forName(driver);

con=DriverManager.getConnection(url+dbName,userName,password);

}

catch(Exception e)

{

System.out.println(e);

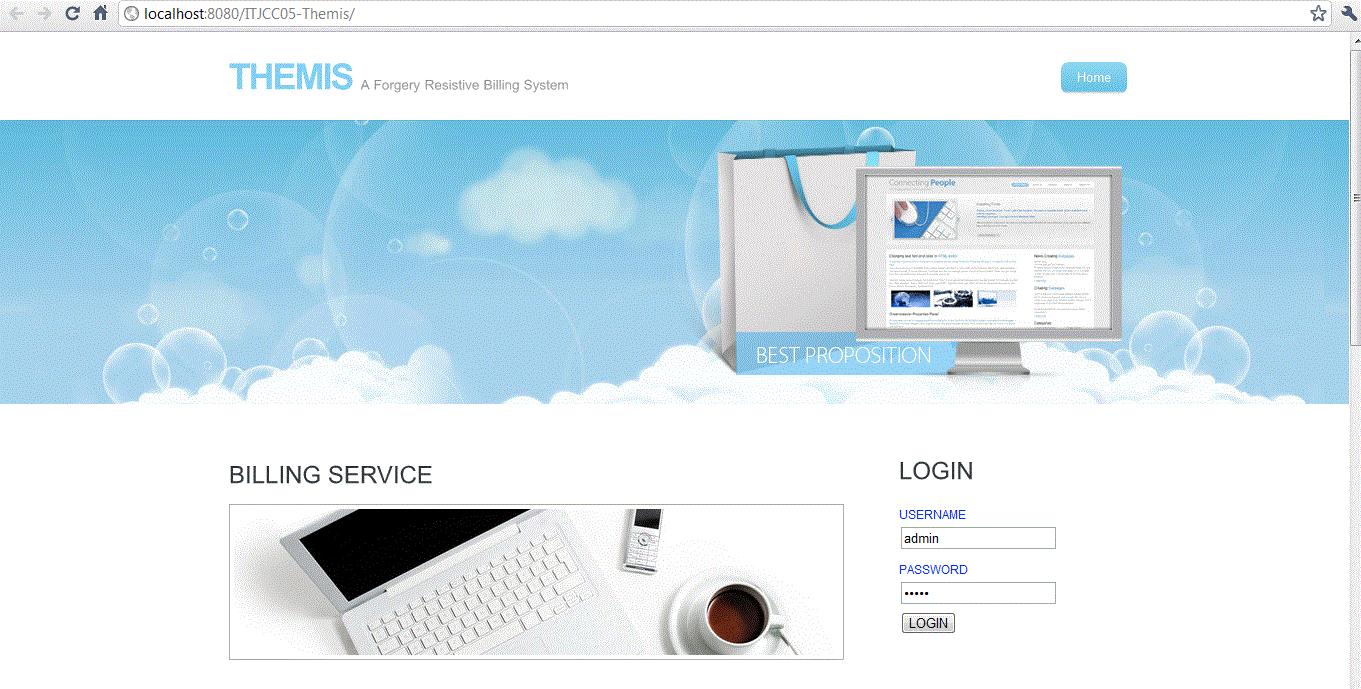
}

return con;

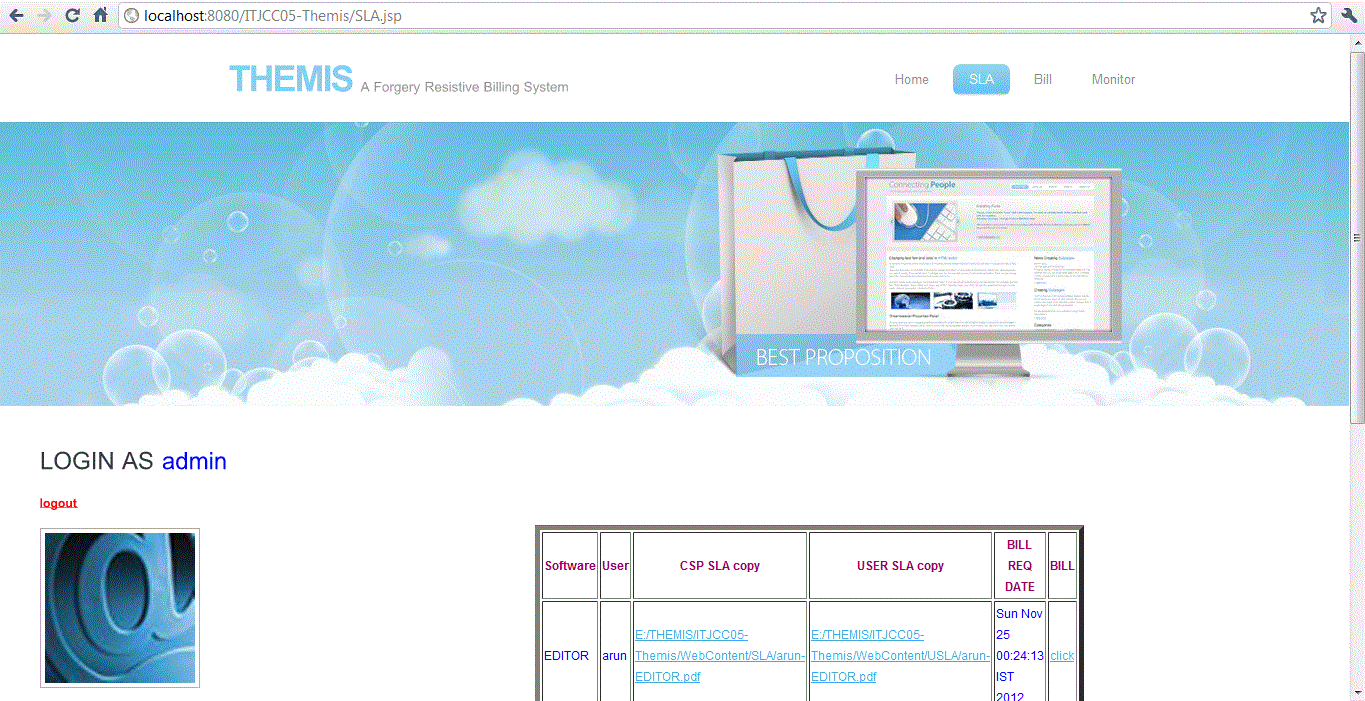
}

}

**Cloud Notary Authority (CNA)**

****

After logging from this page, it will show new form for receiving sla copies from CSP and user.

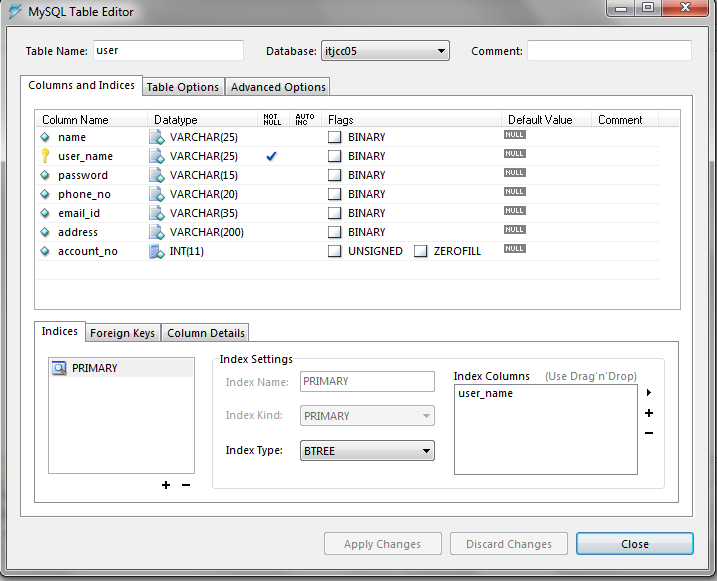


**DATABASE TABLE:**

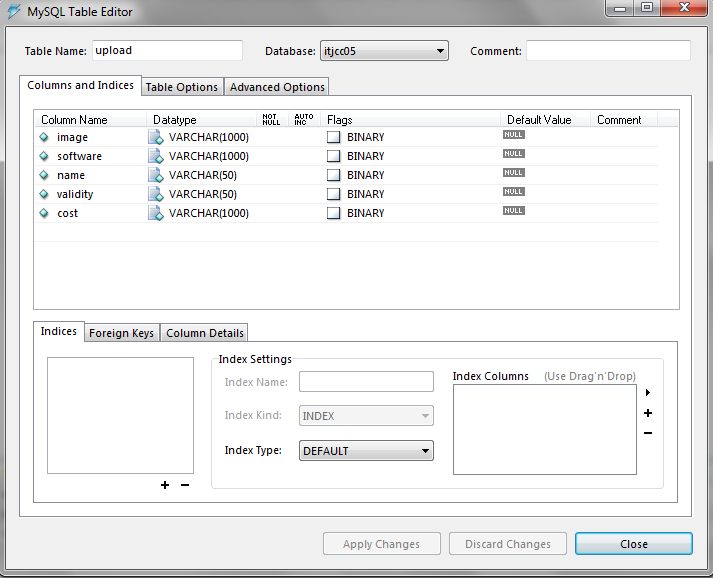
**Database 1:itjcc05**

**Table 1: Registration**

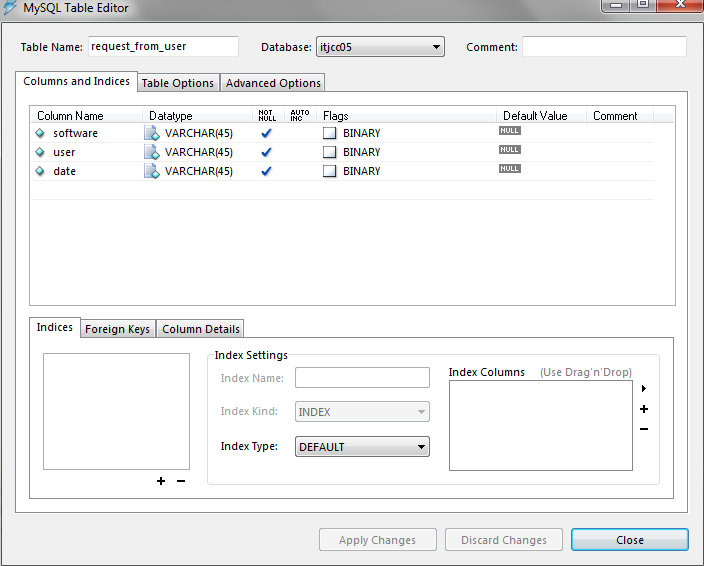
It contains user registration details for authenticating the users.



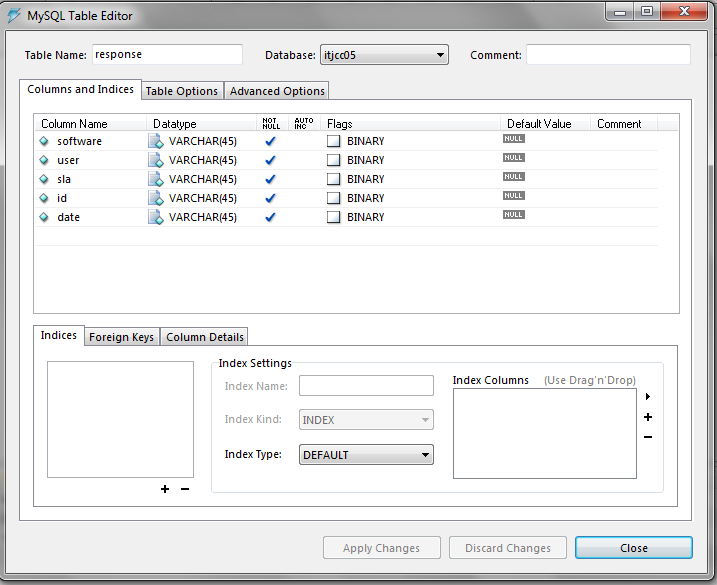
**Table 2: Upload**

****

**Table 3: Request\_from\_user**

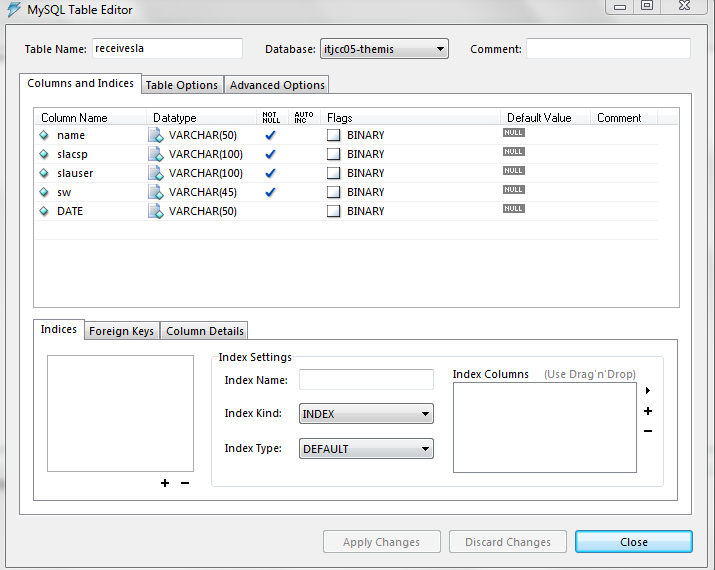
****

**Table 4: Response**

****

**Database 2:itjcc05-themis**

**Table 1: receivesla**

****