**CHAPTER 1**

**INTRODUCTION**

**Chapter 1**

**INTRODUCTION**

Cloud computing is a term that originated in the late 2000s. It is based on the utilization and consumption of computer resources. Cloud computing is the deployment of groups of servers that are remote and the software networks. This is done is such a way that data storage is centralized and there is online access to the computer services or resources.

Cloud computing depends on sharing the said resources. This aims coherence and also the flexibility of scaling. This is similar to a utility over a network.

Cloud computing, most often referred to as "the cloud", focuses on optimizing the effectiveness of the shared resources. This is because cloud resources are both shared by multiple users and dynamically reallocated depending on the demand. This works best in allocating resources to users.

Thanks to all this, there is an option to companies and organizations to scale up or scale down, depending on whether the computing needs increase or demands decrease.

**1.1 CLOUD SECURITY**

Cloud computing security or cloud security is one of the most trending and most evolving branch of computer network security, and more so the information security. It includes a set of policies, technologies, and controls which are put together so as to protect user data, applications, and also the cloud computing infrastructure.

Coming to security issues in cloud, there are two types of attacks – outside attacks and insider attacks.

**1.1.1 OUTSIDE ATTACKS**

Outside attacks are of two categories: security issues that faced by cloud providers (CSPs for SaaS, PaaS an AaaS) and security issues that are faced by the users (the consumers of the data/ services of the cloud).These issues are tackled through firewalls, encryptions and decryptions etc.

**1.1.2 INSIDER ATTACKS**

An internal attack can be termed as the attack where an individual or a group within an organization tries to gain unauthorized access to the company operations with the intention of exploit. This requires resources, tools and skill of great magnitude such that the attack launched is sophisticated. The attacker also should make sure to leave on evidence.

The possibly intruders can be well-skilled and revengeful employees or users who might benefit from the data and resources the attack will uncover.

To handle internal attacks, intrusion detection systems are implemented. All forms of attacks are logged and reviewed regularly.

**1.2 ACCESS CONTROL**

Access control can be defined as the method of controlling or limiting the access of a user to any computer system including the resources that are physical as well as virtual. In the world of cloud computing, it is defined as the process by which users are provided with access and also certain privileges to computer systems, physical and virtual resources or information.

In these systems, users are expected to provide credentials before they are granted the access. In physical systems, these credentials can be illustrated in various methods, but credentials in virtual environment are strictly not transferrable. This is what provides the most security.

**1.3 MOTIVATION**

The present scenario in the computing world includes high performance networks, affordable computers and storage devices and also hardware virtualization. The factors that have led to the growth of cloud computing are service-oriented architecture, autonomic and utility computing etc.

**1.4 ISSUES**

There are a number of security issues/concerns associated with cloud computing and these issues fall into two broad categories – outside threats and internal attacks.

Following some simple security principles like segregating duties or roles of the employees and controlling the access levels will help achieve the overall security as deemed required.

**1.5 PROBLEM STATEMENT**

Implementation Of Risk Based Access Control Module In Cloud Computing

**1.6 LITERATURE SURVEY**

Cloud security involves various factors like risks, threat matrixes, and security parameters. For the same, we have referred to various papers and online materials.

Our work is mainly derived from the three papers listed here.

The respective concepts are as follows:

1. **A Dynamic Trust Model Based on Naive Bayes Classifier for Ubiquitous Environments**

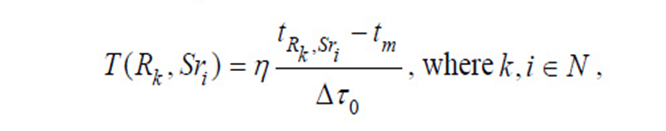
Weiwei Yuan, Donghai Guan, Sungyoung Lee\*, and Youngkoo Lee

Department of Computer Engineering, Kyung Hee University, Korea {weiwei, donghai, sylee}@oslab.khu.ac.kr, yklee@khu.ac.kr

From this paper, we have adopted the Naïve Bayes Classifier Trust Model and modified it to fit our requirements. We have also adopted the trust model factors –the factors that are required in the calculation of risk and trust.

The equation used for the calculation of an important risk value is from the same paper, quoting:

*T* ( *Rk* , *Sri* ) is used to denote the time based operator for recommender *Rk* to service requester *Sri* .



**Fig 1.6.1: Base Equation**

Here *T* ( *Rk* , *Sri* ) denotes the time when last communication between *Rk* and *Sri* happened. And *η* is time adapting operator. Suppose our measurement for time is based on a time window [ *t* *m* , *tn* ] , let ∆*τ* 0 = *t* *m* − *tn* .

Where:

This equation has been modified to:

Risk = (η \* SecurityLevel risk )/(YOE risk \* T)

Where,

* + η = faulty session/total sessions
  + SecurityLevelrisk = is the risk factor depending on the Security level
  + YOE risk =  is the risk factor depending on the Years Of Experience
  + T = Window period for security risk analysis. Value of T will be in terms of months.

1. **A Trust-and-Risk Aware RBAC Framework: Tackling Insider Threat**

Nathalie Baracaldo University of Pittsburgh School of Information Sciences nab62@pitt.edu

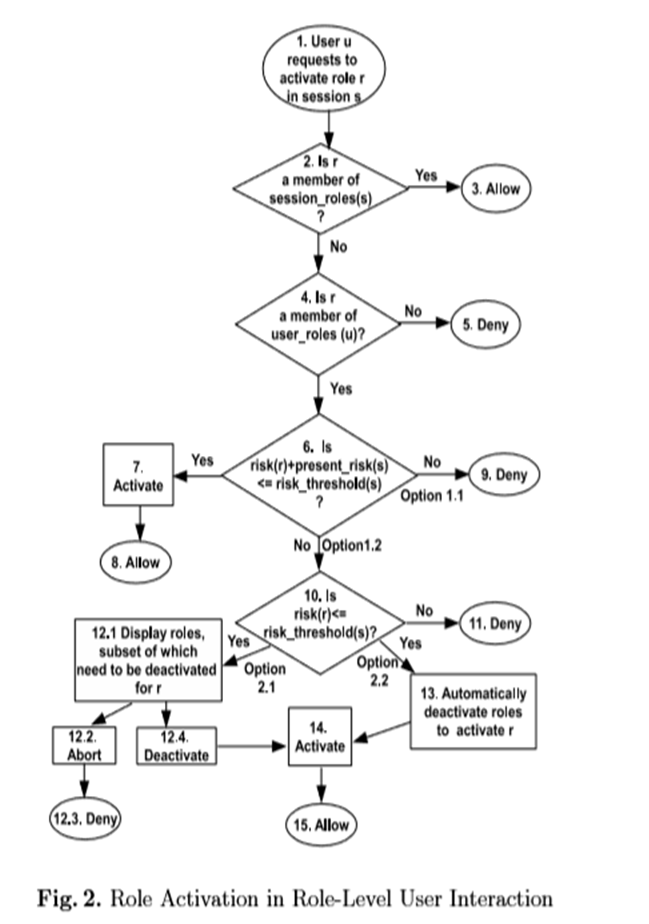
James Joshi University of Pittsburgh School of Information Sciences jjoshi@sis.pitt.edu

This paper has helped us design our entire module around insider attacks and also define the static, dynamic and adaptive modules.

1. **Risk-Aware RBAC Sessions**

Khalid Zaman Bijon, Ram Krishnan, and Ravi Sandhu

Institute for Cyber Security & Department of Computer Science, Institute for Cyber Security & Department of Electrical and Computer Engineering University of Texas at San Antonio

From this paper, we have taken the flow of the module where we determine the module behavior with any user based on a series of decisions.

CHAPTER 2

REQUIREMENT ANALYSIS

**Fig 1.6.2: Flow Chart**

**CHAPTER 2**

**EXPERIMENTAL SET UP**

**Chapter 2**

**2.1 HARDWARE SET UP**

**PROCESSOR:** Any processors of the present generation

**MAIN MEMORY:** 256 MB and above

**2.2 SOFTWARE SET UP**

**FRONT END TECHNOLOGY**: Java Swings

**IDE:** Eclipse

**OPERATING SYSTEM:** Any operating system capable of running JRE

**BACK END TECHNOLOGY**: MySQL Workbench

**2.2.1 JAVA SWINGS**

The GUI widget toolkit for Java is Swings. It belongs to Oracle's Java Foundation Classes (JFC). This is an application programming interface for providing a graphical user interface (GUI) for the Java programs. Swings are written in Java and they are platform independent. A single thread programming model is followed here. We are utilizing Java Swings to ensure the platform independence and the light weight feature which facilitates faster execution.

**2.2.2 ECLIPSE**

This integrated development environment contains a work space and an extensible plug-in system. Eclipse is written in Java. It provides the required functionality within the runtime system, on top of it. Equinox is the basis for the runtime system.

**2.2.3 MYSQL WORKBENCH**

MySQL is the visual database developed by Oracle Corporation. It integrates SQL development, database design, creation and maintenance. It provides a myriad of features like allowing the connections to servers, schema synchronization and more.

**CHAPTER 3**

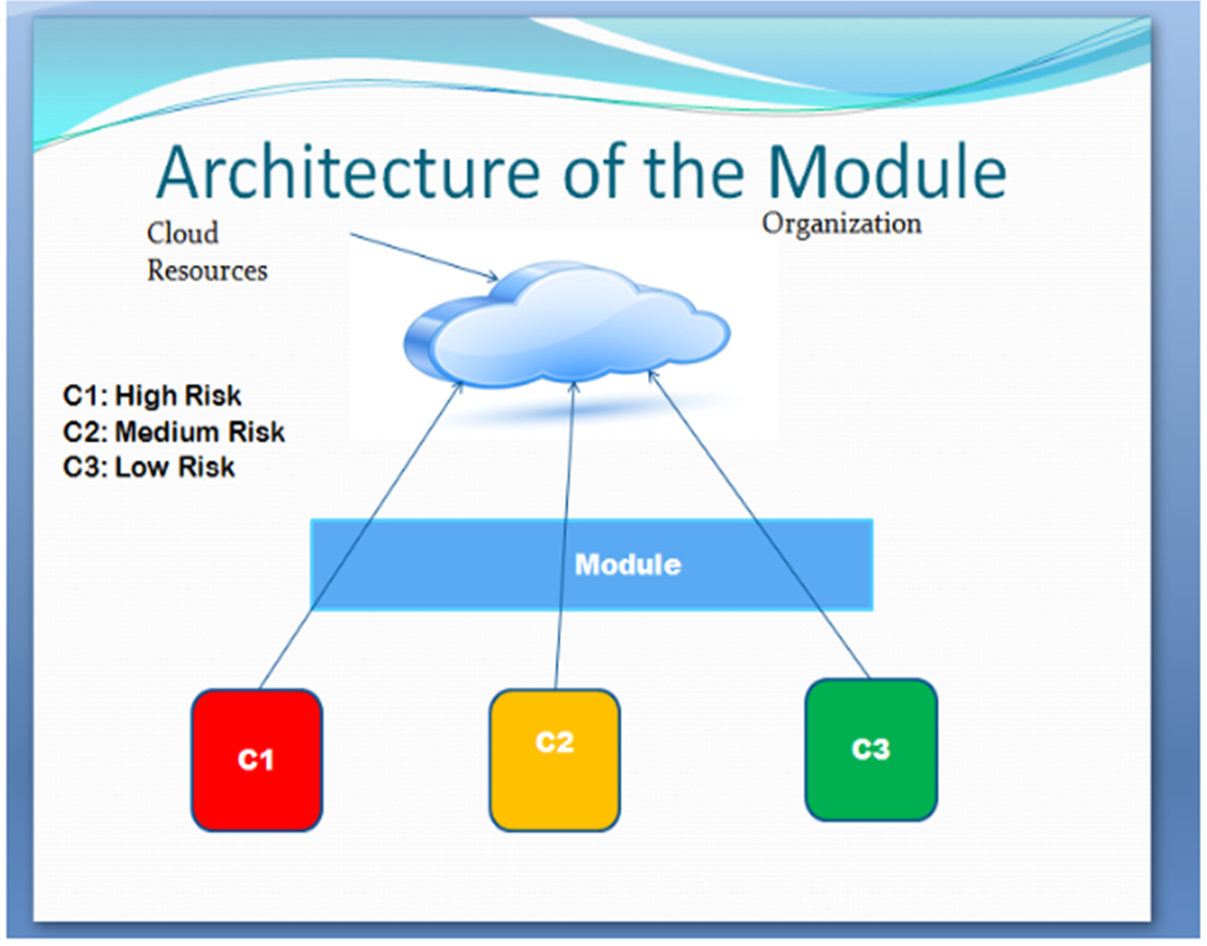
**ARCHITECTURE AND DESIGN**

**Chapter 3**

**ARCHITECTURE AND DESIGN**

**3.1 ARCHITECTURE**

Our module is placed in between the users and the organization’s resources placed in the cloud. A user can access the resources only after being assessed by the module.

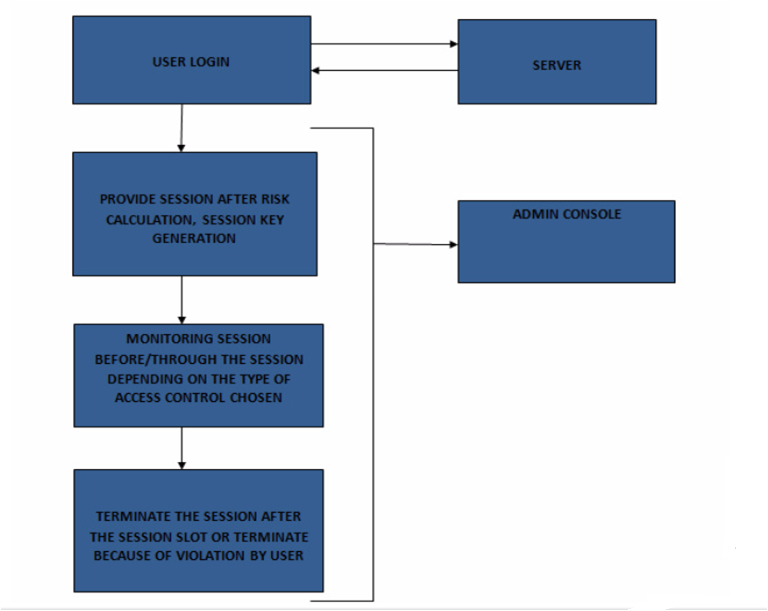


**Fig 3.1 : Architecture of the module**

**3.2 DESIGN**

We start our module with the user login which, on authentication, leads to the module interface. After the necessary computations, the user is allowed access and on termination, the module ends with a log out. There is also an admin console module which works in parallel with the above module and monitors the users.

Our module has two bifurcations – the user console and the admin console. The admin console monitors the user console.



**Fig 3.2 : Design of the module**

The user console begins with the user login authentication which gets sent to the server. After authentication, the module offers the user a set of options, which will be explained further. Once the permissions for the file(s) are chosen, the risk values are calculated. Based on these computations, decisions regarding whether to provide a session or not are made.

The session is either monitored or not, depending on the type of access control methodology chosen.

At the end of the session, the user behavior is logged.

The admin console consists of parameters to supervise the users. There are options to disable accounts, terminate connections and monitor the users in general.

**CHAPTER 4**

**IMPLEMENTATION**

**Chapter 4**

**IMPLEMENTATION**

This chapter provides a complete idea of the implementation of the module.

* 1. **USER REGISTRATION**

Every new user has to register to start with. Basic details like employee ID, name, date of birth, designation, contact number and email ID are collected. These details are stored in a data base. The user is now a valid user who is provided with a unique username and password.

* 1. **USER LOGIN**

Every user has to install the module on his system. The module, when the user logs in successfully, checks if it is already installed or not, in a certain location. Users are expected to maintain the location where the module was installed.

* 1. **GRANT OF SESSION**

To grant the session, risk parameters are analyzed and computations are performed. We have two types of risk values – Threshold Risk Value based on the risk parameters collected in 4.1, and Current Risk Values based on file permissions. Current Risk Values are calculated before a session and checked against the Threshold Risk Value. If it’s lesser than or equal to the Threshold Risk Value, we grant a session, else we don’t.

* 1. **ON EVENT OF GRANT OF A SESSION**

A session key is generated. The user is presented with options to either:

1. Create a file
2. Select a file and the permissions from the list of files that will be displayed to the user.

Each permission will have weightage based on which Current Risk Value will be calculated.

* 1. **PROVISION OF PERMISSIONS**

When the user opts for permissions, we calculate the risk and grade it against the Threshold Risk Value that’s already assigned to the user. If it’s lesser than or equal to the Threshold Risk Value, we grant a session, else we don’t.

* 1. **MONITORING THE USER**

After granting the session, we don’t monitor the user through the session unless it’s an adaptive session.

* 1. **THE SESSION TERMINATION**

If the user doesn’t exhibit risky behavior, we allow complete running of the session. Else, we terminate the session. In cases of termination and disabling of account, the user has to approach the network admin for enabling and is expected to have justifications as to what led to the unusual behavior.

**4.8 RISK PARAMETERS**

**4.8.1 THRESHOLD RISK PARAMETERS**

We make use of 8 parameters to calculate the final risk value.

|  |  |  |
| --- | --- | --- |
| **RISK VALUE** | **MINIMUM VALUE** | **MAXIMUM VALUE** |
| YEARS OF EXPERIENCE | 0.05 | 0.7 |
| DESIGNATION/SECURITY LEVEL | 0.0 | 0.7 |
| FAULTY SESSIONS/TOTAL SESSIONS | 0 | 1 |
| VOUCHING | IF VOUCHED, RISK IS LESS | IF NOT VOUCHED, RISK IS MORE |
| LOCATION OF ACCESS | WITHIN THE ORGANISATION, THE FLAG IS 0 | OUTSIDE THE ORGANIZATON, THE FLAG IS 0 |
| TIME OF ACCESS | WITHIN WORK HOURS= 0 | BEYOND WORK HOURS = 1 |
| PRESET BIASING | ADMIN HAS TRUST MEANS THE FLAG GETS SET TO 0 | ADMIN DOESN’T HAVE TRUST MEANS THE FLAG GETS SET TO 1 |
| TIME BASED RISK | 0 | 0.5 |
| FINAL RISK VALUE | 0 | 0.82 |

**Table 4.8.1: Threshold Risk Parameters**

These values below are assumed and assigned in an incremental value.

The values for:

1. **YEARS OF EXPERIENCE**

We consider seven slots for experience.

|  |  |
| --- | --- |
| **YEARS OF EXPERIENCE** | **RISK VALUE** |
| 1 | 0.7 |
| 2 | 0.6 |
| 3 | 0.5 |
| 4 | 0.4 |
| 5 |  |
| 5-10 | 0.3 |
| 10-15 | 0.2 |
| 15+ | 0.1 |

**Table 4.8.1.1: Risk Values for Years Of Experience**

1. **DESIGNATION OR SECURITY LEVELS**

The users can be assigned into security levels based on their designation.

|  |  |
| --- | --- |
| **SECURITY LEVEL** | **RISK VALUE** |
| LEVEL 1 | 0.1 |
| LEVEL 2 | 0.2 |
| LEVEL 3 | 0.3 |
| LEVEL 4 | 0.4 |
| LEVEL 5 | 0.5 |
| LEVEL 6 | 0.6 |
| LEVEL 7 | 0.7 |

**Table 4.8.1.2: Risk Values for Security Levels**

For example,

1. CEO – Risk –0. 1
2. HR Team Lead– Risk – 0.2
3. HR Manager – Risk – 0.3
4. Development Engineer Team Lead – Risk –0. 4
5. Senior Dev. Engineer – Risk – 0.5
6. Graduate Trainee – Risk – 0.6
7. Receptionist – Risk –0. 7
8. **NUMBER OF FAULTY SESSIONS/NUMBER OF TOTAL SESSIONS**

Faulty session means a session where a user did something that led to termination of the session before the session expires. Like trying to access something that isn’t within his permissions etc. For example, if a user has the permission to only append to a file, but he tries to edit it, we cut the session off. We track the clicks for this.

Total sessions = n

Faulty sessions = m

Required value = m/n

Here, higher the value of this fraction the higher is the risk.

Eg: m=10 n = 18

Risk value = 10/18 = 0.5556

So, this fraction will increase if the user is making such moves. That is why this is one of the deciding factors. This value is always between 0 and 1.

1. **VOUCHING**

At the time of registration, if anybody vouches for the user, this value will be lie between 0.0 to 0.6. This is based on the security level as follows:

|  |  |
| --- | --- |
| **VOUCHER SECURITY LEVEL** | **RISK VALUE** |
| LEVEL 1 | 0.0 |
| LEVEL 2 | 0.1 |
| LEVEL 3 | 0.2 |
| LEVEL 4 | 0.3 |
| LEVEL 5 | 0.4 |
| LEVEL 6 | 0.5 |
| LEVEL 7 | 0.6 |

**Table 4.8.1.3: Risk Values for Vouching**

1. **LOCATION OF ACCESS**

If the access is from outside the organization and Virtual Private Networks, the trust is less and the risk is more. . The access should be within a predefined range of IPs. If any other IP is detected, this value will be 1. If not, its 0.

1. **TIME OF ACCESS**

If the access is beyond the standard work timing, say after logging in 10 hours of work, we assume that there is some risk. If the access is beyond the logged-in hours, this value is 1. If it is well within regular work hours, its 0.

1. **PRE-BIAS FACTOR**

The admin has an option to keep an eye on a user; he is allowed to be biased. If the admin is skeptical, he can set this flag as 1 for a user. If not, its 0.

**4.8.2 CALCULATION OF THRESHOLD RISK VALUE**

After all this data is entered, we calculate the Threshold Risk Value.

This equation derives reference from the paper:

**A Dynamic Trust Model Based on Naive Bayes Classifier for Ubiquitous Environments**

Weiwei Yuan, Donghai Guan, Sungyoung Lee\*, and Youngkoo Lee

Department of Computer Engineering, Kyung Hee University, Korea {weiwei, donghai, sylee}@oslab.khu.ac.kr, yklee@khu.ac.kr

We have modified it according to our needs.

Risk = (η \* Designation risk )/ (YOE risk \* T)

Where,

η = faulty session/total sessions

T = Window period for security risk analysis. Value of T will be in terms of months.

For example, if the risk analysis is done every two months, the value of T=2.

**4.8.2.1 CALCULATION OF MINIMUM TIME BASED RISK**

For calculating minimum risk,

η = 0 (no faulty sessions)

Designation risk = 0.1 (CEO)

YOE risk = 0.05 (15+ years experience)

T = 2

Therefore,

Risk = (0 \* 0.1)/ (0.05 \* 15 \* 2)

= 0

So minimum risk = 0.

**4.8.2.2 CALCULATION OF MAXIMUM TIME BASED RISK**

For calculating maximum risk,

η = 1 (all faulty sessions)

Designation risk = 0.7 (Receptionist)

YOE risk = 0.7 (1 year experience)

T = 2

Therefore,

Maximum risk = (1 \* 0.7)/ (0.7 \* 2)

= 0.5

Now, to calculate total final risk i.e. overall risk value,

**4.8.3 CALCULATION OF FINAL OVERALL RISK**

Overall risk = (VouchingRV + Location Of AccessRV + Time Of AccessRV + Time BasedRV ) / Total number of parameters

1. **MINIMUM OVERALL RISK:**

Vouching RV = Yes, by an employee of Level 1. So VouchingRV = 0.0

LOA =Inside organization, so RV = 0

TOA = Within work hours, So RV = 0

Pre biased = Not biased, so RV = 0

Time based risk = 0 (from previous calculations)

So, now:

Overall risk = (0 +0+0+0+0)/5

= 0

**Variation in Static Module:**

In static module, we do not consider Location of Access and Time of Access as parameters since the value is rigidly set. Ignoring those parameters, we have:

Overall Risk = (0+0+0)/3

=0

1. **FOR MAXIMUM OVERALL RISK:**

Vouching RV = No, so RV = 0.6

LOA =Outside organization, so RV = 1

TOA = Beyond work hours, So RV = 1

Pre biased = Biased, so RV = 1

Time based risk = 0.5 (from previous calculations)

So now,

Overall risk = (0.6+1+1+1+0.5)/5

= 4.1/5

=0.82

That is 82% risk even at worst case in dynamic and adaptive modules.

**Variation in Static Module:**

In static module, we do not consider Location of Access and Time of Access as parameters since the value is rigidly set. Ignoring those parameters, we have:

Overall Risk = (1+1+0.5)/3

=2.5/3

=0.83333

Thus, in static module, we have the maximum possible risk of 83.33%.

So basically, our risk parameters are built such that even worst case risk in dynamic and adaptive modules is 80% even if theoretically its 100%.

**4.8.4 CURRENT RISK VALUE**

To calculate Current Risk Value (i.e. the risk based on permissions the user is asking for), we follow the following rules.

Now, our minimum Threshold Risk Value is 0 and maximum is 0.82.

So, the combination of these permissions shouldn’t cross 0.82.

Creating a fileRV = 0.0

Uploading a fileRV = 0.0

Reading a fileRV = 0.15

Appending a fileRV= 0.20

Editing a fileRV= 0.25

Printing a fileRV = 0.30

Downloading a fileRV = 0.50

We provide every user with the basic permissions of creating a file and uploading a file. So even if we give printing and downloading as the permissions, the risk is 0.80 which is still 0.2 units lesser than the maximum overall risk. We have assigned downloading permission the maximum risk because there, the user has an option to overwrite, change the complete content and also delete content and make it an empty file. Any combination of two permissions won’t cross 0.80.

**4.8.5 IMPLEMENTATION VARIATIONS ACROSS STATIC, DYNAMIC AND ADAPTIVE MODULES**

**4.8.5.1 STATIC MODULE**

1. We already know the Threshold Risk Value. So, we give two permissions that won’t cross the Threshold Risk Value. The users can select two permissions such that their Current Risk Value won’t cross Threshold Risk Value. We can check if even one single permission will still be more than the assigned Threshold Risk Value with an ‘if’ condition.
2. We assign the permissions, and generate a session key.
3. Once the session key is generated the text editor opens with either a new file or old file with granted permissions. This signals the beginning of a new session.
4. At the end of fifty minutes, we inform the user that the session will end in ten minutes.
5. At the last three minutes, we display one more warning.
6. The session is terminated at the 59th minute, 59th second.
7. If the user wants a new session, he has to start from step 1.

**4.8.5.2 DYNAMIC MODULE**

1. We calculate the Threshold Risk Value just after the user logs in. Even if the Threshold Risk Value was calculated one session ago, we calculate it again because the ratio of faulty sessions/total sessions would have changed. After this step, we check the Current Risk Value and decide on whether to assign the permissions or not.

2. After this step, the same steps of static from STEP 2 will continue.

**4.8.5.3 ADAPTIVE MODULE**

Adaptive module involves an extra parameter of monitoring the important actions of the user via the clicks.

1. We calculate Threshold Risk Value every 10 minutes based on click parameters.

For example, a user is allowed the following in ten minutes.

1. Read a maximum of three files.
2. Write/Edit/Append a maximum of three files.
3. Print a maximum of two files

Etc.

We keep a tab on this via clicks counter.

When we check at every ten minute interval, if there is any discrepancy, we terminate the connection.

**4.8.6 ADMIN CONSOLE**

Admin console involves the monitoring of all these parameters, users and sessions. The admin has the privilege to make edits to the user data, user risk parameters, to terminate any session showcasing unusual behavior and to disable accounts too. The admin has the complete control over the module. The console consists of the parameters and privileges required for efficient management.

**4.9 ALGORITHM**

**4.9.1 CLIENT SIDE**

1. On the server side,

MySQL has to be running.

Step 1: User logs in with the credentials.

Step 2: On successful authentication, user is allowed access to the Main Menu. If not, access is denied and the same message is displayed.

Step 3: On successful login, Client Console is displayed. Two options named ‘User’s Applications’ and ‘Help’ are visible on the tab bar.

Step 4: If ‘User’s Applications’ is selected, the following options are available.

User’s Applications

* List Files
* Fetch Files
* Upload Files
* Editor Application
* Logout

Step 5: If the user chooses the option ‘List Files’, a list of files available on the server is displayed.

Step 6: ‘Fetch Files’ option allows the user to input a file’s name from the list of files and fetch it from the server.

Step 7: The ‘Upload Files’ is to upload a file on the server.

Step 8: ‘Editor Application’ opens the text editor application, CDocViewer. Here, the user can perform the following operations:

Under the File tab we have:

* + New
  + Open
  + Save
  + Save As
  + Page Setup
  + Print
  + Exit

Under the Edit tab we have:

* + Undo
  + Redo
  + Select All
  + Copy
  + Paste
  + Cut

Step 9: The ‘Logout’ option allows a legal and secure log out.

Step 10: If ‘Help’ option is chosen, it displays:

About – This gives information about the module.

Contact – email IDs of the developers

**4.9.2 ADMIN SIDE**

Step 1: Admin logs in after successful authentication. If authentication fails, access is denied and the same message is displayed.

Step 2: On successful authentication, Admin Console is displayed with two options in the tab bar, ‘Users’ and ‘Help’.

Step 3: Choosing ‘Users’ gives the following options:

User Registration and Updation

Change Password

Account Status

Disable/Enable Account

Step 4: Selecting the option ‘New User Registration’ allows the admin to register a new user with the parameters required being:

Name

User Name

Password

Email ID

Employee ID

Filling these details and clicking on the ‘Save’ option available on the pop up window stores the data into the database.

Step 5: The option ‘Change Password’ allows changing of password on provision of the following parameters:

User Name

EmployeeID

New Password

Confirm Password

Step 6: The ‘Account Status’ option displays the current status of all the users.

Step 7: Selecting the option ‘Disable/Enable Account’ allows the user to change account status of any user with the input being User Name.

Step 8: If ‘Help’ option is chosen, it displays:

About – This gives information about the module.

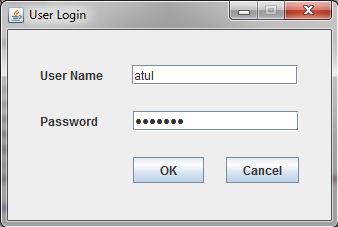
Contact – email IDs of the developers

**CHAPTER 5**

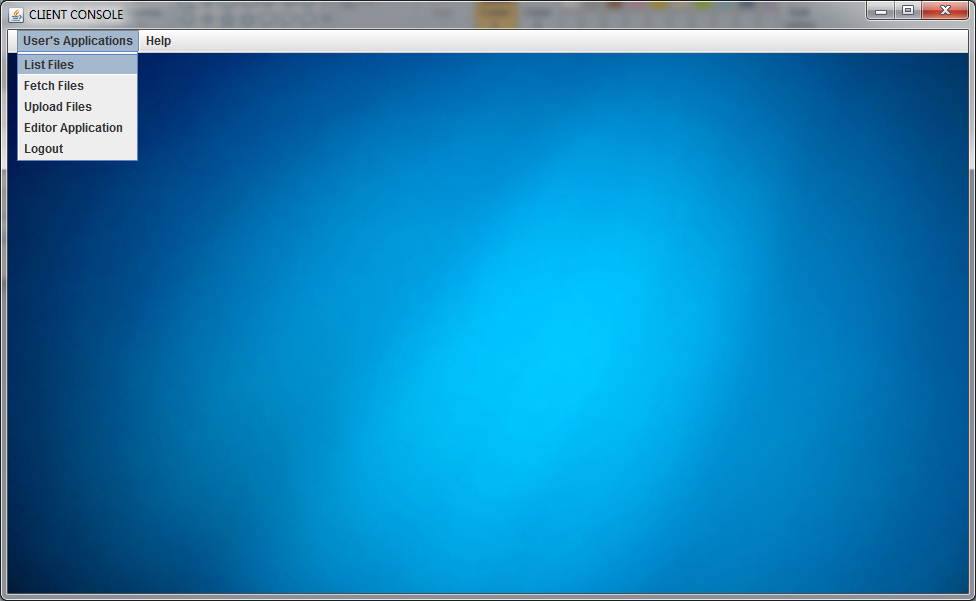
**RESULTS**

Our module, on implementation, looks as shown here.

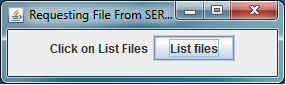
**5.1 CLIENT CONSOLE**



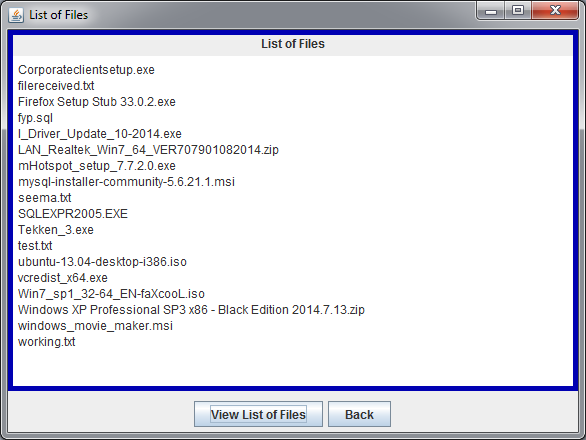
**Fig 5.1.1: User Login**



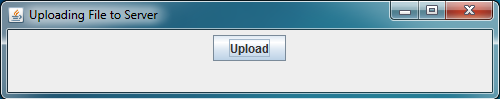
**Fig 5.1.2 : ‘List Files’ option**



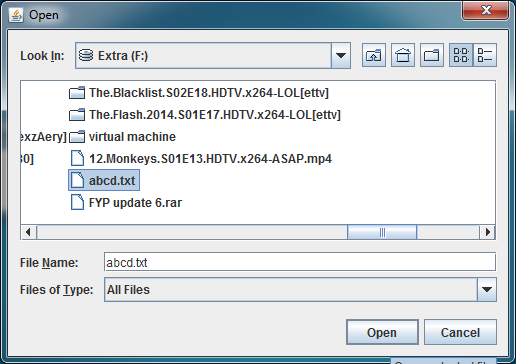
**Fig 5.1.3 : Listing of the files**



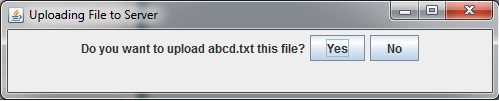
**Fig 5.1.4: List of Files**



**Fig 5.1.5: ‘Upload’ option**



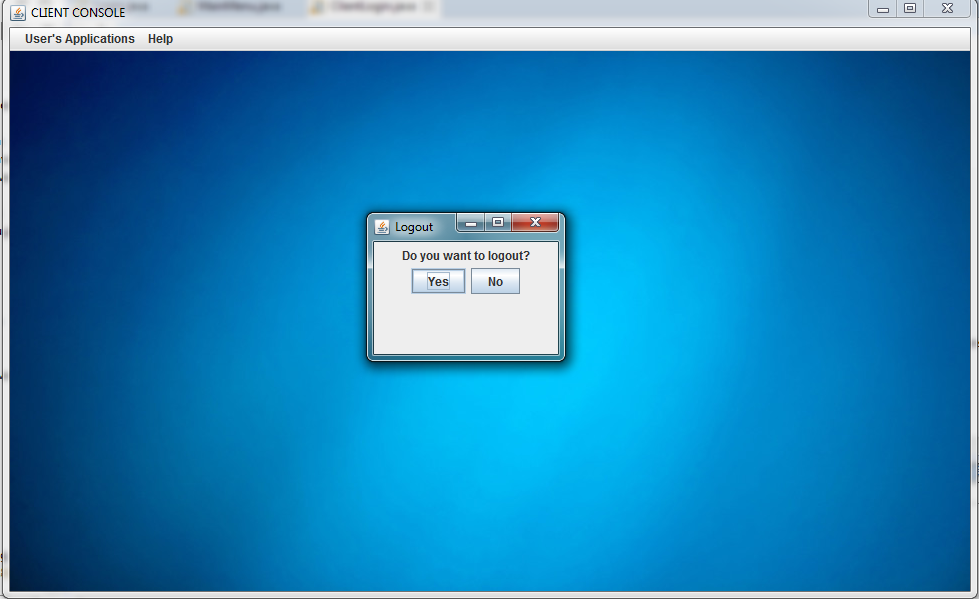
**Fig 5.1.6: Selection of file to upload**



**Fig 5.1.7: Confirmation of uploading of file**

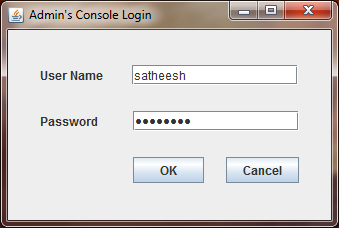


**Fig 5.1.8: CDocViewer**

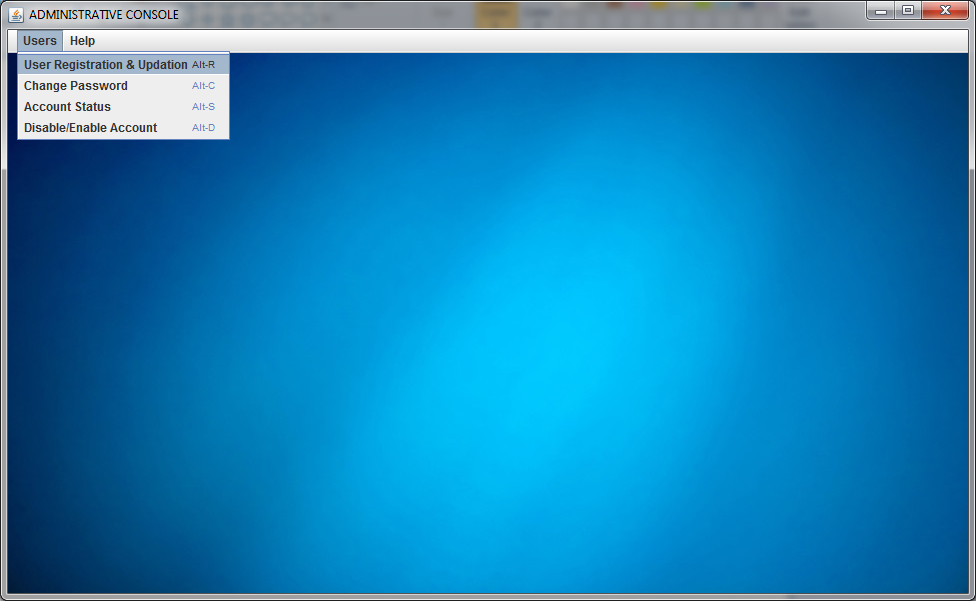


**Fig 5.1.9: Logout**

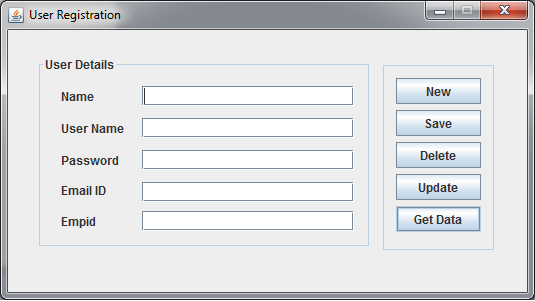
**5.2 ADMIN CONSOLE**

****

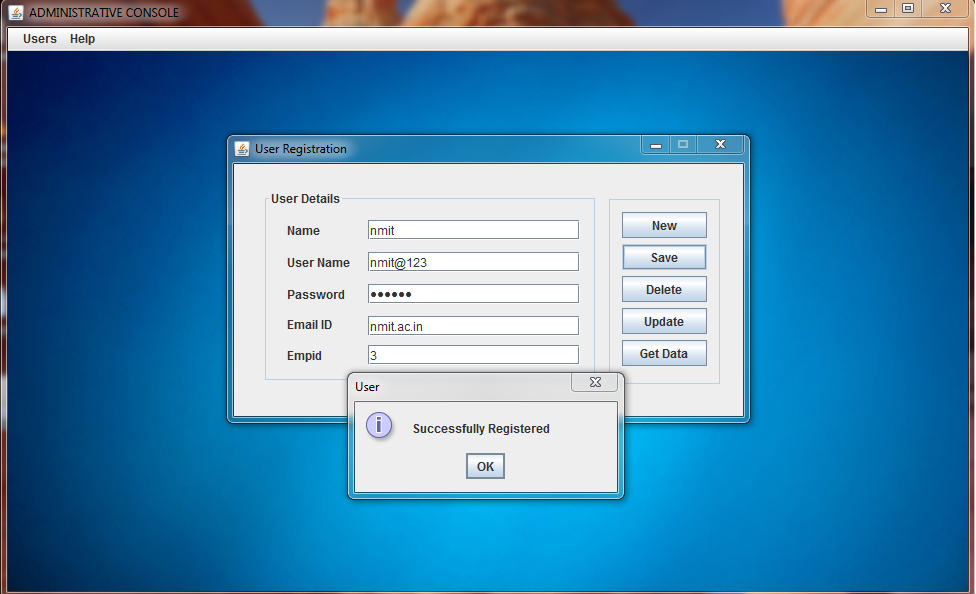
**Fig 5.2.1: Admin Console**



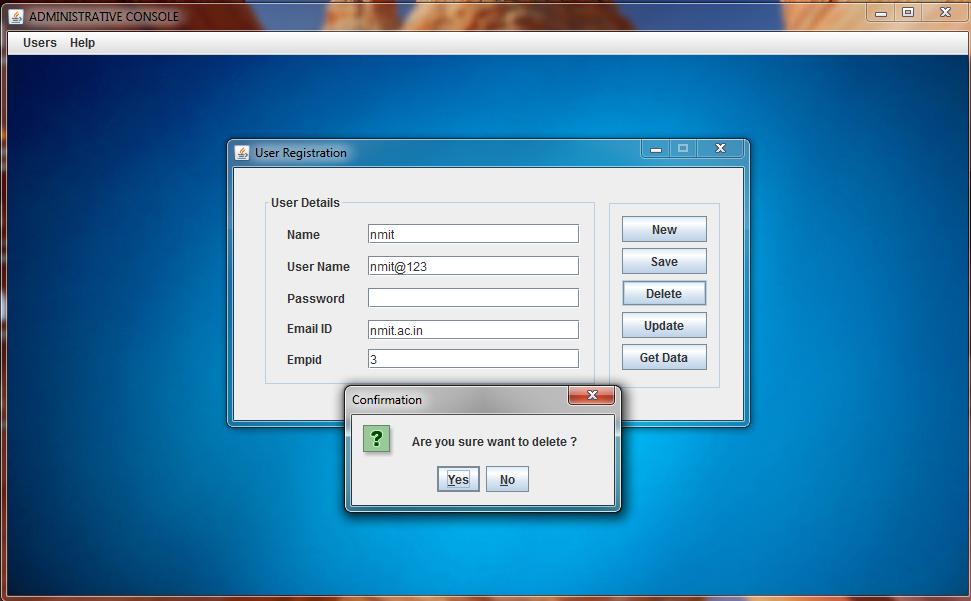
**Fig 5.2.2 : User Registration & Updation**



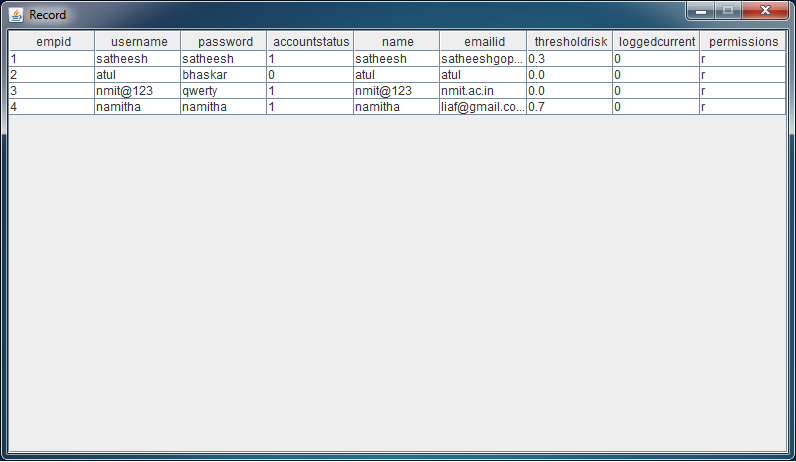
**Fig 5.2.3 : User Registration**



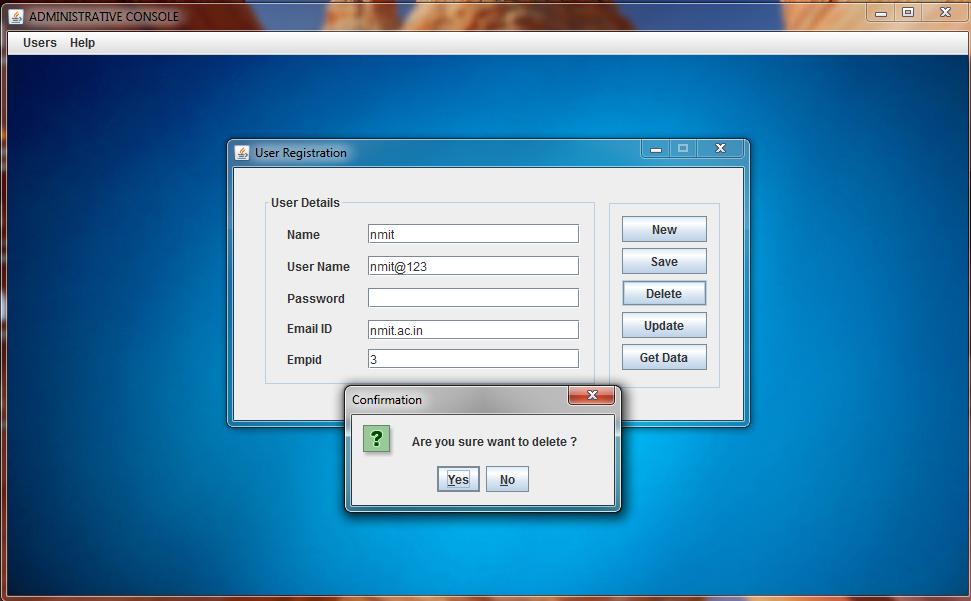
**Fig 5.2.4 : Successful registration**



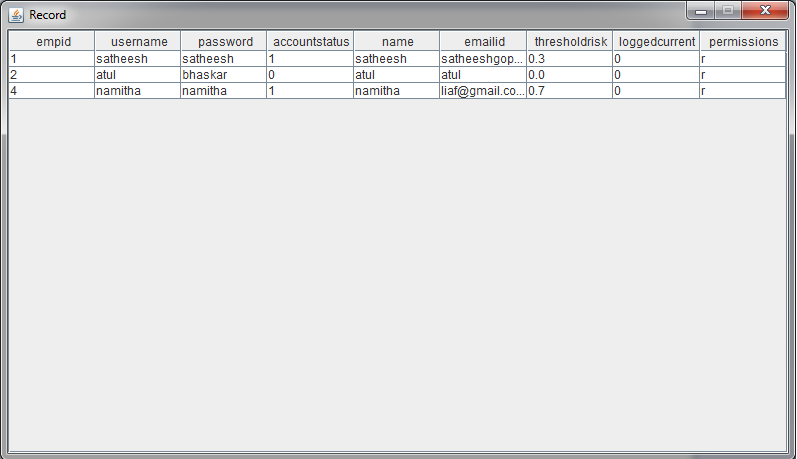
**Fig 5.2.5: Deletion**



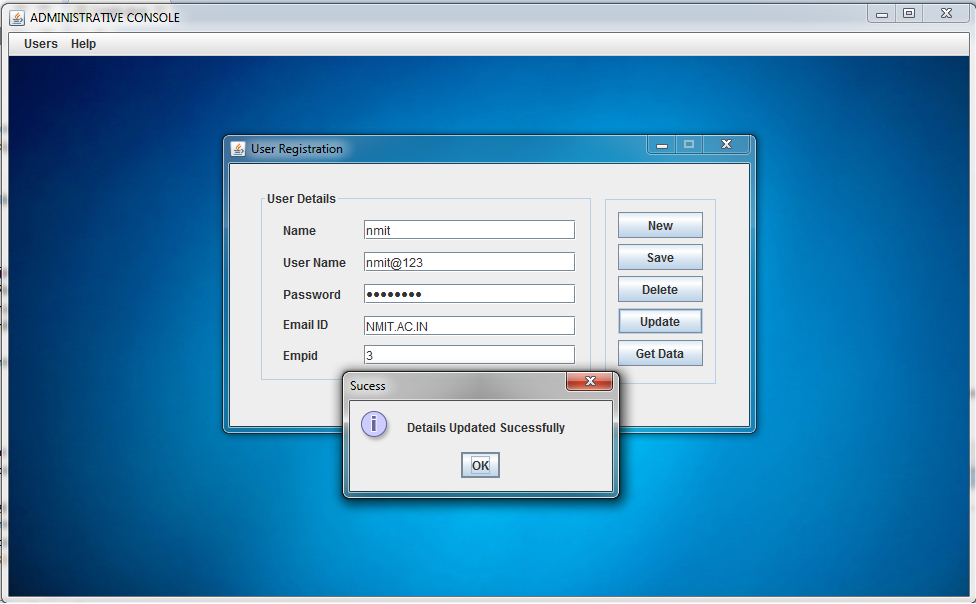
**Fig 5.2.6: Database Updation**



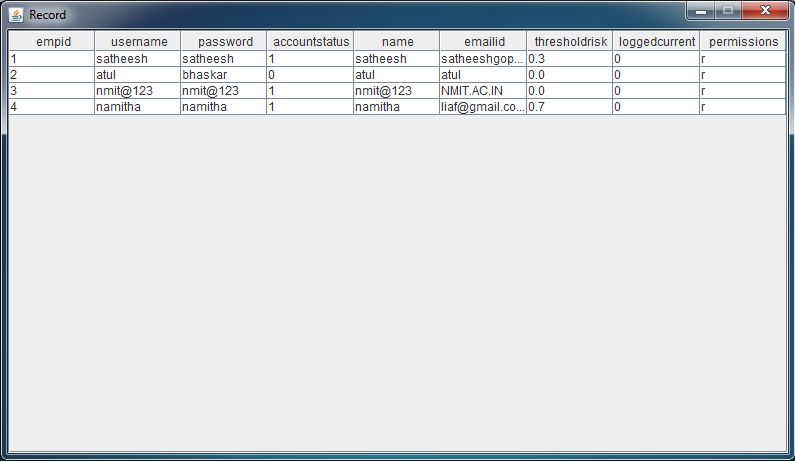
**Fig 5.2.7: Database Updation-Confirmation of Deletion**



**Fig 5.2.8: Database updation**



**Fig 5.2.9: Database Updation - deletion**



**Fig 5.2.10: Updated Database**

**CHAPTER 6**

**CONCLUSION AND FUTURE ENHANCEMENT**

**6.1 CONCLUSION**

Our module has shown successful access control in the three variations of it. The effectiveness of the module lies in the fact that the maximum risk is still at 82% and we believe that proves to be extremely useful in battling insider attacks. The module considers even the minutest possible risk and calculates the required computations on simple equations. This aids in speed optimization and lesser memory consumption. Also, the risk parameters are extremely flexible and can be modified as per organizational needs.

**6.2 FUTURE ENHANCEMENT**

1. We have implemented only the static module. The dynamic and adaptive modules require slight modification and enhancement in the program code since the static code is the foundation for working of the module.
2. Our module involves granting of four permissions only. Future work can be done on increasing the number of permissions.
3. The type of file is text only in our module. Future enhancements can allow various kinds of files for access.

**CHAPTER 7**

**REFERENCES**

1. **A Dynamic Trust Model Based on Naive Bayes Classifier for Ubiquitous Environments**

Weiwei Yuan, Donghai Guan, Sungyoung Lee\*, and Youngkoo Lee

Department of Computer Engineering, Kyung Hee University, Korea {weiwei, donghai, sylee}@oslab.khu.ac.kr, yklee@khu.ac.kr

1. **A Trust-and-Risk Aware RBAC Framework: Tackling Insider Threat**

Nathalie Baracaldo University of Pittsburgh School of Information Sciences nab62@pitt.edu

James Joshi University of Pittsburgh School of Information Sciences jjoshi@sis.pitt.edu

1. **Risk-Aware RBAC Sessions**

Khalid Zaman Bijon, Ram Krishnan, and Ravi Sandhu

Institute for Cyber Security & Department of Computer Science, Institute for Cyber Security & Department of Electrical and Computer Engineering University of Texas at San Antonio

1. www.github.com
2. www.stackoverflow.com
3. www.rgagnon.com/javadetails.html
4. www.docs.oracle.com/javasetutorials.html