Investigation of Security Aspect of Cloud Computing

Sahar Mahdie Klim

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
January 2014
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Elvan Yılmaz
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Işık Aybay
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Asst. Prof. Dr. Gürcü Öz
Supervisor

Examining Committee

1. Prof. Dr. Erden Başar ______________________

2. Asst.Prof. Dr. Gürcü Öz ______________________

3. Asst. Prof. Dr. Ahmet Ünveren ______________________

ABSTRACT

Cloud Computing is an evolving computing pattern in which resources of the computing infrastructure are provided as services over the Internet. This pattern also fetches forth many new challenges for data security and access control when users outsource sensitive data for sharing on cloud servers, which are not within the same trusted domain as data owners. To keep sensitive user data confidential against untrusted servers, existing solutions usually apply more security modes.

In this study, we focused on the security aspect of the cloud computing. We proposed a unique extremely of decentralization information accountability context to remain follow from particular usage of the clients' information within the cloud; it aims to ensure more secure usage of data by providing supplementary security mode in the application of system. So we concentrated on three security modes: Accountability, Auditing Mechanism and Automatic Session Expire Technic for the web application. In our application there are three parties, namely: Admin, Owner and User. We reinforce data security over cloud through the mentioned security modes.

In our system only legally registered clients are eligible to access to stored data. Moreover, an advanced auditing mechanism is implemented, in which the owner can monitor the data while the admin can observe and control all the system. Furthermore, we included expiry session feature to improve the application security.

Above all, this study is suggesting a degree of object-related approach which allows inclusion of our registration and logging procedure together with users' information and policy. We also leveraged the Java Archive Rar (JAR) programmable capabilities to both create a dynamic and traveling object, and to ensure that any access to users' data will trigger authentication and automated logging local to the JARs. And as mentioned above to strengthen user's control, we provided distributed auditing mechanisms. Finally we provided extensive experimental studies that demonstrate the efficiency and effectiveness of the proposed approaches.

Keywords: Cloud computing, security, accountability, Java Archive Rar (JAR)

ÖZ

Bulut bilişim (cloud computing) alt yapı kaynakları Internet üzerinden sağlayan ve gelişmekte olan bir bilişim modelidir. Bu modelde kullanıcılar ve veri sahipleri aynı güvenli alan içerisinde olmayabildiklerinden, kullanıcılar hassas verilerini bu sistem üzerinde paylaşırken veri güvenliği ve erişim kontrolü ile ilgili farklı sorunlar yaşamaktadırlar. Hassas verileri güvenilir olmayan sunucularda korumak için mevcut çözümler daha güvenilir bir şekilde uygulanmaktadır. Bu çalışmada, bulut bilişim sistemlerinin güvenlik yönleri üzerinde durulmuştur. Önerilen ek güvenlik yöntemleriyle kullanıcıların verilerinin sistemde daha güvenli bir şekilde tutulması ve kullanımlarının sağlanması hedeflenmiştir.

Bu amaç için sistem yöneticisi, veri sahibi ve kullanıcılar için üç farklı güvenlik yöntemi üzerinde durulmuştur. Sisteme kayıt yaptırarak girebilme, sisteme girişlerin denetlenmesi ve sistemde tutulan verilere kullanım süresinin verilmesi yöntemleriyle bulut bilişim sistemlerinin güvenliği artırılmaya çalışılmıştır. Oluşturulan sistemde, sadece yasal olarak kayıt yaptıran kullanıcıların verilere giriş izni vardır. Aynı zamanda, oluşturulan denetim mekanizması ile, veri sahibi girişleri gözlemleyebilirken, sistem yöneticisi de tüm sistemi gözlemleyip kontrol edebilmektedir. Bunlara ek olarak, sistemde saklanan dosyalara geçerlilik süresi verilerek dosyaların sistemde saklanma süreleri kısıtlanmış ve verilerin güvenliği artırılmıştır.

Bütün bunların yanında, bu çalışmada kullanıcı kaydı ve girişi işlemleri için nesneye dayalı yöntemler kullanılmaktadır. Aynı zamanda, Java Arşiv Rar (JAR) programlama yöntemleri kullanılarak dinamik ve taşınabilir nesneler yarıtılmış kullanıcı verilerine kimlikle erişim sağlamış ve kullanıcıların girişleri kontrol altına alınmıştır. Çalışmanın sonunda oluşturulan sistem üzerinde bazı deneyler yapılarak sistemin etkinliği, önerilen yöntemlerin etkisi gözlenmiştir.

Anahtar kelimeler: Bulut bilişim, güvenlik, denetimlilik, Java Arşiv Rar (JAR)

To my parents who taught me the value of scholarship and who offered me endless love and care

To my lovely brothers and sisters with whom I shared unforgettable moments

To my friends who encouraged and supported me

ACKNOWLEDGMENT

First of all I would like to thank to our most beloved ALLAH, for supporting me in my master study and in this research particularly.

I express my deepest gratitude and respect to my supervisor Asst. Prof. Dr. Gürcü Öz for her guidance and support throughout my research.

I am also thankful to all my teachers for their encouragement to me and to my colleagues to gain knowledge.

I am in debit to my line manager in Misan University, in Iraq, Dr. Eng. Adel A. SH. Al-Kareemawi who supported me very much to pursue my high education. Thanks to his support I am standing where I am now.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

API	Application Programming Interface
CIA	Cloud Information Accountability
CSP	Cloud Service Providers
DB	Database
GAE	Google Application Engine
HTML	Hypertext Markup Language
IaaS	Infrastructure as a Service
IBE	Identity-Based Encryption
ISP	Internet Service Provider
JAR	Java Archive Rar
JDBC	Java Database Connectivity
JRE	Java Runtime Environoment
JSE	JavaStandard Edition
JSP	Java Script Programming
JVM	Java Virtual Machine
MySQL	My Structured Query Language
OOPP	Object Oriented Programming Paradigm
PaaS	Platform as a Service
SaaS	Software as a Service
SDO	Self-Defending Object
URL	Uniform Resource Locator

Chapter 1

INTRODUCTION

The cloud computing is new technology widely studied in recent years, that comes from network computing, distributed computing parallel computing, virtulization technology, computing utilities, and various computer technologies with its additional characters such as large scale computation of data storage, virtualization, high expansibility, high dependability and low cost service (Liu, 2012). Also cloud computing could be a combination of all resources to enable the resource sharing in terms of scalable infrastructures, middleware and application development platforms, and value-added business applications (Lo, Huang, & Ku, 2010). The main purpose of computing in the cloud is to create a stronger use of well-distributed resources, mix them to realize higher output and which are ready to be used for solving massive scale computation issues (Jadeja & Modi, 2012). It has a lot of resources and private information and therefore they are easily threatened by attackers. Hence it must be protected against both inside and outside (Lee, Park, Eom, & Chung, 2011).

The easiness and durability of this approach however, are also come with security threats and privacy. A major limitation that allows the use of cloud services is the doubt of the user resulting from confidential data leakage and loss of privacy in the cloud (Pearson & Charlesworth, Accountability as a Way Forward for Privacy Protection in the Cloud, 2009).

This chapter is devoted to illustrating a general introduction to cloud computing system, security in cloud computing, computing in the cloud components, cloud classification, cloud architecture, literature review and lastly the aim of this research work.

1.1 Cloud Computing

In the literature some efforts have been done by researchers to define the "Cloud Computing". In what follows, we mention some of the most important definitions of this new intervention in the field of information technology.

Cloud computing denotes the distributing of computing methods that occur through utilizing of high speed network. Data processing is moved from private PCs or servers to the remote computer clusters (big data centers in hand by the Cloud Server Provider (CSP)). Any user who possibly has a super computer at hand may access the information and obtain the computing capability at any time, from any place, even when you merely want to buy the resources that you have used irrespective of who offer the resources and in what way they do it (Zhang & Zhang, 2009). Cloud computing denotes the applications delivered as services through the internet with the use of hardware and software of the systems in the data centers that provide those services (Armbrust, et al., 2009). Cloud is a huge grouping of simply usable and retrievable virtualized resources (like hardware, development platforms and/or services).

These resources may be transferable reconfigured to regulate the variable load (scale), allowing also for a balanced resource allocation (Patidar, Rane, & Jain, 2012). Figure 1.1 shows a cloud computing example with its three layers which are explained in details in section 1.3.

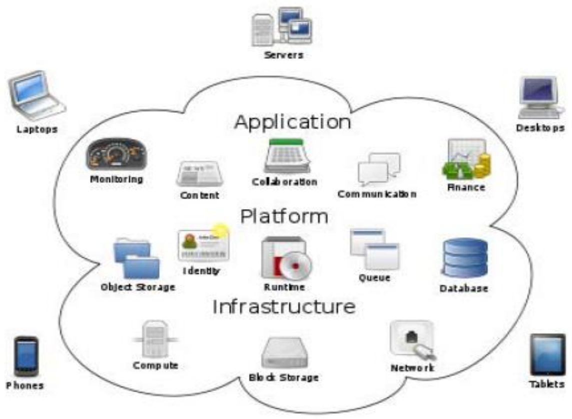


Figure 1.1. Cloud Computing (Patidar, Rane, & Jain, 2012) (Sharma, Soni, & Sengar, 2012)

Advantages of the cloud computing lie in that it covers on-requested self-service, ubiquitous of access to the network, location separate selection of resource, flexibility of rapid resources, usage-based on cost and risk transfer among others. Due to its great flexibility and low cost, costumers prefer to turn their local complex data management system into the cloud (Cao, 2012). It also keeps beneficiaries' data confidential. Below, we simply illustrate the benefits of cloud computing in more details (Donkena & Gannamani, 2012):

i. ***Scalability*****:** If company come to know that there's an increase in demand of resources, then cloud computing will do a great help. Instead of getting new equipment which are usually installed or put together, company can buy extra

CPU cycles or storage from a third party to enhance such purpose and such reducing the cost. Once they need to meet their desires for extra equipment, they can stop the use of cloud provider’s services and hence they don’t have to handle equipment unnecessarily (Ali & Ayub, 2012)**.**

ii. ***Simplicity*****:** By not buying new equipment and configuring them allows Information Technology (IT) staff to get into the business. The cloud makes it possible to start applications immediately and the cost is very less if the company would have to find an on-site solution (Ali & Ayub, 2012).

iii. ***More internal resources*****:** By shifting non-critical data needs to the use of cloud computing, companies hence permit their IT department with more emphasis on business where they do not need to rent or manage more (Ali & Ayub, 2012).

iv. ***Security*****:** Vendors have strict policies for ensuring security. They have proved cryptographic ways to authenticate users. Additionally, they'll permanently cypher their data before storing it on the cloud. By these measures their data is safer on cloud than in-house (Ali & Ayub, 2012).

Disadvantages of the cloud computing originate in its complex and insecure use. To guarantee the reliability of this technology, it is crucial to supply the needed robust security, privacy. Without such reliability, the costumers are unlikely to entrust management of their data to servers in the cloud. To promote the adoption of cloud information outsourcing, in this research work, possibilities of a more secure and trusted information outsourcing is explored.

We aim to employ the most necessary information services that include reliable information management systems, especially with the use of upper level service performance and durability.

1.2 Cloud Computing Security

Security is a key to cloud computing success. Several surveys currently show security about cloud computing to be the most important challenge of this field. Till some years past, every business methods of organizations were based on their personal infrastructure. In what way it might be easy to outsource services, it sometimes of non-critical data/applications or non-public infrastructures of computing in the cloud but with a different modified story presently. The usual perimeter of the network is broken and organizations feel they have to lose management of their information (Jacobo, 2012). Security of the system for database and server uses the service or application of cloud computing, which is the first specified required condition which every reliable one on the server and database should satisfies. After then, associate enterprise is favored to use the associated service of computing in the cloud that is provided by the server part. Hence actualizing the aim of lowering the expected required budget and price of storage associated with the manipulation and demand for either corporate or individual uses. Every private which depend on computing services in the cloud may store their information within the storage that is provided by the Internet Service Provider (ISP) on the remote part through the net and use the computing service to obviously reduce the cost. That is why what comes through the assurance of confidentiality, authentication and integrity is extremely necessary for those data transactions, data manipulation and service provided by computing in the cloud on the remote part through networking (Tsai, Lin, Chang, & Chen, 2010).

1.3 Structure and Components of Clouds

The idea obviously used to illustrate a collective structure and components of clouds is a 3-layered idea which is: (Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS) (Patidar, Rane, & Jain, 2012):

1.3.1 Infrastructure as a Service (IaaS)

This allows client to supply processing power, random memory of access, disk storage and network capabilities et cetera. The user may use the specify materials so as to develop, deploy and run arbitrary software using the provisioned computational resources (Paladi, 2012). Infrastructure-as-a-Service (IaaS) layer is responsible for providing on-demand virtual infrastructures to third-parties use of physical resources like memory, storage and processors. This virtual infrastructure usually allocates resources from data centers in-hand, managed by the cloud supplier and then employed by customers through the net (Salah, Alcaraz-Calero, Zeadally, Almulla, & Alzaabi, 2011)

1.3.2 Platform as a Service (PaaS)

This makes available a platform of computing by using infrastructure of the cloud since it possesses all the applications specifically needed by the customer deployed on it. Therefore the consumer do not have to be compelled to go through the hassle of buying and putting in place the hardware and software necessary for it. With this process, service developers could have every of the systems and environments needed for the life cycle of software, either the developing, testing, deploying or hosting of internet applications. Key examples are Google Application Engine (GAE) and Microsoft's Azure (Jadeja & Modi, 2012). The platform layer includes software. For instance, it includes all of the APIs for a specific programming language or virtualized operating system (OS) of a server (Zargar, Takabi, & Joshi, 2011).

1.3.3 Software as a Service (SaaS)

The consumer is provided with the capability to use provider’s application running on a cloud infrastructure. The client does not manage cloud infrastructure like servers, operating system, storage and network. The services are typically accessed with a web browser (Ali & Ayub, 2012).

.Figure 1.2 shows us those three layers which we covered above:

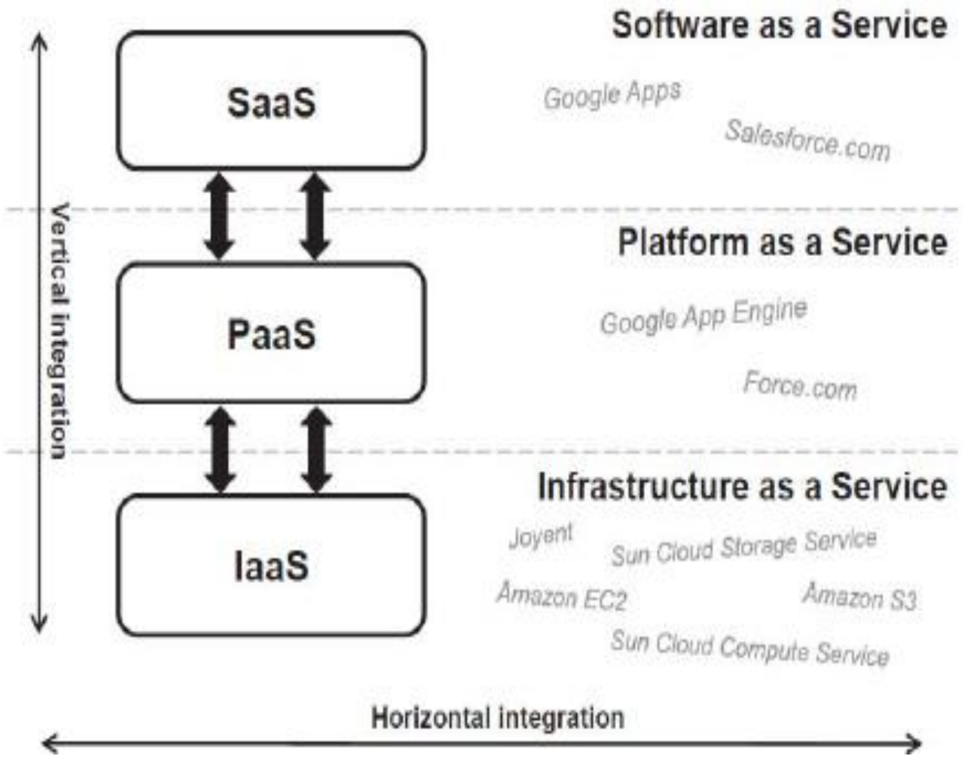


Figure 1. 2. The Three Layers of Cloud Computing SaaS, PaaS and Iaas (Patidar, Rane, & Jain, 2012)

1.4 Classification of Clouds

Generally cloud could be classified base on who the owner of the cloud, where the data centers are. Cloud environment always contains of either a single cloud or multiple clouds. Hence, differences can be established between single-cloud environments and multiple-cloud environments. The succeeding subsections gives appropriate categorization of single-cloud environments in respect to the cloud data center ownership and a categorization of multiple-cloud environments based on which type of clouds is combined as shown in Figure 1.3 (Patidar, Rane, & Jain, 2012).

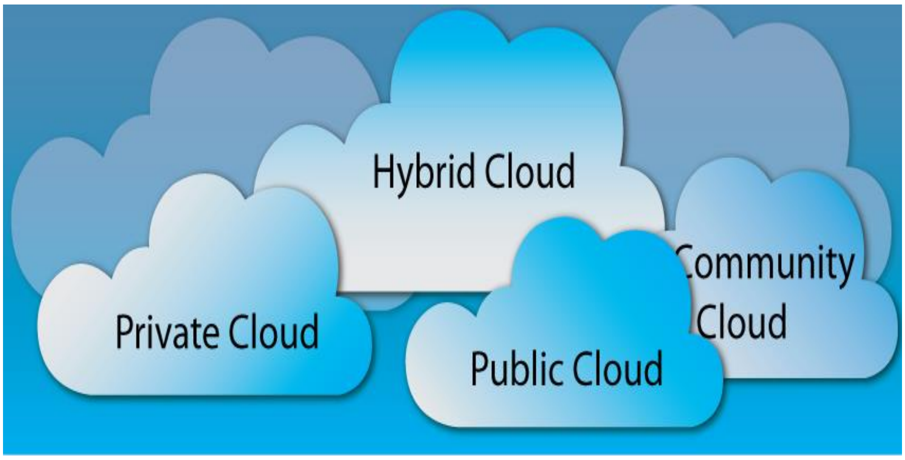


Figure 1.3. Classification of Cloud (Nussbaum, Cloud Deployment Models, 2012)

1.4.1 Private Clouds

Just one organization runs within the cloud. It can be regulated by the organization or outsourced to a 3rd party (Jacobo, 2012). Private cloud is referred to internal datacenters of a business or different organization, not created available to the overall public (Armbrust, et al., 2009).

Private cloud is also called as internal cloud or corporate cloud. Private cloud is providing resources, storage of data to a limited number of hosted services. This cloud may be managed and operated by the organization behind a firewall. Private cloud can access who is positioned within the boundaries of an organization (Donkena & Gannamani, 2012), as shown in Figure 1.4.

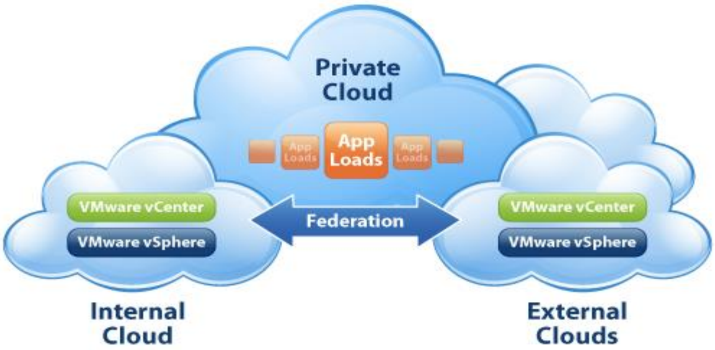


Figure 1.4. Private Clouds (Hexistor, 2012)

1.4.2 Public Clouds

Clouds are assigned publicly to any foundation. It is handled by an organization selling the service (Jacobo, 2012). In public clouds, infrastructure and services are both rendered at distance over the Internet. These clouds supply the best level of efficiency in common resources; however, they are less secure and however vulnerable than private clouds (Sharma, Soni, & Sengar, 2012). The cloud infrastructure is made available to the general public or a large group of industries and is owned by an organization selling the cloud services (Mell & Timothy Grance, 2011), as shown in Figure 1.5, also we want to mention that our system implementation used this type of cloud.

Figure 1.5. Public Clouds (Cotney, 2012)

1.4.3 Community Clouds

Clouds are distributed among multiple foundations with similar aims. It may be managed by the foundation or outsourced to a third party (Jacobo, 2012). The cloud infrastructure is shared among a number of organizations with similar requirements and interests. It can be in-house (outsourced community cloud) or with a third party (outsourced community cloud) on the premises (Ali & Ayub, 2012). The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security needs, policy, and compliance considerations). It may be controlled by the foundation or a third party and may exist on premise or off premise (Mell & Timothy Grance, 2011), as shown in Figure 1.6.

Figure 1.6.Community Clouds (Nussbaum, Dissecting the Cloud IV – Community Clouds, 2012)

1.4.4 Hybrid Clouds

Hybrid cloud computing is a platform which interprets between private cloud and public cloud. It is publishing by foundation, which do not want to put everything in the external cloud (public cloud) while hosting some servers in their own internal cloud infrastructure. The cloud providers are able to process applications which can work seamlessly between those boundaries (MOLLET, 2011)**.** Many cloud infrastructures with various deployment models are combined (Jacobo, 2012). This type of cloud infrastructure is a composition of two or more clouds i.e. private, community or public (Saleem, 2011), as in Figure 1.7. And Figure 1.8 show us cloud computing features.

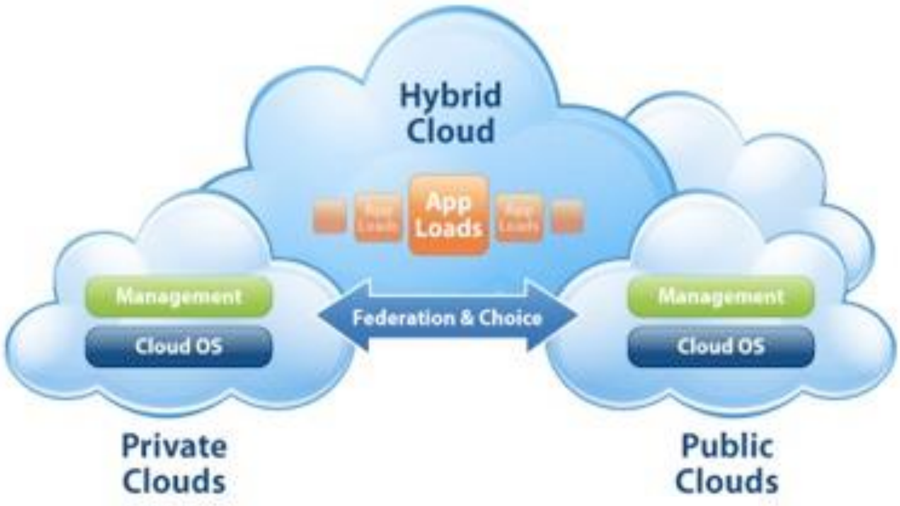


Figure 1.7. Hybrid Clouds (MOLLET, 2011)

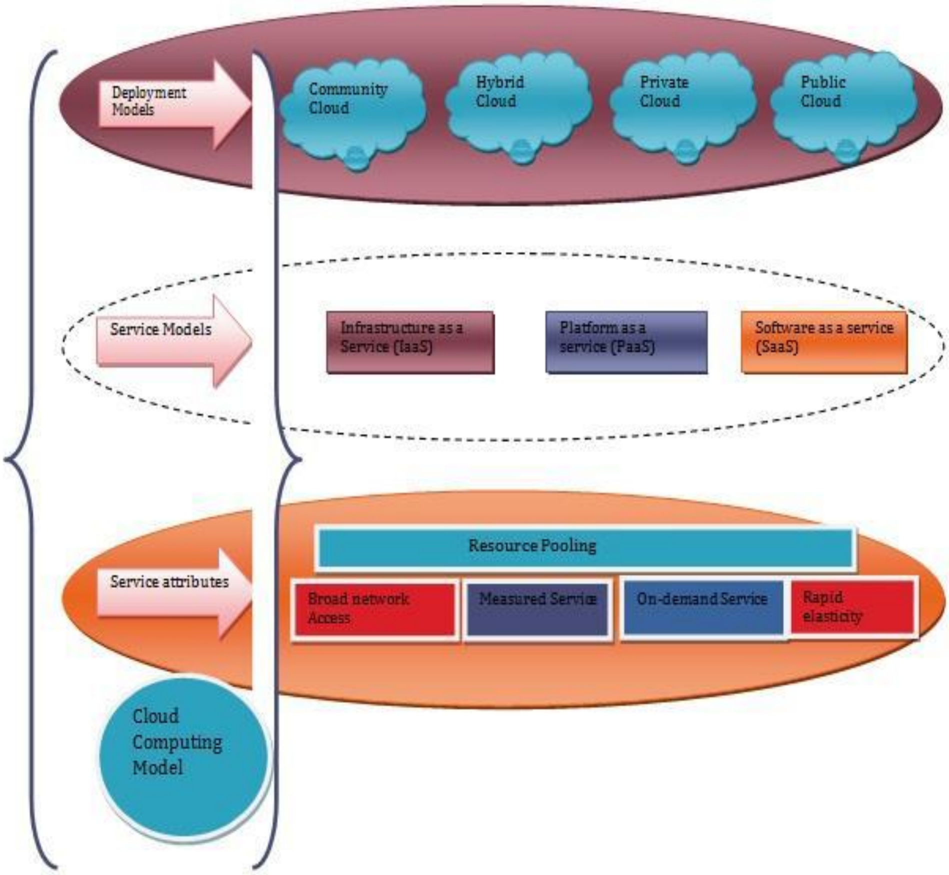


Figure 1.8. Cloud Computing and its Features (Donkena & Gannamani, 2012)

1.5 Cloud Architecture

Cloud architecture is the design of the software package systems to be shared within the delivery of cloud computing which generally involves multiple cloud elements communicating with each other over a loose coupling mechanism like a messaging queue (Mrs.S.Selvarani & Dr.G.Sudha Sadhasivam, 2010). Elastic provision implies intelligence in the use of tight or loose coupling as applied to mechanisms which is shown in Figure 1.9 and Figure 1.10.

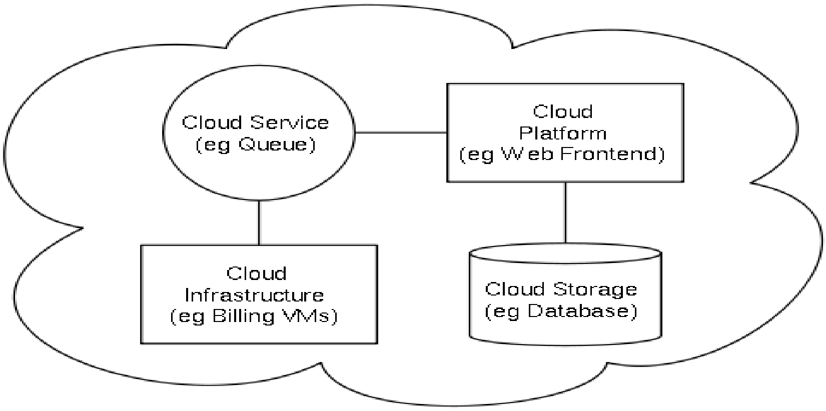


Figure 1.9. Cloud Computing Architecture (Selvarani & Sadhasivam, 2010)

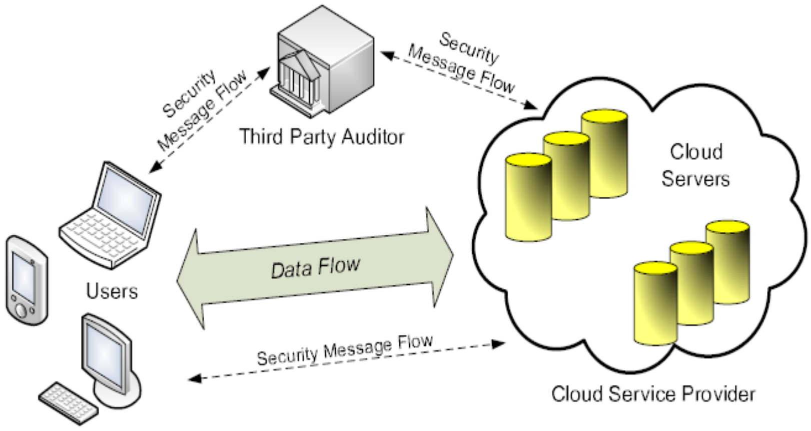


Figure 1.10. The Architecture of Cloud Data Storage Service
(Wang C. , Wang, Ren, & Lou, 2010) (Wang Q. , Wang, Li, Ren, & Lou, 2009)

1.6 Literature Review

There are many studies that have been done in the area of cloud computing security. In this section, we briefly mention some of these studies:

R. Corin et al. proposed a language that is responsible for agents to share information with usage policies in a decentralized architecture, and presented a logic data access and agent accountability in a setting in where data can be created, distributed, and re-distributed. The compliance with usage policies is not enforced. However, agents are also audited by an authority at a capricious moment in time, or the owner of the data attaches a usage policy to the data, which contains a logical speciation of what actions are allowed by the data and under which conditions it can be re-distributed. Also they designed a logic that allows auditing agents to prove their accountability is defined in many categories, including agent accountability and information accountability. This logic allows for a different kind of accountability and the soundness of the logic is seemingly demonstrated.

Peter Buneman et al. proposed data mapping algorithms to map provenance metadata. Coordinated databases associated with Bioinformatics and different disciplines are the results of an excellent deal of manual annotation, correction and transfer of data from different sources by developing net-based of (mostly purpose) database systems which offers some assistance in monitoring of the source especially when information moves through the databases.

The purpose and techniques generally used is investigated in this paper by recording available data that is copied between databases. Data made available through the creation, attribution and version history of such data is crucial for assessing both its scientific and the integrity value in the execution of the technique using to estimate the feasibility of database support for provenance administration. The empirical results confirm that the procedure can be followed efficiently and controlled using this suggested approach.

Giuseppe Ateniese et al. offered a model for Provable Data Possession (PDP) which enables a consumer that has stored his data at untrusted server to confirm that the server possesses the initial data without retrieving it and without having the server access the entire file. The design generates an assured probabilistic proofs of possession by sampling random sets of blocks from the server, which decreasing the I/O costs. The challenge and response protocol transmits a tiny low and constant quantity of data that reduces communication within the network. The PDP design for remote data checking supports of large data sets in widely-shared storage systems.

Paul T. Jaeger et al. focused in their paper on a range of policy aspects of cloud computing and (proprietary algorithms) development of cloud computing as a beneficial development for individual, corporate, and governmental computer users. This paper is intended to identify and encourage discussion about the policy issues related to cloud computing.

Radha Jagadeesan1 et al. noted an operational design and developed analysis mechanisms which they described analyses that support both the design of accountability systems and the validation of auditors for finitely accountability systems by using interpretation into games. Game-based has logics for multi agent systems with perfect data such as Alternating Temporal Logic (ATL), and coalition logics with parameters such as (a) the sincere partners (b) the guarantees supplied by network communication and (c) the precision requested of the audit protocol. The break down suggests a justified reason for the need to apply a more secure acknowledgement (ACKS) in enhancing the aims of these peer review. Their study provided formal foundations to explore the tradeoffs underlying the design of accountability systems including: the power of the auditor, the efficiency of the audit protocol, the requirements placed on the agents, and the requirements placed on the communication infrastructure.

Siani Pearson et al. proposed privacy manager in the cloud for enhance security in the cloud by using a commercial digital imaging website to share pictures online using Global Positioning System (GPS) which enabled cameras with translucent databases. Its performance metrics solution is not suitable for all cloud applications. With access to limited computing resources, there is a tradeoff between the extent, and they got existing cloud services that could be used in an obfuscated fashion (call it obfuscation rather than encryption because some of the information present in the original data is in general still present in the obfuscated data) without any cooperation from the service provider.

Siani Pearson et al. suggested mapping legal and regulatory approaches for privacy and security in cloud by using simulation programming that will be in any language that helps to store data on the cloud. Here they are used Service Level Agreements (SLAs) with accountability and co-design involving technological approach as performance metrics. They help in reducing user's privacy violation and enhance user's control of their data.

Smitha Sundareswaran et al. proposed a completely unique and extremely not decentralized data accountability procedure to keep following the particular application of the user's information within the cloud. This specially suggest a procedure which is based on object-related method that allows incorporating the procedures of log-in methods and data policies of owner's in executing web-based application using the commands of Java Archive Rar (JAR) programmable capabilities that each one creates. A non-static object status always ensures that any access to users' information triggers authentication and logging to the JARs is automatically executed. In improving user's administration, they conjointly offer to distribute auditing mechanisms with the Java based simulator, application server and database for storing of files. Integrated Development Environment, IDE's tools to support java application framework using log creation time, authentication time, time taken to perform logging, log merging time as performance metrics. The scalability, efficiency, and granularity of the approach are illustrated by the outcomes of the study. This paper is the ground on which the thesis is established.

Ravi Shankar V et al. proposed use of the Self Defending Object (SDO) as a framework to securing very sensitive image in the cloud. The SDO is an added security to the one offered by Object Oriented Programming Paradigm (OOPP). By using experimented environment, the SDO was built based on JAR taken from Java programming language, java, crypto, sealed object and class of java programming language which was also used for the confidentiality aspect of the proposed SDO web-based. In the simulation, the researchers have the user interacting with the cloud, they use the SDO as a security platform for the user to be authenticated to the cloud applications, they have some image stored in the cloud and they also have the cloud provider as simulation parameters. According to the name of the proposed algorithm, it offers users to mention the kind of access he/she wants to use to access the image in the cloud. The author specifies three methods for accessing the cloud: restricted, free, and paid access. The SDO also uses logger to keep session of the activities of the users after granting access to the cloud and the recorded sessions is sent to the users email address when he/she finishes the session for security purpose against the future. Finally, the proposed SDO is a solution to safeguard the security of image in the cloud built on JAAS (which it is pluggable security framework best suited for SDO in distributed environment such as cloud) of the Java programming language. It is a security measure that uses a self - immune approach, and it helps to avoid unauthorized access to an image stored in the cloud.

1.7 Aim of the Thesis

When clients and owners use cloud computing services, their information is stored remotely on the cloud with the use of some applications they do not own or control. This make them hesitant about adapting cloud computing for personal and/or professional needs. Therefore, in this study we investigate the security aspect of cloud computing with the aim to enhance the accountability of data management system and so protect the privacy of clients.

In another word, to handle this drawback, we tend to propose completely unique and extremely fragmented data accountability and auditing principles to keep monitoring the specific usage of owners' and users' information within the cloud. We used object-based tools, in our implementation. In our application we tend to balance a JAR programmable ability which each of them produce a non-static object and also confirms the access to users' information prompts authentication and automatic work native to the JARs.

The remaining part of this thesis is structured into Chapter 2 which covers architecture and design of the system, problem statement, system structure, application components, database tables with the security aspects of the system. Then Chapter 3 discuss about the implementation of the system, the used software's and hardware's and description of the systems. Finally in Chapter 4 discussions is based on the performance of the system, experiment results, comparison with previous works before making recommendations for further studies within the concluding Chapter.

Chapter 2

ARCHITECTURE AND DESIGN OF THE SYSTEM

In this thesis we designed a web application system based on "Ensuring Distributed Accountability for Data Sharing in the Cloud", a study which is done by Smitha Sundareswaran, Anna C. Squicciarini and Dan Lin (2012).

2.1 Problem Statement

As cloud computing has a considerable range of privacy and security concerns, clients' and owners' privacy needs to be ensured by adapting efficient additional methods regarding the usage as well as the management of their data. In this system the owner and user are expected to respect the system policies and the usage conditions when computing their data. For example:

A professional artist hops to sell her pictures by using the cloud service. For her marketing strategy in the cloud, she had the following specifications:

i. Only the users who paid are allowed to obtain her pictures by the cloud.

ii. Her picture can initially be viewed by prospective buyers before making payment to getting downloading access.

iii. She wants to ensure that the cloud service providers do not distribute her information with others, in such that accountability supplied to specific clients are similarly expected from the cloud service providers. Having the aforementioned specifications as tips, we point out the common specifications and hence format a strategy that ensures information accountability within the cloud.

Problems in existing system:

First, data handling is going to be directly managed by server or the Cloud Service Provider (CSP) to various components within the cloud. The components might also assign the responsibilities to each another. When the data are downloaded by the user, the Admin can have granted access rights, like view, download by the users and uploading by the owners, on the data. Figure 2.1 presents components of our system.

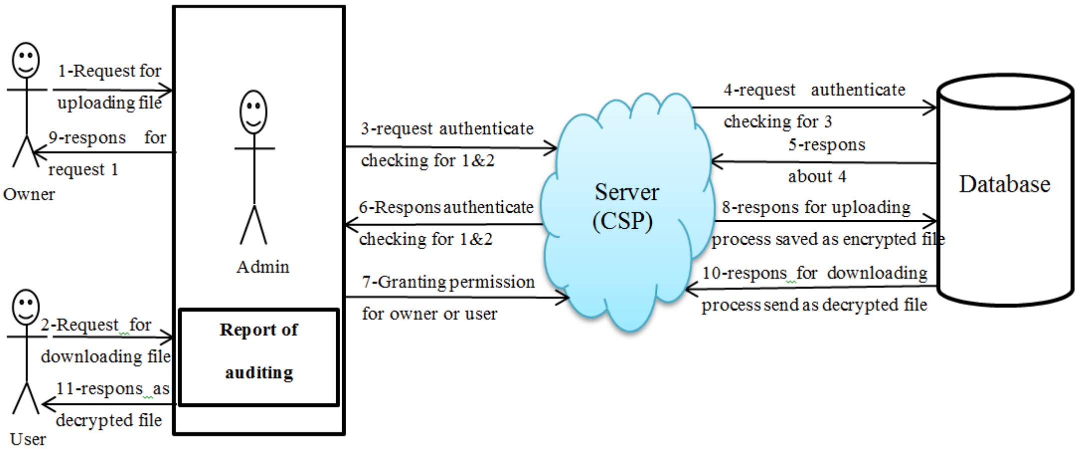

Owner
1-Request for uploading file
9-respons for request 1
User
2-Request for downloading file
11-respons as decrypted file
Admin
Report of auditing
3-request authenticate checking for 1&2
6-Respons authenticate checking for 1&2
7-Granting permission for owner or user
Server (CSP)
4-request authenticate checking for 3
5-respons about 4
8-respons for uploading process saved as encrypted file
10-respons for downloading process send as decrypted file
Database

Figure 2.1. Main Diagram for The System

Simply, when the owner wants to upload a file or user want to download a file he/she should ask Admin permission to do that then the Admin will check authentication for those parties to allow them access the database or not.

2.2 System Structure

Our system can be structured through either Data Flow Diagram or Sequence Diagram, as explained below.

2.2.1 Data Flow Diagram

The data flow diagram which is also called bubble chart is a simple graphical design that shows the flow of information that can be used in the system in terms of the inputs data and the outputs.

The figures below (Figure 2.2, Figure 2.3 and Figure 2.4) show the data flow diagrams for administrator, owner and user modules. The data flow start with registration or authorization checking for those parties to use the system when they are logged in, they will have access to specific options defined by the authorization policy as shown below.

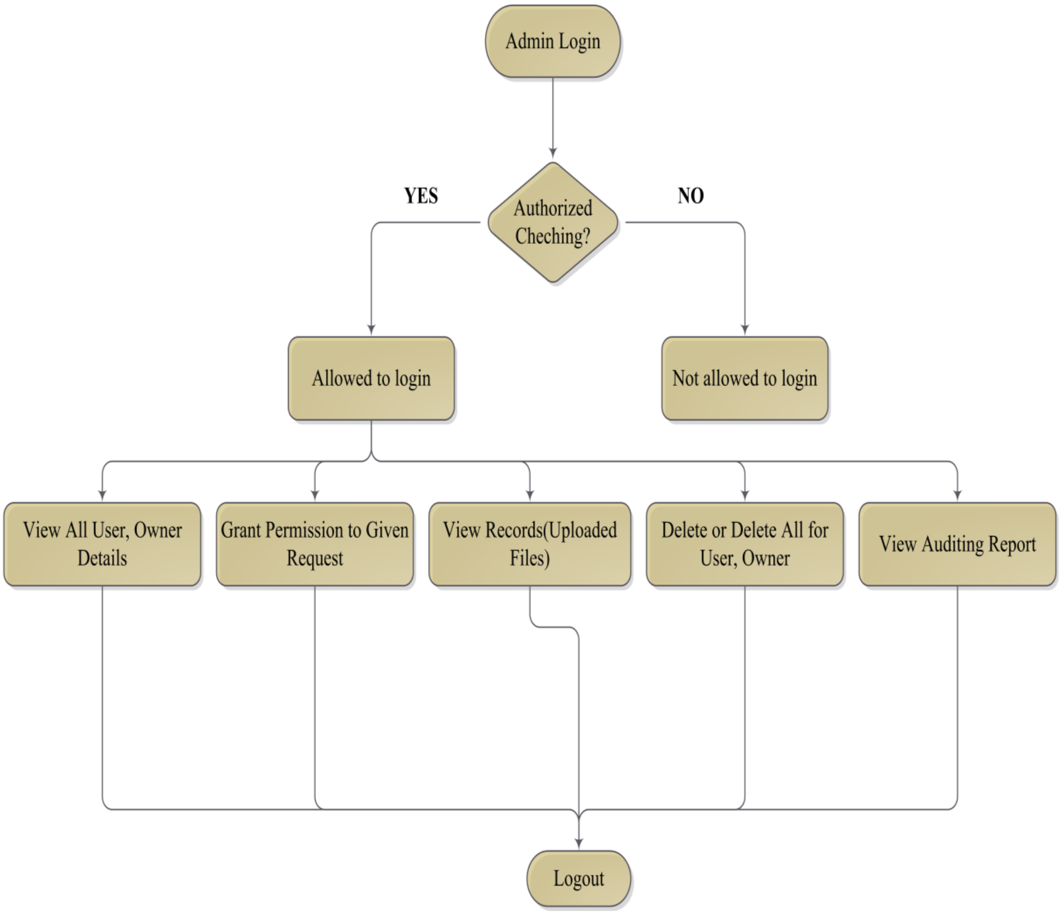

Admin Login
YES
Authorized Cheching?
NO
Allowed to login
Not allowed to login
View All User, Owner Details
Grant Permission to Given Request
View Records(Uploaded Files)
Delete or Delete All for User, Owner
View Auditing Report
Logout

Figure 2.2. Data Flow Diagram for Admin

Figure 2.2 shows the flowchart of data flow for the admin module. According to our web application, the process starts with verifying the accessibility authorization for admin to the system. If the output result is "No", the admin will not be allowed to login. However, if the admin is authorized, he/she can access his/her own options which appear on his page which includes viewing all users and owners' data, viewing all requests, viewing all records, viewing auditing reports, and the ability for delete component or records. More details about practical part of this process can be found in in Figure 2.9.

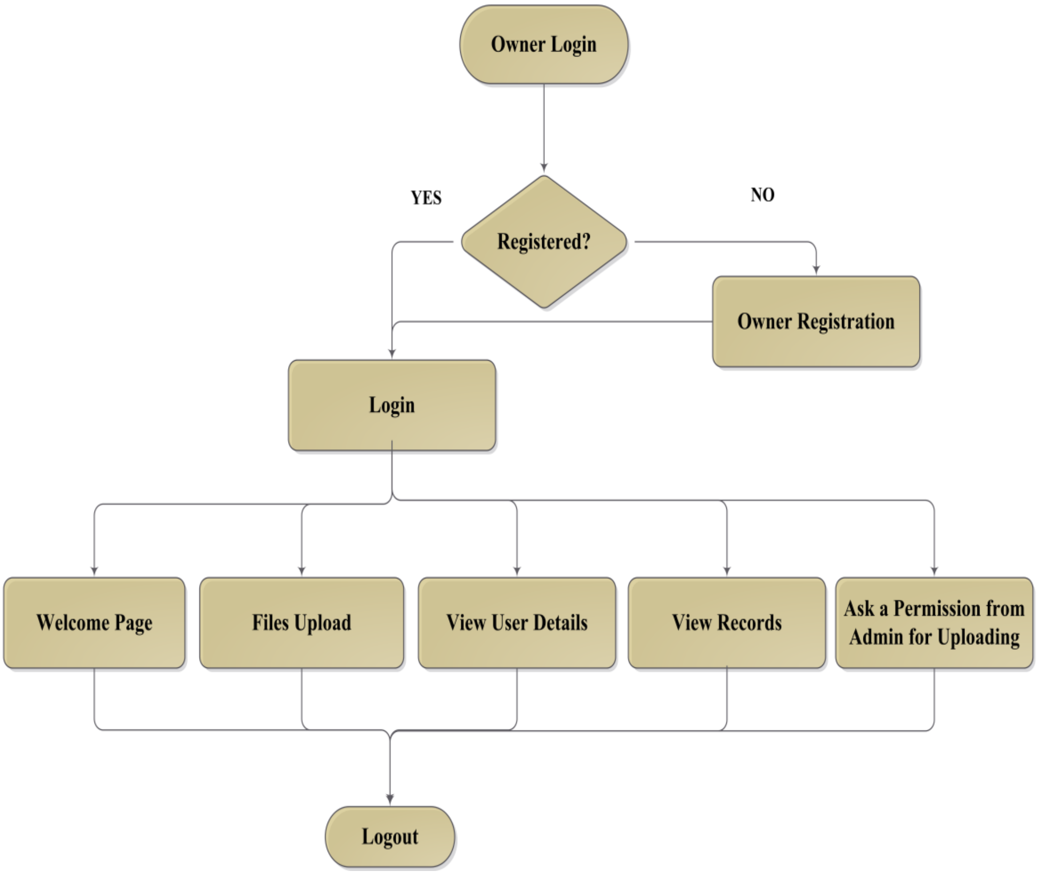


Figure 2.3. Data Flow Diagram for Owner

Figure 2.3 shows us the flowchart of data flow for the owner module. According to our web application, the process starts with verifying the accessibility authorization for owners to the system. If the output result is "No", the owners will not be allowed to login. But, if the owners are authorized they can access their own options which appear on their pages which include files upload, viewing users' details, viewing records and the ability for viewing the records as auditing for his data. More details about practical part of this process can be found in in Figure 2.10.

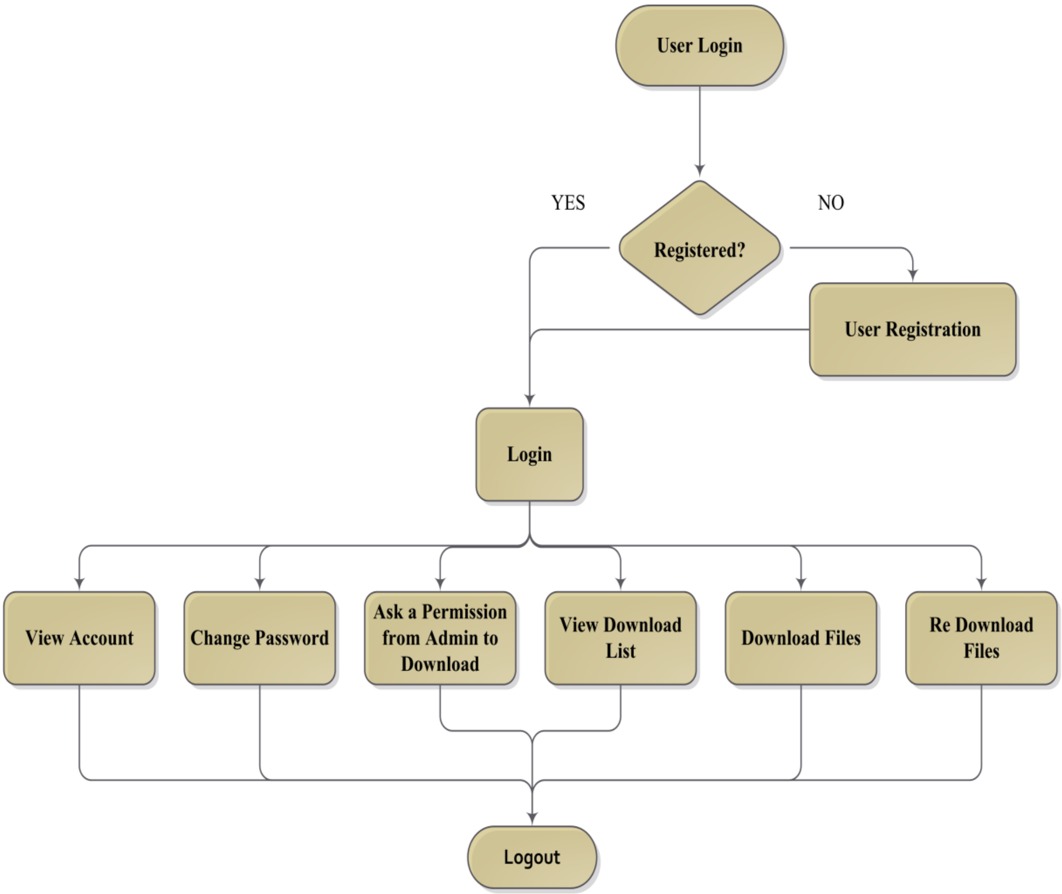


Figure 2.4. Data Flow Diagram for User

Figure 2.4 shows us the flowchart of data flow for the user module. According to our web application, the process starts with verifying the accessibility authorization for users to the system. If the output result is "No", the users will not be allowed to login. However, if the users are authorized they can access their own options which appear on his pages which include viewing his account, changing his password, viewing download list, download files and re-download the files. More details about practical part of this process can be found in Figure 2.11.

2.2.2 Sequence Diagram

The figures below (Figure 2.5, Figure 2.6 and Figure 2.7) show the sequence diagram for administrator (admin), owner and user modules. These diagrams show us the sequences of our system modules' operations.

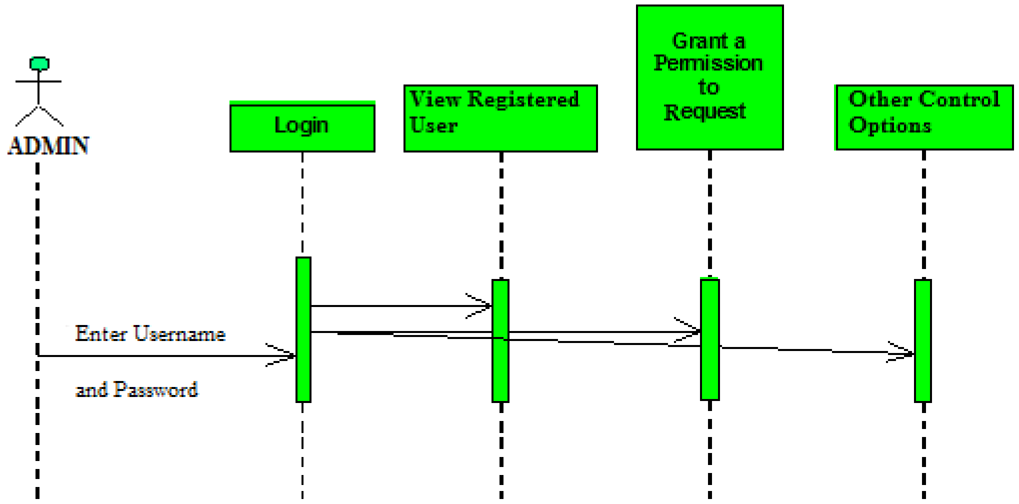


Figure 2.5. Sequence Diagram for Admin

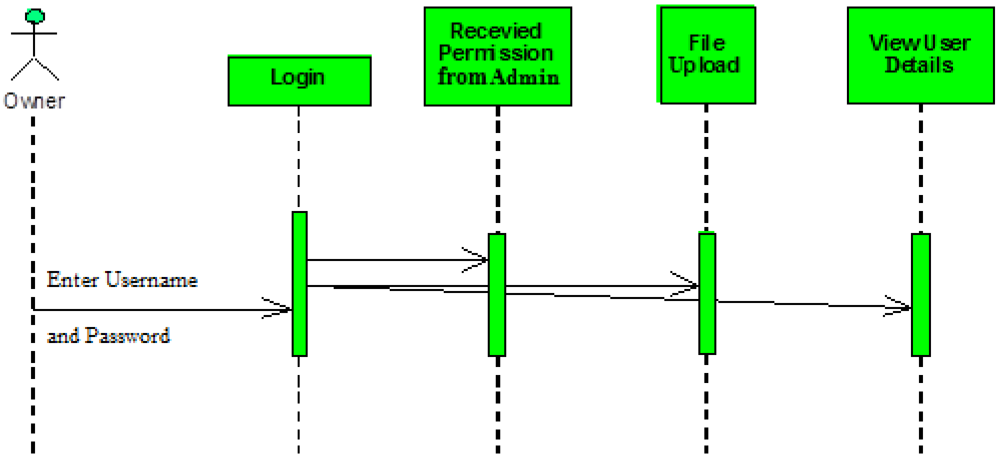


Figure 2.6. Sequence Diagram for Owner

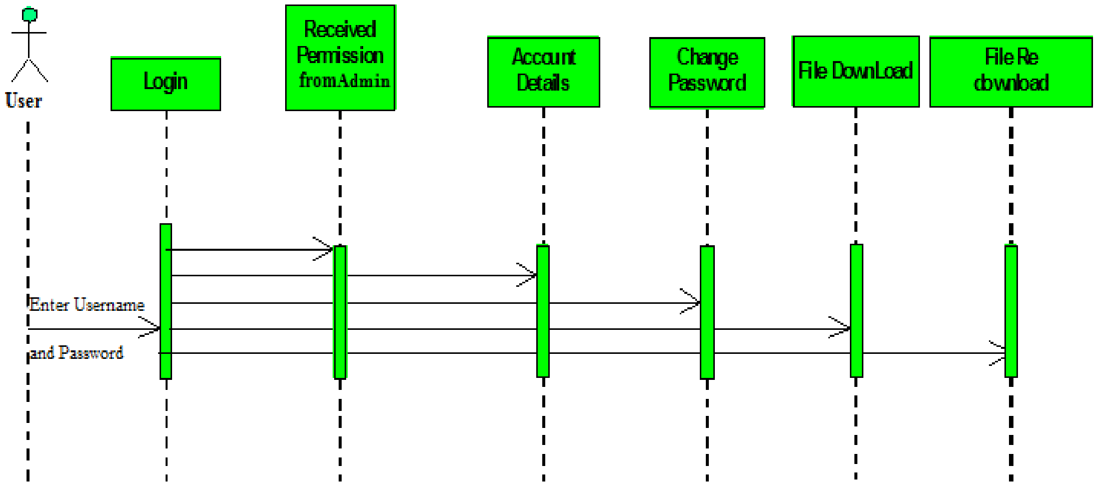


Figure 2.7. Sequence Diagram for User

2.3 Application Components

Generally, in the existing cloud computing systems the privacy of clients is at risk because their information is processed and stored by remotely servers which they cannot control. In our application we seek to design a model in which the accountability and auditing of costumers data are ensured and supplemented by extra security procedures or methods.

Our system contains the following modules as shown in Figure 2.8. These are administrator, owner and user modules or components.

Administrator module is responsible for :
- Granting permission to a user or owner.
- Auditing mechanism.
- Application settings .

Owner module is responsible for :
-Registration information.
- File uploading information.
- Monitoring data.

User module is responsible for :
-Registration information.
-View and download the files

Figure 2.8. Modules of the Application

We can figure the above modules using a diagram shown in Figure 2.1 which shows the various levels of interactions of the users and owners through the administrator who coordinates accesses to the database by those parties through authentication processes.

2.3.1 Administrator Module

The administrator module is employed by the administrator (admin) of the portal. Admin will accept or reject requests from owners, and additionally admin will accept or reject requests from users. The requests are within the kind of user and owner registration. This module has the following functionalities:

i. The admin controls records of different owners and different users and owners whose are accessing the cloud services.

ii. The admin can delete the users and owners and files as well.

iii. Admin is able to audit the users' and owners' actions.

We provided the admin in our application an auditing property which allows him to record all users' and owners' actions to discover any data tamper by owners or users. We can observe all those functionalities by practical part in Figure 2.9.

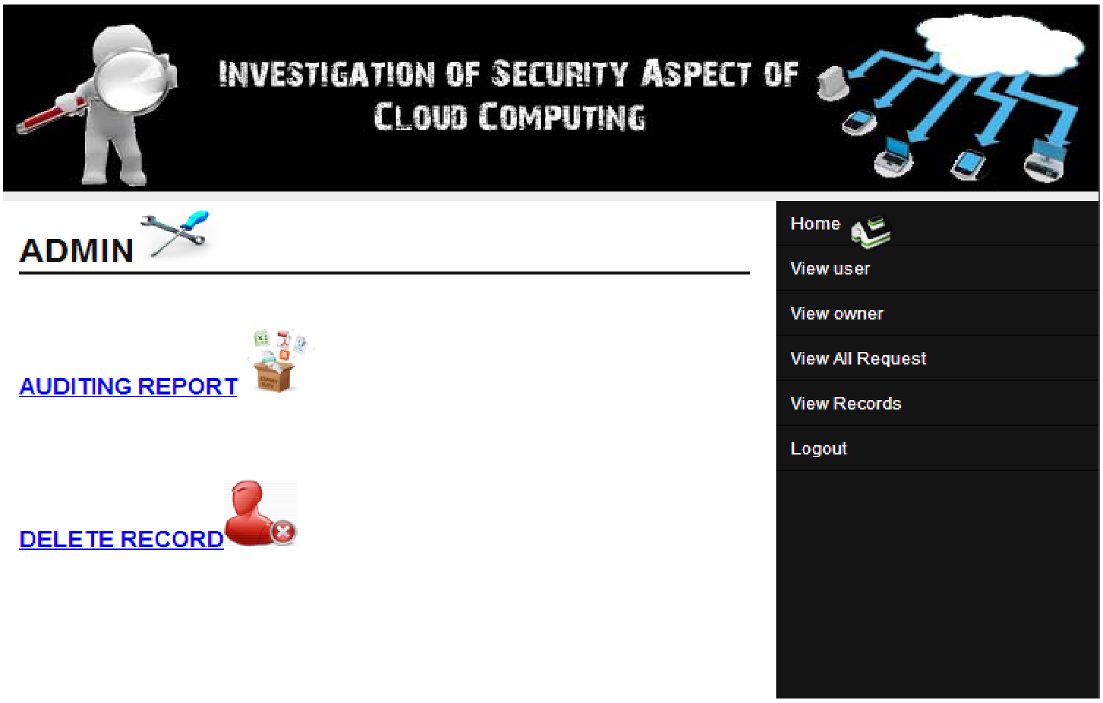


Figure 2.9. Admin Module Page

2.3.2 Owner Module

This module describes all about owners, by using this module any owner can do some operations like getting new account, uploading file into the account where any owner can also access the view records by clicking this option which its' position in his page. This module consists of the following functionalities:

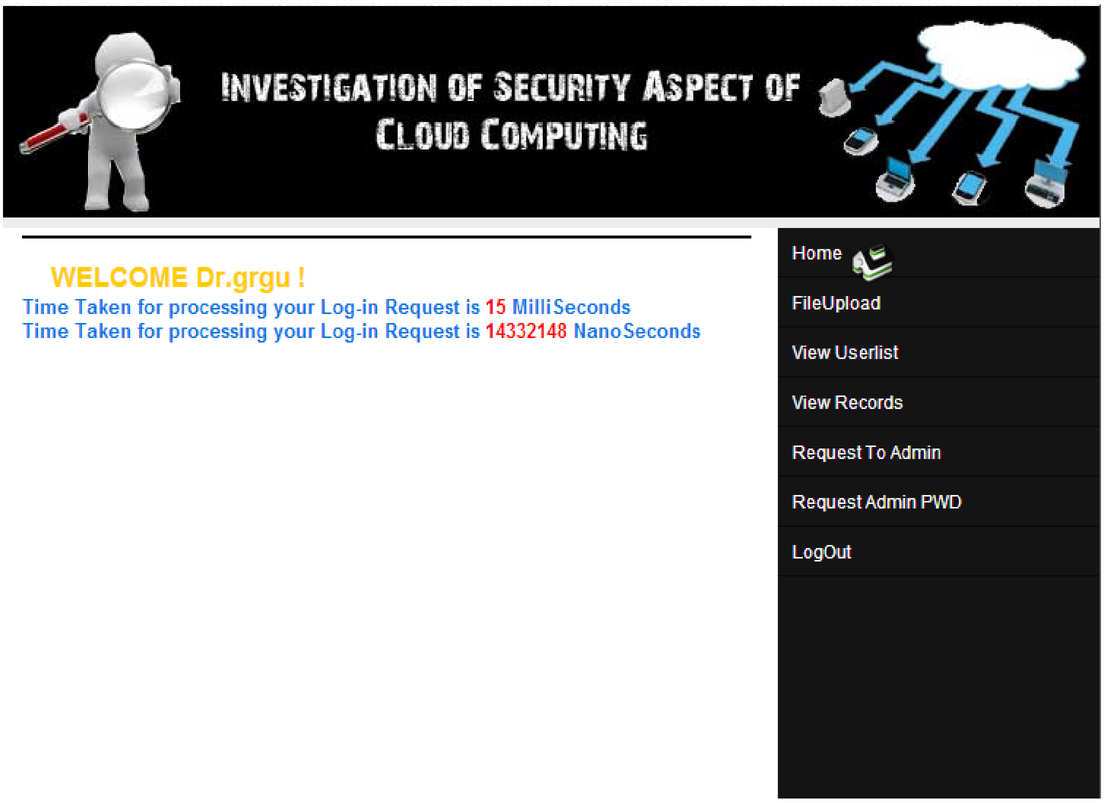
i. **Get New Account:** By the use of this, practicality owner will foundation a new account into cloud services by registering as new owner.

ii. **File Upload:** By using this functionality owner can upload files into his account. And also can delete those files by entering the file expiry date. But before this process the owner should ask permission from the admin by clicking the request to admin option.

iii. **View Records:** by using this practical application owners will get all their details about file reports like accepted transactions, rejected transactions and unfinished transactions.

iv. **Request Admin PWD:** It is an extra security option through the uploaded data can be managed and protected by a code (password) generated automatically by admin upon the request of the owner.

Also we can observe all those functionalities in practical part in Figure 2.10 below:



WELCOME Dr.grgu !
Time Taken for processing your Log-in Request is 15 MilliSeconds
Time Taken for processing your Log-in Request is 14332148 NanoSeconds

Figure 2.10. Owner Module Page

2.3.3 User Module

This module describes all about users. By using this module any user can see all owners' files but he/she cannot modify those files or tamper with it by just viewing and downloading without getting admin permission. Or we can say all users can do some operations like get a new account, view the account information, view files of owners, download a file from the account and view records. This module consist the following functionalities:

i. **Get New Account:** By using this functionality, any user will produce a new account into cloud services.

ii. **View Account Information:** By using this functionality, the users can view all their account details, this may be viewed by admin who is having account in our application.

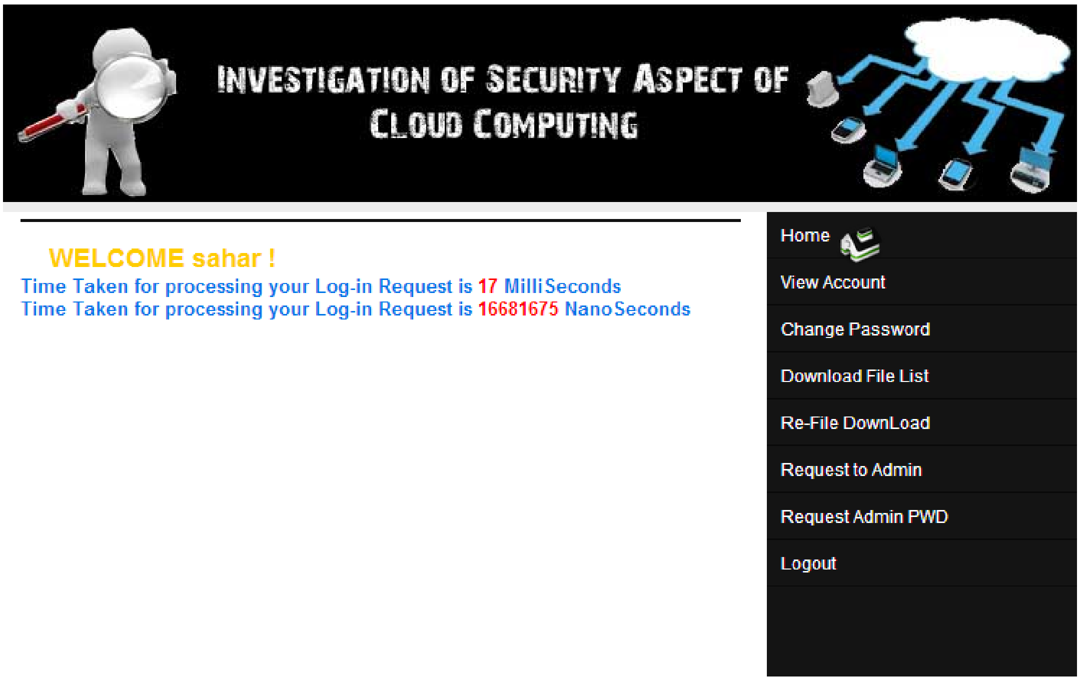
iii. **File Download:** By using this functionality users can download the files into their account by entering the file key and details of their card to pay for downloading.

iv. **View File:** By using this functionality, the users can get all their details about viewing file reports.

v. **Re-download File:** Using this functionality, users can get all the files which were downloaded earlier by entering file identification (ID) after selecting the re-download option in their page.

vi. **Request Admin password (PWD):** It is an extra security option through the downloaded data can be managed and protected by a code (password) generated automatically by admin upon the request of the user.

Also we can observe all those functionalities in practical as shown in Figure 2.11 below:



WELCOME sahar !

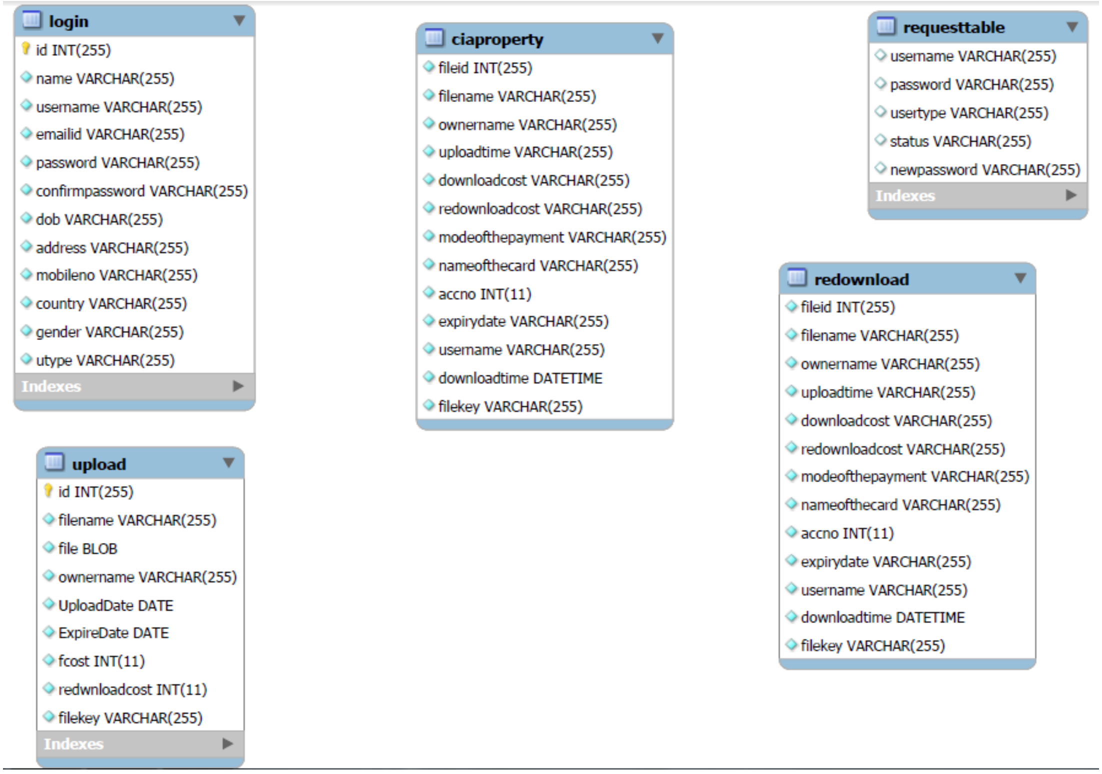
Time Taken for processing your Log-in Request is 17 MilliSeconds

Time Taken for processing your Log-in Request is 16681675 NanoSeconds

Figure 2.11. User Module Page

2.4 Database Tables

The created database named as ‘db_saharproject.sql’ is made by the Administrator. Every information essential in the creating system is found in the database. The database manages all user details, owner details and file details, requesting access details, re-download file details. Database entity diagrams given in Figure 2.12 below:



login
id INT(255)
name VARCHAR(255)
username VARCHAR(255)
emailid VARCHAR(255)
password VARCHAR(255)
confirmpassword VARCHAR(255)
dob VARCHAR(255)
address VARCHAR(255)
mobileno VARCHAR(255)
country VARCHAR(255)
gender VARCHAR(255)
utype VARCHAR(255)
Indexes

upload
id INT(255)
filename VARCHAR(255)
file BLOB
ownername VARCHAR(255)
UploadDate DATE
ExpireDate DATE
fcost INT(11)
redwnloadcost INT(11)
filekey VARCHAR(255)
Indexes

ciaproperty
fileid INT(255)
filename VARCHAR(255)
ownername VARCHAR(255)
uploadtime VARCHAR(255)
downloadcost VARCHAR(255)
redownloadcost VARCHAR(255)
modeofthepayment VARCHAR(255)
nameofthecard VARCHAR(255)
accno INT(11)
expirydate VARCHAR(255)
username VARCHAR(255)
downloadtime DATETIME
filekey VARCHAR(255)

requesttable
username VARCHAR(255)
password VARCHAR(255)
usertype VARCHAR(255)
status VARCHAR(255)
newpassword VARCHAR(255)
Indexes

redownload
fileid INT(255)
filename VARCHAR(255)
ownername VARCHAR(255)
uploadtime VARCHAR(255)
downloadcost VARCHAR(255)
redownloadcost VARCHAR(255)
modeofthepayment VARCHAR(255)
nameofthecard VARCHAR(255)
accno INT(11)
expirydate VARCHAR(255)
username VARCHAR(255)
downloadtime DATETIME
filekey VARCHAR(255)

Figure 2.12. Database Tables

All tables are explained below:

i. Cloud Information Accountability (CIA) Property Table :

Table 2.1 which is the Cloud Information Accountability table ‘cia property’ that used to store the download information, include of file and properties’ data with integers. When new file is designed by the administrator, a specific integer value for every new file is automatically attributed to the system. This value is hidden and will not appear in the system and it also contains various fields like file name, owner name, upload time, download cost, re-download cost, as described in the table below:

Table 2.1. Cia Properties

Field Name	**Data Type**
FileId	Int(11)
Filename	Varchar(255)
Owner name	Varchar(255)
Upload time	Varchar(255)
Downloadcost	Varchar(255)
Redownloadcost	Varchar(255)
Modeofthe payment	Varchar(255)
Accno	Int(11)
Expiry date	Varchar(255)
Username	Varchar(255)
Download date	Datetime
Filekey	Varchar(255)

ii. Login Table :

The ‘login’ table (Table 2.2) holds all site login tables like the id, name and password. A type of integer which is also a primary key is known as the ‘id’ field. The task of updating settings which are saved in ‘Login’ table as well as all site settings update are simply the responsibilities of the administrator.

Table 2.2. Login Table

Field Name	**Data Type**
Id(Primary Key)	Int(11)
Name	Varchar(255)
User name	Varchar(255)
Emailid	Varchar(255)
Password	Varchar(255)
Confirmpassword	Varchar(255)
Dob	Varchar(255)
Address	Varchar(255)
Mobileno	Int(11)
Country	Varchar(255)
Gender	Varchar(255)
Utype	Varchar(255)

iii. Redownload Table :

The ‘re-download’ table (Table 2.3) holds all re-download information; fileid, file name, file, ownername, upload date, download cost, re-download cost, file key, mode of the payment, name of the card, account number, expiry date, username and download time. When the owners complete their uploading to the system, and the user re-dwonload the file every of their data is stored in‘re-download’ table, where the integer is also a field in 'fileId'.

Table 2.3. Redownload File Table

Field Name	**Data Type**
FileId	Int(11)
Filename	Varchar(255)
Owner name	Varchar(255)
Upload time	Varchar(255)
Downloadcost	Varchar(255)
Redownloadcost	Varchar(255)
Modeofthe payment	Varchar(255)
Nameofthecard	Varchar(255)
Accno	Int(11)
Expiry date	Varchar(255)
Username	Varchar(255)
Downloadtime	Datetime
Filekey	Varchar(255)

iv. Request table Table :

The ‘request table’ table (Table 2.4) include of request table data like username, password as well as the user type. This is a type of varchar field known as the ‘username’ field. The name of the user is requested in ‘username’ field.

Table 2.4. Requesttable Table

Field Name	**Data Type**
User name	Varchar(255)
Password	Varchar(255)
Usertype	Varchar(255)
Status	Varchar(255)
Newpassword	Varchar(255)

v. Upload Table :

The ‘upload’ table (Table 2.5) holds all owners information; id, file name, file, ownername, upload date, expire date, fcost, redownloadcost and file key. In the 'upload' table, information that is completely uploaded to the system by the owner is stored. A type of integer which is a primary key is known as the ‘id’ field. The ownername holds the name of the owner name and upload date holds the uploading file date and fcost and file key and redownloadcost file details whenever an owner upload a file details and stored in uploaded table.

Table 2.5. Upload Table

Field Name	**Data Type**
Id(Primary Key)	Int(11)
Filename	Varchar(255)
File	Blob
Owner name	Varchar(255)
UploadDate	Varchar(255)
ExpireDate	Varchar(255)
Fcost	Int(11)
Redownloadcost	Int(11)
Filekey	Varchar(255)

2.5 Security Aspects of the System

The JAR programmable abilities is made flexible as to have a non-static object in ensuring that any access to owners' data in the cloud is liable to prompt authentication with an automatic logging to the JARs. With the help of JAR programmable capabilities, our application provide the security aspects in our application in cloud with the use of more strategic methods of coupling content through access management with Identity-Based Encryption (IBE) which provided by server hosting company (Net4), this is used to enhance more powerful protection for the data encrypted files as well saving it against chosen plaintext and cipher text attacks. After a successful authentication process, the user is allowed to access the information contained within the JAR programmable abilities.

2.5.1 Auditing Mechanism

This mechanism is a security mode adds to the options available for both the admin and the owner. Through this mechanism the admin can follow up and control the actions of owners and users while the owner for his part use this mechanism to follow up the data or actions of the users.

The auditing mechanism involves tow modes:

i. Push Mode.
ii. Pull Mode.

Push mode: In this mode, the actions of the users are simultaneously pushed to the data owner or to the admin in an automated fashion, so they can respond, based on the options available to them. That is to say, the admin can record all the actions of both the users and the owners while the owner can monitor the users who access the data uploaded by the owner. This mode serves essential function within the logging architecture which permits timely detection and correction of any loss or injury to the log files.

Pull mode: This mode allows the admin to retrieve the data resulted from the actions taken by both of the owners and users actions. Also the owner can retrieve and track the actions of those who access his data. We have the summarized auditing mechanism in the Figure 2.13 below:

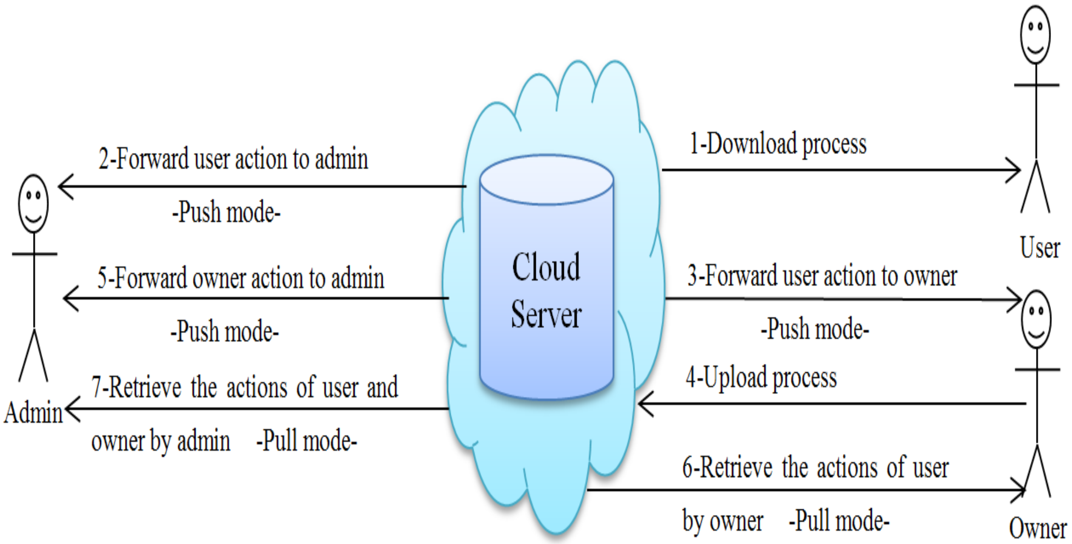

2-Forward user action to admin
-Push mode-
5-Forward owner action to admin
-Push mode-
7-Retrieve the actions of user and
owner by admin -Pull mode-
Admin
Cloud Server
1-Download process
User
3-Forward user action to owner
-Push mode-
4-Upload process
6-Retrieve the actions of user
by owner -Pull mode-
Owner

Figure 2.13. Push and Pull modes in Audting Mechanism

2.5.2 Auditing Technique for Solving Dispute and Copying Attacks

Pull mode in the auditing mechanism enable us to solve any dispute may be raised by the data user it happens that users claim they cannot access the data which the users paid for, in this case the admin can retrieve use the pull mode to verify the truth of this claim, which will be as this record (User Name, User Email, File Id, File Name, Owner Name, Download Cost, Download Time). Similarly if the owner makes claim that his data are accessed illegally in this case the admin can do the same thing.

Copying Attack:

The most axiom attack is when the attacker copies entire JAR files. The attacker might assume that doing these permits accessing the information within the JAR file without being detected by the data owner. However, such attack is going to be detected by our auditing mechanism.

Recall that each of JAR file needed to send action records to the admin particularly, with the push mode so that the admin can send the action record to the data owners periodically. That is, although the data owner is not acquainted with the functioning of the extra copies of its JAR files, he can still be ready to accept log records from every available copy. When attackers send copies of JARs to areas wherever the admin cannot see, the copies of JARs can presently become inaccessible (Sundareswaran, Squicciarini, & Lin, 2012). This can be as a result that every JAR is needed to write down redundancy data to the admin regularly. If the JAR cannot contact the admin, the access to the content within the JAR will be disabled. Thus, the logger part provides additional transparency than conventional log files encryption; it permits the data owner to find when an attacker has created copies of a JAR.

2.5.3 Accountability

Accountability mechanisms are procedures and tools which focus on keeping the data usage transparent and track able, often technical tools including software. But also in organizational and/or legal procedures and other mechanisms – by which accountability practices are supported and implemented, we can say that accountability is an important but complex notion that encompasses the obligation to act as a responsible steward of relaying personal information to others.

This include taking responsibility for the protection and appropriate use of that information, to be transparent (give account) about how this has been done and to provide remediation and redress.

This notion is increasingly seen as a key market enabler in global environments and in helping overcome barriers to cloud service adoption. However, the relative complexity of the service provision chain makes it very challenging both legally and technically to providing accountability for and in the cloud. Our system seeks to ensure the accountability of data usage through the authorization login and the auditing mechanism, none of the parties can use the data illegally or manipulate them. So, all the parties are held accountable for their actions. Thanks to push and pull modes of the auditing as motioned previously.

Chapter 3

IMPLEMENTATION OF THE SYSTEM

In this chapter we are going to explain the software tools and environment which we used to developed our system that include the used software, hardware and database and how we connect them by using database connectivity.

3.1 The Used Softwares

Firstly we need to mention that we used windows 7 operating system to develop our application, but this application can be also applied to Windows 95/98/2000/XP. The said application is based on:

i. User Interface: HTML, Java, Jsp

ii. Java Development Kit (JDK 1.6)

iii. NetBeans IDE 7.2 v

iv. Mysql 5.5v

v. Tomcat application server 6

vi. Database Connectivity: JDBC

3.1.1 User Interface

The user interface is the space where interaction between the user and the system occurs. We developed it by using html and java languages. In this interface the users' activities are processed in terms of:

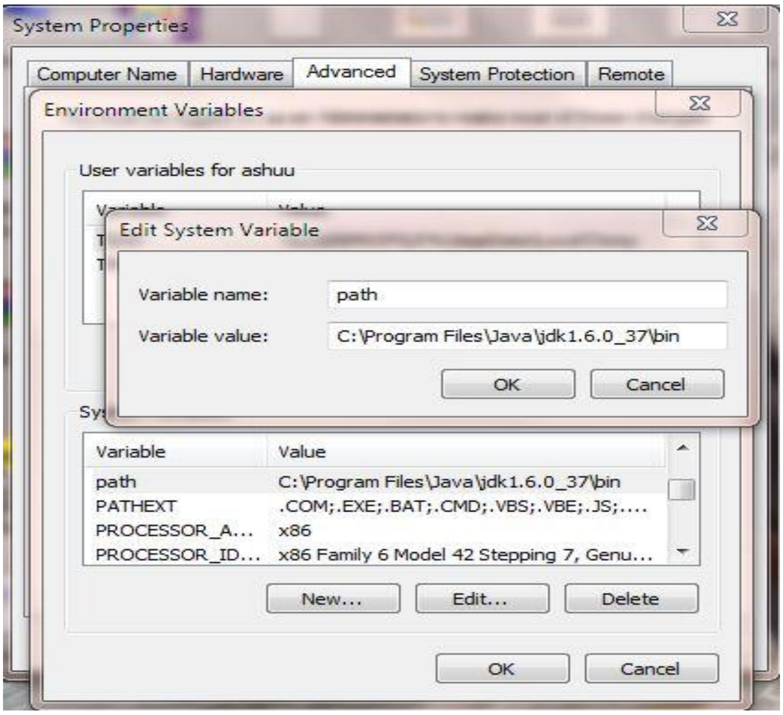
- **i.** Input, allows the users to manipulate a system.
- **ii.** Output, allows the system to show the effect of the users' manipulation.

3.1.2 Java Development Kit (JDK 1.6)

Java is technology-based programming language which is in flat form. It has many principle concepts of Object Oriented Programming(OOP) language principles. So we choose java based on the requirement of the application because it has may facilities for the programme to build the application.We can download this product (JDK) from oracle site (www.oracle.com) and then install it. We applied java for server side while the html language is used in client side they are combined through Javascript Programming (JSP) for warning messages.

By accepting the license agreement java installation will be completed as JDK which contains Java Runtime Environment (JRE) and Java Standard Edition(JSE) development kit which also will be installed when you click on the finish button. After the finish of installation of JDK we need to set the path for the JVM (java virtual machine) to run the program.

To set the path for JVM we should follow the prompts after right clicking on (my computer->properties->advanced system setting->Environmental variables->new->variable name->type “path”->variable value->) as shown in Figure 3.1.



Variable name: path

Variable value: C:\Program Files\Java\jdk1.6.0_37\bin

Variable	Value
path	C:\Program Files\Java\jdk1.6.0_37\bin
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	x86
PROCESSOR_ID...	x86 Family 6 Model 42 Stepping 7, Genu...

Figure 3.1. Setting the Path in System Properties

3.1.3 NetBeans

The NetBeans is an Integrated Development Environment (IDE) written in java, my eclipse, eclipse, java builder and can run on windows for developing primarily with java, but also with other languages in particular PHP, C, C++, HTML and many others languages.

The NetBeans IDE is used in according to the following steps:

i. Start NetBeans IDE.

ii. In the IDE, choose File > New Project (Ctrl-Shift-N), as shown in Figure 3.2.

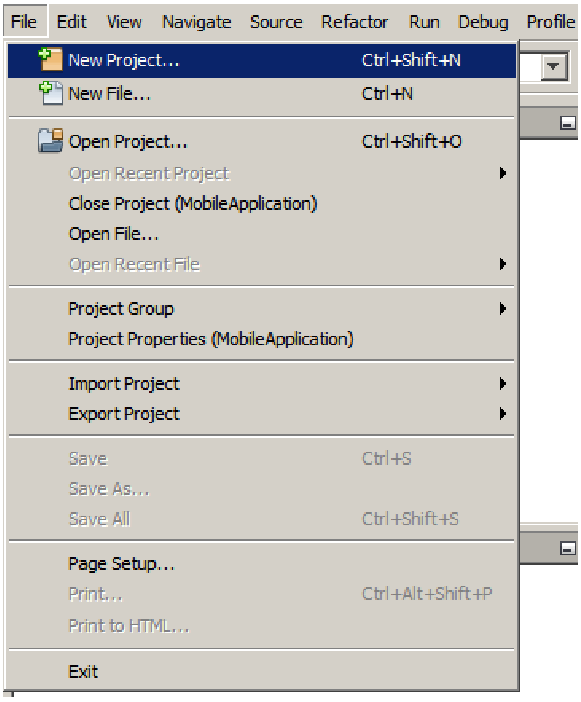
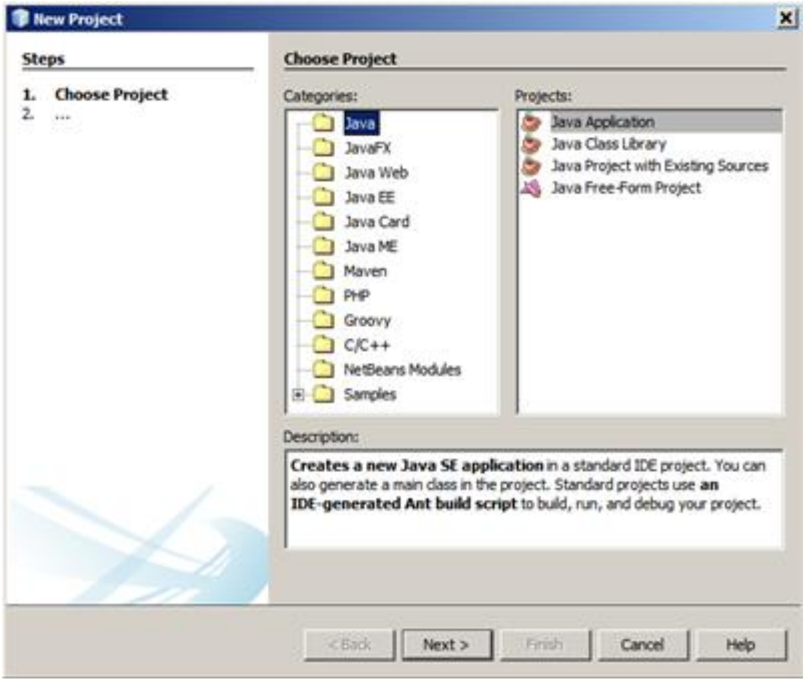


Figure 3.2. Creat New Project in NetBeans

iii. In the New Project wizard, expand the java category and select java application as shown in Figure 3.3. Then click Next.

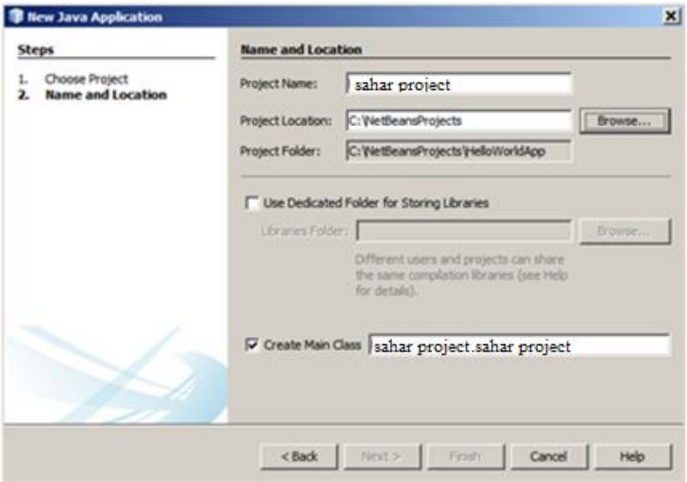


Creates a new Java SE application in a standard IDE project. You can also generate a main class in the project. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.

Figure 3.3. Creat New Project in NetBeans Details

iv. In the name and location textbox of the wizard, do followings:

- In the project name field, type the name of your project,(example: sahar project).
- Don't select the option (dedicated folder for storing libraries).
- In create main class field, write the name of your project as shown in the example (sahar project.sahar project).
- Then click Finish as shown in Figure 3.4.



Project Name: sahar project

Project Location: C:\NetBeansProjects

Project Folder: C:\NetBeansProjects\HelloWorldApp

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

☑ Create Main Class sahar project.sahar project

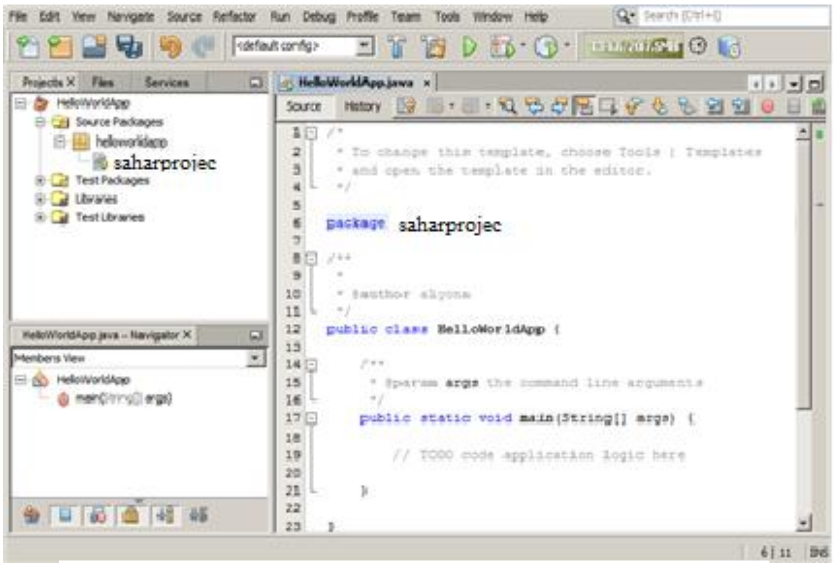
Figure 3.4. Another Steps for Java Installation

After those steps the project is created and opened in the NetBeans. So you can see the following components:

i. The Projects window, which contains a tree diagram showing the components of the project, including source files and libraries that your code relies on and so on.

ii. The source editor window with a file named sahar project is opened; this is our project file.

iii. The navigator window is used to rapidly cross between elements within the selected class. We can observe all these components in Figure 3.5.

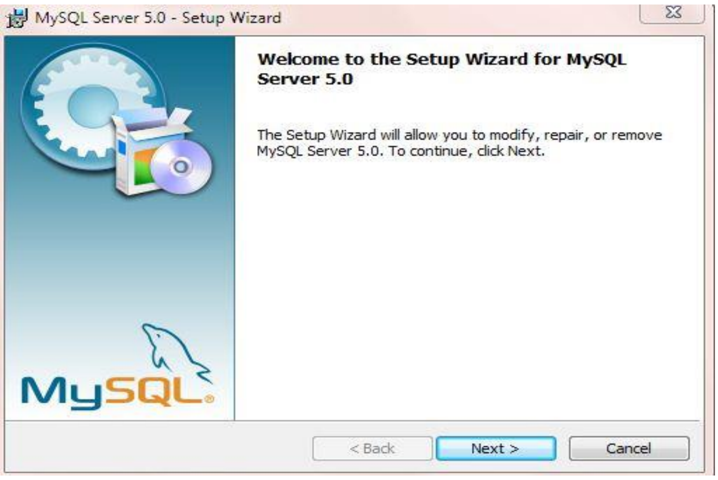


```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package saharprojec

/**
 *
 * @author alyona
 */
public class HelloWorldApp {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {

        // TODO code application logic here

    }

}
```

Figure 3.5. NetBeans Window

3.1.4 MYSQL 5.5v

MySQL is a database that is presently the most popular open source database. It is very commonly used in conjunction with PHP scripts and acts as a bank end for many applications and also to create powerful and dynamic server-side applications. We can likewise get this product from oracle site and Figure 3.6 shows us its installation.



Welcome to the Setup Wizard for MySQL Server 5.0

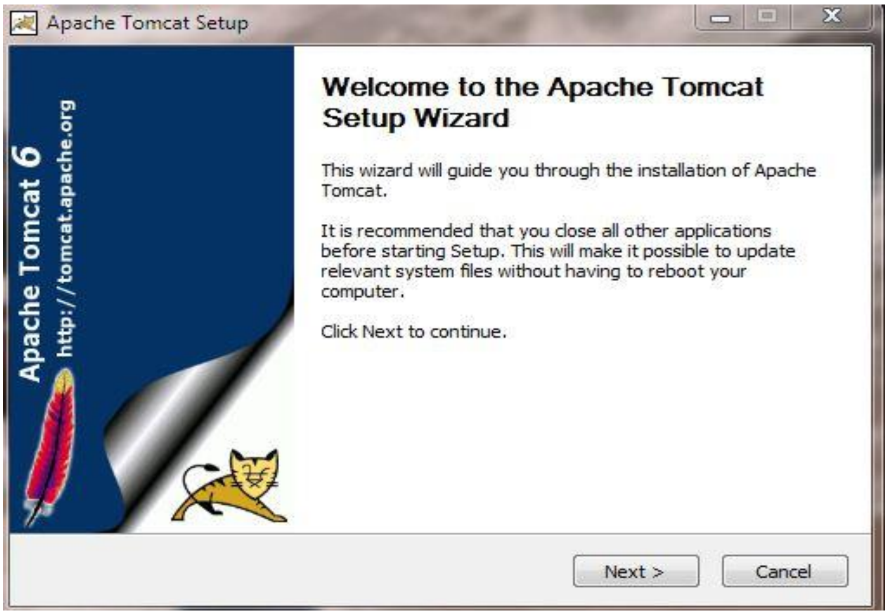
The Setup Wizard will allow you to modify, repair, or remove MySQL Server 5.0. To continue, click Next.

Figure 3.6. MySql Setup

After clicking on the Next icon and accept terms and conditions, then click on Next icon to finish the setup file as you are installing the sever and naming the userid and password for server as db_sahar and password@sahar to complete the installation.

3.1.5 Tomcat Application Server 6

Application server (servlet container that is used in the official reference implementation for the java servlet and java server pages technologies) is processing a request from the browser and getting the response from the server. In the application servers of java, the server act like an extended virtual machine for running application programs, transparently handling connections to the database one-sidedly and frequent connections to the web client on the other hand. The Figure 3.7 below shows tomcat server setup window.



Welcome to the Apache Tomcat Setup Wizard

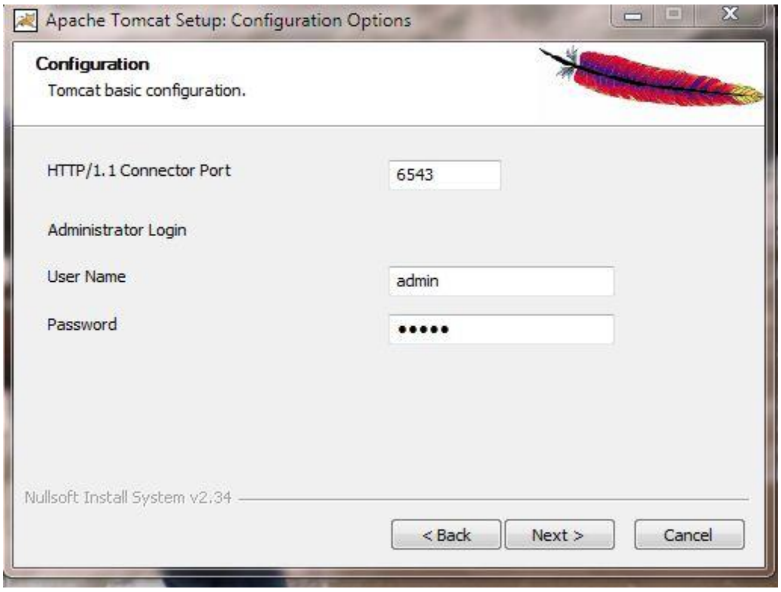
This wizard will guide you through the installation of Apache Tomcat.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Figure 3.7. Installation Steps for Tomcat

Click on the Next icon to continue the installation process after agreement on the license and the setting-up port details are made by entering the user id with password for authentication as shown in Figure 3.8 below:


Apache Tomcat Setup: Configuration Options

Configuration

Tomcat basic configuration.

HTTP/1.1 Connector Port: 6543

Administrator Login

User Name: admin

Password: •••••

Figure 3.8. Next Step for Tomcat Installation

3.1.6 Database Connectivity (JDBC)

A Java database connection is an integral part of SQL and it is the means by which a server and its user interfaces applications communicate between each other. The consumer uses a database connection to send queries and retrieving the data from the server. It is used for the following purposes:

i. Load the driver

ii. Define URL connections

iii. Establish a connection with database

iv. Send SQL statment

v. Execute a query

vi. Process the results

vii. Close the connection

JDBC class involved in Java-sql pakage is used for the fallowing code of JDBC in our project: The code below is used as JDBC in our project:

```
Class.forName("com.mysql.jdbc.Driver");
//Connection
conn3=DriverManager.getConnection("jdbc:mysql://localhost:3306/DB_saharproject","saharproject","admin@sahar");
Statement st=conn3.createStatement();
ResultSet rs2=st.executeQuery(sql)
```

3.2 The Required Hardware

The hardware, in the computer world, refers to the physical components that make up a computer system. And it is one of the common features which interact with the software by operating system to implement different applications.

Below we listed the minimum requirements needed for our project:

Processor	-	Intel Duel core
RAM	-	512 MB
Hard Disk	-	60 GB
Floppy Drive	-	1.44 MB
Key Board	-	Standard Windows Keyboard
Mouse	-	Optical Mouse
Monitor	-	LCD color

The Figure 3.9 shows the Complementary portions included in our application.

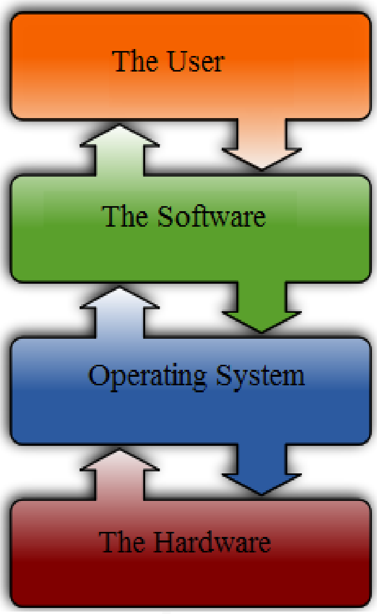


Figure 3.9. System Layers

3.3 System Implementation

Our system mainly discusses client's side web application which consists of owner and user access options. This application is aimed to guarantee to the user and the owner more secure and more accountable data accessibility. Figure 3.10 shows us the interface page which is in order to login or register as new user or owner in this system.

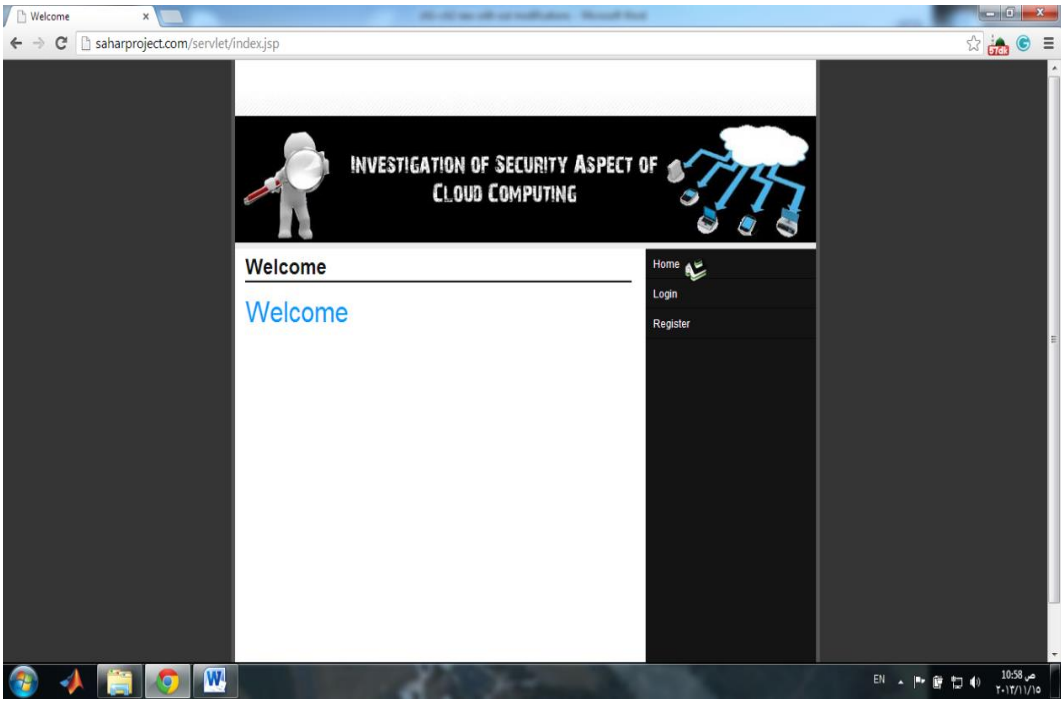

Welcome
saharproject.com/servlet/index.jsp
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Welcome
Welcome
Home
Login
Register
EN
10:58 ص
٢٠١٣/١١/١٥

Figure 3.10. Welcome Page

When clients click on register option in welcome page as shown in Figure 3.10 they will access a new page that enables them to register as shown in Figure 3.11. Once they access the page, it is required to fill in a registration form which includes: name field, username field, email, password, conform password, birth date, address, mobile number, country, gender and user type.

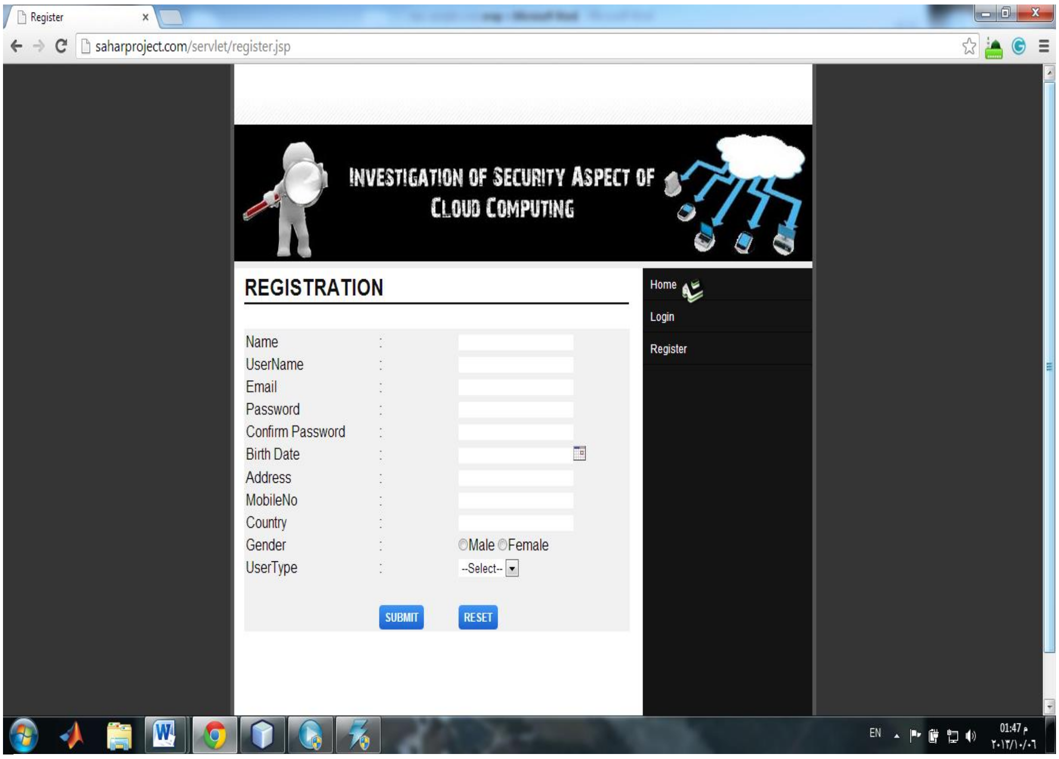


Figure 3.11. Registration Page

After the client completed the needed information in the registration form they move to the login page where they are required to fill in the following fields the username, password and user type as shown in the Figure 3.12.

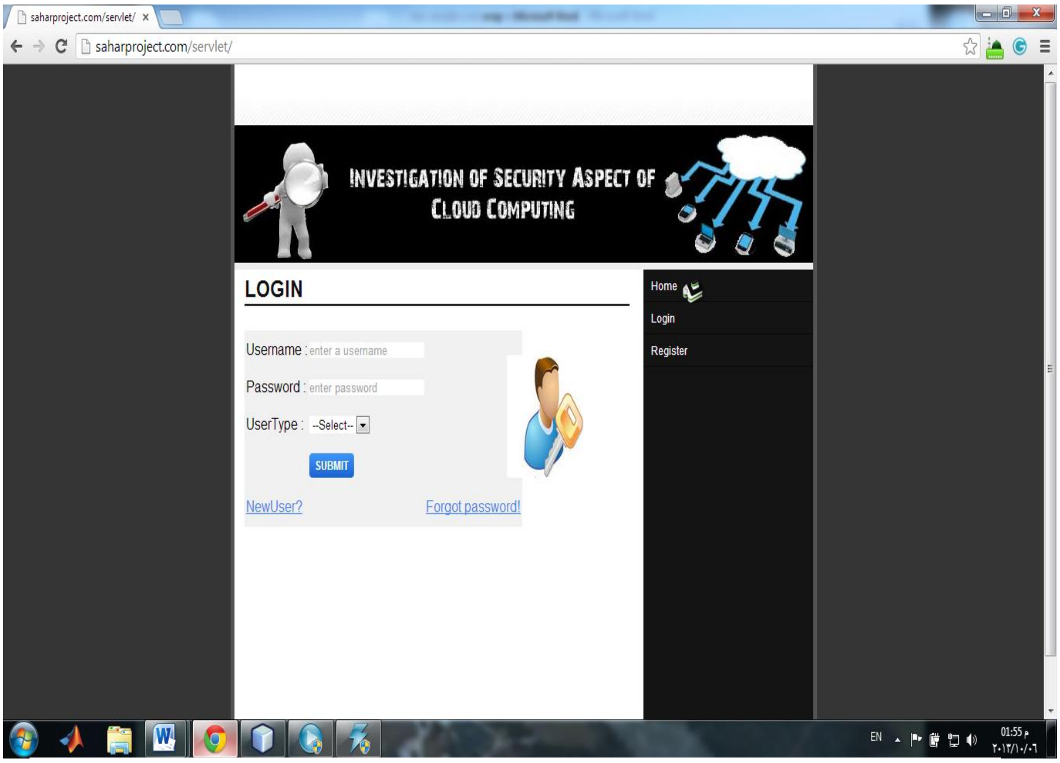
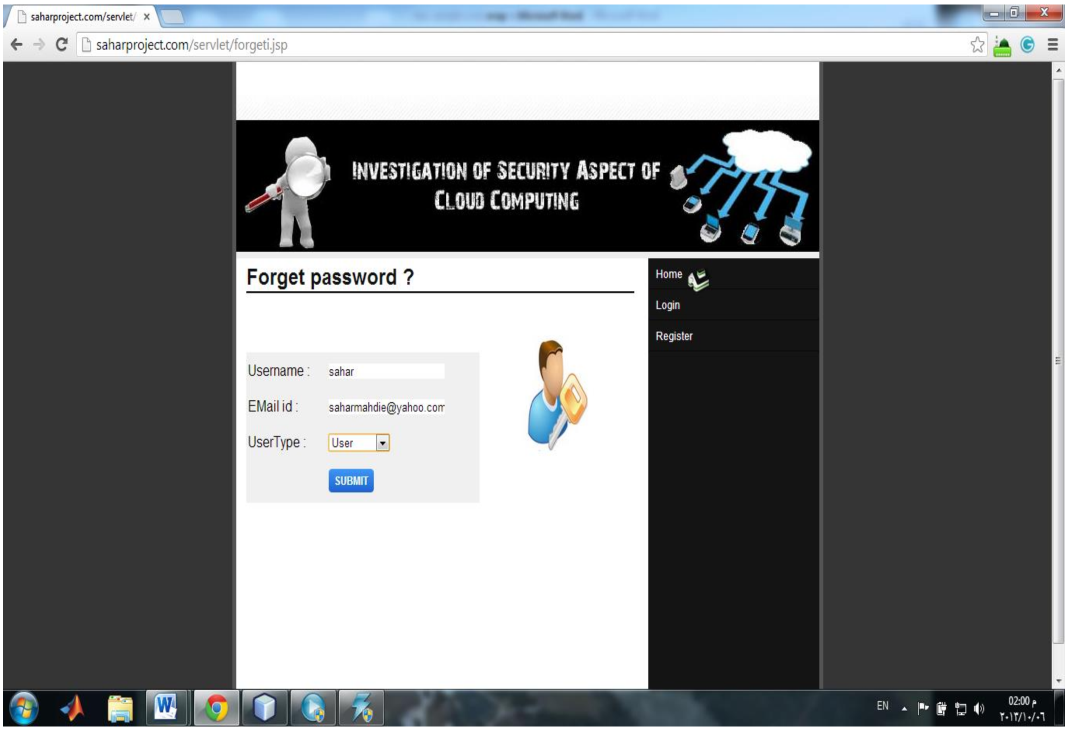

saharproject.com/servlet/
saharproject.com/servlet/
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
LOGIN
Username : enter a username
Password : enter password
UserType : --Select--
SUBMIT
NewUser?
Forgot password!
Home
Login
Register
EN
01:55 م
٢٠١٣/١٠/٠٦

Figure 3.12. Login Page

After clients completed entering the needed information in the login page the system checks whether the credentials are included in the data base or not. If they are found then clients will be navigated into their pages if not error page will be displayed.


saharproject.com/servlet/
saharproject.com/servlet/forgeti.jsp
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Home
Login
Register
SUBMIT

Forget password ?

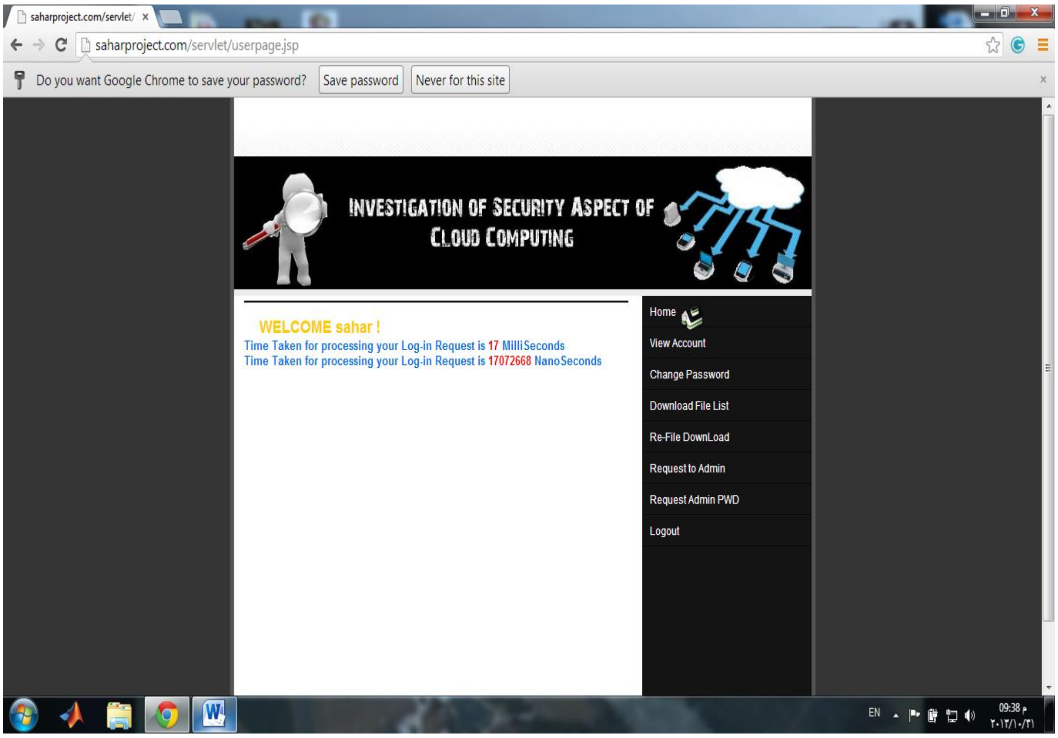
Username : sahar

EMail id : saharmahdie@yahoo.com

UserType : User

Figure 3.13. Forget Password Page

In case the clients forget their password they need to click on the forget password option which transfers them to (forget password jsp page). They are required to enter their user name, email and user type then the password will be automatically sent to their email.


saharproject.com/servlet/
saharproject.com/servlet/userpage.jsp
Do you want Google Chrome to save your password?
Save password
Never for this site
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Home
View Account
Change Password
Download File List
Re-File DownLoad
Request to Admin
Request Admin PWD
Logout
EN
09:38

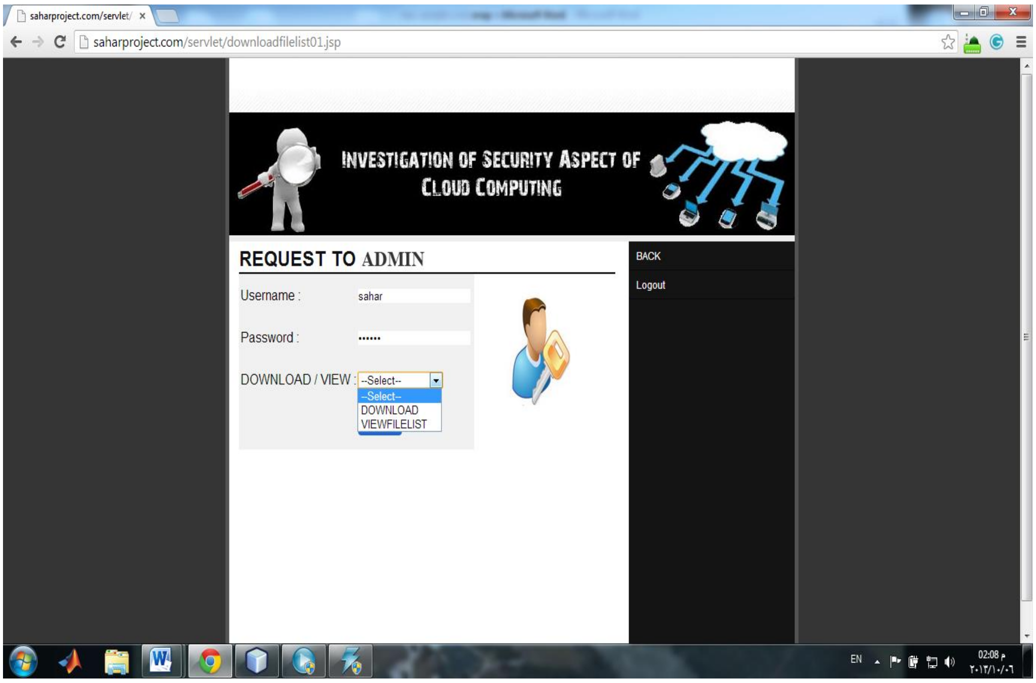
WELCOME sahar !

Time Taken for processing your Log-in Request is 17 MilliSeconds

Time Taken for processing your Log-in Request is 17072668 NanoSeconds

Figure 3.14. User Welcome Page

Upon the users login to the welcome page they can use all the available options in their page. These options include: view account, change password, download files and re-download files after getting the permission from the Admin as shown in Figure 3.14.



saharproject.com/servlet/
saharproject.com/servlet/downloadfilelist01.jsp
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
BACK
Logout
EN
02:08 م
٢٠١٣/١٠/٠٦

REQUEST TO ADMIN

Username : sahar

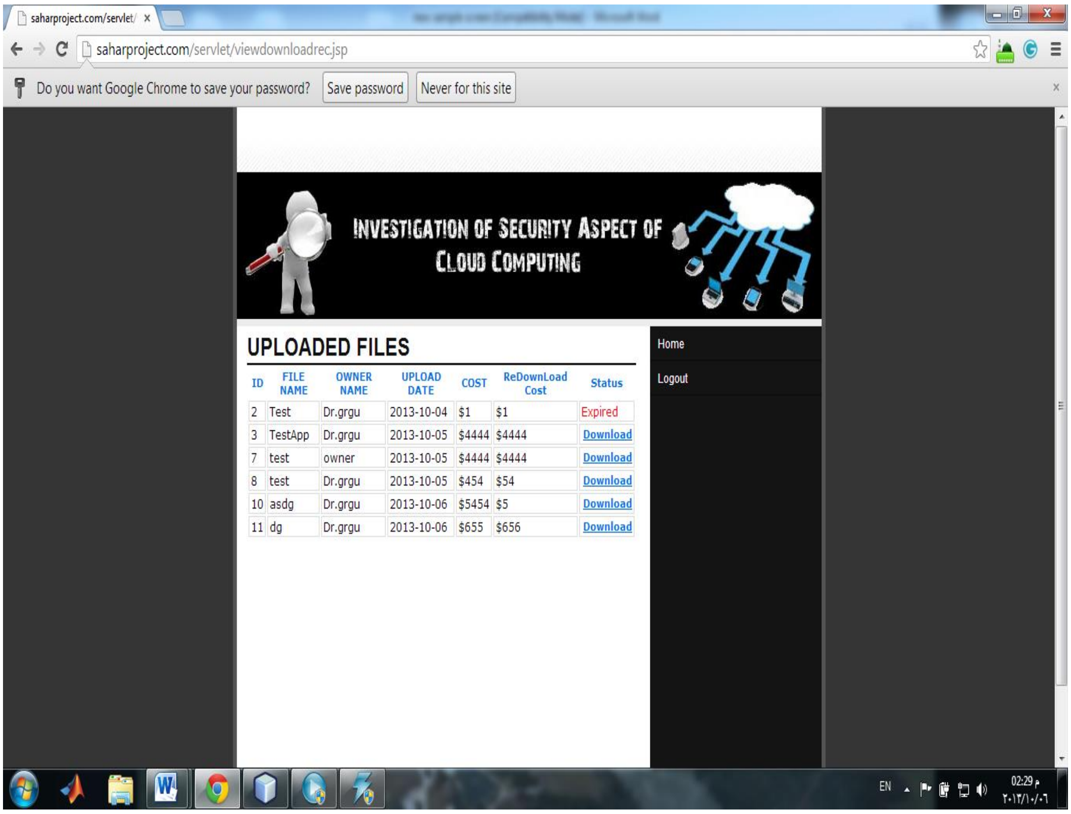
Password : ••••••

DOWNLOAD / VIEW : --Select--

--Select--
DOWNLOAD
VIEWFILELIST

Figure 3.15. Request to Admin

When the users wants to view or download the list of available files or view any of them the users should click on the option (request to admin) which will refer them to Request Admin Page, Where the users can download or viewing files list as shown in the Figure 3.15.



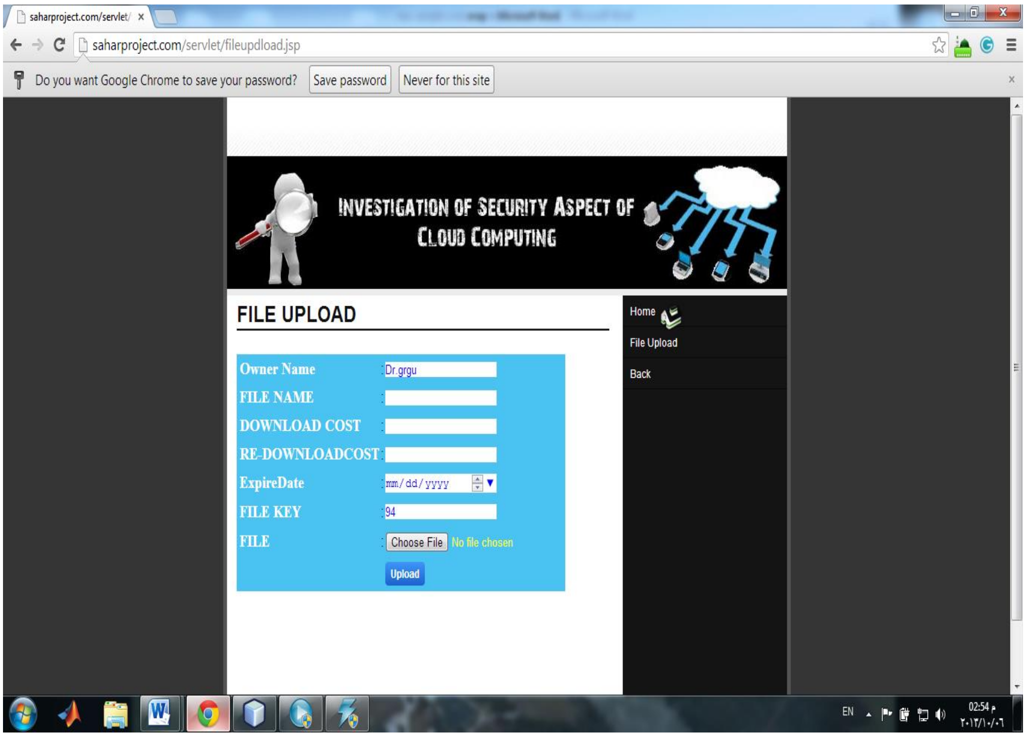
saharproject.com/servlet/
saharproject.com/servlet/viewdownloadrec.jsp
Do you want Google Chrome to save your password?
Save password
Never for this site
INVESTIGATION OF SECURITY ASPECT OF
CLOUD COMPUTING
Home
Logout
EN
02:29 م
٢٠١٣/١٠/٠٦

UPLOADED FILES

ID	FILE NAME	OWNER NAME	UPLOAD DATE	COST	ReDownLoad Cost	Status
2	Test	Dr.grgu	2013-10-04	$1	$1	Expired
3	TestApp	Dr.grgu	2013-10-05	$4444	$4444	Download
7	test	owner	2013-10-05	$4444	$4444	Download
8	test	Dr.grgu	2013-10-05	$454	$54	Download
10	asdg	Dr.grgu	2013-10-06	$5454	$5	Download
11	dg	Dr.grgu	2013-10-06	$655	$656	Download

Figure 3.16. Download File Details

The files uploaded by the owner are displayed once the user click on the option download file as shown in the Figure 3.16 the user can only download the active files. It is noted that expired times are caused by data owners.


saharproject.com/servlet/
saharproject.com/servlet/fileupdload.jsp
Do you want Google Chrome to save your password?
Save password
Never for this site
Home
File Upload
Back
Upload
EN
02:54

INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING

FILE UPLOAD

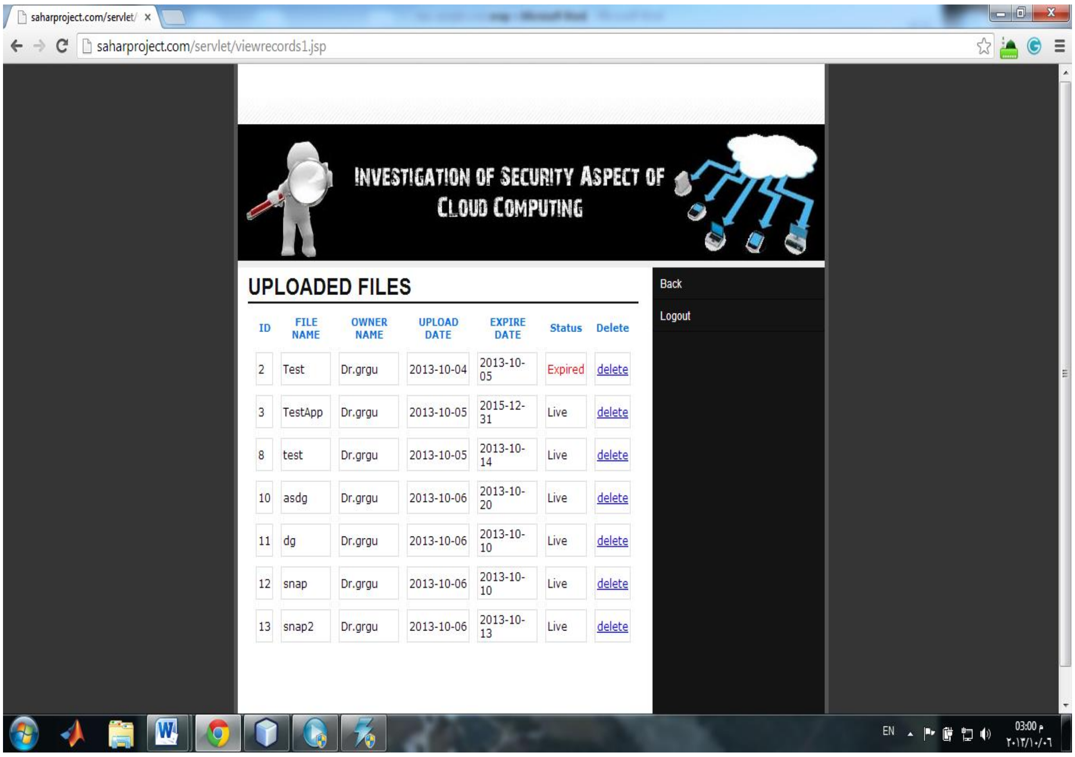
Owner Name	: Dr.grgu
FILE NAME	:
DOWNLOAD COST	:
RE-DOWNLOADCOST	:
ExpireDate	: mm/dd/yyyy
FILE KEY	: 94
FILE	: Choose File No file chosen

Figure 3.17. File Upload Page Details

After explanations of some important pages for user module in the previous figures now we chose to display file upload page as shown in Figure 3.17 this is one of the important owners' module pages that shows us uploading file details by filling requested information textbox in this page. After that the owner finishes uploading file to the system within cloud.

This process is done after the owner's login to the system then they can navigate all their privileges and options through his page as well as the user module, as shown in Figure 2.10 the owners' options was uploading files, viewing users list, view records and monitoring his data by viewing user list and their actions on his data.

The view records and delete option functionalities are shown in Figure 3.18.


saharproject.com/servlet/
saharproject.com/servlet/viewrecords1.jsp
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Back
Logout
EN
03:00 م
٢٠١٣/١٠/٠٦

UPLOADED FILES

ID	FILE NAME	OWNER NAME	UPLOAD DATE	EXPIRE DATE	Status	Delete
2	Test	Dr.grgu	2013-10-04	2013-10-05	Expired	delete
3	TestApp	Dr.grgu	2013-10-05	2015-12-31	Live	delete
8	test	Dr.grgu	2013-10-05	2013-10-14	Live	delete
10	asdg	Dr.grgu	2013-10-06	2013-10-20	Live	delete
11	dg	Dr.grgu	2013-10-06	2013-10-10	Live	delete
12	snap	Dr.grgu	2013-10-06	2013-10-10	Live	delete
13	snap2	Dr.grgu	2013-10-06	2013-10-13	Live	delete

Figure 3.18. View Records and Delete Option Page

The last page which we want to mention is the owner page. The interface page of the owner is mostly similar to the user. For example, the owner follows same registration process that was explained for the user. However, we would like to mention the view records and delete option pages which are the important option of the owner. Through this option, the owner can view uploaded files with delete option as shown in Figure 3.18.

Figure 2.9, Figure 2.10 and Figure 2.11 show us all options related to the Admin, owner and user pages respectively. And also all pages details which designed for our system can be found in Appendix A.

3.4 Description of the System

The application is mainly about a cloud-based system we can access it through this link (www.saharproject.com) which is mainly used for storing files in cloud-based server with smooth and fast accessibility. So the client can access through tablets or PCs which are connected to the cloud application across the globe.

We can simply summarize the description of our system as follows:

i. Anyone can register into the website as owner or user and access the service of our application via the link (www.saharproject.com).

ii. If clients register as owners they have services like uploading files and sharing the files. Moreover, an owner can monitor his data by using auditing mechansim.

iii. If clients register as users, they can access the list of files which are uploaded by the owner. The users can buy any files they want, by following the download instructions.

iv. The Admin will be supervising the application process so he/she can view all the records and delete them. And also he/she has the authority to delete the owners and users as well and also give permissions to specific owners or users.

3.5 Security Aspects

In our system we mainly concentrate on the security aspect of the cloud system. We ensured the privacy and the security of the client by applying three mechanisms:

-We added accountability whereby clients could access the system only through authorization.

-We added an auditing mechanism which distinguishes our system from the other existing one. In our advanced monitoring operation the owner can observe the user and the admin can observe both of them.

-We added an automatic session expiry feature. In this security mode the owner and the users are logged out from the system if they are not active for 30 seconds.

Chapter 4

THE PERFORMANCE OF THE SYSTEM

We conducted a number of experiments to test the performance of the implemented system. We ran clients' side of the program in our department's laboratory (CMPE, LAB 137). The wired internet and the server is located in INDIA with the following details:

Server location: INDIA, domain bought at "net4india"(www.net4.in)
Hosting space in the server: 4 GB
Server IP: 118.67.248.245, URL: http://saharproject.com.com
Database Details: MYSQL, Database Name: DB_saharproject
port number: 3306

4.1 Experiment Results

Our experiments focused on three points: Authentication time, uploading time, downloading and uploading time when the system is used by a number of clients at the same time.

The first point involves measuring the authentication time (logging time). The aim that stands behind this experiment is to examine the delay in the accessibility to the system. This experiment was conducted in CMPE research laboratory using wired Internet with PC's properties: Intel Duel core CPU @ 3.06 HZ, RAM 4.00GB (3.25 GB usable), and with 32 bit Windows 7. In this experiment we measured how much time it takes for a client to logging the system. The authentication was measured in different periods of a day (in the morning, at noon, in the afternoon, at night). Five measurements were taken to calculate the average for each period. The authentication time is calculated in the program through the use of current time function. Once the client clicks on the submit button after entering his/her name and password, the time calculation starts and continues until the end of checking the clients' information process. Then the time taken in this operation appears on the clients' page. The experiment gives the following results:

i. Morning experiment time result for authentication is taken as 0.0192 s.
ii. Noon experiment time for authentication is noted as 0.016 s.
iii. Afternoon experiment time for authentication is noted as 0.015 s.
iv. And night experiment time for authentication is recorded as 0.017 s.

From the experiment results it is noticeable that the authentication is taking some time and is changing during the different periods of the day. The Afternoon measurements show the shortest authentication time.

In the second group of experiments we measured uploading time under the same conditions. To achieve this aim we used JPEG files with different sizes ranging between 100 KB to 1000 KB. Each file was uploaded five times in four different periods of the day (morning, noon, afternoon and night) and then we took the average of five measurements. Here we have measured log creation time for different file sizes in the system. The measurement outcomes show that the uploading time gradually increases according to the size of the file; i.e. the bigger the file size the more time it takes to be uploaded. Measurement results for uploading time is given in Table 4.1 and presented in Figure 4.1.

Table 4.1. Measurement Results for Uploading Time in the Morning

Measurements at 08:00-09:00	
23/10/2013	
Files Size (KB)	**Log Creation Time(s)**
100	1.474
200	1.848
300	2.153
400	2.459
500	2.648
600	2.907
700	3.259
800	3.633
900	3.813
1000	4.3

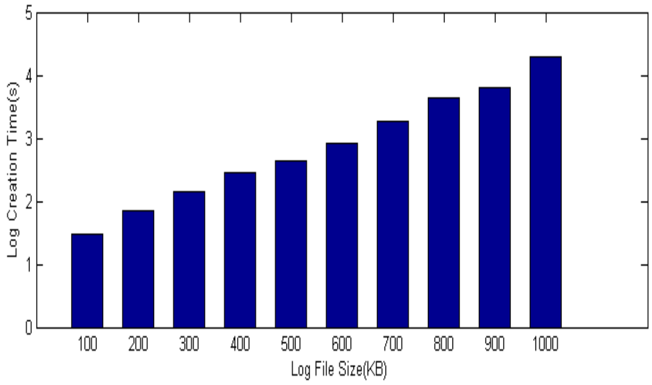


Figure 4.1. Measured Uploading Time in the Morning

We repeated the same experiment at noon time. Table 4.2 shows the results of the experiment and the results are presented in form of graph in Figure 4.2. Results show that the uploading time gradually increases according to the size of the file; i.e. the bigger file size the more time it takes to be uploaded (if we compare Table 4.1 and Table 4.2 we can see that there is no big difference in the log creation time in the morning and at noon time).

Table 4.2. Measurement Results for Uploading Time at Noon

Measurements at 12:00-13:00	
22/10/2013	
Files Size (KB)	**Log Creation Time(s)**
100	1.27
200	1.548
300	1.899
400	2.293
500	2.525
600	2.643
700	2.953
800	3.166
900	3.261
1000	3.667

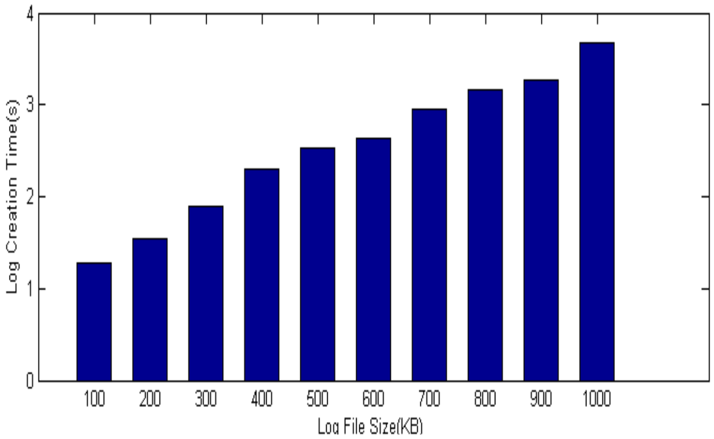


Figure 4.2. Measured Uploading Time at Noon

We also repeated same experiments in the afternoon time. Table 4.3 shows the outcomes of the experiment and the results are presented in form of graph in Figure 4.3. Results show that the uploading time gradually increases according to the size of the file; i.e. the bigger file size the more time it takes to be uploaded (if we compare Table 4.3 with the previous two tables (Table 4.1 and Table 4.2) we can see that there is no big difference in the log creation time in those periods (in the morning, at noon times and in the afternoon).

Table 4.3. Measurement Results for Uploading Time in the Afternoon

Measurements at 16:00-17:00 pm	
22/10/2013	
Files Size (KB)	**Log Creation Time(s)**
100	1.176
200	1.772
300	1.957
400	2.1
500	2.231
600	2.398
700	3.005
800	3.469
900	3.585
1000	3.755

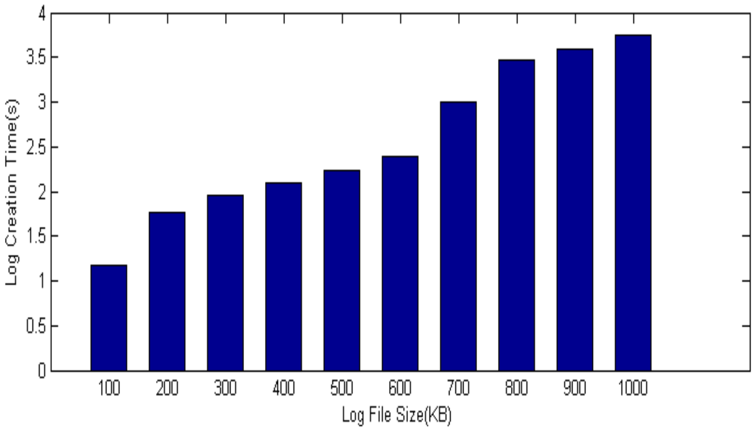


Figure 4.3. Measured Uploading Time in the Afternoon

Lastly, in the second group of experiments, we repeated the same experiment at night time. Table 4.4 shows the results of the experiments and the results are presented in form of graph in Figure 4.4. Results show that the uploading time gradually increases according to the size of the file; i.e. the bigger the file size the more time it takes to be uploaded (if we compare Table 4.4 with the previous three tables (Table 4.1, Table 4.2 and Table 4.3) we can see that at the night period the uploading time is the shortest compared to the other times.

Table 4.4. Measurement Results for Uploading Time at Night

Measurements at 10-11 pm	
22/10/2013	
Files Size (KB)	**Log Creation Time(s)**
100	1.208
200	1.523
300	1.74
400	2.115
500	2.371
600	2.512
700	2.678
800	2.989
900	3.413
1000	4.054

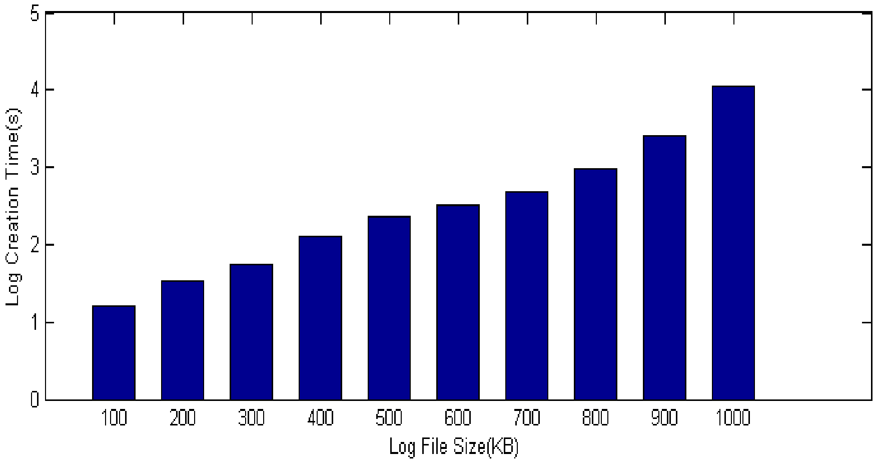


Figure 4.4. Measured Uploading Time at Night

Generally for second experiment we noticed that the best results of file uploading time are shown in the experiment that are conducted between afternoon and night periods as the uploading process time was the shortest.

Appendix B shows us the code used to obtain the previous results, and Figure 28 in appendix A includes details about the practical part of the experiment.

In the third group of experiments we tried to measure the uploading and downloading time when more than one client (either user or owner) are working at the same time. This experiment is conducted in CMPE LAB 137 using wired internet with PC's properties: Intel Duel core CPU @ 2.93 GHZ, RAM 4.00 GB (3.25 GB usable) and system type of 32 bit operating system. As shown in Figure 4.5. We measured the uploading time for four cases: when one owner is logged into the system, when there are five owners online, then when there are ten owners online and finally when there are fifteen owners active at the same time. In the experiments, each owner uploaded 100 KB JPEG file in the noon period of the day. This process was repeated five times and we took the average of five measurements for uploading time.

All measurement results are collected in Table 4.5 and Figure 4.5. From the results we can see the uploading time increases when the number of owners working in the system increases, because the server carries more of requests when the owners number are increasing.

Table 4.5. Measurement Results of Uploading Time for Different Number of Owners

Measurements at 12:00-13:00 pm	
23/10/2013	
No. of Owners	**Uploading Time(s)**
1	1.26
5	1.5194
10	1.6038
15	1.8418

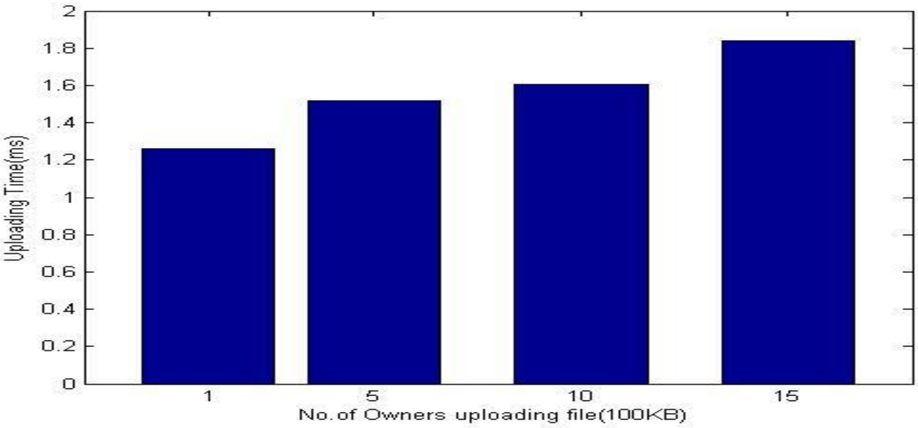


Figure 4.5. Uploading Time for Different Number of Owners with 100KB File Size

Finally, we measured the downloading time for the users in four cases: when one user is logged into the system, when there are five users online, then when there are ten users online and finally when there are fifteen users active at the same time. In the experiments, each user requests to download a 100 KB JPEG file at noon period of the day. This process was repeated five times and we took the average of five experiment results for downloading.

The measurements for downloading are shown in Table 4.6 and Figure 4.6. From the results we can see downloading time increases when the number of users working in the system increases, because the server and the database are carry more of requests when the users number are increasing.

Table 4.6. Measurement Results of Downloading Time for Different No. of Users

Measurements at 12:00-13:00 pm	
28/10/2013	
No. of users	**Request Downloading Time(s)**
1	0.013
5	0.0192
10	0.0326
15	0.0427

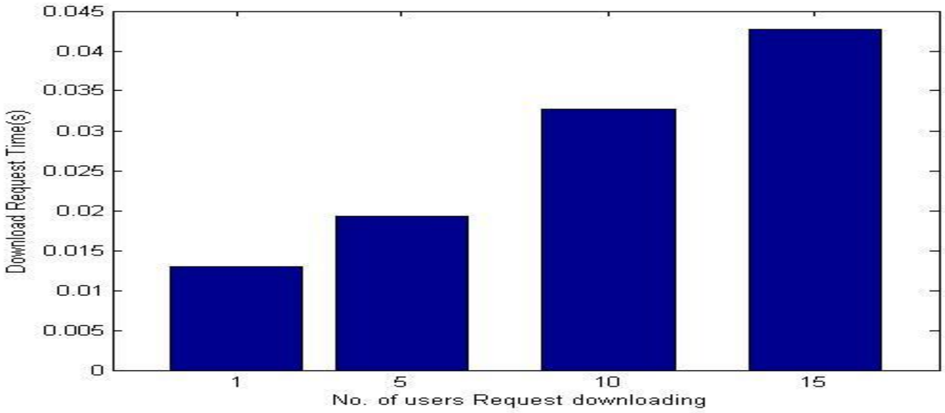


Figure 4.6. Downloading Time Versus Number of Users for 100KB File Size

We noticed from the previous results that the uploading time is more than the downloading time, because with uploading we need to do many checking processes such as checking if owner has the permission to upload, checking if there is a free place in the database then file will be inserted into the database. While in downloading, we calculate the time difference between receiving a response for a sent request, saving time not included. Downloading time includes checking if user has permission to download the file and to search file in the data file in database, so downloading time is smaller than uploading time.

4.2 Comparisons with Other Studies

This study is based on the paper by (Sundareswaran, Squicciarini, & Lin, 2012) which adopted the Linux-based system. Our system is developed for windows operating system. The distinctive characteristics of our system are:

i. We measured downloading time to test performance of the system from different aspects which is not done in our reference paper (Sundareswaran, Squicciarini, & Lin, 2012).

ii. According to approximate results of uploading time, which is gotten from reference paper, it has approved that our uploading time results are better than the refrence paper results as shown in Table 4.7 of the related figure (Figure 4.7) which is given below.

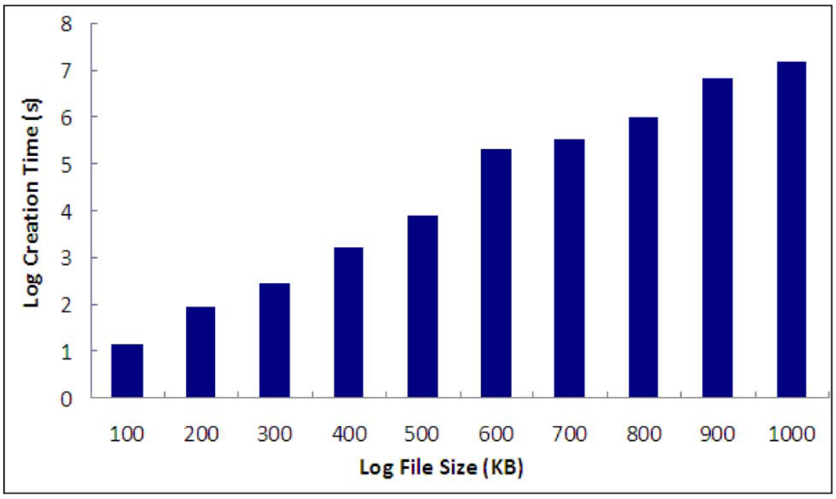


Figure 4.7. Time of Uploading for Different File Sizes (Sundareswaran, Squicciarini, & Lin, 2012)

Table 4.7. Comparison Uploading Time Result in Our Study with Other Study

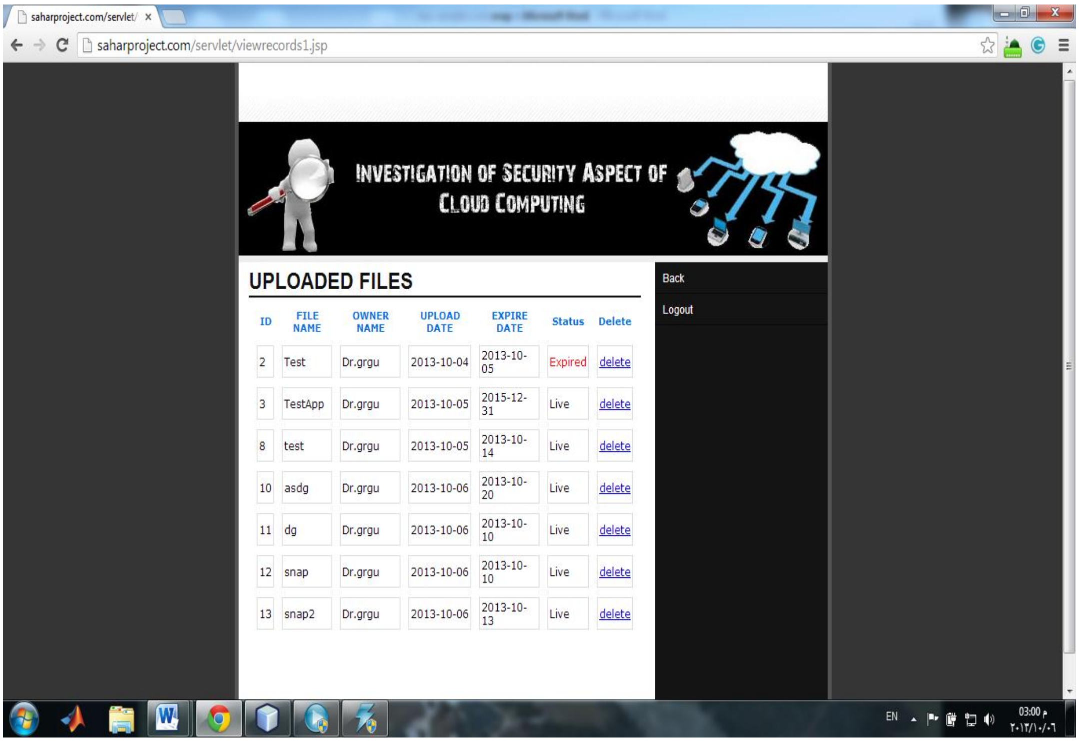
Approximate Results for (Sundareswaran, Squicciarini, & Lin, 2012)		**Our Study Results -Morning Period-**
Files Size (KB)	**Log Creation Time(s)**	**Log Creation Time(s)**
100	1.14	1.474
200	2	1.848
300	2.4	2.153
400	3.15	2.459
500	3.96	2.648
600	5.3	2.907
700	5.5	3.259
800	6	3.633
900	6.8	3.813
1000	7.2	4.3

Table 4.7 shows the longest uploading times which took place in the morning periods in our study. It also shows the approximate results obtained from the chart in our reference paper (Sundareswaran, Squicciarini, & Lin, 2012).

We can see that the longest times measured in our experiment are shorter and therefore better than the outcomes concluded in the chart in Figure 4.7.

iii. File validity

In our study we also give the owners the ability to define a time limit for the file validity. After the time limit restriction expires, the data becomes inaccessible for the users (see Figure 4.8).


saharproject.com/servlet/viewrecords1.jsp
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Back
Logout

UPLOADED FILES

ID	FILE NAME	OWNER NAME	UPLOAD DATE	EXPIRE DATE	Status	Delete
2	Test	Dr.grgu	2013-10-04	2013-10-05	Expired	delete
3	TestApp	Dr.grgu	2013-10-05	2015-12-31	Live	delete
8	test	Dr.grgu	2013-10-05	2013-10-14	Live	delete
10	asdg	Dr.grgu	2013-10-06	2013-10-20	Live	delete
11	dg	Dr.grgu	2013-10-06	2013-10-10	Live	delete
12	snap	Dr.grgu	2013-10-06	2013-10-10	Live	delete
13	snap2	Dr.grgu	2013-10-06	2013-10-13	Live	delete

Figure 4.8. File Validity

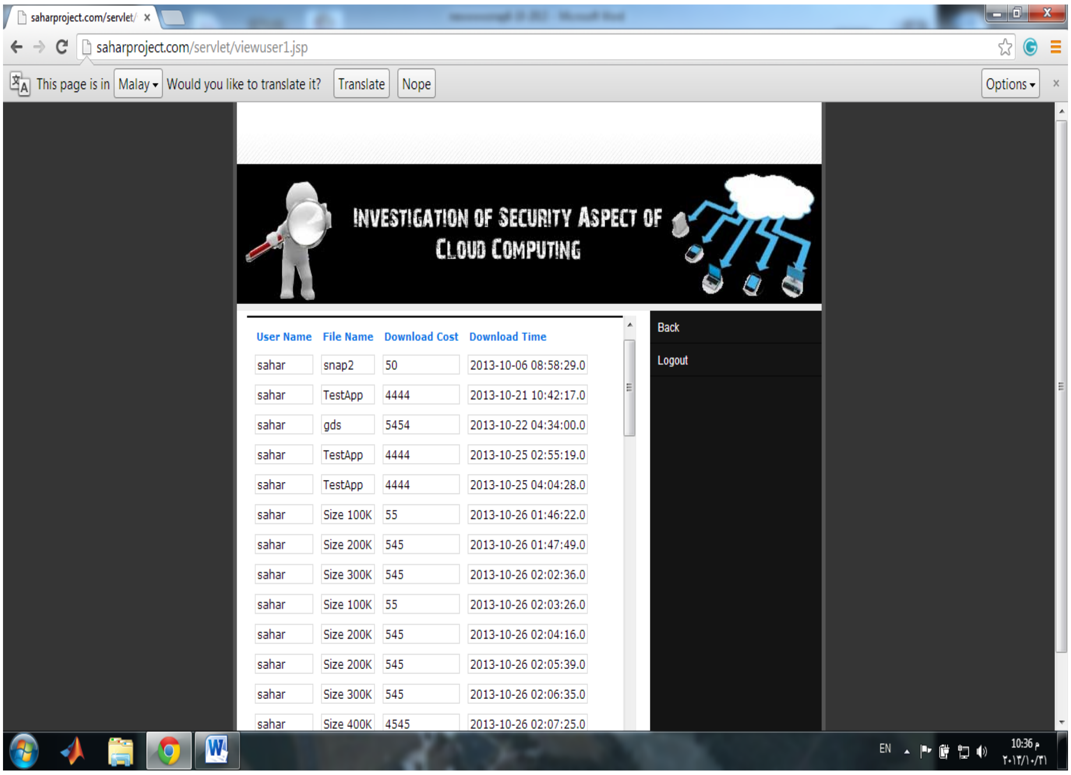
iv. Auditing Mechanism

In the reference paper the auditing mechanism is implemented by the owner, who can retrieve user actions according to the following: user name, user action, time and location.

However, in our study this mechanism is implemented by the owners and the admin components. Owners can retrieve user information and user actions through this record: user name, file name, download cost and download time) as shown in Figure 4.9.

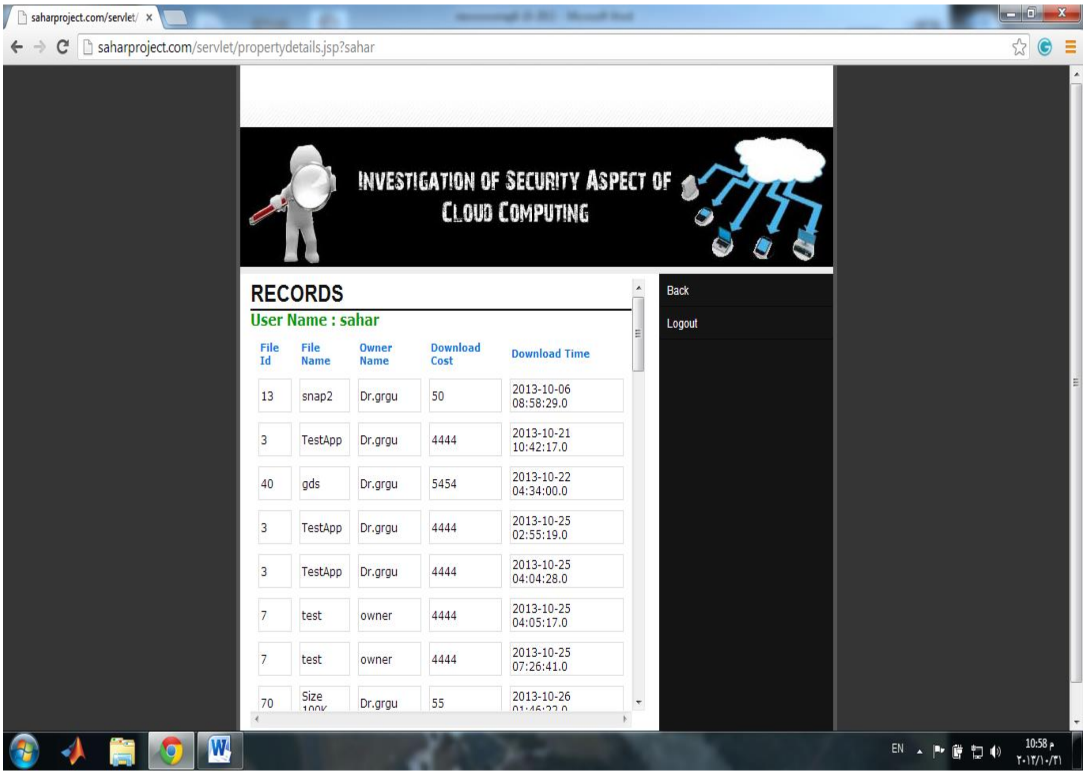
Secondly, the auditing mechanism is implemented by the admin as well. The admin can track both owners and users and can retrieve their action records at any time. In the reference paper, an admin based auditing mechanism was not mentioned. Figure 4.10 shows how the admin can choose the user name after clicking the auditing option in the admin page (see Figure A.37 in Appendix A) to retrieve the following user information: file name, file owner name, download cost and download time.

Finally, Figure 4.11 shows the option that we added on the admin page giving the admin the ability to retrieve the owner actions according to this record: owner name, file name, upload date, expire date and status.


saharproject.com/servlet/
saharproject.com/servlet/viewuser1.jsp
This page is in Malay Would you like to translate it? Translate Nope
Options
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Back
Logout
EN
10:36

User Name	File Name	Download Cost	Download Time
sahar	snap2	50	2013-10-06 08:58:29.0
sahar	TestApp	4444	2013-10-21 10:42:17.0
sahar	gds	5454	2013-10-22 04:34:00.0
sahar	TestApp	4444	2013-10-25 02:55:19.0
sahar	TestApp	4444	2013-10-25 04:04:28.0
sahar	Size 100K	55	2013-10-26 01:46:22.0
sahar	Size 200K	545	2013-10-26 01:47:49.0
sahar	Size 300K	545	2013-10-26 02:02:36.0
sahar	Size 100K	55	2013-10-26 02:03:26.0
sahar	Size 200K	545	2013-10-26 02:04:16.0
sahar	Size 200K	545	2013-10-26 02:05:39.0
sahar	Size 300K	545	2013-10-26 02:06:35.0
sahar	Size 400K	4545	2013-10-26 02:07:25.0

Figure 4.9. Auditing by the Owner Page

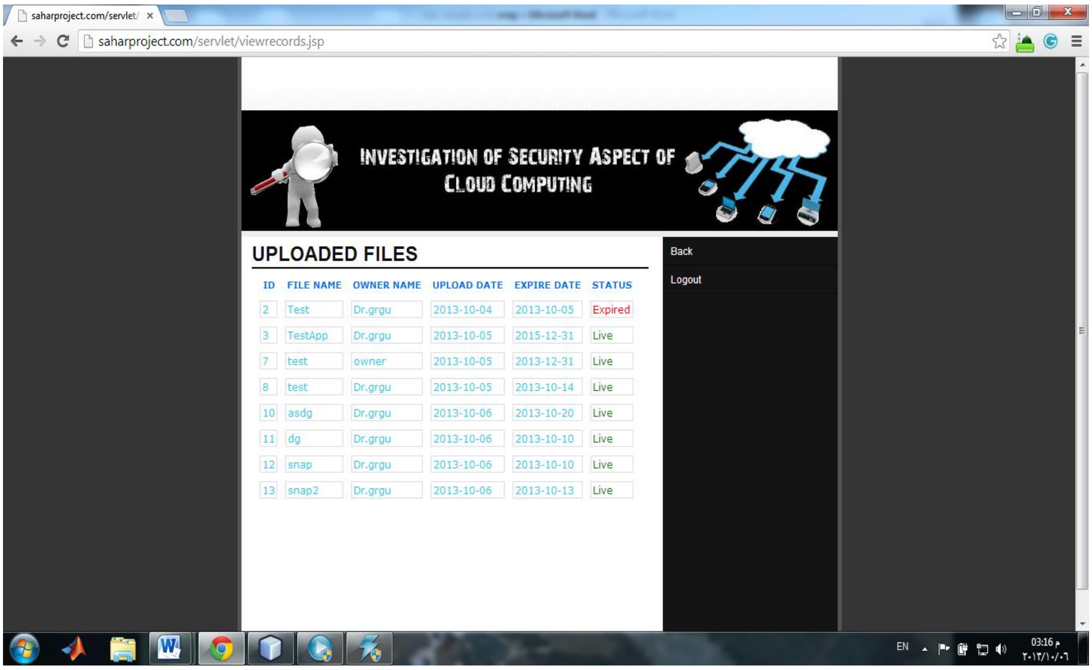

saharproject.com/servlet/
saharproject.com/servlet/propertydetails.jsp?sahar
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Back
Logout

RECORDS

User Name : sahar

File Id	File Name	Owner Name	Download Cost	Download Time
13	snap2	Dr.grgu	50	2013-10-06 08:58:29.0
3	TestApp	Dr.grgu	4444	2013-10-21 10:42:17.0
40	gds	Dr.grgu	5454	2013-10-22 04:34:00.0
3	TestApp	Dr.grgu	4444	2013-10-25 02:55:19.0
3	TestApp	Dr.grgu	4444	2013-10-25 04:04:28.0
7	test	owner	4444	2013-10-25 04:05:17.0
7	test	owner	4444	2013-10-25 07:26:41.0
70	Size 100K	Dr.grgu	55	2013-10-26 01:46:22.0

Figure 4.10. Auditing by the Admin Page to Monitor the User


saharproject.com/servlet/
saharproject.com/servlet/viewrecords.jsp
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Back
Logout

UPLOADED FILES

ID	FILE NAME	OWNER NAME	UPLOAD DATE	EXPIRE DATE	STATUS
2	Test	Dr.grgu	2013-10-04	2013-10-05	Expired
3	TestApp	Dr.grgu	2013-10-05	2015-12-31	Live
7	test	owner	2013-10-05	2013-12-31	Live
8	test	Dr.grgu	2013-10-05	2013-10-14	Live
10	asdg	Dr.grgu	2013-10-06	2013-10-20	Live
11	dg	Dr.grgu	2013-10-06	2013-10-10	Live
12	snap	Dr.grgu	2013-10-06	2013-10-10	Live
13	snap2	Dr.grgu	2013-10-06	2013-10-13	Live

Figure 4.11. Auditing by the Admin Page to Monitor the Owner

Chapter 5

CONCLUSION

This study focused on the security aspect of the cloud computing system. It aims to ensure more secure usage of data by providing a supplementary security mode in the application of the system. So we concentrated on three security aspects in this study: accountability, auditing mechanism and automatic session expire technique. We designed a web application which involves three parties: admin, owner and user. It reinforces data security over cloud through the following mechanisms:

It requires accountability for data usage. Only legally registered clients are legible to access stored data. Moreover, an advanced auditing mechanism is implemented, in which the owner can monitor the data while the admin can observe and control all the system. Furthermore, we included an expire session technique to improve the security of the application and added a file validity property, which provides owners further control over their data.

We conducted a number of experiments with our system. These experiments focused on three points: authentication time, uploading time and downloading and uploading time with more than one client at the same time.

From the results we can say that our system performance is more efficient than the system performance mentioned in the reference paper.

The results show that the interaction process in our system lasts a shorter time in relation to authentication, uploading and downloading processes. For further studies, we recommend more focus on strengthening the capacity of server storage with proper certification for accessing data. In addition, further research is needed to adapt cloud computing applications to mobile-based systems.

REFERENCES

Ali, M. U., & Ayub, R. (2012). Cloud Computing as a Tool to Secure and Manage. *School of Computing Blekinge Institute of Technology 371 79 Karlskrona Sweden*, 1 - 57.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., et al. (2009). *Above the Clouds: A Berkeley View of Cloud Computing.* Berkeley: No.UCB/EECS-2009-28 http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html.

Ateniese, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z., et al. (2007). Provable Data Possession at Untrusted Stores. *CCS'07, October 29–November 2, 2007, Alexandria, Virginia, USA*, 598 - 610.

Buneman, P., Chapman, A. P., & Cheney, J. (2006). Provenance Management in Curated Databases. *SIGMOD 2006, June 27–29, 2006, Chicago, Illinois, USA.*

Cao, N. (2012). Secure & Reliable Data Outsourcing in Cloud Computing. *Worcester polytechnic institute In partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Electrical and Computer Engineering*, 1-123.

Corin, R., Etalle, S., Hartog, J. d., Lenzini, G., & Staicu, I. (2005). A Logic for Auditing Accountability in Decentralized Systems. *Dep. of Computer Science, University of Twente, The Netherlands,CWI, Center for Mathematics and Computer Science Amsterdam*, 1 - 17.

Cotney, b. A. (2012). *Armedia Blog.* Retrieved 9, 6, 2013, from « The New Data Visualization: This is Not Your Father’s Pie Chart: http://www.armedia.com/blog/2012/03/federal-cloud-computing-challenges-part-1-cloud-deployment-models/

Donkena, K., & Gannamani, S. (2012). Performance Evaluation of Cloud Database and. *School of Computing Blekinge Institute of Technology 371 79 Karlskrona Sweden*, 1 - 47.

Hexistor. (2012). *Hexistor data protection services.* Retrieved 9, 6, 2012, from Hexistor: http://www.hexistor.com/virtualization/hybrid-data-protection/

Jacobo, R. (2012). Security in the Cloud: The threat of coexist with an unknown tenant on a public environment. *Submitted as part of the requirements for the award of the MSc in Information Security at Royal Holloway, University of London*, 1 - 71.

Jadeja, Y., & Modi, K. (2012). Cloud Computing - Concepts, Architecture and. *IEEE Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on*, 877- 880.

Jaeger, P. T., Lin, J., & Grimes, J. M. (2008). Cloud Computing and Information Policy:Computing in a Policy Cloud. *www.chinacloud.cn/upload/2009-04/temp_09043010187252.pdf*, 1 - 35.

Jagadeesan, R., Jeffrey, A., Pitcher, C., & Riely, J. (2009). Towards a theory of accountability and audit. *published in ESORICS2009,Supported by NSF Career 0347542*, 1 - 18.

Lee, J.-H., Park, M.-W., Eom, J.-H., & Chung, T.-M. (2011). Multi-level Intrusion Detection System and Log Management in Cloud Computing. *IEEE Advanced Communication Technology (ICACT), 2011 13th International Conference*, 552 - 555.

Liu, W. (2012). Research on Cloud Computing Security Problem and Strategy. *Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on*, 1216 - 1219.

Lo, C.-C., Huang, C.-C., & Ku, J. (2010). A Cooperative Intrusion Detection System Framework for. *IEEE Parallel Processing Workshops (ICPPW), 2010 39th International Conference on Digital Object Identifier: 10.1109/ICPPW.2010.46*, 280 - 284.

Mell, P., & Grance, T. (2011). *Reports on Computer Systems Technology.* Gaithersburg, MD 20899-8930: National Institute of Standards and Technology Special Publication 800-145 (Draft).

MOLLET, N. G. (2011). Cloud Computing Security. *Helsinki Metropolia University of Applied Sciences*, 34 pages.

Nussbaum, C. (2012). *Cloud Deployment Models.* Retrieved 9 6, 2013, from AtomRain: http://www.atomrain.com/it/technology/cloud-deployment-models

Nussbaum, C. (2012). *Dissecting the Cloud IV – Community Clouds.* Retrieved.96,2013,-from-Atomrain: http://www.atomrain.com/it/technology/dissecting-cloud-iv-community-clouds

Paladi, N.(2012). Trusted Computing and Secure Virtualization in Cloud Computing. 1 - 64.

Patidar, S., Rane, D., & Jain, P. (2012). A Survey Paper on Cloud Computing. *IEEE Advanced Computing & Communication Technologies (ACCT), 2012 Second International Conference on*, 394 - 398.

Pearson, S., & Charlesworth, A. (2009). *Accountability as a Way Forward for Privacy Protection in the Cloud.* HP Labs, Long Down Avenue, Stoke Gifford, Bristol, UK. BS34 8QZ: HPL-2009-178 To be appeared in Proc. CloudCom 2009, Beijing, Springer LNCS, December 2009.

Pearson, S., Shen, Y., & Mowbray, M. (2009). A Privacy Manager for Cloud Computing. *HP Labs, Long Down Avenue, Stoke Gifford, Bristol BS34 8QZ, UK.*

Salah, K., Alcaraz-Calero, J.M., Zeadally, S., Almulla, S., & Alzaabi, M. (2011). Using Cloud Computing to Implement a Security Overlay Network. *Security & Privacy, IEEE Volume:11, Issue:1,Digital Object Identifier: 10.1109/MSP.2012.88*, 44 - 53.

Saleem, R. (2011). Cloud Computing's Effect on Enterprises. *School of Economics Management,* 1 - 89.

Selvarani, M., & Sadhasivam, D. S. (2010). Improed Cost-Based algorthm for task scheduling in. *Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference on*, 1 - 5.

Sharma, S., Soni, S., & Sengar, S. (2012). Security in Cloud Computing. *National Conference on Security Issues in Network Technologies (NCSI-2012),* 1 - 6.

Sundareswaran, S., Squicciarini, A. C., & Lin, D. (2012). Ensuring Distributed Accountability for Data Sharing in the Cloud. *IEEE Transactions on dependable and secure computing, VOL. 9, NO. 4, July/August 2012*, 555 - 567.

Tsai, C.-L., Lin, U.-C., Chang, A. Y., & Chen, C.-J. (2010). Information Security Issue of Enterprises Adopting. *Networked Computing and Advanced Information Management (NCM), 2010 Sixth International Conference*, 645 - 649.

V., R. S., & Jagadeswari, M. (2013). A Comprehensive Security Model for Image Storage in Cloud. *International Journal of Computer Trends and Technology- volume4Issue3- 2013*, 387 - 390.

Wang, C., Wang, Q., Ren, K., & Lou, W. (2010). Privacy-Preserving Public Auditing for Data Storage. *INFOCOM, 2010 Proceedings IEEE Digital Object Identifier: 10.1109/INFCOM.2010.5462173*, 1 - 9.

Wang, Q., Wang, C., Li, J., Ren, K., & Lou, W. (2009). Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing. *Illinois Institute of Technology, Chicago IL 60616, USA,Worcester Polytechnic Institute, Worcester MA 01609, USA*, 1 -20.

Zargar, S. T., Takabi, H., & Joshi, J. B. (2011). DCDIDP: A Distributed, Collaborative, and Data-driven Intrusion Detection and Prevention Framework for Cloud Computing Environments. *IEEE Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2011 7th International Conference* (pp. 332 - 341). USA: School of Information Sciences University of Pittsburgh Pittsburgh, PA, USA.

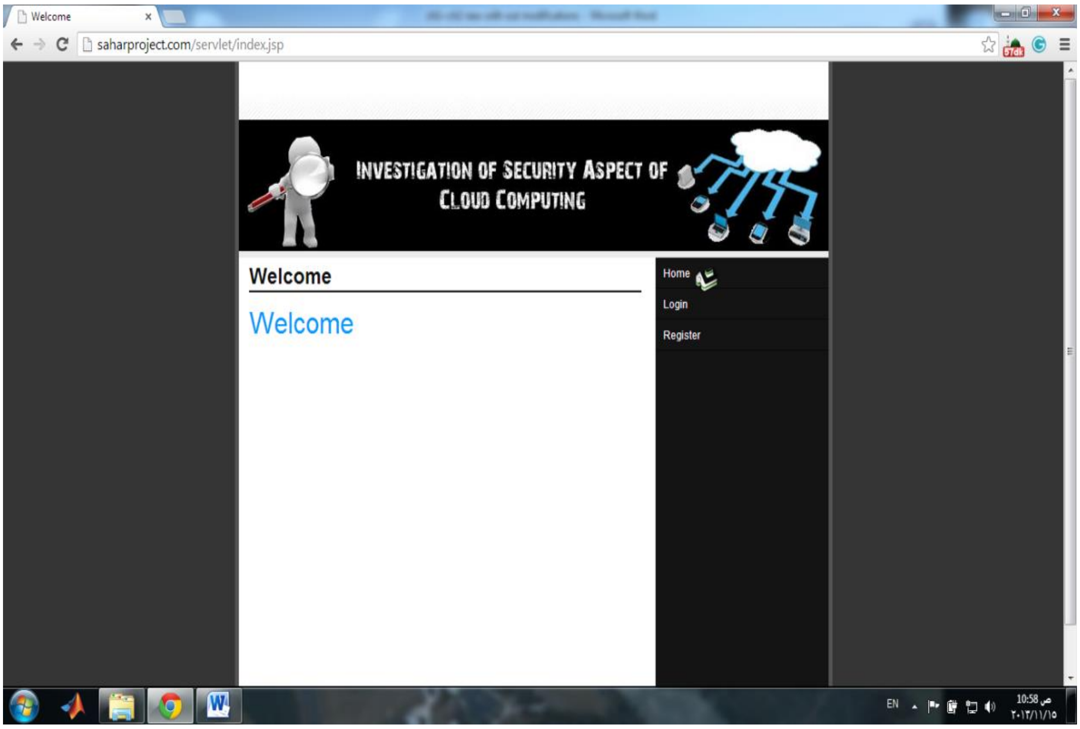
Zhang, Z., & Zhang, X. (2009). Realization of Open Cloud Computing Federation. *Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on*, 642 - 646.

APPENDICES

Appendix A: Figures which Describing All Pages of the System

Below, figures are describing all pages in our Application implementation:

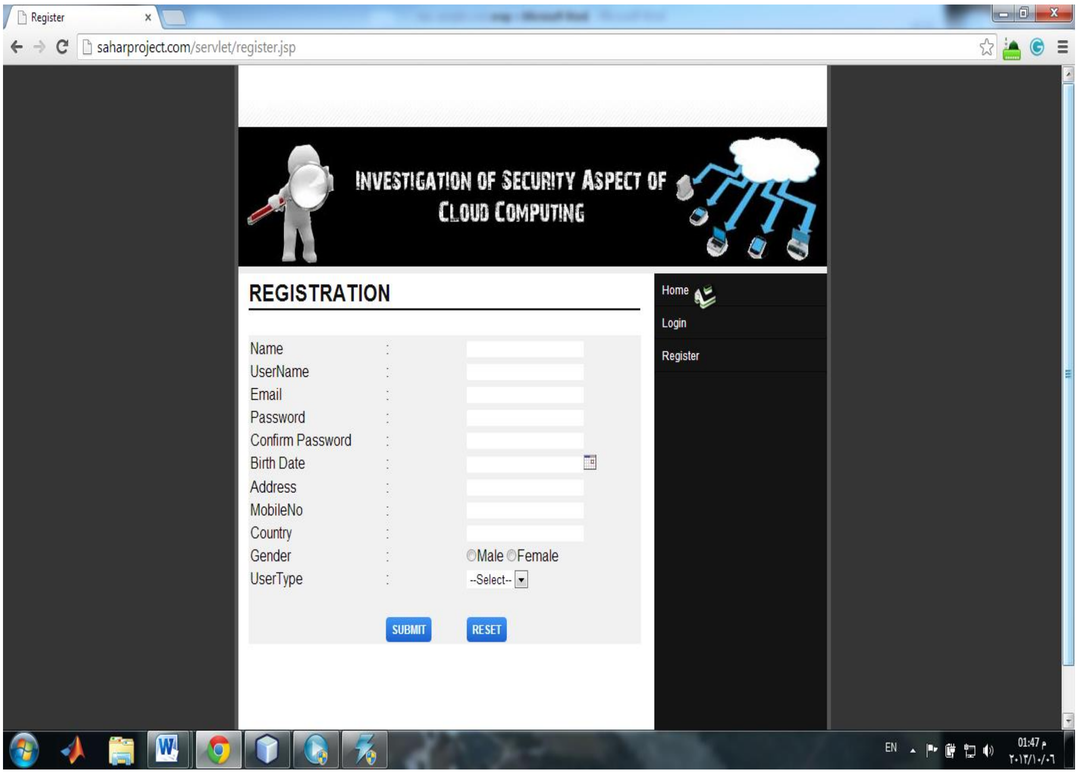
A.1. Home page


Welcome
saharproject.com/servlet/index.jsp
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Home
Login
Register

Welcome

Welcome

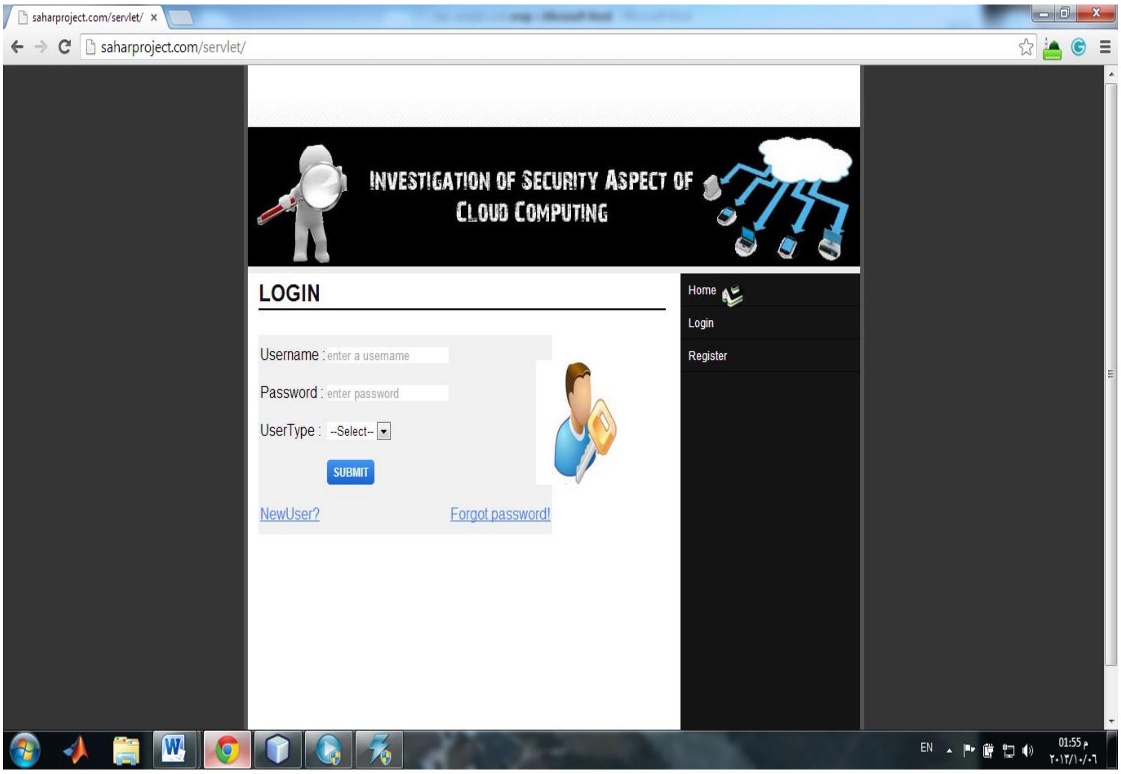
A.2. Registration page


Register
saharproject.com/servlet/register.jsp
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Home
Login
Register
SUBMIT
RESET

REGISTRATION

Name	:	
UserName	:	
Email	:	
Password	:	
Confirm Password	:	
Birth Date	:	
Address	:	
MobileNo	:	
Country	:	
Gender	:	○Male ○Female
UserType	:	--Select--

A.3. Login Page


saharproject.com/servlet/
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Home
Login
Register

LOGIN

Username : enter a username

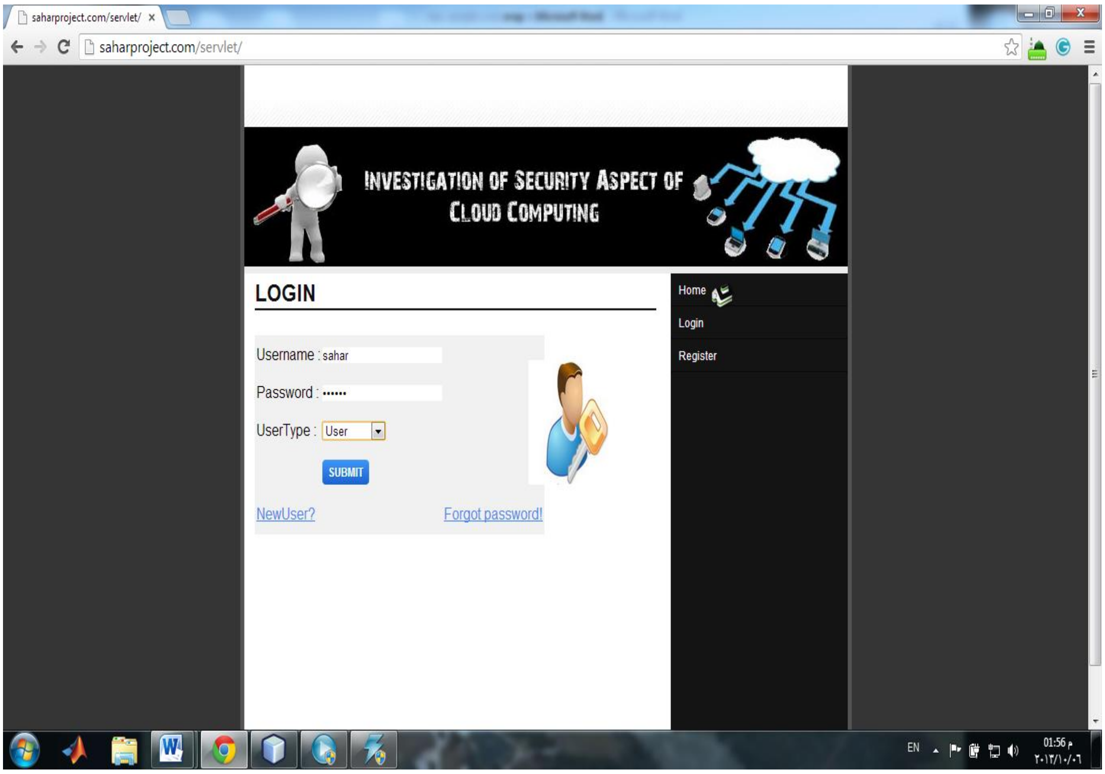
Password : enter password

UserType : --Select--

SUBMIT

NewUser? Forgot password!

A.4. User Login Page:


saharproject.com/servlet/
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Home
Login
Register

LOGIN

Username : sahar

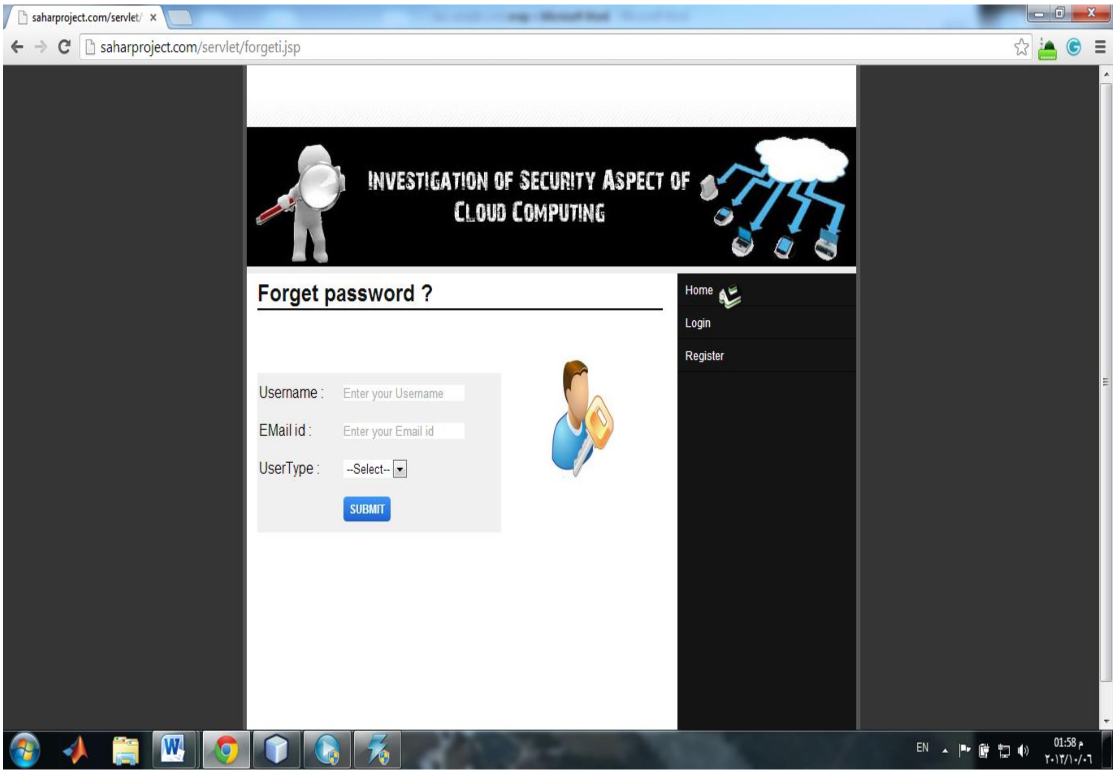
Password : ••••••

UserType : User

SUBMIT

NewUser? Forgot password!

A.5. Forget password page:



saharproject.com/servlet/forgeti.jsp

INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING

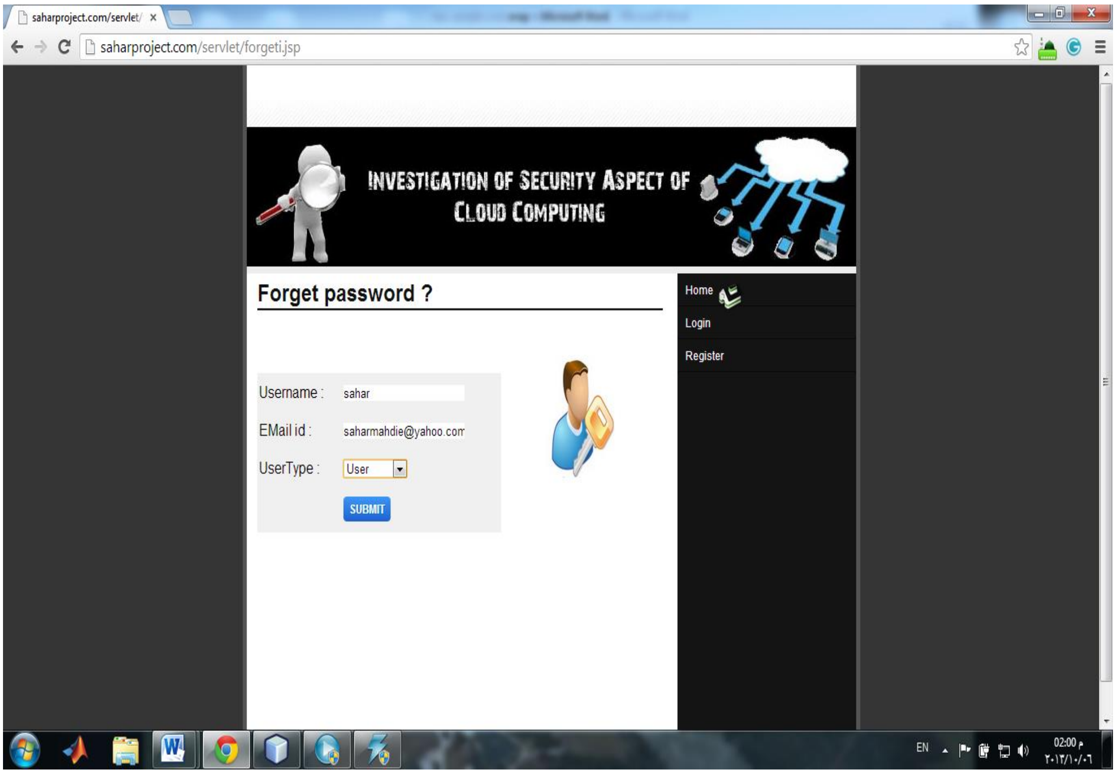
Forget password ?

Username : Enter your Username

EMail id : Enter your Email id

UserType : --Select--

A.6. Forget password details :



saharproject.com/servlet/forgeti.jsp

INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING

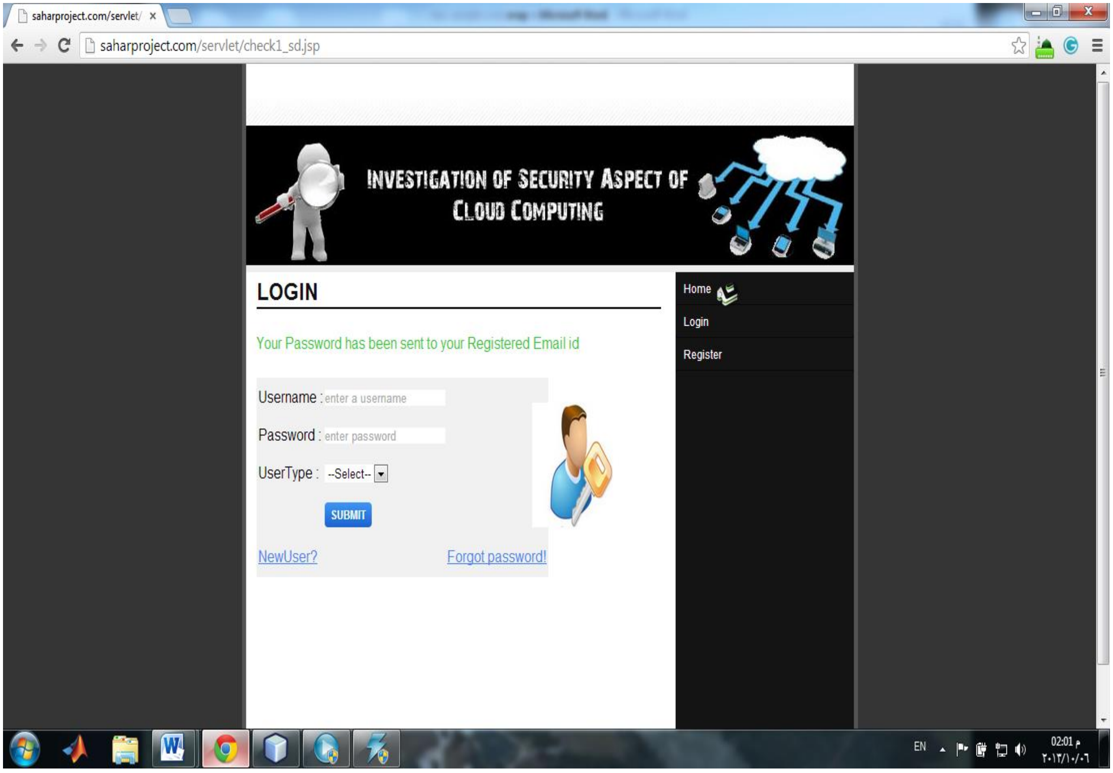
Forget password ?

Username : sahar

EMail id : saharmahdie@yahoo.com

UserType : User

A.7. Last Page for Forget password details:



saharproject.com/servlet/check1_sd.jsp

INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING

LOGIN

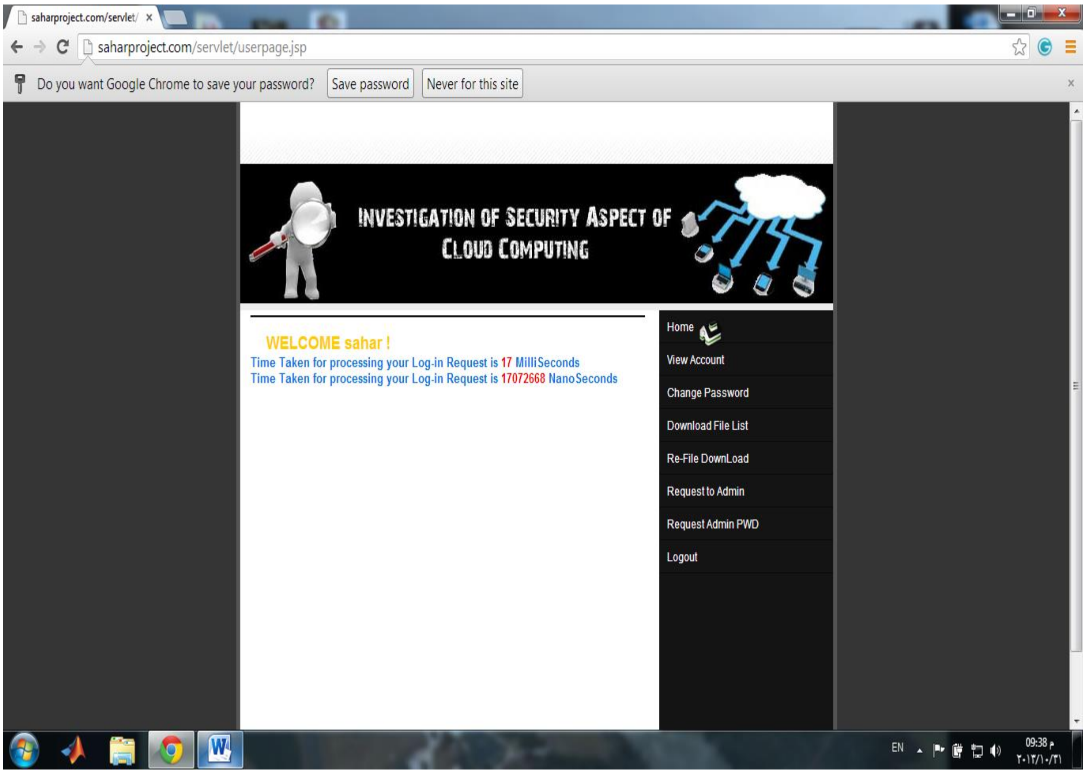
Your Password has been sent to your Registered Email id

Username : enter a username

Password : enter password

UserType : --Select--

A.8. User Welcome Page:



saharproject.com/servlet/userpage.jsp

Do you want Google Chrome to save your password?

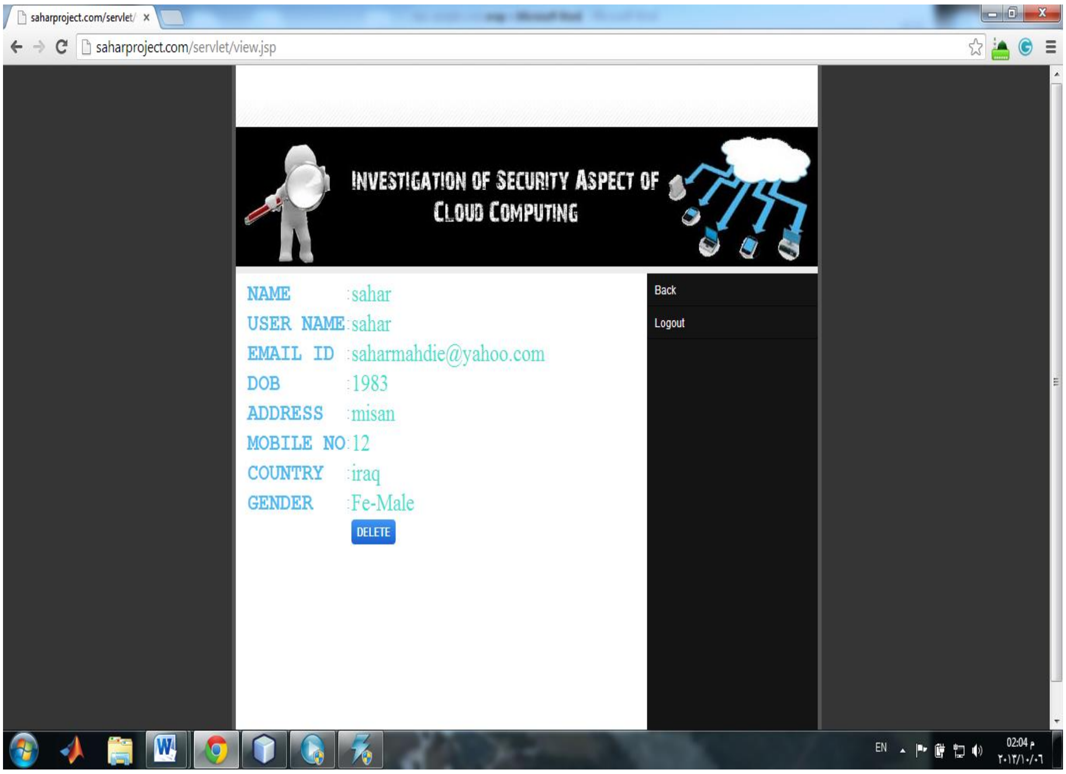
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING

WELCOME sahar !

Time Taken for processing your Log-in Request is 17 MilliSeconds

Time Taken for processing your Log-in Request is 17072668 NanoSeconds

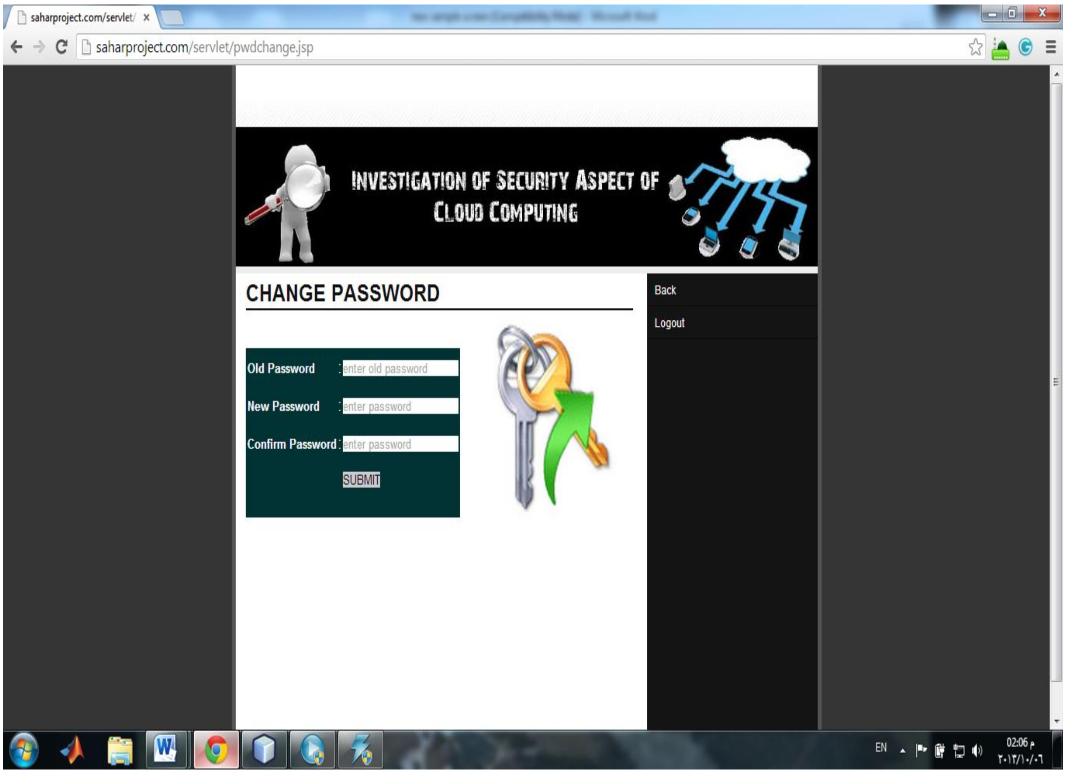
A.9. View Account Page:



saharproject.com/servlet/
saharproject.com/servlet/view.jsp
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Back
Logout
DELETE
EN
02:04 م
٢٠١٣/١٠/٠٦

NAME :sahar
USER NAME:sahar
EMAIL ID :saharmahdie@yahoo.com
DOB :1983
ADDRESS :misan
MOBILE NO:12
COUNTRY :iraq
GENDER :Fe-Male

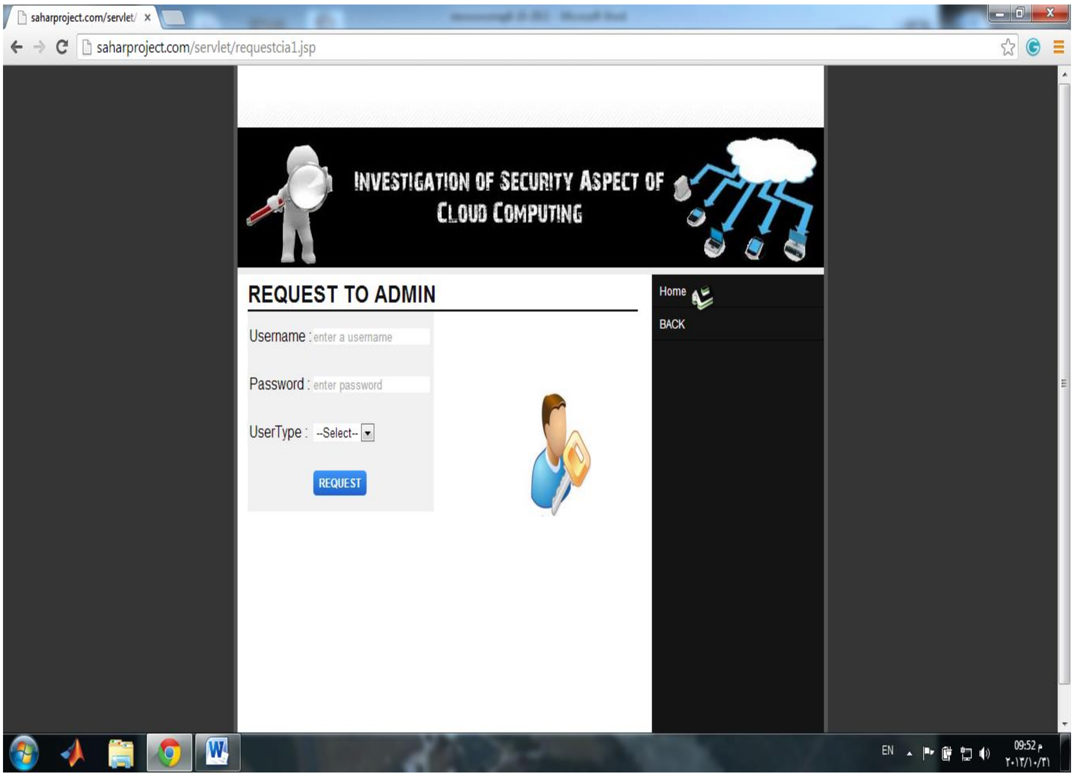
A.10. Change Password Page



saharproject.com/servlet/
saharproject.com/servlet/pwdchange.jsp
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
CHANGE PASSWORD
Back
Logout
SUBMIT
EN
02:06 م
٢٠١٣/١٠/٠٦

Old Password :enter old password
New Password :enter password
Confirm Password :enter password

A.11. Send request by user to get Graunt from ADMIN for view filelist and download purpose:


saharproject.com/servlet/requestcia1.jsp
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Home
BACK
REQUEST
EN

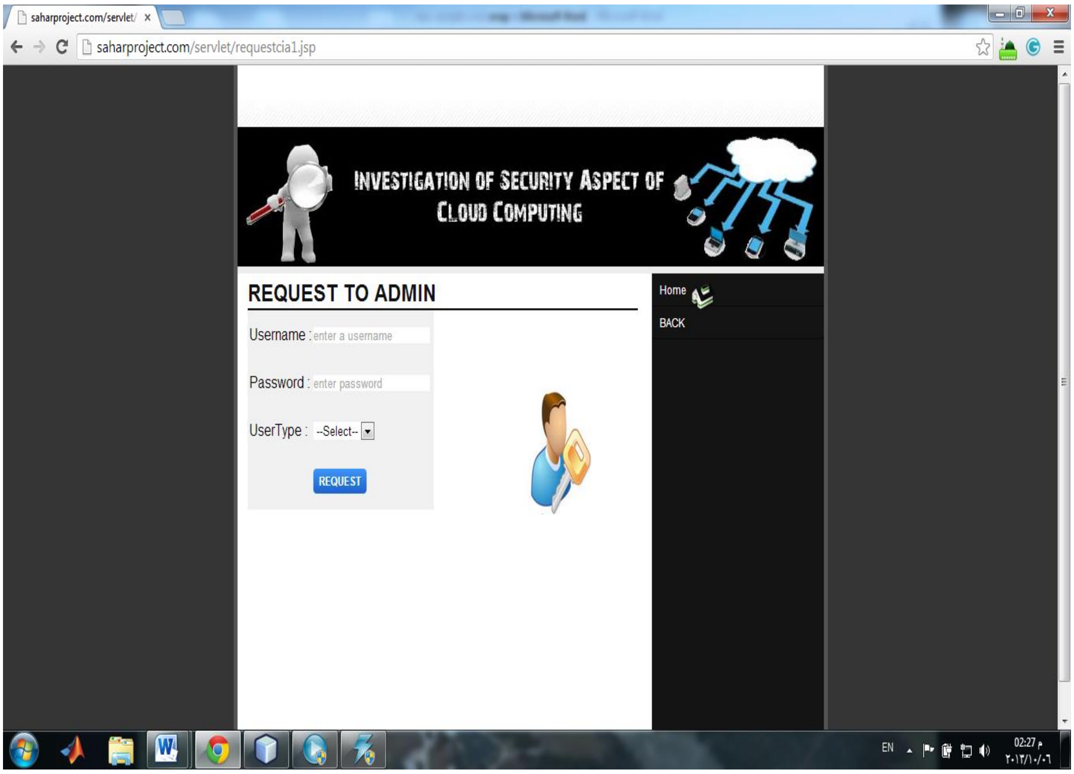
REQUEST TO ADMIN

Username : enter a username

Password : enter password

UserType : --Select--

A.12. User ADMIN request Password page:


saharproject.com/servlet/requestcia1.jsp
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Home
BACK
REQUEST
EN

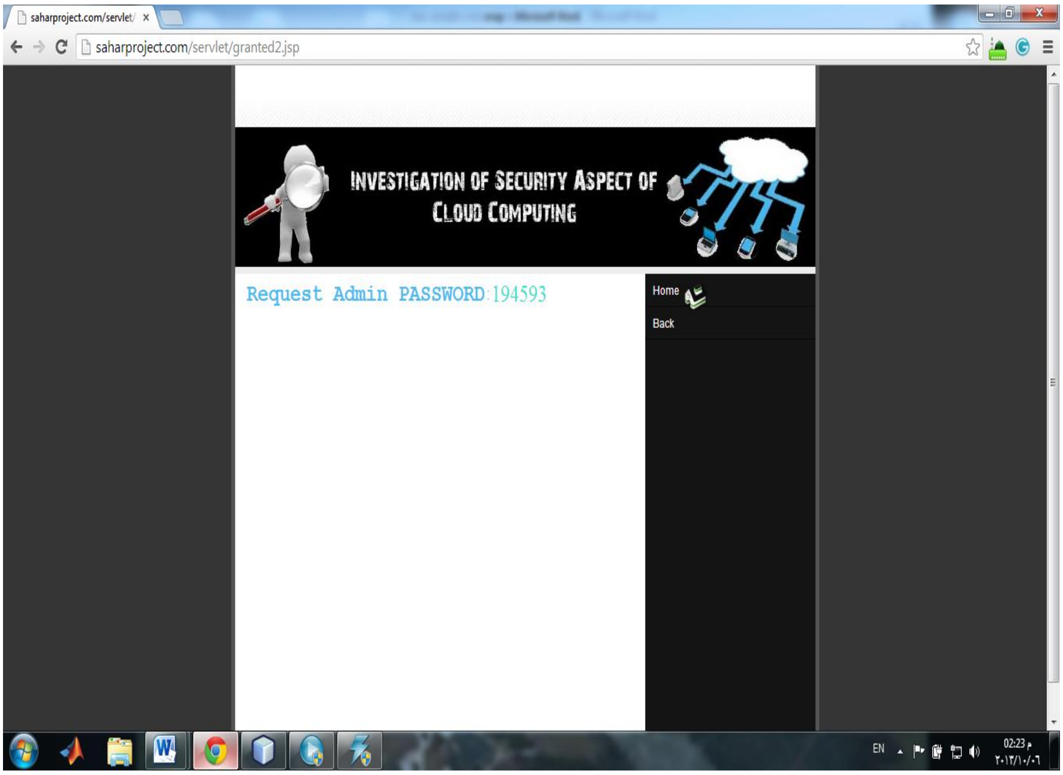
REQUEST TO ADMIN

Username : enter a username

Password : enter password

UserType : --Select--

A.13. User ADMIN Request Password Replay Page:

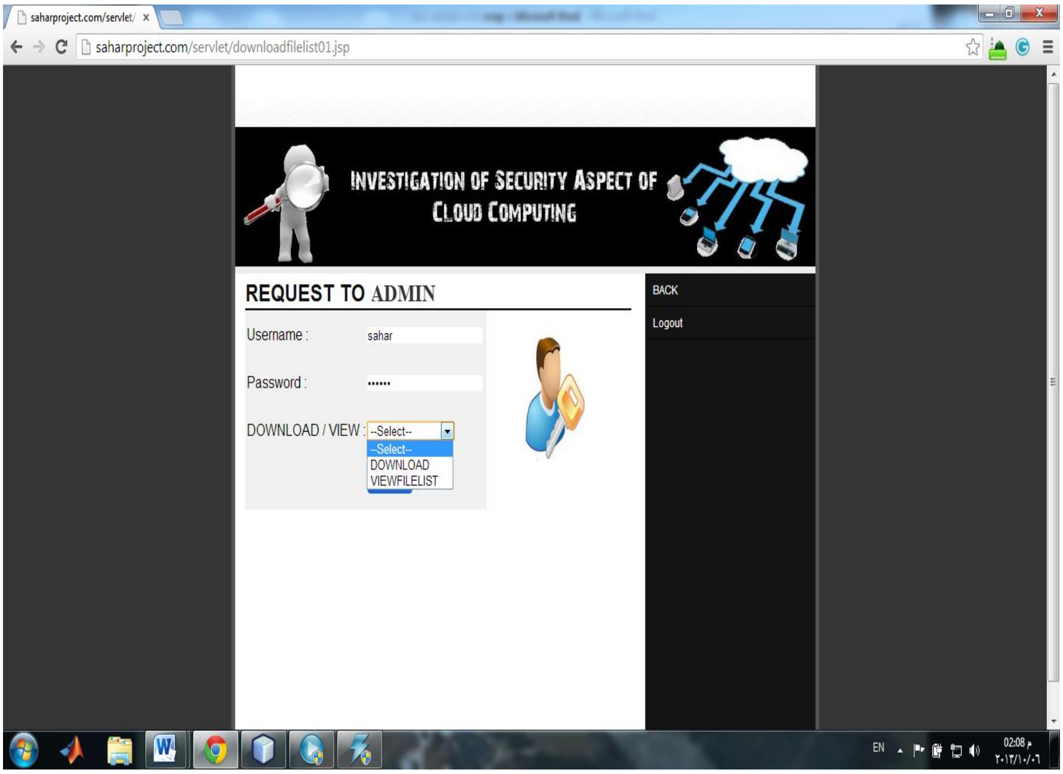


saharproject.com/servlet/granted2.jsp

INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING

Request Admin PASSWORD:194593

A.14. Send Request to ADMIN for View File list and Download:



saharproject.com/servlet/downloadfilelist01.jsp

INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING

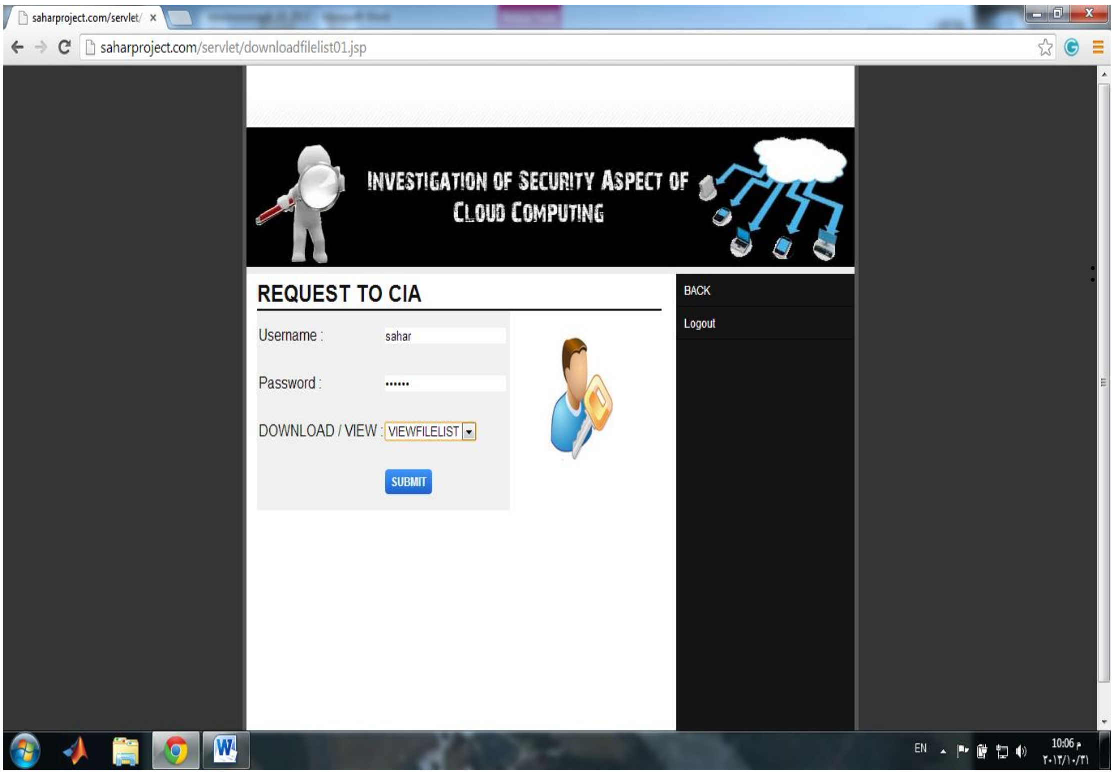
REQUEST TO ADMIN

Username : sahar

Password : ••••••

DOWNLOAD / VIEW : --Select--

A.15. View File List Option by User:



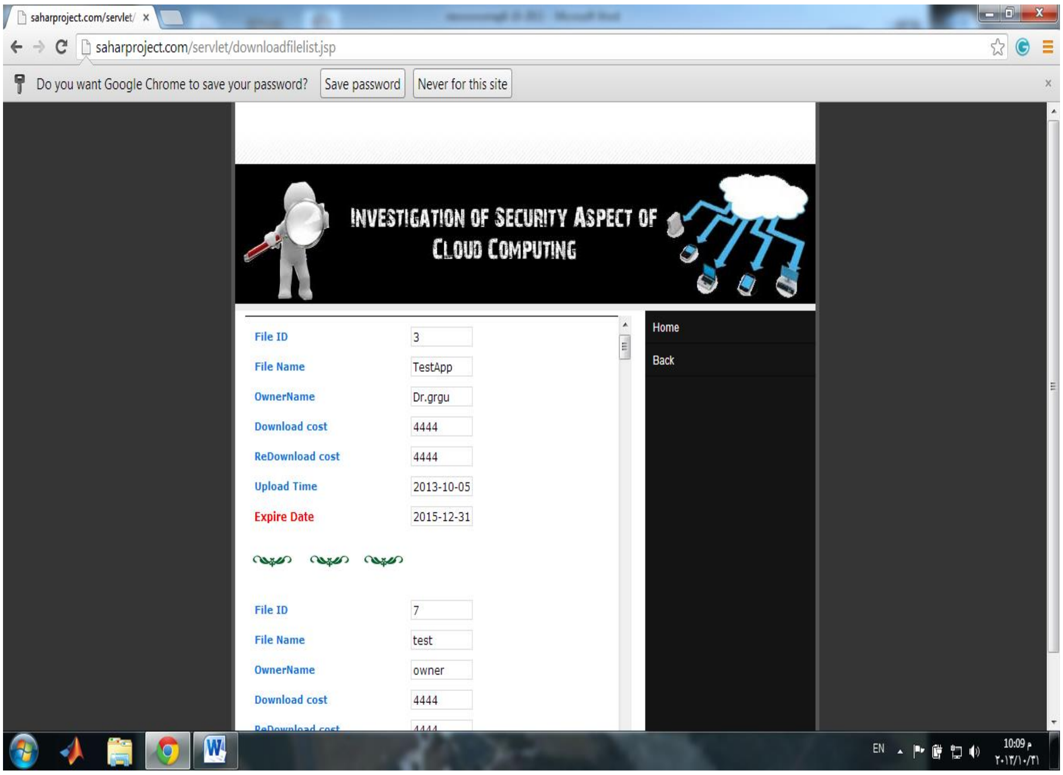
saharproject.com/servlet/downloadfilelist01.jsp

INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING

REQUEST TO CIA

Username :	sahar
Password :	••••••
DOWNLOAD / VIEW :	VIEWFILELIST

A.16. View File List Details:



saharproject.com/servlet/downloadfilelist.jsp

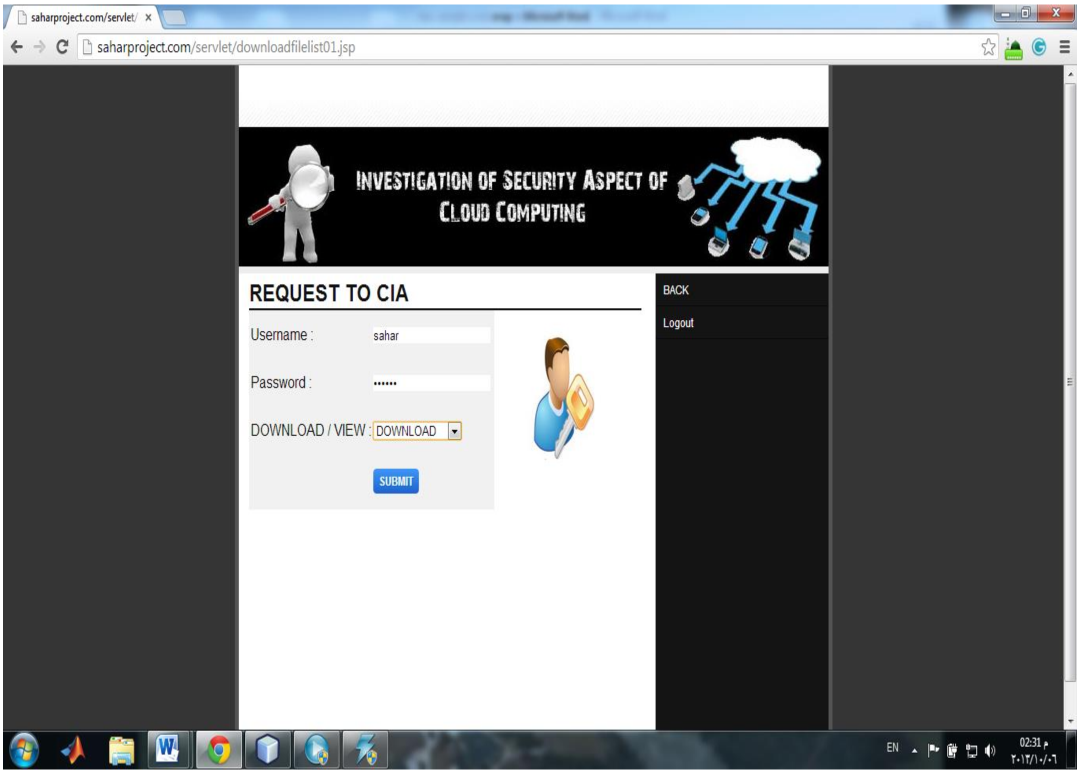
Do you want Google Chrome to save your password? Save password Never for this site

INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING

File ID	3
File Name	TestApp
OwnerName	Dr.grgu
Download cost	4444
ReDownload cost	4444
Upload Time	2013-10-05
Expire Date	2015-12-31

File ID	7
File Name	test
OwnerName	owner
Download cost	4444
ReDownload cost	4444

A.17. File Download:



saharproject.com/servlet/downloadfilelist01.jsp

INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING

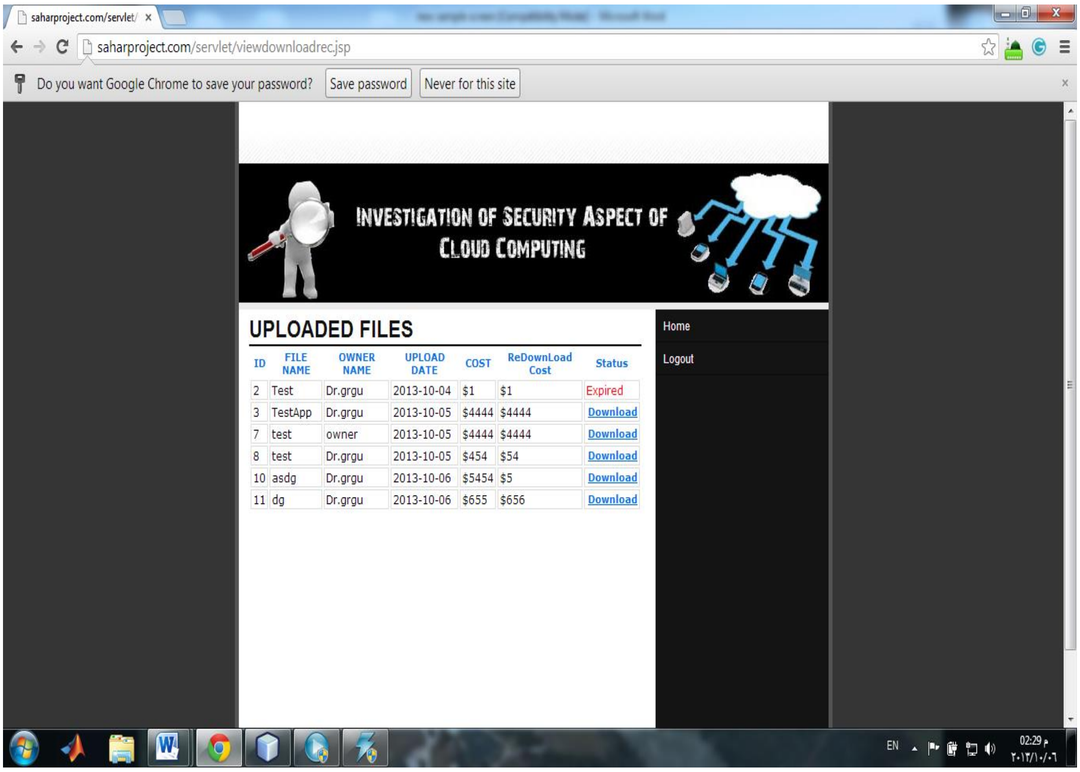
REQUEST TO CIA

Username : sahar

Password : ••••••

DOWNLOAD / VIEW : DOWNLOAD

A.18. File Download Details:



saharproject.com/servlet/viewdownloadrec.jsp

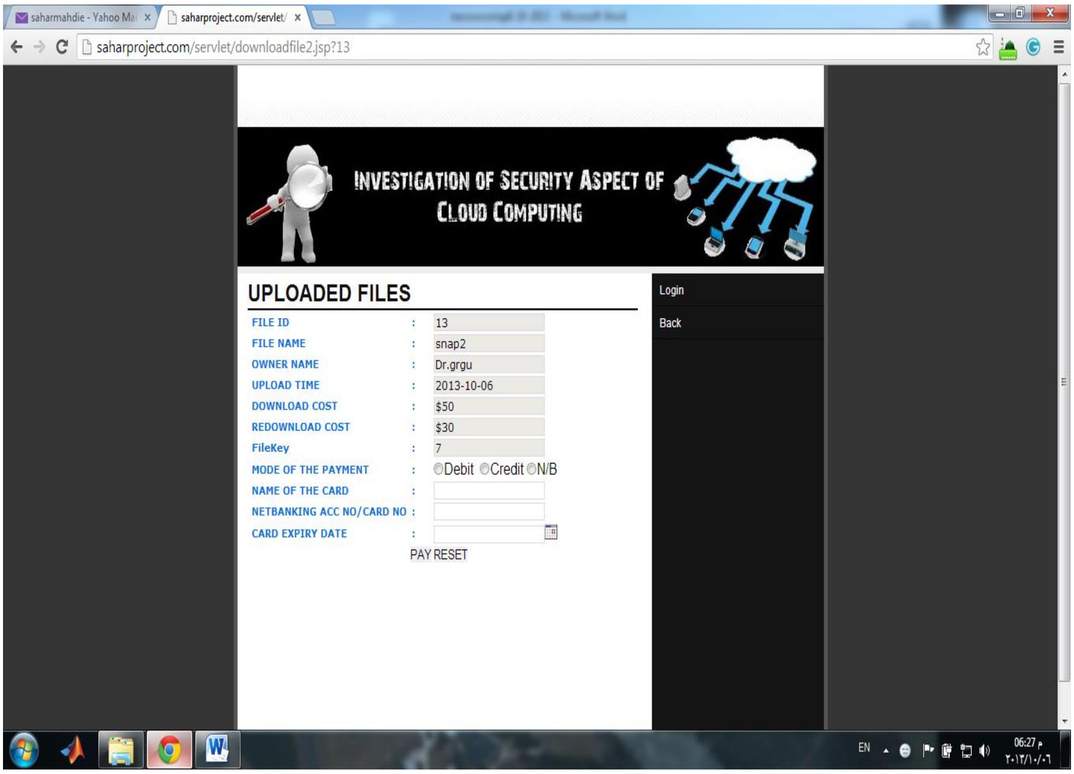
Do you want Google Chrome to save your password?

INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING

UPLOADED FILES

ID	FILE NAME	OWNER NAME	UPLOAD DATE	COST	ReDownLoad Cost	Status
2	Test	Dr.grgu	2013-10-04	$1	$1	Expired
3	TestApp	Dr.grgu	2013-10-05	$4444	$4444	Download
7	test	owner	2013-10-05	$4444	$4444	Download
8	test	Dr.grgu	2013-10-05	$454	$54	Download
10	asdg	Dr.grgu	2013-10-06	$5454	$5	Download
11	dg	Dr.grgu	2013-10-06	$655	$656	Download

A.19. Download Details Page:



saharproject.com/servlet/downloadfile2.jsp?13

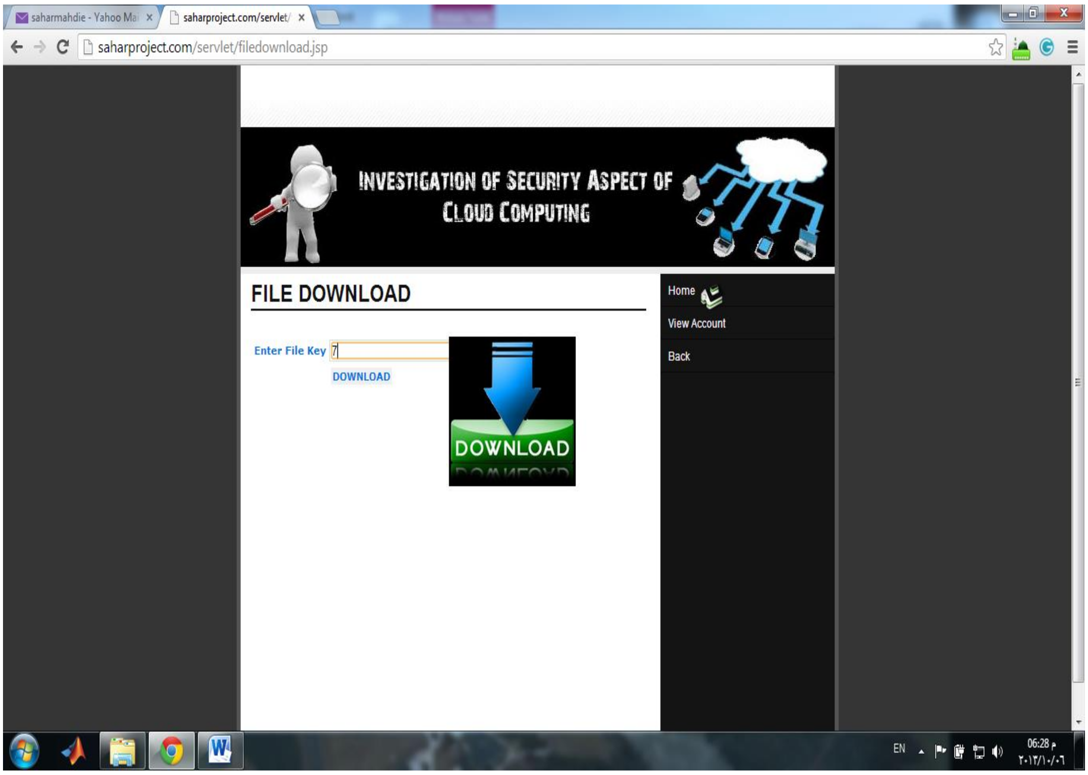
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING

UPLOADED FILES

FILE ID	:	13
FILE NAME	:	snap2
OWNER NAME	:	Dr.grgu
UPLOAD TIME	:	2013-10-06
DOWNLOAD COST	:	$50
REDOWNLOAD COST	:	$30
FileKey	:	7
MODE OF THE PAYMENT	:	○Debit ○Credit ○N/B
NAME OF THE CARD	:	
NETBANKING ACC NO/CARD NO	:	
CARD EXPIRY DATE	:	

PAY RESET

A.20. Final Download Details Page:



saharproject.com/servlet/filedownload.jsp

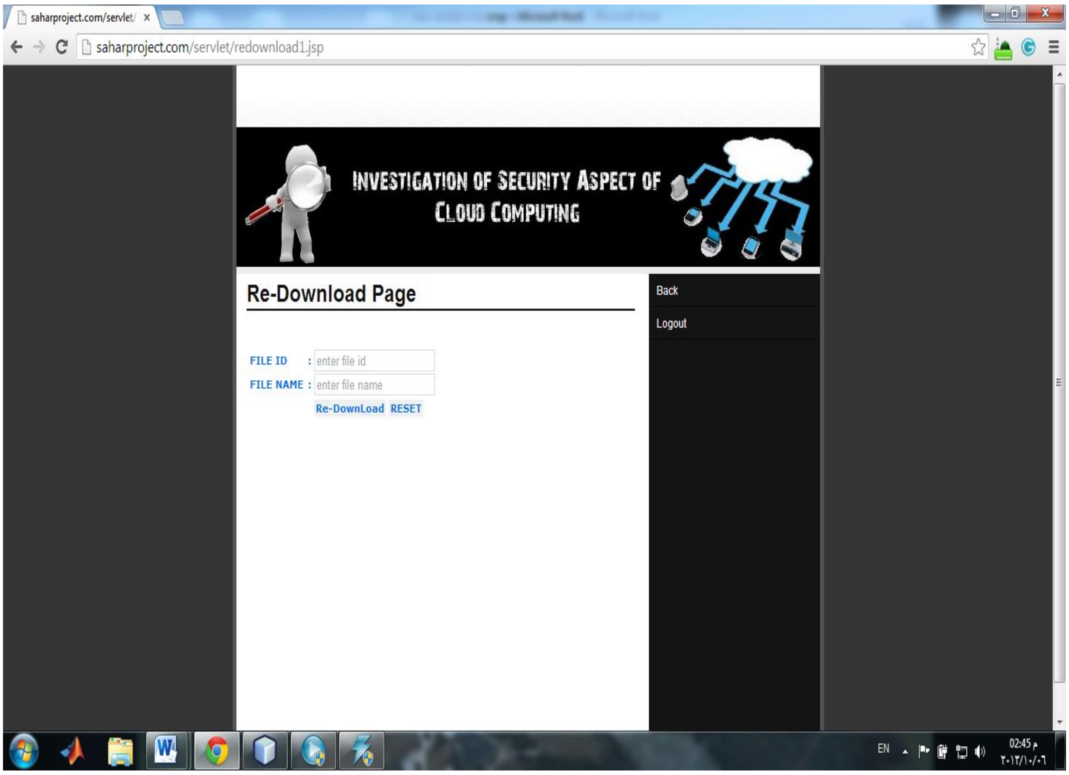
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING

FILE DOWNLOAD

Enter File Key 7

DOWNLOAD

A.21. Re-File Download Page:

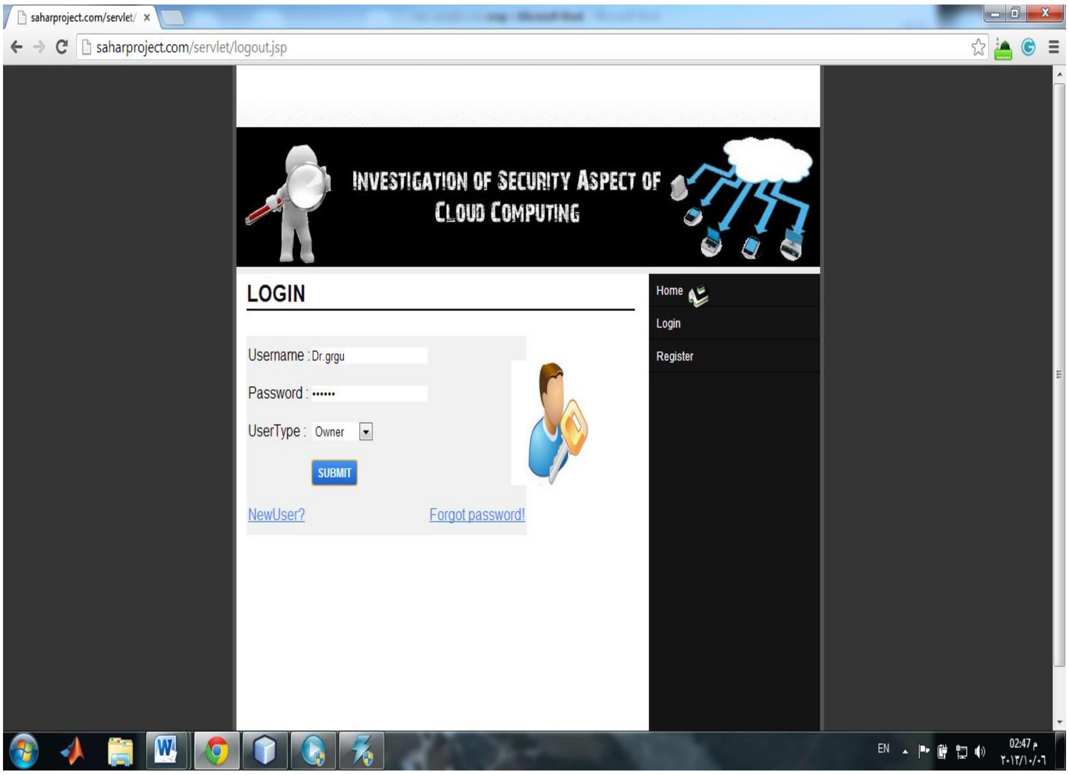

saharproject.com/servlet/
saharproject.com/servlet/redownload1.jsp
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Back
Logout
Re-DownLoad
RESET
EN
02:45 م
٢٠١٣/١٠/٠٦

Re-Download Page

FILE ID : enter file id

FILE NAME : enter file name

A.22. Below Figures are Show Us the Owner Pages


saharproject.com/servlet/
saharproject.com/servlet/logout.jsp
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Home
Login
Register
SUBMIT
NewUser?
Forgot password!
EN
02:47 م
٢٠١٣/١٠/٠٦

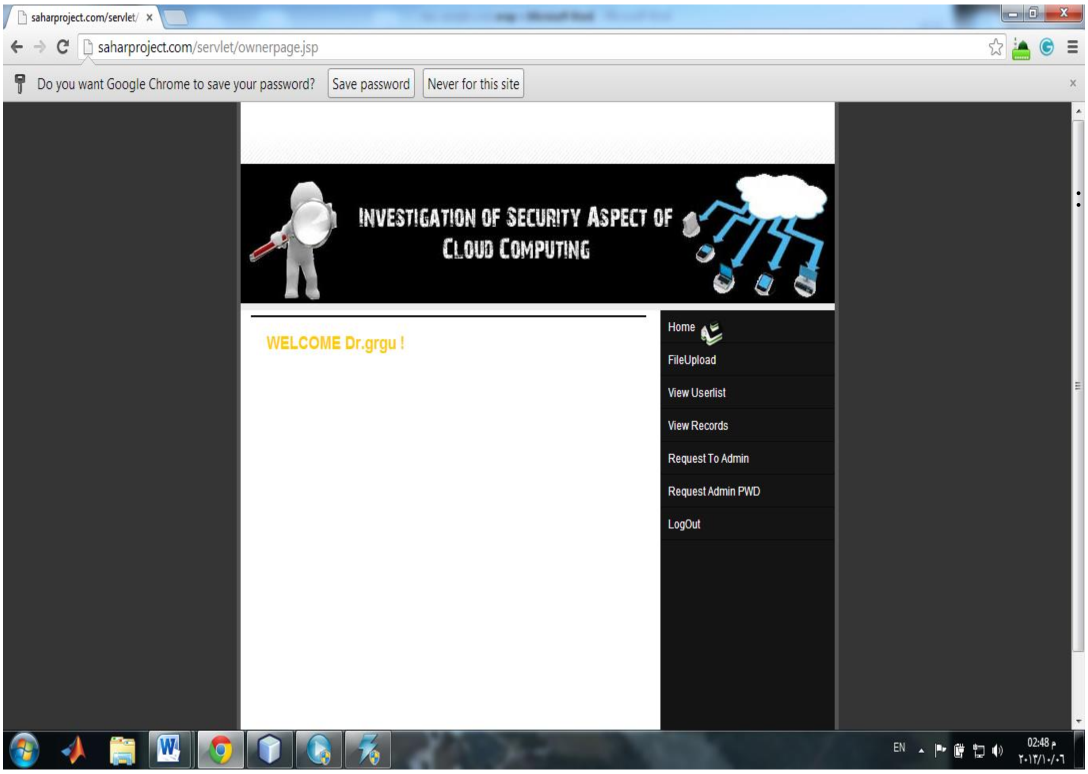
LOGIN

Username : Dr.grgu

Password : ••••••

UserType : Owner

A.23. Owner Welcome Page

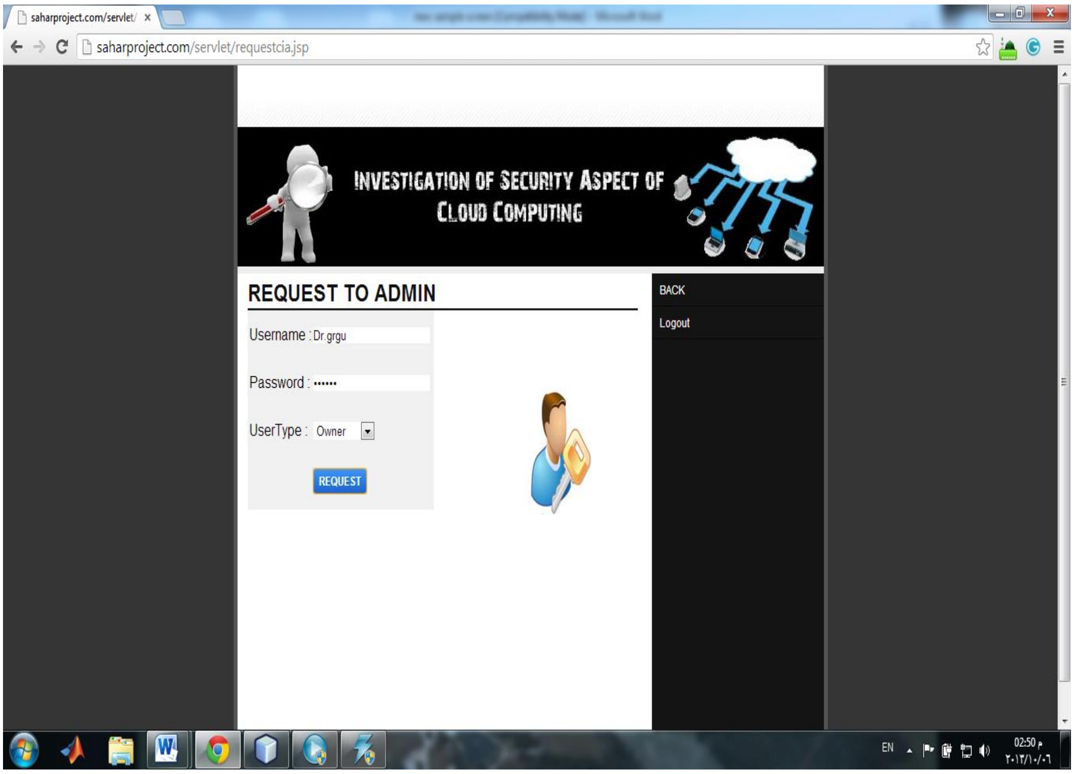

saharproject.com/servlet/ownerpage.jsp

Do you want Google Chrome to save your password? Save password Never for this site

INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING

WELCOME Dr.grgu !

A.24. Send Request by Owner to Get Grant from ADMIN for Upload Purpose


saharproject.com/servlet/requestcia.jsp

INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING

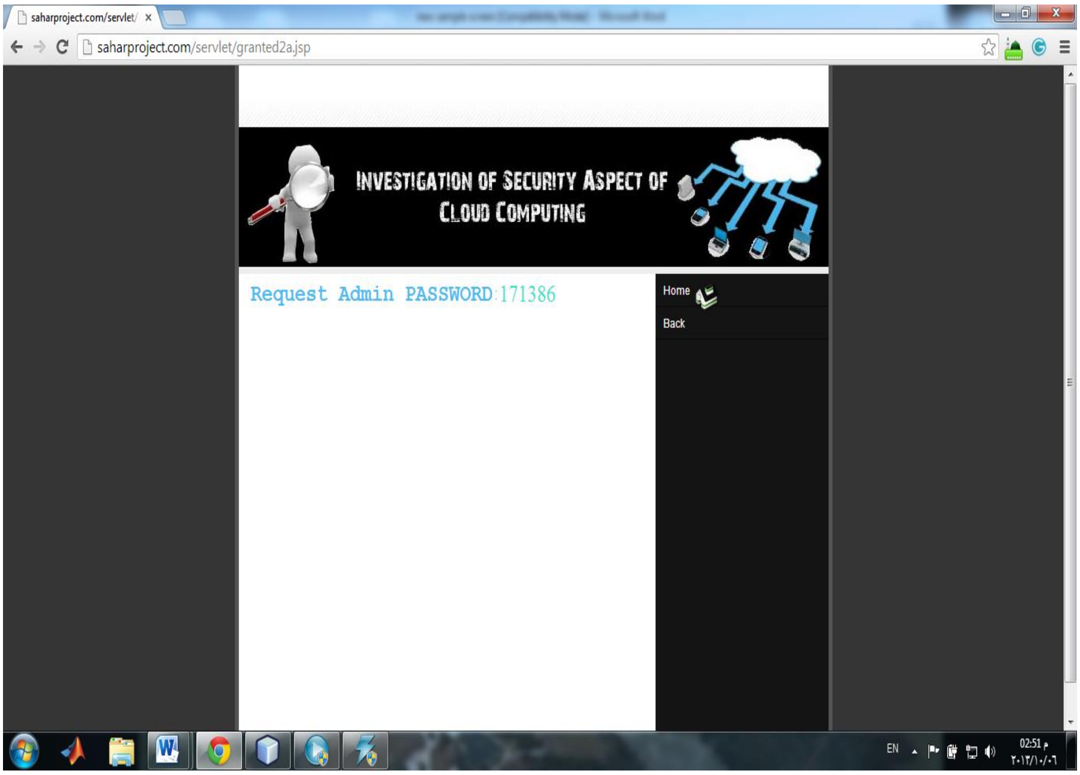
REQUEST TO ADMIN

Username : Dr.grgu

Password : ••••••

UserType : Owner

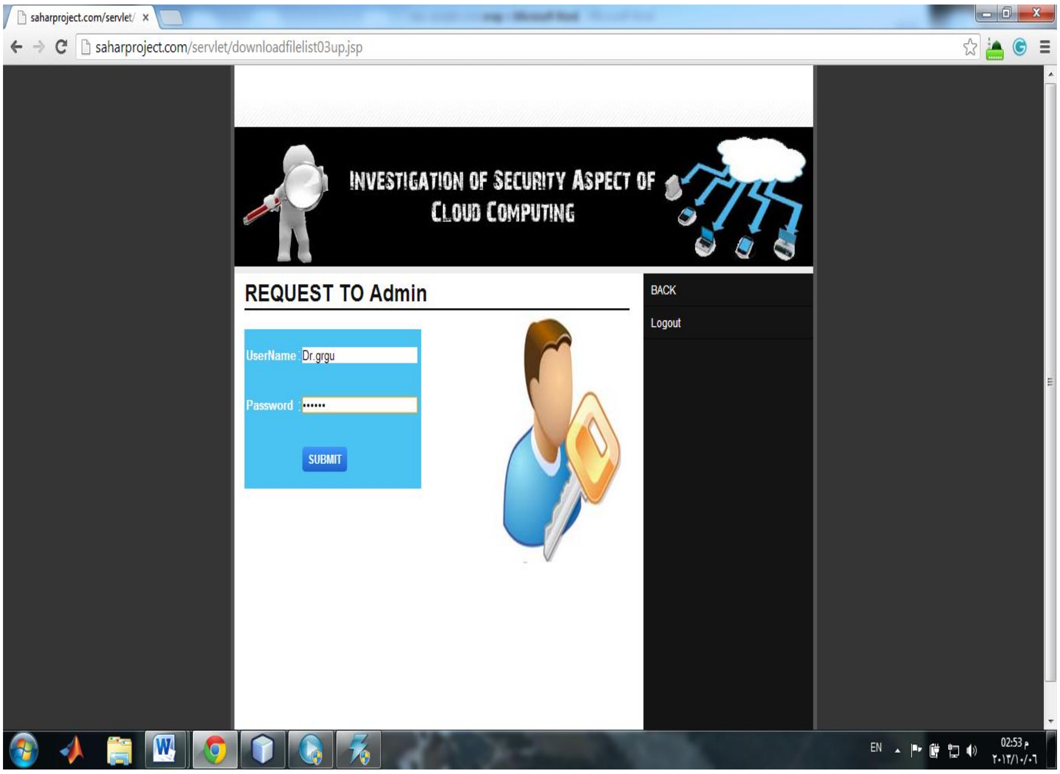
A.25. Owner Pequest Password Page After Got Grant from ADMIN:



saharproject.com/servlet/granted2a.jsp

Request Admin PASSWORD:171386

A.26. File Upload Page:



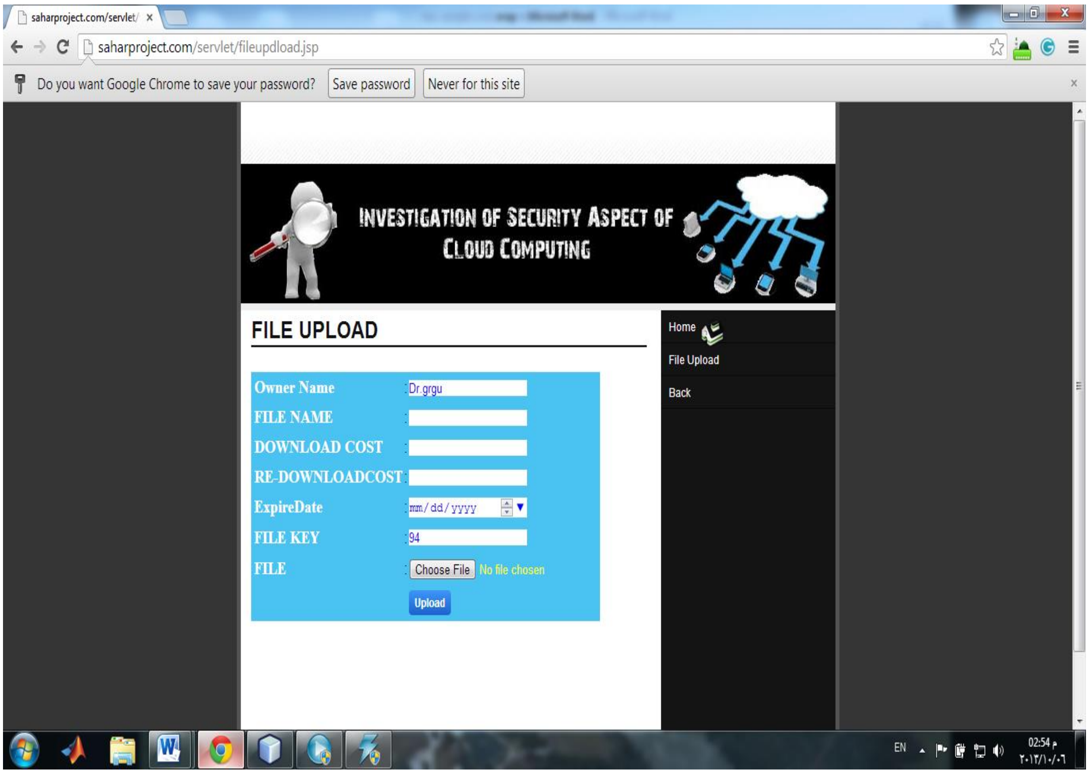
saharproject.com/servlet/downloadfilelist03up.jsp

REQUEST TO Admin

UserName : Dr.grgu

Password : ••••••

A.27. File Upload Page Details:


saharproject.com/servlet/
saharproject.com/servlet/fileupdload.jsp
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Home
File Upload
Back
EN
02:54 م
٢٠١٣/١٠/٠٦

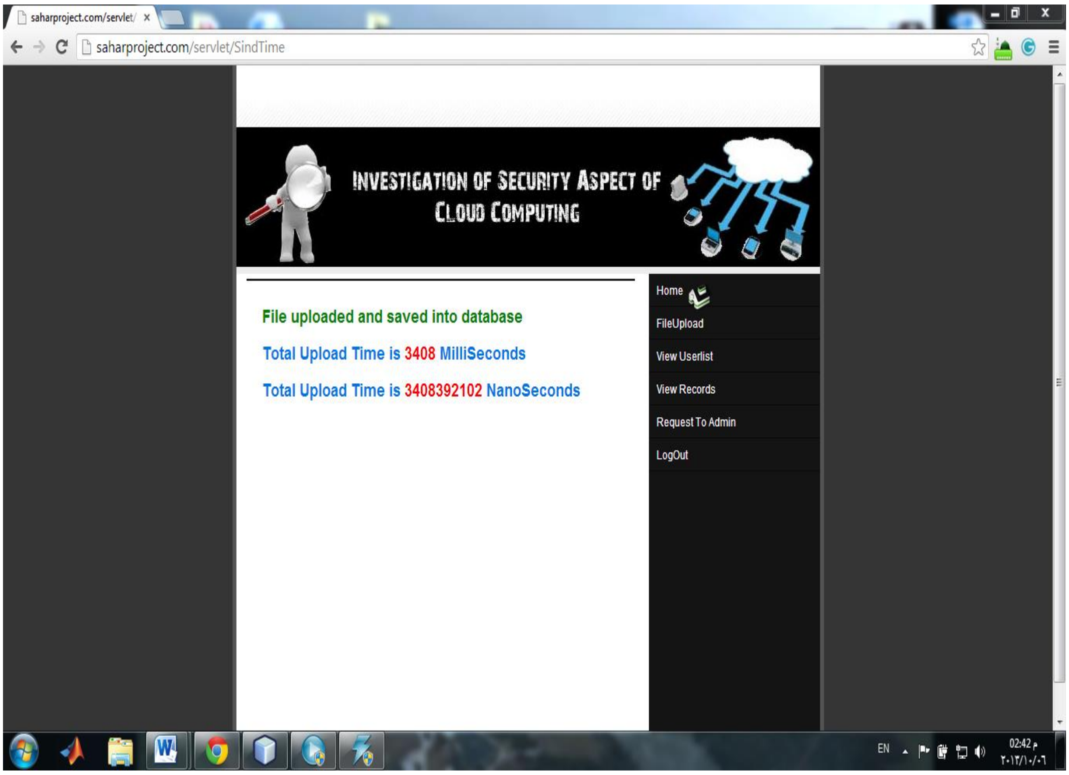
Do you want Google Chrome to save your password? Save password Never for this site

FILE UPLOAD

Owner Name	: Dr.grgu
FILE NAME	:
DOWNLOAD COST	:
RE-DOWNLOADCOST	:
ExpireDate	: mm/dd/yyyy
FILE KEY	: 94
FILE	: Choose File No file chosen

Upload

A.28. Final File Upload Page:

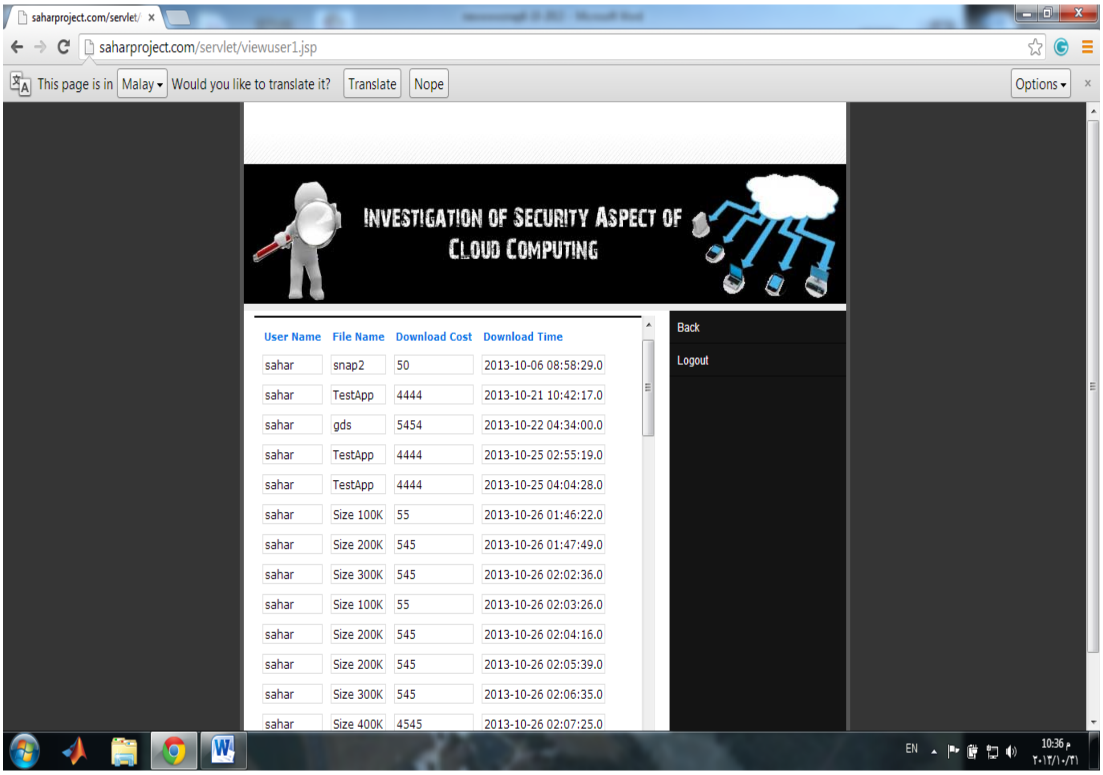

saharproject.com/servlet/
saharproject.com/servlet/SindTime
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Home
FileUpload
View Userlist
View Records
Request To Admin
LogOut
EN
02:42 م
٢٠١٣/١٠/٠٦

File uploaded and saved into database

Total Upload Time is 3408 MilliSeconds

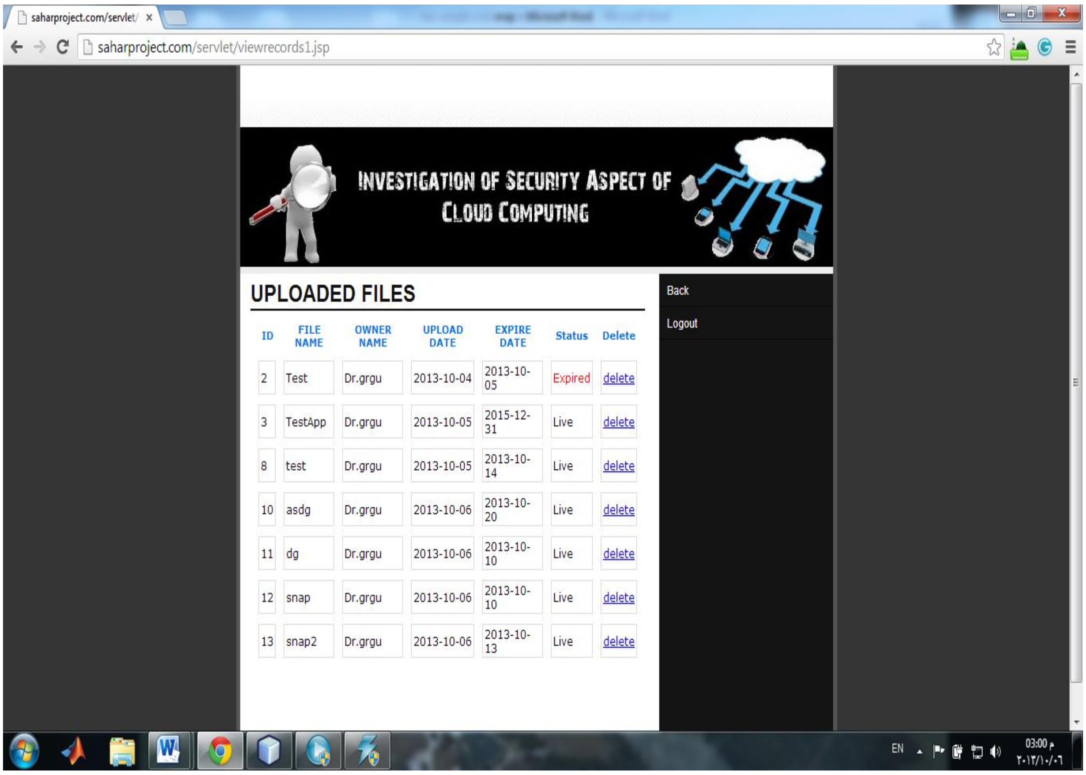
Total Upload Time is 3408392102 NanoSeconds

A.29. View User List Page:


saharproject.com/servlet/
saharproject.com/servlet/viewuser1.jsp
This page is in Malay Would you like to translate it? Translate Nope Options
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Back
Logout
EN
10:36 م

User Name	File Name	Download Cost	Download Time
sahar	snap2	50	2013-10-06 08:58:29.0
sahar	TestApp	4444	2013-10-21 10:42:17.0
sahar	gds	5454	2013-10-22 04:34:00.0
sahar	TestApp	4444	2013-10-25 02:55:19.0
sahar	TestApp	4444	2013-10-25 04:04:28.0
sahar	Size 100K	55	2013-10-26 01:46:22.0
sahar	Size 200K	545	2013-10-26 01:47:49.0
sahar	Size 300K	545	2013-10-26 02:02:36.0
sahar	Size 100K	55	2013-10-26 02:03:26.0
sahar	Size 200K	545	2013-10-26 02:04:16.0
sahar	Size 200K	545	2013-10-26 02:05:39.0
sahar	Size 300K	545	2013-10-26 02:06:35.0
sahar	Size 400K	4545	2013-10-26 02:07:25.0

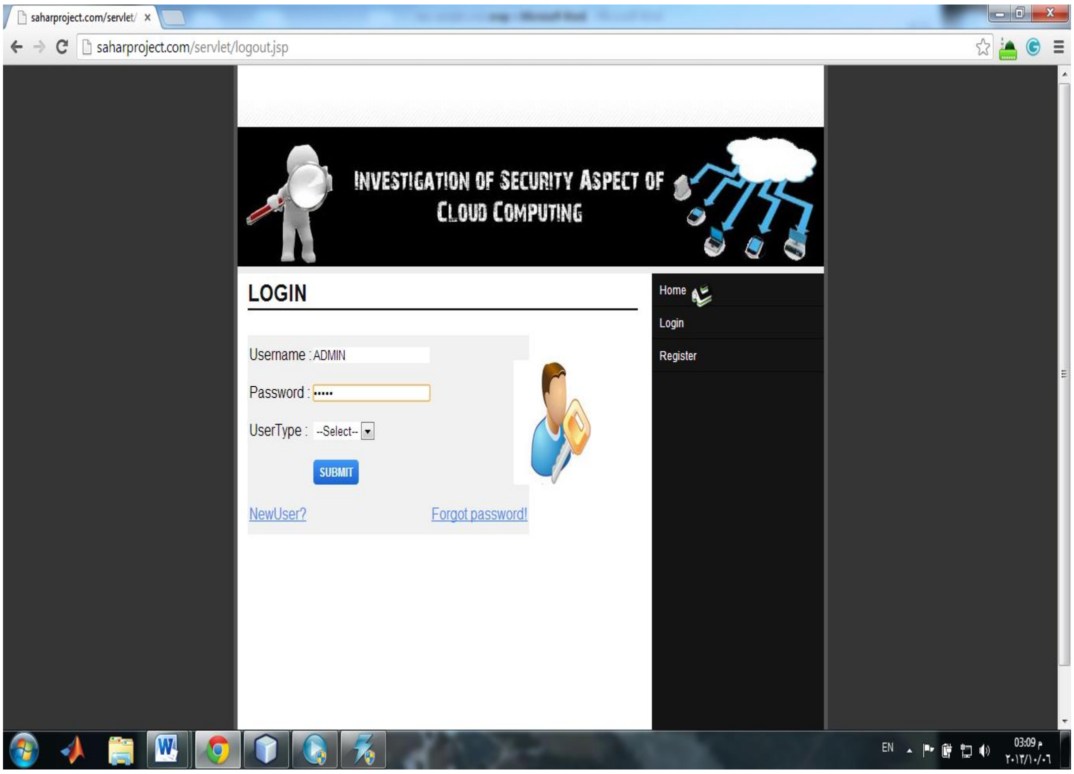
A.30. View Records and Delete Option Page:


saharproject.com/servlet/
saharproject.com/servlet/viewrecords1.jsp
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Back
Logout
EN
03:00 م

UPLOADED FILES

ID	FILE NAME	OWNER NAME	UPLOAD DATE	EXPIRE DATE	Status	Delete
2	Test	Dr.grgu	2013-10-04	2013-10-05	Expired	delete
3	TestApp	Dr.grgu	2013-10-05	2015-12-31	Live	delete
8	test	Dr.grgu	2013-10-05	2013-10-14	Live	delete
10	asdg	Dr.grgu	2013-10-06	2013-10-20	Live	delete
11	dg	Dr.grgu	2013-10-06	2013-10-10	Live	delete
12	snap	Dr.grgu	2013-10-06	2013-10-10	Live	delete
13	snap2	Dr.grgu	2013-10-06	2013-10-13	Live	delete

A.31. Below Figures are Show Us the Admin Pages



saharproject.com/servlet/logout.jsp

INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING

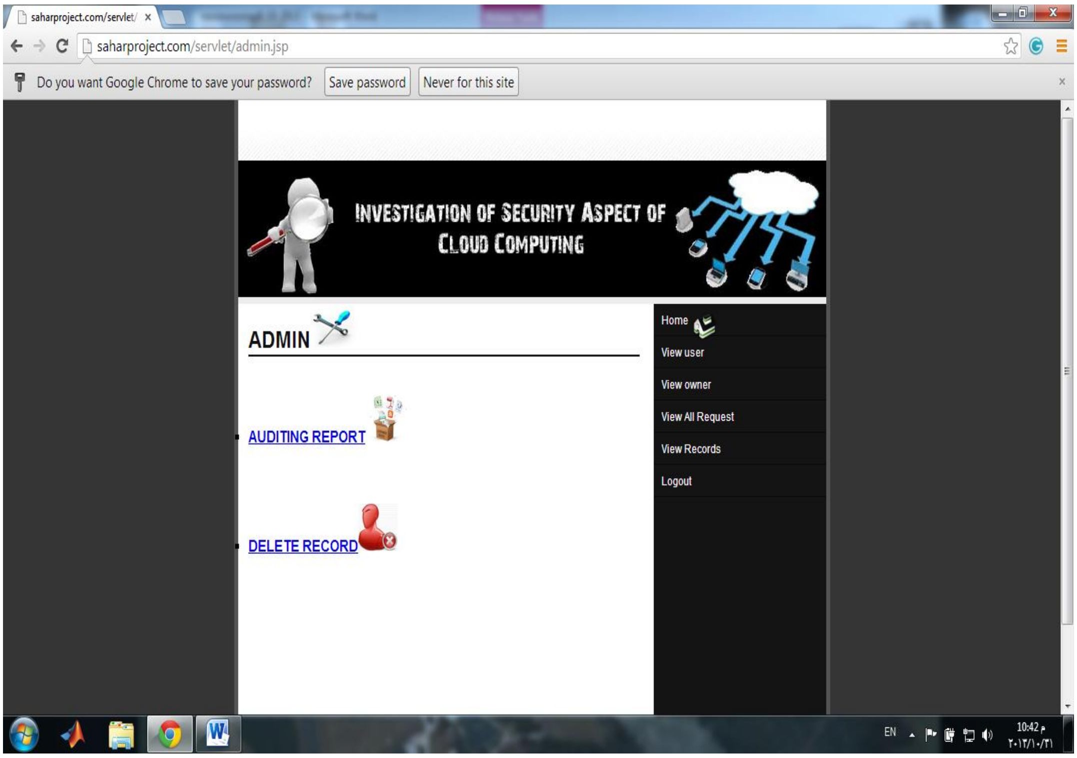
LOGIN

Username : ADMIN

Password : •••••

UserType : --Select--

A.32. ADMIN Welcome Page:



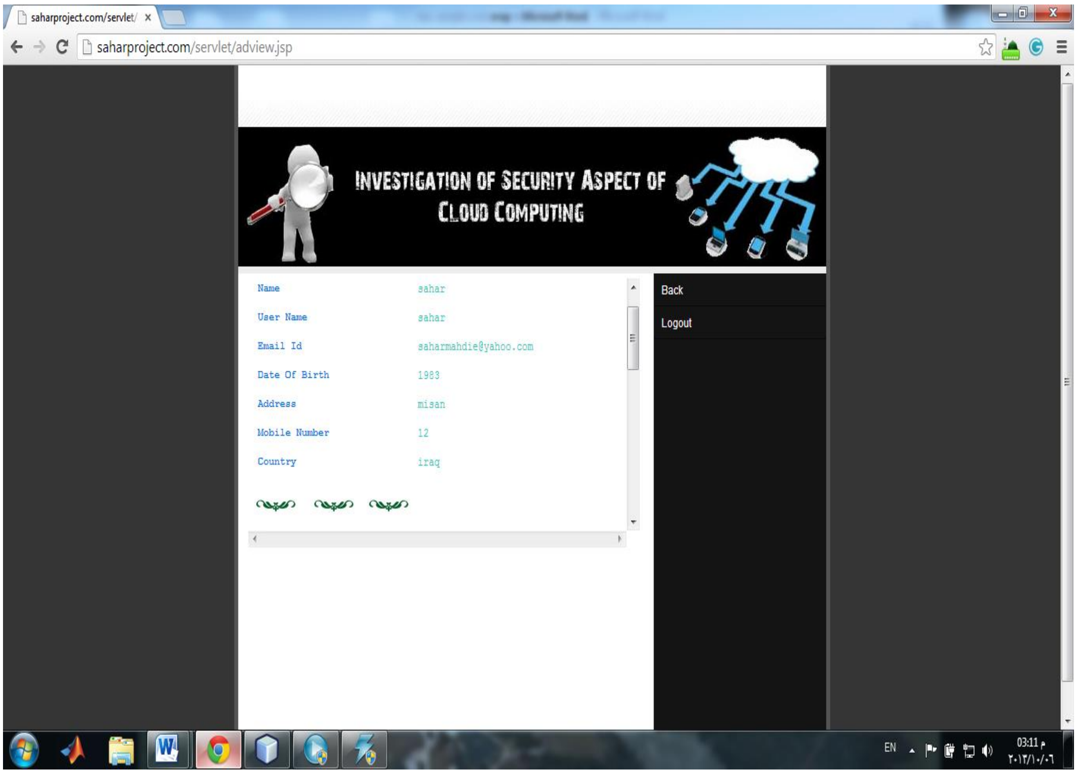
saharproject.com/servlet/admin.jsp

Do you want Google Chrome to save your password?

INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING

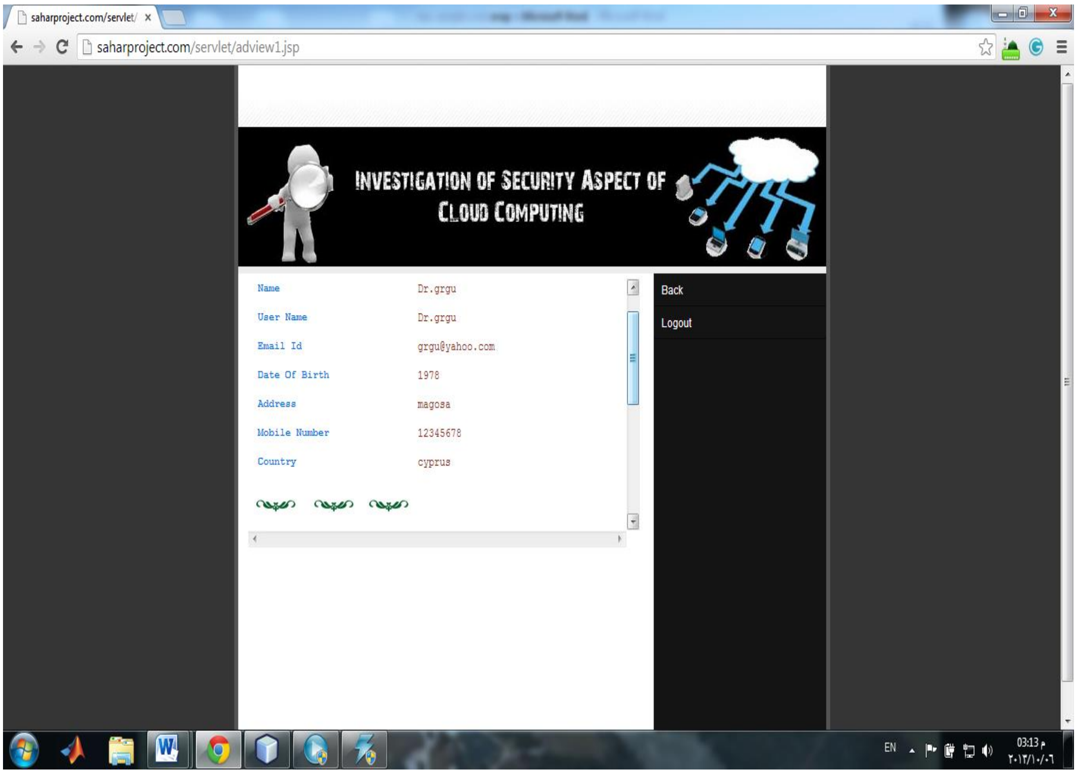
ADMIN

A.33. View User List Page:


saharproject.com/servlet/
saharproject.com/servlet/adview.jsp
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Back
Logout

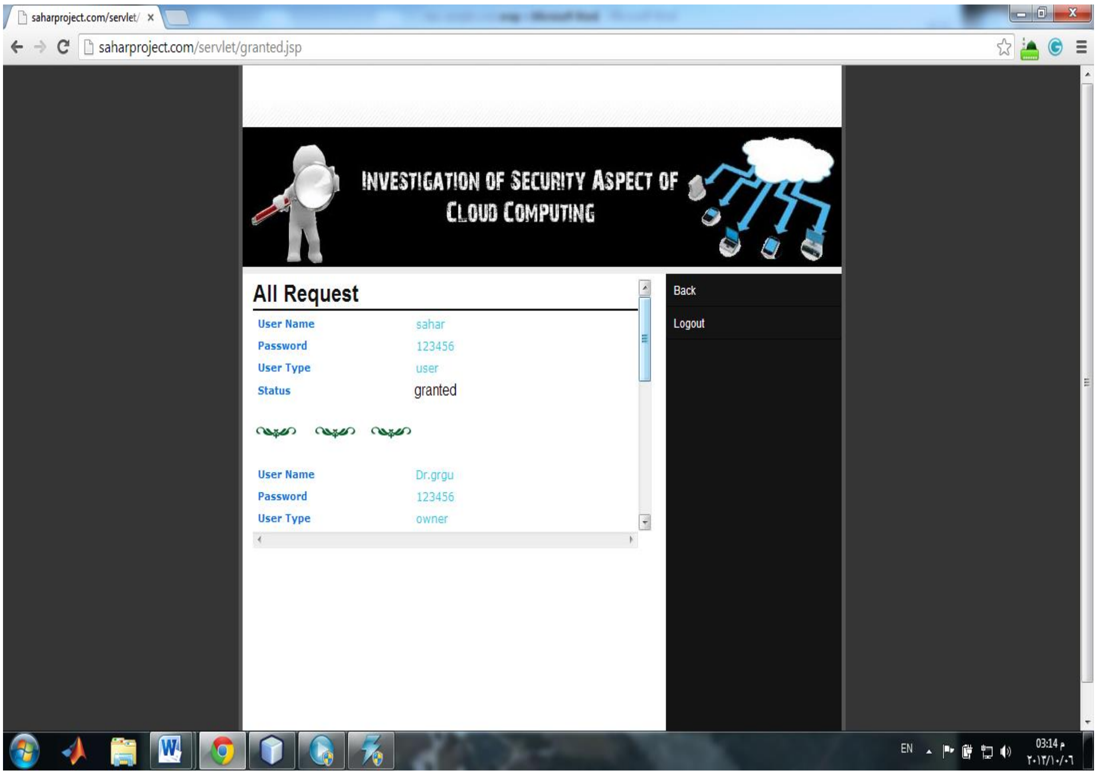
Name	sahar
User Name	sahar
Email Id	saharmahdie@yahoo.com
Date Of Birth	1983
Address	misan
Mobile Number	12
Country	iraq

A.34. View Owner List Page:


saharproject.com/servlet/
saharproject.com/servlet/adview1.jsp
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Back
Logout

Name	Dr.grgu
User Name	Dr.grgu
Email Id	grgu@yahoo.com
Date Of Birth	1978
Address	magosa
Mobile Number	12345678
Country	cyprus

A.35. View all Request (which sent by users and owners to get permision for their actions) Page:

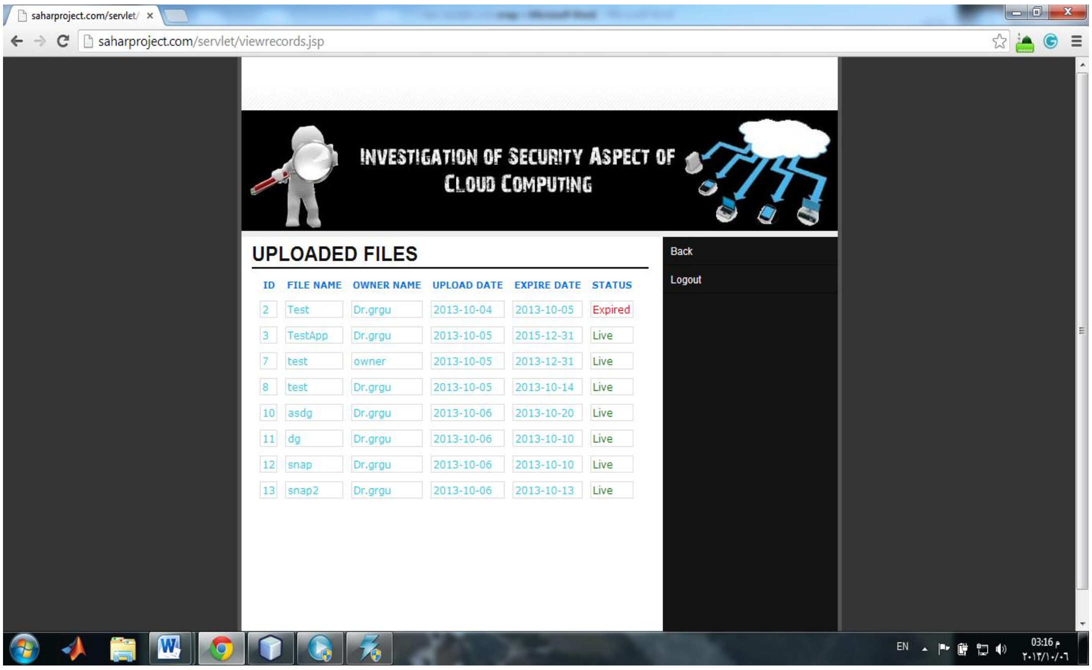

saharproject.com/servlet/
saharproject.com/servlet/granted.jsp
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Back
Logout
EN
03:14 م
٢٠١٣/١٠/٠٦

All Request

User Name	sahar
Password	123456
User Type	user
Status	granted

User Name	Dr.grgu
Password	123456
User Type	owner

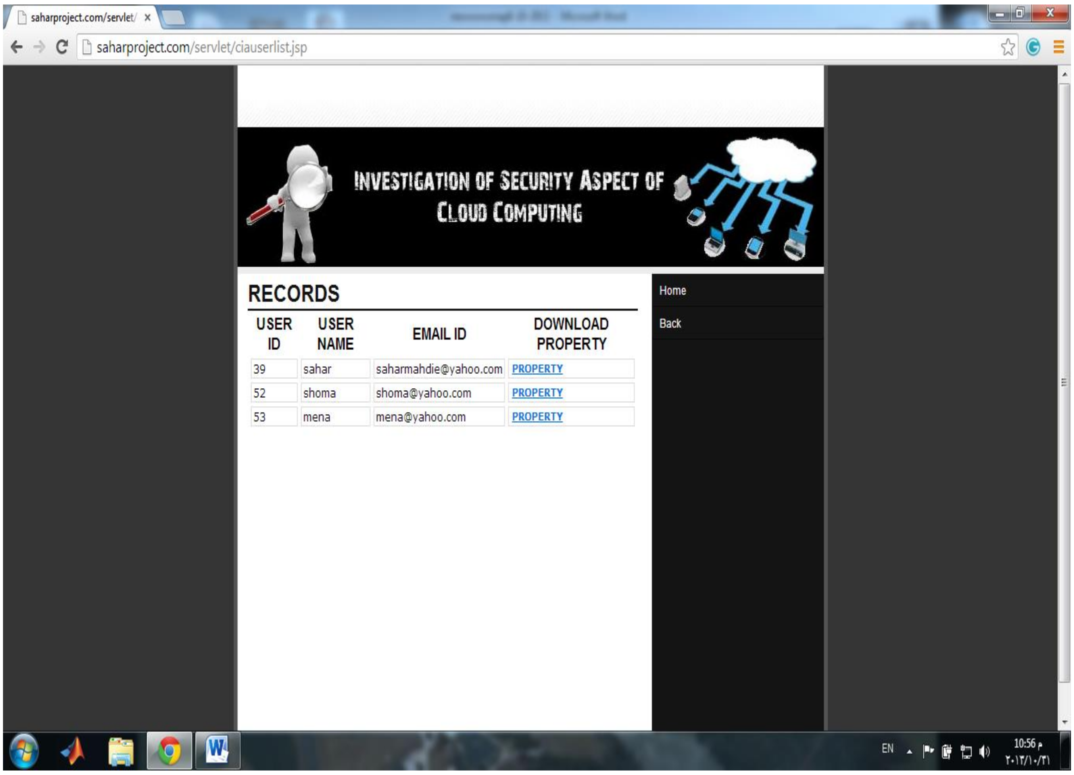
A.36. View all uploaded files by viewing records page (part of auditing mechanism) with showing us expires feature:


saharproject.com/servlet/
saharproject.com/servlet/viewrecords.jsp
INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING
Back
Logout
EN
03:16 م
٢٠١٣/١٠/٠٦

UPLOADED FILES

ID	FILE NAME	OWNER NAME	UPLOAD DATE	EXPIRE DATE	STATUS
2	Test	Dr.grgu	2013-10-04	2013-10-05	Expired
3	TestApp	Dr.grgu	2013-10-05	2015-12-31	Live
7	test	owner	2013-10-05	2013-12-31	Live
8	test	Dr.grgu	2013-10-05	2013-10-14	Live
10	asdg	Dr.grgu	2013-10-06	2013-10-20	Live
11	dg	Dr.grgu	2013-10-06	2013-10-10	Live
12	snap	Dr.grgu	2013-10-06	2013-10-10	Live
13	snap2	Dr.grgu	2013-10-06	2013-10-13	Live

A.37. Auditing Property Page:

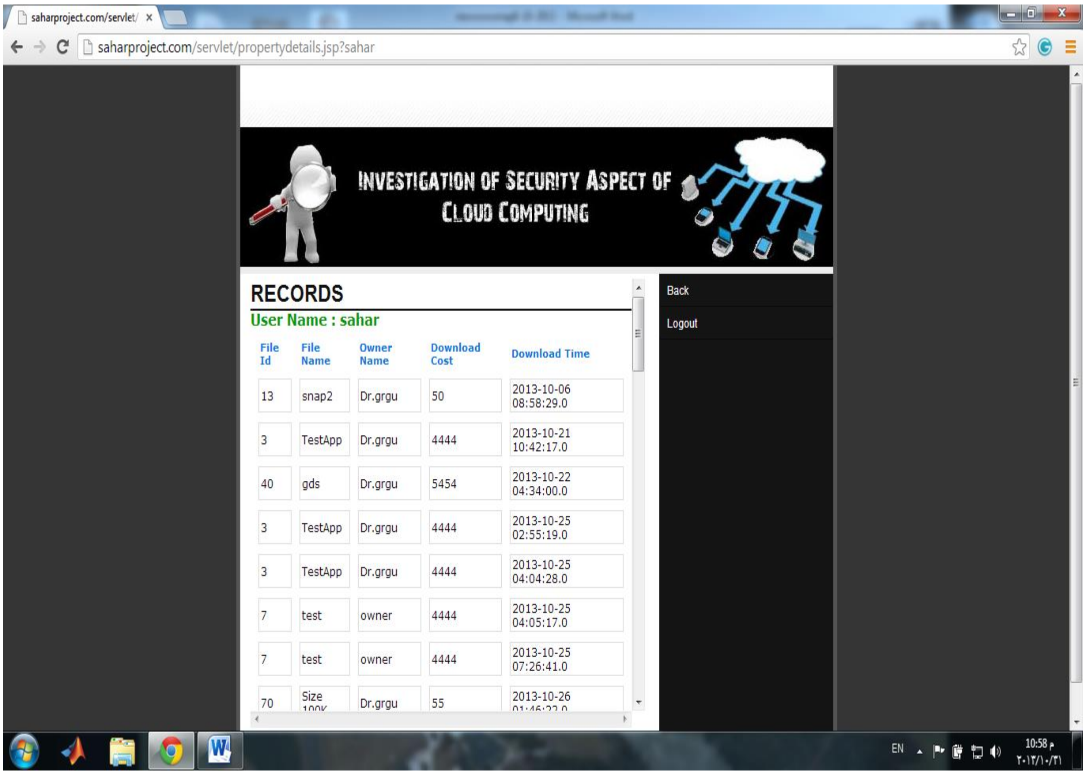


saharproject.com/servlet/ciauserlist.jsp

RECORDS

USER ID	USER NAME	EMAIL ID	DOWNLOAD PROPERTY
39	sahar	saharmahdie@yahoo.com	PROPERTY
52	shoma	shoma@yahoo.com	PROPERTY
53	mena	mena@yahoo.com	PROPERTY

A.38. Auditing property page after choosing one of the users :



saharproject.com/servlet/propertydetails.jsp?sahar

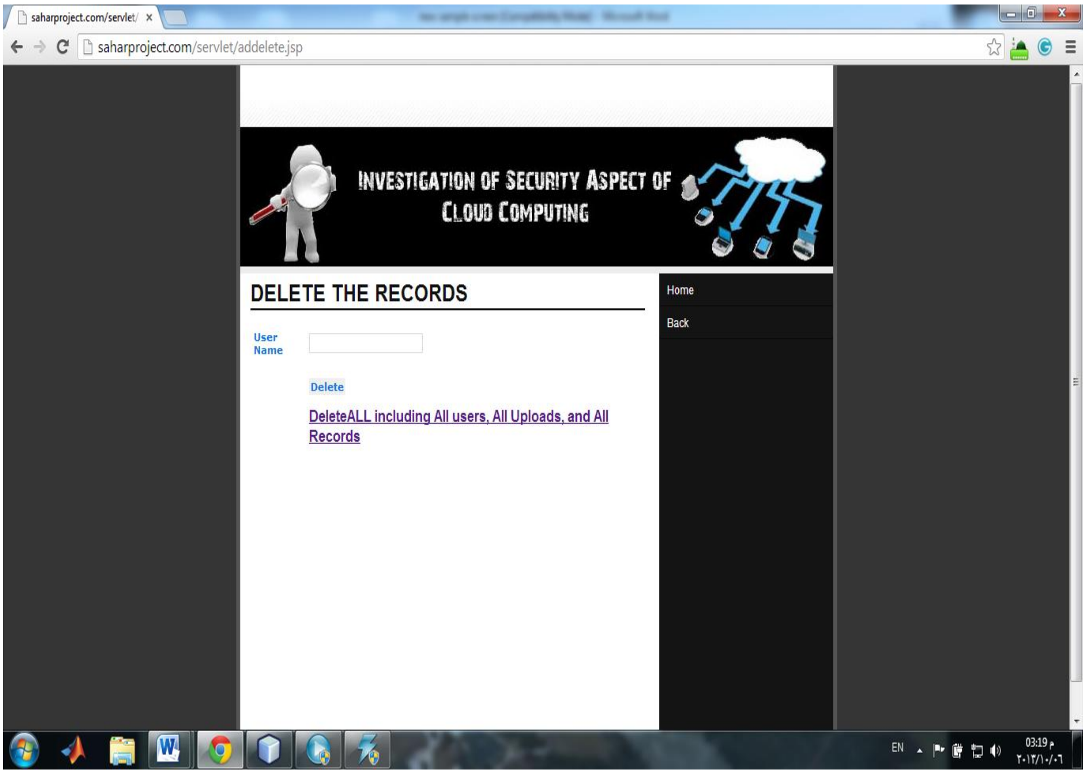
RECORDS

User Name : sahar

File Id	File Name	Owner Name	Download Cost	Download Time
13	snap2	Dr.grgu	50	2013-10-06 08:58:29.0
3	TestApp	Dr.grgu	4444	2013-10-21 10:42:17.0
40	gds	Dr.grgu	5454	2013-10-22 04:34:00.0
3	TestApp	Dr.grgu	4444	2013-10-25 02:55:19.0
3	TestApp	Dr.grgu	4444	2013-10-25 04:04:28.0
7	test	owner	4444	2013-10-25 04:05:17.0
7	test	owner	4444	2013-10-25 07:26:41.0
70	Size 100K	Dr.grgu	55	2013-10-26 01:46:22.0

A.39. Delete records page, delete members page:

If you click delete all then all the user and owner will be removed from the database….



saharproject.com/servlet/addelete.jsp

INVESTIGATION OF SECURITY ASPECT OF CLOUD COMPUTING

DELETE THE RECORDS

User Name

Delete

DeleteALL including All users, All Uploads, and All Records

Appendix B: Uploading code

```
package net.codejava.upload;
import java.io.IOException;
import java.io.InputStream;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.*;
import java.text.SimpleDateFormat;
import javax.servlet.ServletException;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.Part;
WebServlet(urlPatterns="/SindTime" , name="SindTime")
public class SindTime extends HttpServlet {
int StartMinute,Startsecond,StopMinute,StopSecond;
long StartMilliSeconds,StopMilliSeconds,StartNanoSeconds,StopNanoSeconds;
  // database connection settings
        private String dbURL = "jdbc:mysql://localhost:3306/DB_saharproject";
        private String dbUser = "saharproject";

        private String dbPass = "admin@sahar";


        protected void doPost(HttpServletRequest request,

                        HttpServletResponse response) throws ServletException,

IOException {

        //StartMinute = date.get(Calendar.MINUTE);

         // StartSecond = date.get(Calendar.SECOND);

          StartMilliSeconds = System.currentTimeMillis();

          StartNanoSeconds = System.nanoTime();

                // gets values of text fields

                //String firstName = request.getParameter("firstName");

                //String lastName = request.getParameter("lastName");

           String fileName = request.getParameter("filename");

           String ownerName = request.getParameter("ownername");

           String fdcost = request.getParameter("filecost");

           String frdcost = request.getParameter("refilecost");

           //String ownerName="test";

           int fcost=Integer.parseInt(fdcost);

           int fcost1=Integer.parseInt(frdcost);

           String  filekey=request.getParameter("filekey");

            java.util.Date now = new java.util.Date();

         String date=now.toString();

         String DATE_FORMAT = "yyyy-MM-dd hh:mm:ss";

         SimpleDateFormat sdf = new SimpleDateFormat(DATE_FORMAT);
```

```
    String strDateNew = sdf.format(now) ;
        InputStream inputStream = null;       // input stream of the upload file
                // obtains the upload file part in this multipart request
                Part filePart = request.getPart("datafile");
                if (filePart != null) {
                        // prints out some information for debugging
                        System.out.println(filePart.getName());
                        System.out.println(filePart.getSize());
                        System.out.println(filePart.getContentType());
                        // obtains input stream of the upload file
                        inputStream = filePart.getInputStream();
                 }
                Connection conn = null;       // connection to the database
                String message = null;// message will be sent back to client
                try {
                // connects to the database
                DriverManager.registerDriver(new com.mysql.jdbc.Driver());
                conn = DriverManager.getConnection(dbURL, dbUser, dbPass);
                // constructs SQL statement
                //String sql = "INSERT INTO contacts (first_name, last_name, photo)
values (?, ?, ?)";
                String sql="insert into upload
(filename,file,ownername,utime,fcost,redwnloadcost,filekey) values(?,?,?,?,?,?,?)";
                        PreparedStatement statement = conn.prepareStatement(sql);
                        statement.setString(1, fileName);
```

```
                    if (inputStream != null)
      {
                    // fetches input stream of the upload file for the blob column
                        statement.setBlob(2, inputStream);
                    }
                statement.setString(3, ownerName);
                statement.setString(4, strDateNew);
                statement.setInt(5, fcost);
                statement.setInt(6, fcost1);
                statement.setString(7, filekey);
                    // sends the statement to the database server
                    int row = statement.executeUpdate();
                    if (row > 0) {
                        message = "File uploaded and saved into database";
                    }
                } catch (SQLException ex) {
                    message = "ERROR: " + ex.getMessage();
                    ex.printStackTrace();
                }
            finally {
              // StopMinute = date.get(Calendar.MINUTE);
            // StopSecond = date.get(Calendar.SECOND);
             StopMilliSeconds = System.currentTimeMillis();
             StopNanoSeconds = System.nanoTime();
                    if (conn != null) {
```

```
                // closes the database connection
                try {
                    conn.close();
                } catch (SQLException ex) {
                    ex.printStackTrace();
                }}
            // sets the message in request scope
            request.setAttribute("Message", message);
            //request.setAttribute("StartMinute", StopMinute);
            //request.setAttribute("StartSecond", StopSecond);
            request.setAttribute("StartMilliSeconds", StartMilliSeconds);
            request.setAttribute("StartNanoSeconds", StartNanoSeconds);
            //request.setAttribute("StopMinute", StopMinute);
            //request.setAttribute("StopSecond", StopSecond);
            request.setAttribute("StopMilliSeconds", StopMilliSeconds);
            request.setAttribute("StopNanoSeconds", StopNanoSeconds);
            // forwards to the message page

    getServletContext().getRequestDispatcher("/Message.jsp").forward(request,
response);
        }}}
```