



Remote Browser of Encrypted Database

(Providing Secure File Storage and Sign-In, Browsing, Adding Bookmarks)

Mr B.Srinivasulu, #1, Mr Umamaheswararao inkollu., #2, Mr A.Bhima Sanakram #3, Mr Naga Babu .G #4,
JNTU Hyderabad
India

Abstract - *The project consists of an efficient The application stores the search history of the user. It also saves the bookmarks. The user must be registered in the remote data encryption storage by the consent of the third party. The admin will provide the results based on the request sent by the user. The request can be based on password and security issues. User logs in through the third party authentication and browses the read information. The information which is browsed by the user will be stored by the application. The application can also remember the password through which the user gets an one click access once the website i.e, the application opens the website without asking the user details(id, password). The account data is encrypted in the database. Remote Browser of Encrypted Database(RBFED) is an one of the best and better-most application out there to help you manage passwords. it is a surprisingly easy way to manage website credentials. Remote Browser of Encrypted Database is a a single sign in solution that supports both userid and direct sign in to a number of supported websites. Never have to remember dozens of usernames and passwords again! Access your online accounts at the most popular web sites with one sign-in .Just create your Remote Browser for encrypted database account to use the services of our web application. A bit more recently, Remote Browser for Encrypted Database is providing secure file storage and sign-in, browsing, adding bookmarks, adding clients.*

Keywords: *Encrypted Database, security issues, authentication, data encryption storage, systems development method, Error Handling,*

I. Introduction

The application stores the search history of the user. It also saves the bookmarks. The user must be registered in the remote data encryption storage by the consent of the third party. The admin will provide the results based on the request sent by the user. The request can be based on password and security issues. User logs in through the third party authentication and browses the read information. The information which is browsed by the user will be stored by the application. The application can also remember the password through which the user gets an one click access once the website i.e, the application opens the website without asking the user details(id, password). The account data is encrypted in the database.

Problem Definition

It is a local system in which the user performs the searching operations. The user can't access the history, bookmarks and more from different places. The user cannot remember all the passwords in the existing system. So, he can benefit from the proposed system.

Proposed System Features

The remote data encryption storage is an application which provides personal account for each individual user and provides security to the user account. It saves the information of all the different accounts of social networking.

Search : provides an live browser to an user, which serves the user request on the web and provides related accurate results to an user on google search, and also stores the user searching data on to the database, which will make you store history on web application and make use of it whenever u need it. user can also clear his history of visited sites, using clear history option

Database search: which is an custom search for user to look after the clients in the web application

Bookmarks:- user can add his bookmarks and can store it in the database in an encrypt format for future purpose, and can also remove from bookmarks list.

No Multiple Identities. This is a feature of user provided with username and password. Allows user to set up separate 'identities' for each and individual clients to which administrator has given access to share with users

One-Click Access:

This is a truly unique and really very useful feature. Simply it provides direct access to the clients, when user is login with his credentials at the first time of login, can make use of clients without entering passwords for again and again, only registered users with active status can be added by user.

File storage:

It provides secure file storage through the database. online file storage is online file storage for the most part and unlimited data storage.

II. System Overview

The remote data encryption storage is an application which provides personal account for each individual user and provides security to the user account. It saves the information of all the different accounts of social networking.

Search : provides an live browser to an user, which serves the user request on the web and provides related accurate results to an user on google search, and also stores the user searching data on to the database, which will make you store history on web application and make use of it whenever u need it. user can also clear his history of visited sites, using clear history option

Modules

1. Administrator module
2. User module
3. Client module

Administrator Module

Admin manages the entire application and provides services like browsing, managing history, bookmarks, one click access to other accounts and encrypted database.

User Module

User is benefited by using the services of the Client which are approved and provided by the Admin. User also maintains history, bookmarks, and one click access.

Client Module

Clients are the websites approved by the admin so that they can be made available to the end users.

Model View Controller:

In software engineering, Model–View–Controller (MVC) is an architectural pattern that splits interactions between users and applications into three roles: the Model (business logic), the View (user interface), and the Controller (user input). This separation of concerns facilitates the independent development, testing, and maintenance of each role.

History

MVC was first described in 1979 by Trygve Reenskaug, then working on Smalltalk at Xerox PARC. The original implementation is described in depth in the influential paper "Applications Programming in Smalltalk-80: How to use Model–View–Controller". MVC was then called software architecture, in a now-deprecated sense.

Since the late 1990s, MVC is commonly classified as an architectural pattern, i.e. a classic way of structuring software that is used in software architecture in the modern sense of Shaw and Garlan.

Overview

Although MVC comes in different flavours, the control flow is generally as follows:

The user interacts with the user interface in some way (for example, by pressing a mouse button).

The controller handles the input event from the user interface, often via a registered handler or call back, and converts the event into an appropriate user action, understandable for the model.

The controller notifies the model of the user action, possibly resulting in a change in the model's state. (For example, the controller updates the user's shopping cart).

A view queries the model to generate an appropriate user interface (for example the view lists the shopping carts contents).

The view gets its own data from the model. In some implementations, the controller may issue a general instruction to the view to render itself. In others, the view is automatically notified by the model of changes in state (observer) that require a screen update.

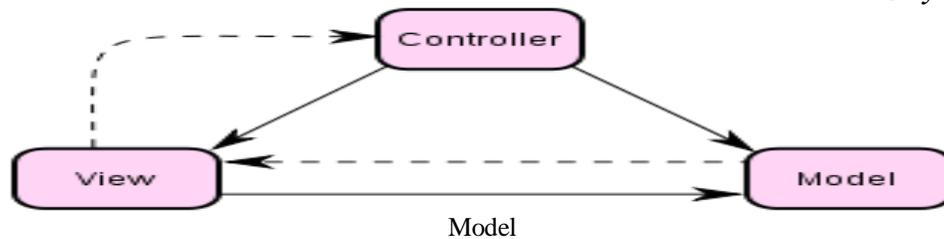
The user interface waits for further user interactions, which restarts the control flow cycle.

Architecture vs framework

Although the MVC pattern is typically associated with frameworks, it is essentially architectural. This means that it can be implemented even without the object-oriented language or the specific class hierarchy. For example, using as little as jQuery's `trigger()` and `bind()`, it is possible to build robust MVC-based applications in the browser using JavaScript. The key is simply to divide up the responsibilities of the MVC components into clearly defined sections of the code. As stated in the overview, the code that embodies the model takes care of the state, business logic, persistence, and notifications. Persistence can be implemented via cookies or Ajax. The notifications can be taken care of by `jQuery.trigger()`. The code that embodies the view takes care of querying the model and rendering the view. The view code can be implemented in a variety of ways, including inserting HTML DOM nodes or changing Cascading Style Sheets (CSS) styles. The code that embodies the controller takes care of the initialization of the model and the wiring up of the events between the view's HTML DOM elements and controller and between the model and the view code, using `jQuery.bind()`.

Example

Here is a simple application of the pattern, implementing Java Servlets and Java Server Pages from Java EE:



The model is a collection of Java classes that form a software application intended to store, and optionally separate, data. A single front end class that can communicate with any user interface (for example: a console, a graphical user interface, or a web application).

View

The view is represented by a Java Server Page, with data being transported to the page in the HttpServletRequest or HttpSession.

Implementations as Web Based Framework

In the design of web applications, MVC is implemented by web template systems as a "view for web" component.

MVC is typically implemented as Model 2 architecture in Sun parlance. Model 2 focuses on efficiently handling and dispatching full page form posts and reconstructing the full page via a front controller. Complex web applications continue to be more difficult to design than traditional applications because of this "full page" effect. More recently "view for web" Ajax driven frameworks that focus on firing focused UI events at specific UI Components on the page are emerging. This is causing MVC to be revisited for web application development using traditional desktop programming techniques.

III. The Working Principle

Functional Requirements:

Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system requires to perform. As the product is being developed it is useful to distinguish between the base line functionality necessary for any system to complete in that product domain and features that differentiate the system. Features may be additional functionality or differ from basic functionality along some quality attribute. A platform based development approach leverages this commonality utilizing a set of reusable assets across the family. These strategies have important implications for software architecture. In particular, it is not just the functional requirements of the product that must be supported by the architecture. The functional requirements of early release need to be explicitly taken into account. Use cases have quickly become a wide spread practice for capturing functional requirements.

To detect buffer overflow first we need to design overflow detection tools. For detecting overflow we need find all buffer size of given program. For doing this we need to browse a program on which we need to verify buffer overflow. We need to design a window for uploading a program. Next we need to verify buffers and their sizes of the given program. Then we have to verify the size of the buffer whenever we try to insert data into buffer and displays a message when a overflow occurs. These are all under buffer overflow detection tools.

Non Functional Requirements:

In systems engineering and requirements engineering, a nonfunctional requirement is that which species criteria that can be used to judge the operation of a system rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior of functions.

Qualities that is nonfunctional requirements can be divided into two main categories:

Execution qualities such as a security and usability which are observable at run time

Evolution quality such as testability, maintainability, extensibility and scalability which are embodied in a static structure of software system

Technical View

JSP

Jsp technology enables you to mix regular static html with dynamically generated content from serves. Separating the static html from the dynamic content provides a number of benefits over serves alone.

JSP compared to Asp

Jip and asp are fairly similar in the functionality that they provide. Jip may have slightly higher learning curve. Both allow embedded code in an html page, session variables Platform i.e., NT, JSP can operate on any platform that conforms to the J2EE specification. Jip allow component reuse by using JavaBeans and Ebbs. Asp provides the use of Com/active controls.

JSP compared to serves

A served is java class that provides special server side service. It is hard to write HTML code in servlets.you need to have lots of println statement to generate HTML.

Description

JSP looks like html, but they get compiled into java servlets the first time they are invoked. The resulting servlet is a combination of the html from the jsp file and embedded dynamic content specified by the new tags. That is not to say that jsp must contain html. Some of them contain only java code; this is particularly useful when the jsp is responsible for a particular task like maintaining application flow.

Everything in Jsp can be broken into 2 categories

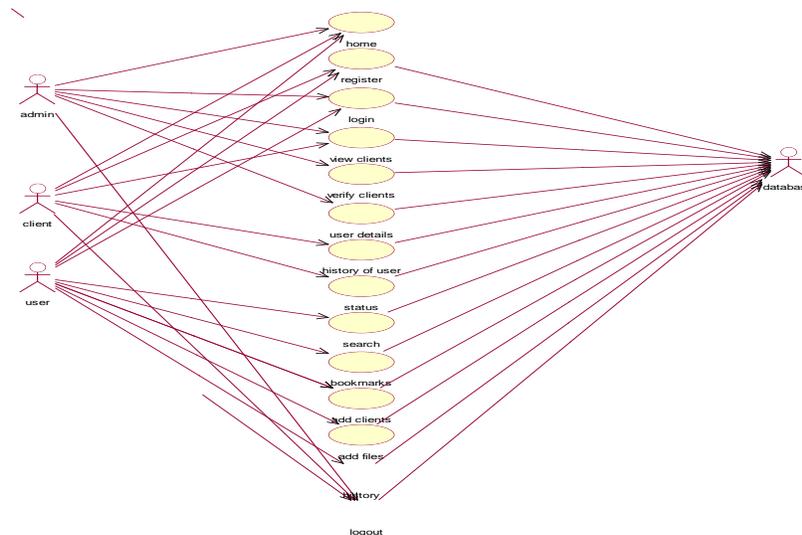
1. Elements that are processed on the server.
2. Template data or everything other than elements that the engine processing the Jsp ignores.

JSP Architecture

Jsp are built on top of sun's servlet technology. Jsp is essentially an html page with special jsp tags embedded. These jsp tags can contain java code. The jsp file extension is .jsp rather than .htm or .html. The jsp engine parses the .jsp and creates a java servlet source file. It then compiles the source file into a class file; this is done the first time and this why the jsp is probably slower the first time it is accessed. Any time after this, the special compiled servlet is executed and is therefore returns faster.

SYSTEM Design

Unified Modeling Language (UML): is a standardized general-purpose modeling language in the field of software engineering. It is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. UML combines best techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies. Use Case Diagram: A Use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals, and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



IV. Implementation Of System

It is the age of information technology and data & database play a very key role in this age. A layperson of these days needs no introduction to databases, whether it is a personal telephone directory or the bank passbook database are omnipresent. In this session we learn about database management systems in general with an emphasis on the relational model of the DBMS. The conventional data processing approach is to develop a program (or many programs) for each application. This result in one or more data files for each application. Some of the data may be common between files. However one application may require the file to be organized on a particular field, while other application may require the file to be organized on another field. A major drawback of the conventional method is that the storage access methods are built in to the program. Therefore, though the same data may be required by two applications, the data will have to be sorted in two different places because each application depends on the way that the data stored.

There are various drawbacks of conventional data file processing environment. Some of them are listed below:

Data Redundancy

Some data elements like name, address, identification code, are used in various applications. Since data is required by multiple applications, it is stored in multiple data files. In most cases,

there is a repetition of data. This is referred to as data redundancy, and leads to various other problems.

Data Integrity Problems

Data redundancy is one reason for the problem of data integrity. Since the same data is stored in different places, it is inevitable that some inconsistency will creep in.

Data Availability Constraints

When data is scattered in different files, the availability of information from a combination of files is constrained to some extent.

Database Management System

A database management system (DBMS) consists of a collection of interrelated data and a set of programs to access the data. The collection of data is usually referred to as the database. A Database system is designed to maintain large volumes of data. Management of data involves:

- Defining the structures for the storage of data
- Providing the mechanisms for the manipulation of the data
- Providing for the security of the data against unauthorized access

Users of the DBMS

Broadly, there are three types of DBMS users:

- The application programmer
- The end user
- The database administrator (DBA)

The application programmer writes application programs that use the database. These programs operate on the data in the database. These operations include retrieving information, inserting data, deleting or changing data.

The end user interacts with the system either by invoking an application program or by writing their queries in a database query language. The database query language allows the end user to perform all the basic operations (retrieval, deletion, insertion and updating) on the data.

The DBA has to coordinate the functions of collecting information about the data to be stored, designing and maintaining the database and its security. The database must be designed and maintained to provide the right information at the right time to authorized people. These responsibilities belong to the DBA and his staff.

Advantages of a DBMS

The major advantage that the database approach has over the conventional approach is that a database system provides centralized control of data. Most benefits accrue from this notion of centralized control.

Redundancy Can Be Controlled

Unlike the conventional approach, each application does not have to maintain its own data files. Centralized control of data by the DBA avoids unnecessary duplication of data and effectively reduces the total amount of data storage required. It also eliminates the extra processing necessary to trace the required data in a large mass of data present. Any redundancies that exist in the DBMS are controlled and the system ensures that these multiple copies are consistent.

Inconsistency Can Be Avoided

Since redundancy is reduced, inconsistency can also be avoided to some extent. The DBMS guarantees that the database is never inconsistent, by ensuring that a change made to any entry automatically applies to the other entries as well. The process is known as propagating update.

The data can be shared

A database allows the sharing of data under its control by any number of application program or users. Sharing of data does not merely imply that existing applications can share the data in the database, it also means that new applications can be developed to operate using the same database.

Standards Can Be Enforced

Since there is centralized control of data, the database administrator can ensure that standards are maintained in the representation of the stored data formats. This is particularly useful for data interchange, or migration of data between two systems.

Security Restrictions Can Be Applied

The DBMS guarantees that only authorized persons can access the database. The DBA defines the security checks to be carried out. Different checks can be applied to different operations on the same data. For instance, a person may have the access rights to query on a file, but may not have the right to delete or update that file. The DBMS allows such security checks to be established for each piece of data in the database.

Integrity Can Be Maintained

Centralized control can also ensure that adequate checks are incorporated in the DBMS to provide data integrity. Data integrity means that the data contained in the database is both accurate and consistent. Inconsistency between two entries can lead to integrity problems. However, even if there is no redundancy, the data can still be inconsistent. For example a student may have enrolled in 10 courses in a semester when the maximum number of courses one can enroll in is 7. Another example could be that of a student enrolling in a course that is not being offered that semester. Such problems can be avoided

in a DBMS by establishing certain integrity checks to be carried out whenever any update operation is done. These checks can be specified at the database level, besides the application programs.

Data Independence

In non-database systems, the requirement of the application dictates the way in which the data is stored and the access techniques. Besides, the knowledge of the organization of the data, the access techniques are built into the logic and code of the application. These systems are data dependent. Consider this example, suppose the university has an application that processes the student file. For performance reason, the file is indexed on the roll number. The application would be aware of the existing index, and the internal structure of the application would be built around this knowledge. Now consider that the some reason, the file is to index on the registration data. In this case it is impossible to change the structure of the stored data without affecting the application too. Such an application is a data dependent one.

Features of RDBMS

- The ability to create multiple relations and enter data into them
- An interactive query language
- Retrieval of information stored in more than one table

Database Design

Having identified all the data in the system, it is necessary to arrive at the logical database design. Database design involves designing the conceptual model of the database. This model is independent of the physical representation of data. Before actually implementing the database, the conceptual model is designed using various techniques.

The requirements of all the users are taken into account to decide the actual data that needs to be stored in the system. Once the conceptual model is designed, it can then be mapped to the DBMS/RDBMS that is actually being used. Two of the widely used approaches are Entity-relationship (E/R) Modeling and Normalization.

The E/R model is an object based model and is based on a perception of the real world that is made up of a collection of objects or entities and the relationships among these. E/R modeling is generally used as a top down approach for new systems.

Entity

Entity is an object or place or event, which can be stored on the system. A physical object can be as employee, customer, and machinery. An abstract object can be as dept, accounting. An event can be as registration or application form. A place can be as city, state. Before a table is created it is known as entity. It is denoted as a rectangle diagram.

Attribute

Attribute is describing the entity. Example an entity employees can contain empno, ename, sal, hiredate etc. It is represented by a circle.

Relation

A "Relation" is a two-dimensional table. It consists of 'rows' which represent records and 'columns' which show the attributes of the entity. A relation is also called a file, it consists of a number of records, which are also called as tuples. Record consists of a number of attributes, which are also known as fields or domains.

In order for a relational structure to be useful and manageable, the relation tables must first be "normalized".

Some of the properties of a relation are

No duplication -In the sense that no two records are identical

Unique Key-Each relation has a unique key by which it can be accessed

Order -There is no significant order of data in the table.

In case we want the names of all the employees whose grade is 20, we can scan the employee relation noting the grade. Here the Unique key is the employee number.

Normalization

Normalization is a process of simplifying the relationship between data elements in a record. It is the transformation of complex data stores to a set of smaller, stable data structures.

Normalized data structures are simpler, more stable and are easier to maintain. Normalization can therefore be defined as a process of simplifying the relationship between data elements in a record.

Purpose for Normalization

- To permit simple retrieval of data in response to query and report requests.
- To simplify the maintenance of the data through updates, insertions and deletions.
- To reduce the need to restructure or reorganize data when new application requirements arise.

Steps of Normalization

It consists of basic three steps

- First Normal Form, which decomposes all data groups into two-dimensional records.
- Second Normal form, which eliminates any relationships in which data elements do not fully depend on the primary key of the record.
- Third Normal Form which eliminates any relationships that contain transitive dependencies.

System Testing And Implementation:

Test plan:

Once the source code has been generated, the software must be tested to uncover as many errors as possible before delivery to the customer. Software testing is a critical element of software assurance and represents the ultimate review of specification design and code generation.

Testing Objectives

Testing is a process of executing a program with the intent of finding an error.

A good test case is one that has a high probability of finding an as yet undiscovered.

A successful test is one that uncovers an as yet undiscovered error.

All tests should be traceable to customer requirements.

Tests should be planned large before testing begins.

Testing should begin "In the Small" and progress towards "In the Large".

Unit Testing

The first step in the testing is the unit testing. Unit test is normally considered as an adjunct to the coding step. After the coding has been developed, received and verified for correct syntax, unit testing begins. The standalone modules were tested individually for their correct functionality, with the corresponding data. This ensures the reliability of the modules when integrated. Each and every module is tested independently with sample data and it was found that all modules are properly functioning. Using the unit test plans, prepared in the design phase of the system as a guide, important control paths are tested to uncover errors within the boundary of the modules. Boundary conditions were checked, all independent paths were exercised to ensure that all statements in the module are checked at least once and all error handling paths were tested. Each unit was thoroughly tested to check if it might fall in any possible situation. This testing was carried out during the programming itself. At the end of this testing phase, each unit was found to be working satisfactory, as regard to the expected output from the module.

Integration Testing

Functional testing was performed on the system by raising the demand with an eye to check all the validations. The total processing of the system is satisfactory with the following results.

All the validations are clearly notified to the user regarding customer orders, stock maintenance, supplier quotations, and bill preparation etc.

Almost all the functional errors, data storage errors and all types of logical errors are tested successfully.

Acceptance Testing

User acceptance test of a system is the factor for the success of the system. The system under consideration was listed for user acceptance by keeping constant touch with the perspective user of the system at the time of design, development and making changes whenever required for unit testing. The requirements of the customer are gathered at regular intervals at the developing site itself. The problems that are to be visualized through this tool are been gathered by the customer and are reported.

The user at the user's site carried this test. Live data entered and the system's output was compared with what was manually prepared. Here the system has met the user's requirement in the following fields:

Data Entry

Error Handling

Reporting and corrections

Data Access Protections

System Output

Implementation

Implementation includes all those activities that take place to convert the old system to the new system. The new system will replace the existing system. The aspects of implementation are as follows.

Conversion, Post Implementation Review.

Conversion

Conversion means changing from one system to another. The objective is to put the tested system into operation. It involves proper installation of the software package developed and training the operating staff.

The software has been installed and found to be functioning properly. The users how to be trained to handle the system effectively. Sample data provide to the operating staff and were asked to operate on the system. The operating staffs now have a clear out look of the software and are ready for practical implementation of the package.

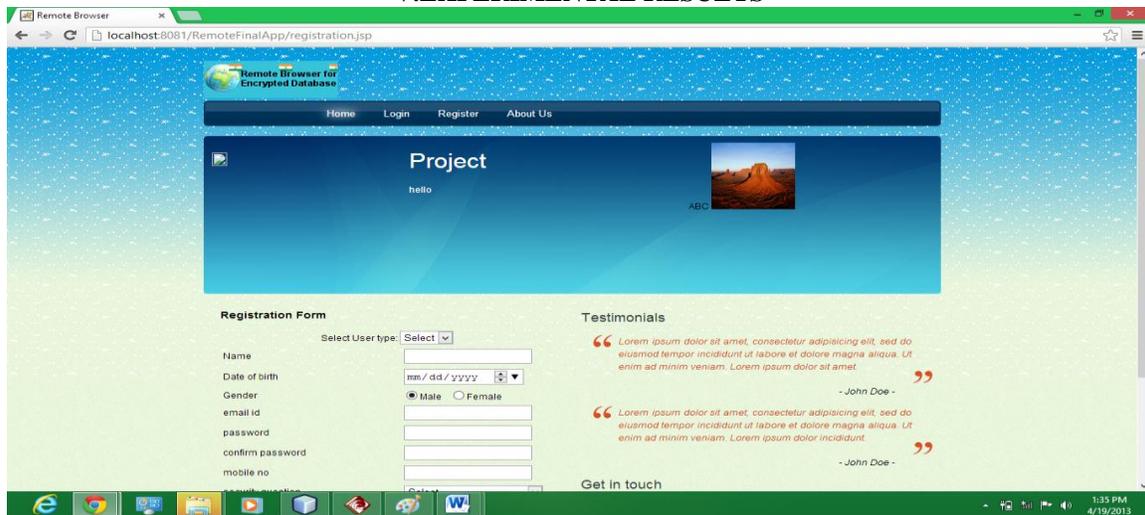
Positive test cases

The positive flow of the functionality must be considered.

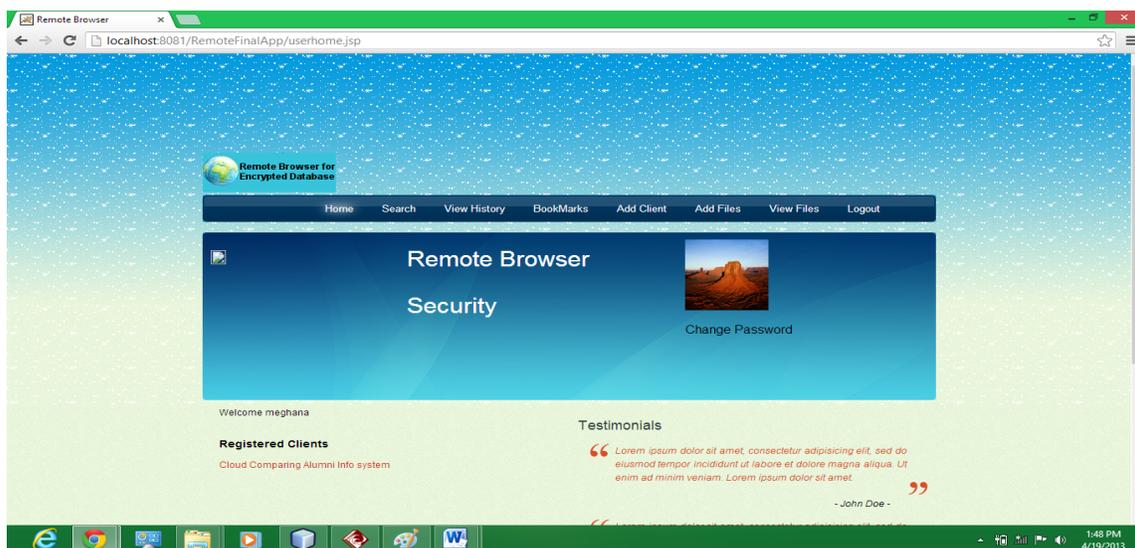
Valid inputs must be used for testing.

Must have the positive perception to verify whether the requirements are justified

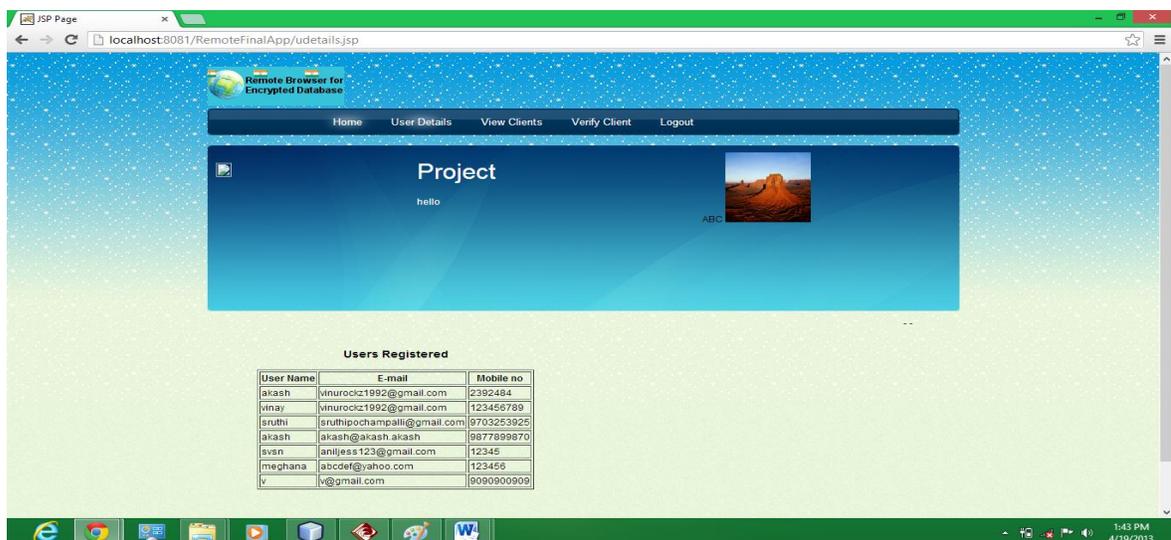
V.EXPERIMENTAL RESULTS



Register for client



Admin view clients



User search

VI. Conclusion

This system has been developed successfully incorporate all the requirements. Appropriate care has taken during database design maintain database integrity and to avoid redundancy of data. This site was developed in such a way that any further modifications needed can be easily done. User feels freely while using this site. In this all technical complexities are hidden. This site is a more user friendly. The quality futures like correctness, efficiency, usability, maintainability, portability, accuracy, errors, tolerance, expandability and communicatively all are successfully done. The advanced technology which we are using is Ajax and Java it provides user-friendly interface, more flexibility, efficient design for an user which reduce the traffic travels between the client and the server, response time is faster so increases performance and speed of an web application. Remote Browser for Encrypted Database, Web Application is a creative idea with a very bright future with further scope for advancements. The opportunities provided from this medium are immense and many organizations can make use of this medium for better their practices. With the help of social networking they can advertise or communicate in a more efficient way.

VII. Future Enhancement

There is always a room for improvement in any software package, however good and efficient it may be. The important thing is that the website should be flexible enough for further modifications. Considering this important factor, the web site is designed in such a way that the provisions are given for further enhancements. At present this website provides all the information using static pages and reservation forms.

In future we can enhance our project by providing options like connecting people through social networking websites we can improve our login validations.

You can also expect customized form of web Applications for your mobile phones with which you can make your online browsing more easy and comfortable.

We may change our Application design for providing more efficiency through the upcoming technologies which are going to be launched in the future

We have countable number of client's and more users for the services which are providing to the user, we can expect more client's and user's in the future for the advanced services like, live messenger for online chatting, mail services to send emails to other users which are connected with our web application, online video chat application to user's, online drive for sharing documents, grouping service to share the views, files in an particular group selected by the user. online saving live documents into the website, we can also provide softwares to users in the future like.. online document editor, online image editor, and more.

References

- [1] D. Danchev, "ZDNET: french hacker gains access to twitter's admin panel," April 2009. [Online]. Available: <http://www.zdnet.com/blog/security/french-hacker-gains-access-totwitter-admin-panel/3292>
- [2] P. Allen, "Obama 's Twitter password revealed after french hacker arrested for breaking into U.S. president's account," March 2010. [Online]. Available: <http://www.dailymail.co.uk/news/article-1260488/Barack-Obamas-Twitter-password-revealed-French-hacker-arrested.html>
- [3] F. Rocha and M. Correia, "Lucy in the sky without diamonds: Stealing confidential data in the cloud," in Proceedings of the First International Workshop on Dependability of Clouds, Data Centers and Virtual Computing Environments, Hong Kong, ser. DCDV '11, June 2011.
- [4] M. Van Dijk and A. Juels, "On the impossibility of cryptography alone for privacy-preserving cloud computing," in Proceedings of the 5th USENIX conference on Hot topics in security, ser. HotSec'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 1–8. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1924931.1924934>
- [5] J. Pepitone, "Dropbox's password nightmare highlights cloud risks," June 2011.
- [6] M. Ben-Salem and S. J. Stolfo, "Modeling user search-behavior for masquerade detection," in Proceedings of the 14th International Symposium on Recent Advances in Intrusion Detection. Heidelberg: Springer, September 2011, pp. 1–20.
- [7] B.M. Bowen and S. Hershkop, "Decoy Document Distributor: <http://sneakers.cs.columbia.edu/ids/fog/>," 2009. [Online]. Available: <http://sneakers.cs.columbia.edu/ids/FOG/>

Biography:



Mr B.Srinivasulu, Post Graduated in Computer Science Engineering (M.Tech) From JNTUH,2010, and graduated in Computer Science Engineering (B.Tech)from JNTU Hyderabad, 2008. He is working presently as Assitant Professor in Department of Computer Science & Engineering in **St.Martin's Engineering College**, RR Dist, A.P, INDIA. He is has 2+ years Experience.His Research Interests Include Software Engineering, Network Security & Cloud Computing. Email: srinivas.nivas47@gmail.com



Mr Umamaheswararao inkollu., Post Graduated in Computer Science Engineering (M.Tech) From JNTUH,, and graduated in Computer Science Engineering (B.Tech)from JNTU Hyderabad, He is working presently as Assitant Professor in Department of Computer Science & Engineering in **St.Martin's Engineering College**, RR Dist, A.P, INDIA. He is has 2+ years Experience.His Research Interests Include Software Engineering, Network Security & Cloud Computing.

Email: inkoluchanti@gmail.com



Mr A.Bhima Sanakram Post Graduated in Computer Science & Technology (M.Tech), Andhra University, 2010, and graduated in Information Technology (B.Tech) from JNTU Hyderabad, 2008. He is working presently as Assistant Professor in Department of Computer Science & Engineering in St. Martin's Engineering College, RR Dist, A.P, INDIA. He is has 2+ years Experience. His Research Interests Include Software Engineering & Cloud Computing. Email: dhana1206@gmail.com



Mr Naga Babu G, graduated in B.Tech (CSE) from JNTU Hyderabad, 2011. He is working presently as Assistant Professor in Department of Computer Science & Engineering in St. Martin's Engineering College, RR Dist, A.P, INDIA. He is has 2+ years Experience. His Research Interests Include Software Engineering & Cloud Computing. Email: nag.garipati@gmail.com