GLOBAL JOURNAL OF ENGINEERING, DESIGN & TECHNOLOGY
(Published By: Global Institute for Research & Education)
www.gifre.org

Creation of a Secure Chat System

Adewale O Adebayo*, Komolafe Oyindolapo, Ogu Chiemela, Omotosho Toluwalase
School of Computing and Engineering Sciences, Babcock University, P.M.B.21244 Ikeja, Lagos, Nigeria

Abstract

Secure chat system is essential for effective and efficient communication in succeeding organizations. Current ‘free’ chat systems make breach risk of confidentiality probable, and organizations lose possession of logged chat messages. Their designs are also not readily available for examination and subsequent improvement. There is therefore the need for the design and creation of a private chat system which this research addressed. Design and creation research strategy was adopted and data collection was through existing documents and structured interviews. Incremental software development method was adopted. The outcome provides secure private chat system that minimizes the breach risk of confidentiality through encryption and localization, and grants user organization possession of logged chat messages.

Keywords: *Chat System, Private Chat, Secure Chat, Create Chat System, Design Chat System*.

1.0 Introduction

Any organization having any number of employees needs a communication system. A chat system, which could be intranet or internet based, can be used to share information, make inquiry, among others. Chat is to participate in a synchronous text, video, audio, or multicast exchange of remarks with one or more people over a computer network (Wiki-online Chat, 2013). There is the need to ensure confidentiality of communication to breed honest and frank chatting free from fear of eavesdropping and breach of privacy. Organizations should also keep possession of their chat messages. The capability to securely chat with a colleague in an organization is to a great advantage against competitors who do not have such facility. A Secure Chat System is a system which enhances communication between two or more people within an organization or over the internet in a way that seriously attempts to be free from risk of interception by or involvement of unauthorized persons.

There are a number of chat system available; Voodoo (Parniak, 2009), MUSHclient (Gammon, 2005), Google Talk (Google Talk Beta, 2011; Google Talk Help, 2013), Skype (Higginbotham, 2008), Facebook (Yadav, 2006), among others. These ‘free’ chat system providers use clients’ information for marketing and may sell it to prospective buyers who need them. Possession of chat messages is also lost to the provider. Having a private chat system will reduce all these probable confidential, privacy, and possession security risks. In addition, chat system design is not readily available as a guide and as basis for further improvement, to intending developers.

The goal of this work is, therefore, to design and implement private chat application software that will enable effective and efficient text-based communication between users. The chat system will privatize and secure the transfer of information and communication. Security of this chat system would be limited to password and username for authentication, Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) to detect if the other person is human, and Encryption of messages to improve confidentiality. It will also grants user organisation possession of logged chat messages.

The design and creation research strategy (Oates, 2009) was adopted. The data collection method was existing documents and structured interviews. The data collected was analyzed qualitatively. Various literatures on the subject matter, including on Voodoo, MushClient, and Skype chat systems, were examined with the intention of incorporating learnt ideas. The software development methodology used was Incremental Development because it provides for the development of business information systems in a series of small implementable sub-stages. It also promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change (Hughes & Cotterell, 2009; Whitten, Bently & Dittman, 2001). The application development environment includes: MySQL (which supports the relational model and allows the integration and association of database tables), XAMPP Server (All Platforms, Apache, MySQL, PHP, Perl - a virtual web server used to simulate connections to external software applications such as Apache and Windows), and programming languages of choice (PHP - runs natively on Apache server and can easily extend or integrate with Java, .Net, Microsoft Exchange Server, web services and more, JavaScript - scripting language that Web server administrators use to manage the server and connect its pages to other services, such as back-end databases and search engines for users looking up information, Ajax - brings desktop applications’ features into the web), and CSS - allows document authors to specify the presentation of elements on a web page separately from the structure of the document). Object oriented concept was adopted in the development of the system. PHP language was used to build the login features, CSS used to build the design and presentation, and HTML was used for the structure of the system.

2.0 The Secure Chat System

The specification and requirements, design, and implementation of the proposed private chat system are provided in succeeding sections.

2.1 System Specification and Requirements

Current ‘free’ chat systems have unique features and characteristics, and work over the internet. Their setback is that they offer limited privacy and confidentiality, and organisations lose possession of logged chat exchanges to the chat system providers. While some chat systems employed advanced technology and others implement degrading features. It can be concluded that chat system follow a seeming conventional pattern (Chamberglain, William & Jarred, 2008; Finkler & Dao, 2001).

Having critically observed both the limitations and unique features of the studied chat systems, the following are the main features and functionalities suggested: Security (users should be prompted for registered username and password for authenticated, automatically logged out after the session period has expired, and each message sent should be encrypted and decrypted at the receiver’s end; the use of CAPTCHA should included; the system database should be hosted on a local server or the intranet of the organization), Message broadcast (fast means of sending messages or information to multiple employees at the same time, should be enabled), and User-friendly interface (which can be used by both technicians and non-technicians should be built).

User requirements specify the services the application should provide for the users of the system. System requirements give the detailed description of the system services (Hughes & Cotterell, 2009; Whitten, Bently & Dittman, 2001). The following are the user and system requirements for the project: User should be able to do a private chat with other user; User should be able to broadcast messages to other users; The user interface shall be user-friendly and extremely low learning curve; The system shall allow authentication and grant access to valid users; The system shall enforce encryption of messages and decryption once it gets to the recipient; Each chat room displays the discussion thread and list of on-line users; Each chat entry is marked with the user name and the time stamp of the entry; Only the administrator can add new users; The user should see all other users that post messages in the last specified number of minutes.

The Functional Requirement are as follows: In the event of accidental exit of application, the session will be restored with all messages when the user re-executes the application; The chat frame shall have main chat window, user list, text-box for chat entry and a send button; Chat history should be saved as logs on private host, and text files will be used; Each entry of the log should be marked with the username and timestamp of entry.

The Non-functional Requirements are as follows: Interoperability is a property that depicts the ability of diverse systems and organizations to work together (inter-operate); Robustness is the ability of a computer system to cope with errors during execution or the ability of an algorithm to continue to operate despite abnormalities in input, output, among others; Response time is the time the system takes to react to a given input; Safety is the state or condition of being protected against any event which could be considered non-desirable; Documentation is the process of providing evidence.

2.2 System Design

In the preliminary or general design, the features of the new system are specified. The costs of implementing these features and the benefits to be derived are estimated. In the structure or detailed design stage, the design of the system becomes more structured. Structure design is a blue print of a computer system solution to a given problem having the same components and inter-relationship among the same components as the original problem. Input, output and processing specifications were drawn up in detail. In the design stage, the programming language and the platform in which the new system will run were also decided. Input, output and processing specifications were also drawn up in detail subsequently.

2.2.1 Process Design

The two actors of the chat system are the Administrator and the User. The administrator is the super user and has control over all the activities that can be performed. The administrator can add new users and give them their default password. The administrator can also create a new department in the advent of a new department. The administrator can view all registered users with their details. The administrator can delete a user, and can also generate reports (Figure 1).

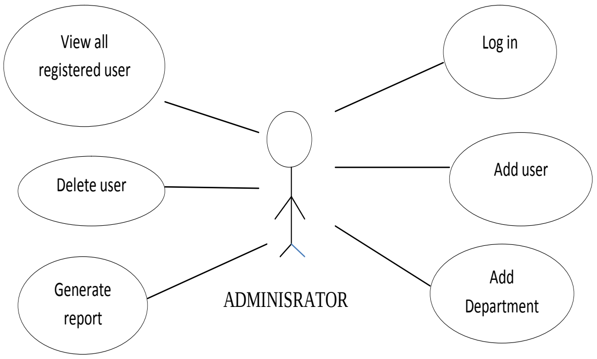


Figure 1: Administrator Use Case

A standard user after a successful login can change his/her password, edit his/her profile. The user can broadcast messages to all other users, read broadcasted messages and also delete. The user can also chat with online users (Figure 2).

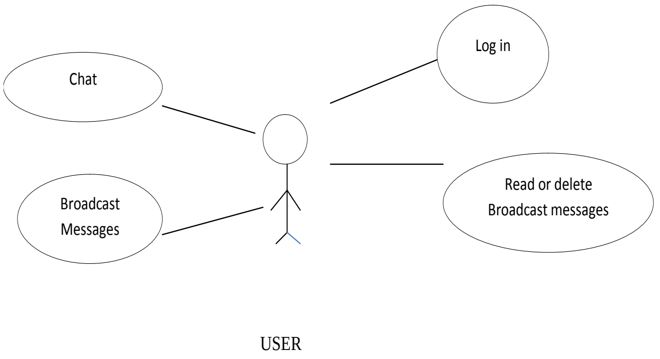


Figure 2: User Use Case

2.2.2 Application Design

The application has five modules based on the requirements specifications. The modules are Database, Login, Navigation, Data Manager, and Generation modules. Database Module - The design of a database has to do with the way data is stored and how that data is related. The software will incorporate a database of online users, messages, and broadcast of an authenticated user. This database would generate error reports when an empty input is entered or when a conflict or error in authentication occurs. Login Module - This will enable validation of user such that only authenticated users with valid credentials will be granted access to the software. This module will also incorporate functionalities for change of passwords. All users will be created on the virtual sever. Navigation Module - This module comprises all navigation items; buttons and links and their associated pages or functions. The site-map functionality is also encompassed in this module. Data Manager Module - This module provides the interface on which the user will input required data. The module is also responsible for mapping the inputted data to their respective tables on the database. Generation Module - This module is responsible for the allocation of messages to their respective receiver on the message table. It performs certain check functions to ensure that no clash whatsoever occurs before mapping out.

Class Diagram shows the list of all the classes to be used in the design of the project (Figure 3).

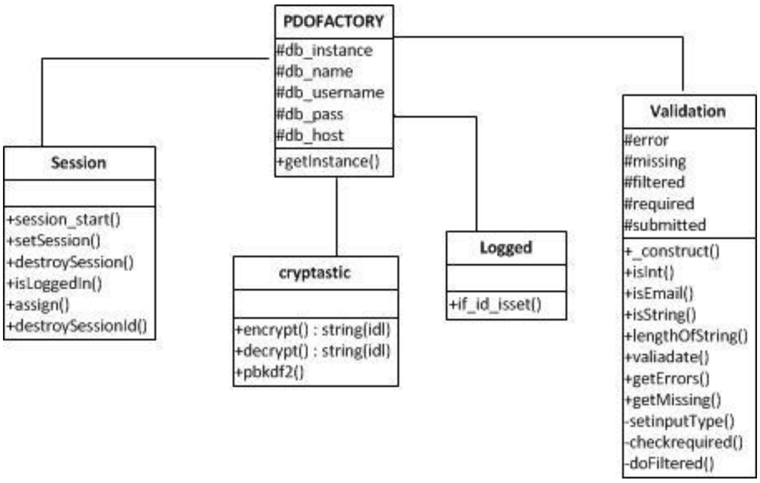
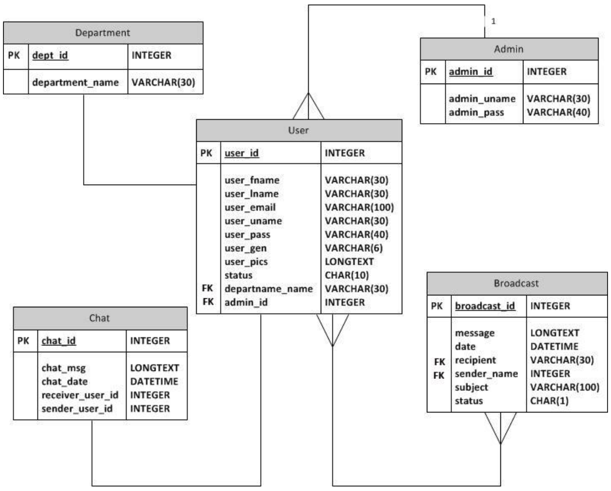


Figure 3: Class Diagram

2.2.3 Database Design

The entities involved in this project are Administrator, user information (User), Department, Broadcast and Chat. The entity relationship model gives a brief description of each of this entities and their individual attributes. This makes the arrangement of data in the database much more coordinated and the database design model easy to implement (Date, 2003).

The Administrator represents the authenticated user with the administrative privileges. The Administrator attributes are: admin_id (the primary key. This key field identifies an administrator. Data validation is switched on ensuring that the field is unique, i.e. no two records have the same admin_id), Admin_pass (key field to hold the password of the admin), and Admin_uname (This field holds the admin’s name). The User represents authenticated user without administrative privileges. The User attributes are: user_id (This is the primary key. This key field identifies a person in the chat system. Data validation is switched on ensuring that the field is unique), User_pass (key field to hold the password of the user), User_lname and user_fname (These two fields hold the user’s name), user_email (The email field is not essential and is just there so that if users have any problems, then the administrator can contact them), Status (The status field tracks whether the user is currently online. This field is set by the login and logout scripts), User_uname (This field holds the user’s application name), User_gen (this holds the gender of the user), Department_name (the field identifies the department a particular user belongs to. It is a foreign key to the department table), User_pics (this field holds the reference to the location where the user’s profile picture is saved), and Admin_id (this is a foreign key referencing the admin table, to identify the admin that registered the user). The Department represents the various departments that present in an organization. The Department attributes are: Department_id (primary key), and Department_name (holds the name of the department). The Broadcast represents broadcast messages. The Broadcast attributes are: broadcast_id (primary key), message (This field holds the actual text of the message that was broadcasted), sender_name (foreign key, reference to user_tbl table; the user that sends the broadcast's name), Recipient (the users receiving the broadcast messages), Subject (the topic of the message), Time (the current time the message was sent), and Status (if the message has been read or not by the recipient). The Chat represents actual messages during chat sessions. The Chat attributes are: chat_id (This field is used as the primary key for the Chat table), chat_date (This field holds the date and time the message was created), chat_message (This field holds the actual text of the message), Receiver_user_id (this field identifies the receiver of the message being sent), and user_id (foreign key, reference to user table). The attendant entity relationship diagram is presented in Figure 4.



Department		
PK	dept_id	INTEGER
	department_name	VARCHAR(30)

Admin		
PK	admin_id	INTEGER

	admin_uname
	admin_pass
		VARCHAR(30)
		VARCHAR(40)

User		
PK	user_id	INTEGER

FK
FK
	user_fname
	user_lname
	user_email
	user_uname
	user_pass
	user_gen
	user_pics
	status
	departname_name
	admin_id
		VARCHAR(30)
		VARCHAR(30)
		VARCHAR(100)
		VARCHAR(30)
		VARCHAR(40)
		VARCHAR(6)
		LONGTEXT
		CHAR(10)
		VARCHAR(30)
		INTEGER

Chat		
PK	chat_id	INTEGER
	chat_msg chat_date receiver_user_id sender_user_id	LONGTEXT DATETIME INTEGER INTEGER

Broadcast		
PK	broadcast_id	INTEGER

FK
FK
	message
	date
	recipient
	sender_name
	subject
	status
		LONGTEXT
		DATETIME
		VARCHAR(30)
		INTEGER
		VARCHAR(100)
		CHAR(1)

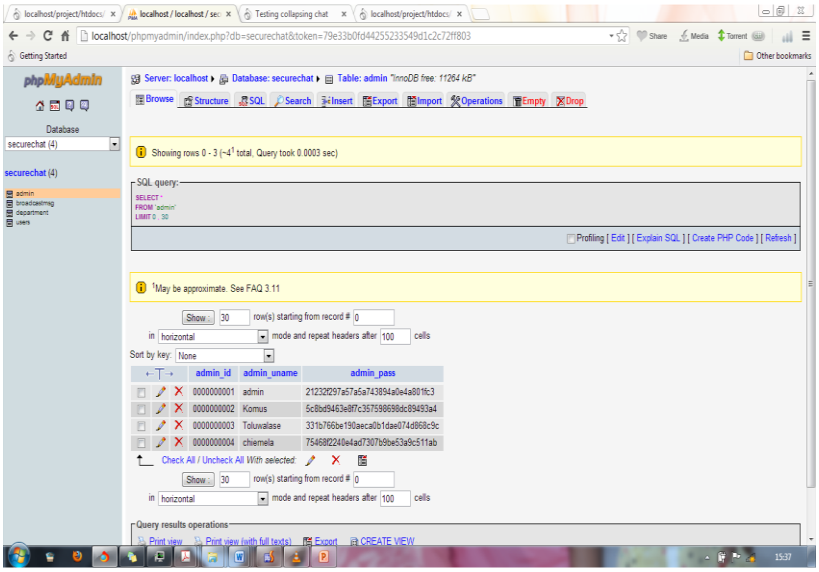
Figure 4: Entity Relationship Diagram

2.3 Implementation and Testing

This section focuses on the implementation of the functional application and the test for defects and necessary properties such as performance and reliability. These tests involve the execution of the application with test data to ensure that all requirements were implemented correctly and that the required quality characteristics were present in the finished product to ensure the deployment of a flawless system.

2.3.1 Component Testing

The major components of the system that were considered are the database and the interface. Component testing was done on individual components in the system to ensure that the system is completely exclusive of faults. Database Testing - An ideal test for this database would be one that ensures that all domain constraints are properly implemented. In order to reduce the complexity of the system, most of the contents of the application were stored in the codes while the dynamic contents were stored in the database. The most critical components of a database are the tables because they are the actual content-holders created to retain a set of related data. The database management system (DBMS) confirms whether all integrity rules governing the tables have been implemented before any data is stored. The domain constraint is the most important of these rules as it gives the allowable values for the attributes of any data. The screen shots of the database tables (Figures 5 to 9) coincide with, and therefore confirm that domain constraints are duly implemented.



localhost/project/htdocs/
localhost / localhost / sec
Testing collapsing chat
localhost/project/htdocs/
localhost/phpmyadmin/index.php?db=securechat&token=79e33b0fd44255233549d1c2c72ff803
Share
Media
Torrent
Getting Started
Other bookmarks
phpMyAdmin
Database
securechat (4)
securechat (4)
admin
broadcastmsg
department
users
Server: localhost
Database: securechat
Table: admin "InnoDB free: 11264 kB"
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Empty
Drop

Showing rows 0 - 3 (~4[1] total, Query took 0.0003 sec)

SQL query:

```
SELECT *
FROM `admin`
LIMIT 0 , 30
```

□ Profiling [Edit] [Explain SQL] [Create PHP Code] [Refresh]

[1]May be approximate. See FAQ 3.11

Show : 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

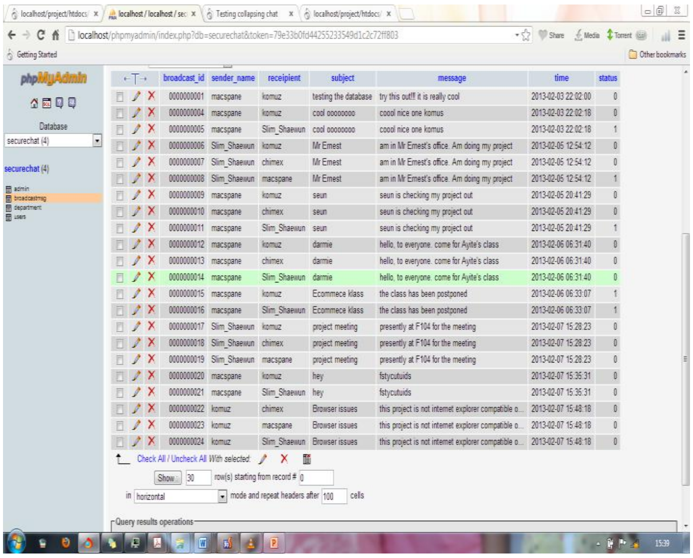
Sort by key: None

←T→			admin_id	admin_uname	admin_pass
□		✕	0000000001	admin	21232f297a57a5a743894a0e4a801fc3
□		✕	0000000002	Komus	5c8bd9463e8f7c357598698dc89493a4
□		✕	0000000003	Toluwalase	331b766be190aeca0b1dae074d868c9c
□		✕	0000000004	chiemela	75468f2240e4ad7307b9be53a9c511ab

Show : 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Figure 5: The Administrator Table


localhost / localhost / sec
Testing collapsing chat
localhost/project/htdocs/
localhost/phpmyadmin/index.php?db=securechat&token=79e33b0fd44255233549d1c2c72ff803
phpMyAdmin
Database
securechat (4)
admin
broadcastmsg
department
users
Query results operations

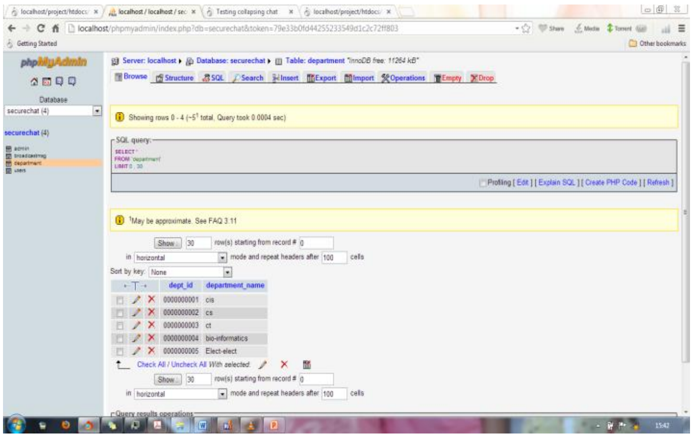
broadcast_id	sender_name	receipient	subject	message	time	status
000000001	macspane	komuz	testing the database	try this out!! it is really cool	2013-02-03 22:02:00	0
000000004	macspane	komuz	cool oooooooo	coool nice one komus	2013-02-03 22:02:18	0
000000005	macspane	Slim_Shaewun	cool oooooooo	coool nice one komus	2013-02-03 22:02:18	1
000000006	Slim_Shaewun	komuz	Mr Ernest	am in Mr Ernest's office. Am doing my project	2013-02-05 12:54:12	0
000000007	Slim_Shaewun	chimex	Mr Ernest	am in Mr Ernest's office. Am doing my project	2013-02-05 12:54:12	0
000000008	Slim_Shaewun	macspane	Mr Ernest	am in Mr Ernest's office. Am doing my project	2013-02-05 12:54:12	1
000000009	macspane	komuz	seun	seun is checking my project out	2013-02-05 20:41:29	0
000000010	macspane	chimex	seun	seun is checking my project out	2013-02-05 20:41:29	0
000000011	macspane	Slim_Shaewun	seun	seun is checking my project out	2013-02-05 20:41:29	1
000000012	macspane	komuz	darmie	hello, to everyone. come for Ayite's class	2013-02-06 06:31:40	0
000000013	macspane	chimex	darmie	hello, to everyone. come for Ayite's class	2013-02-06 06:31:40	0
000000014	macspane	Slim_Shaewun	darmie	hello, to everyone. come for Ayite's class	2013-02-06 06:31:40	0
000000015	macspane	komuz	Ecommece klass	the class has been postponed	2013-02-06 06:33:07	1
000000016	macspane	Slim_Shaewun	Ecommece klass	the class has been postponed	2013-02-06 06:33:07	1
000000017	Slim_Shaewun	komuz	project meeting	presently at F104 for the meeting	2013-02-07 15:28:23	0
000000018	Slim_Shaewun	chimex	project meeting	presently at F104 for the meeting	2013-02-07 15:28:23	0
000000019	Slim_Shaewun	macspane	project meeting	presently at F104 for the meeting	2013-02-07 15:28:23	0
000000020	macspane	komuz	hey	fstycutuids	2013-02-07 15:35:31	0
000000021	macspane	Slim_Shaewun	hey	fstycutuids	2013-02-07 15:35:31	0
000000022	komuz	chimex	Browser issues	this project is not internet explorer compatible o...	2013-02-07 15:48:18	0
000000023	komuz	macspane	Browser issues	this project is not internet explorer compatible o...	2013-02-07 15:48:18	0
000000024	komuz	Slim_Shaewun	Browser issues	this project is not internet explorer compatible o...	2013-02-07 15:48:18	0

Check All / Uncheck All With selected:

Show : 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Figure 6: The Broadcast Table


localhost / localhost / sec
Testing collapsing chat
localhost/project/htdocs/
localhost/phpmyadmin/index.php?db=securechat&token=79e33b0fd44255233549d1c2c72ff803
phpMyAdmin
Database
securechat (4)
admin
broadcastmsg
department
users
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Empty
Drop
Profiling [Edit] [Explain SQL] [Create PHP Code] [Refresh]
Query results operations

Server: localhost ▸ Database: securechat ▸ Table: department "InnoDB free: 11264 kB"

Showing rows 0 - 4 (~5[1] total, Query took 0.0004 sec)

SQL query:

```
SELECT *
FROM `department`
LIMIT 0 , 30
```

[1]May be approximate. See FAQ 3.11

Show : 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Sort by key: None

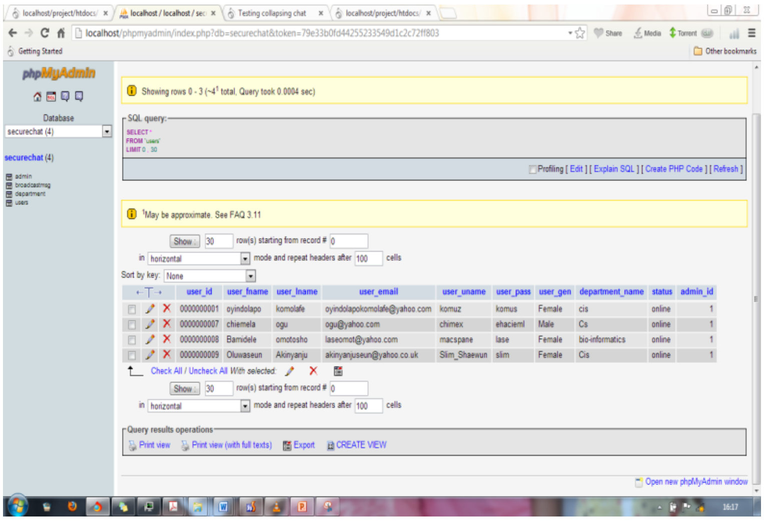
dept_id	department_name
000000001	cis
000000002	cs
000000003	ct
000000004	bio-informatics
000000005	Elect-elect

Check All / Uncheck All With selected:

Show : 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Figure 7: The Department Table


localhost / localhost / sec
Testing collapsing chat
localhost/project/htdocs/
localhost/phpmyadmin/index.php?db=securechat&token=79e33b0fd44255233549d1c2c72ff803
phpMyAdmin
Database
securechat (4)
admin
broadcastmsg
department
users
Profiling [Edit] [Explain SQL] [Create PHP Code] [Refresh]
Query results operations
Print view
Print view (with full texts)
Export
CREATE VIEW
Open new phpMyAdmin window

Showing rows 0 - 3 (~4[1] total, Query took 0.0004 sec)

SQL query:

```
SELECT *
FROM `users`
LIMIT 0 , 30
```

[1]May be approximate. See FAQ 3.11

Show : 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Sort by key: None

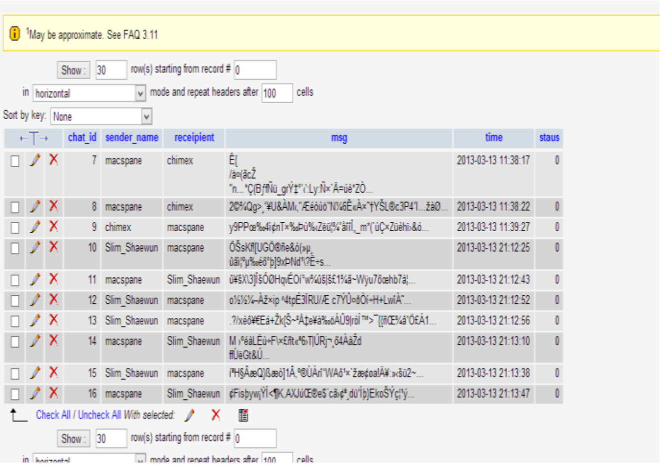
user_id	user_fname	user_lname	user_email	user_uname	user_pass	user_gen	department_name	status	admin_id
0000000001	oyindolapo	komolafe	oyindolapokomolafe@yahoo.com	komuz	komus	Female	cis	online	1
0000000007	chiemela	ogu	ogu@yahoo.com	chimex	ehacieml	Male	Cs	online	1
0000000008	Bamidele	omotosho	laseomot@yahoo.com	macspane	lase	Female	bio-informatics	online	1
0000000009	Oluwaseun	Akinyanju	akinyanjuseun@yahoo.co.uk	Slim_Shaewun	slim	Female	Cis	online	1

Check All / Uncheck All With selected:

Show : 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Figure 8: The User Table



¹May be approximate. See FAQ 3.11

Show : 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Sort by key: None

←T→			chat_id	sender_name	receipient	msg	time	staus
□	✎	✗	7	macspane	chimex	Ê{ /ä¤(ãcŽ "n...*Ç(BƒfÑü_grÝ‡"¿:Ly:Ñ×`Å=úè*ZÒ...	2013-03-13 11:38:17	0
□	✎	✗	8	macspane	chimex	2©¾Qg>¸"¥U&ÀM‹,"Æéòùö"N¼6Ê«À×˜†YŠL®c3P4'l...žàØ...	2013-03-13 11:38:22	0
□	✎	✗	9	chimex	macspane	y9PPœ‰4i¢nT×‰Þù‰Zëü¦¾˜âiïÌ_m*(`ùÇ×Zûèhi›&ó...	2013-03-13 11:39:27	0
□	✎	✗	10	Slim_Shaewun	macspane	ÕŠsKf[UGÖ®ñe&ò(»µ¸ ûãi¦ºµ‰éõ˚þ]9xÞNd°\?Ê÷s...	2013-03-13 21:12:25	0
□	✎	✗	11	macspane	Slim_Shaewun	û¥šX\3]ÌšÓØHqvÉOi°w¾ûš|š£1¾ã~Wÿu7õœhb7à¦...	2013-03-13 21:12:43	0
□	✎	✗	12	Slim_Shaewun	macspane	o½½¼–Àž×ip ^4tpÉ3ÏRU/Æ c7ÝÜ¤ðÒi÷H+LwiÀ˜...	2013-03-13 21:12:52	0
□	✎	✗	13	Slim_Shaewun	macspane	.?/xèõ¥€Eá+Žk{Š~³Å‡e¥ä‰öÁÛ9|ròÌ™>¯{{ñŒ¾ã˜Ö£Å1...	2013-03-13 21:12:56	0
□	✎	✗	14	macspane	Slim_Shaewun	M ›ºéàLÊü÷F\×£ñt«ª6›T|ÚRj¬¸õ4ÀàŽd ffÛëGt&Û...	2013-03-13 21:13:10	0
□	✎	✗	15	Slim_Shaewun	macspane	i ªH§ÂæQ)ßæò]1Å¸ª®ÙÁri˜WAô¹×`žæ¢oa!Ä¥:»‹šû2~...	2013-03-13 21:13:38	0
□	✎	✗	16	macspane	Slim_Shaewun	¢FisþywjŸÌ<¶K.AXJúŒ®eS˘cã‹¢ª¸dü˚Íþ)EkoŠÝç¦ªý...	2013-03-13 21:13:47	0

Check All / Uncheck All *With selected:* ✎ ✗

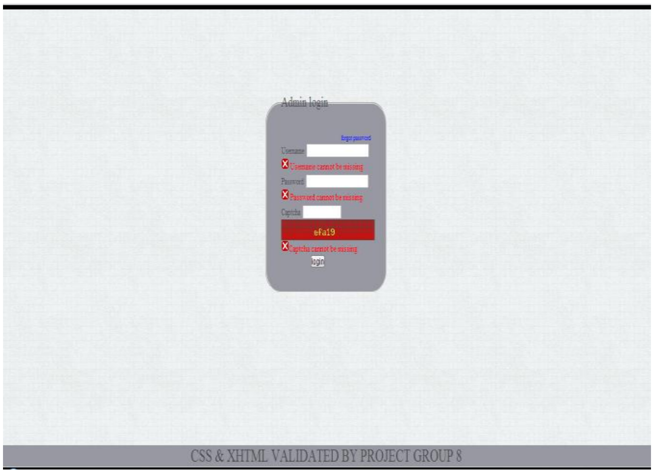
Show : 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Figure 9: The Chat Table

The interface of the system was tested using the Black Box Testing technique, which is based on an analysis of the specification of the system without reference to its internal workings. This was performed by those individuals who have no real knowledge of how the software works to also test ease of usage. Features of the system that are expected to be sensitive to data were tested in this phase.

Login Module Testing was conducted. From the system analysis conducted previously, access denial is expected for any login attempt without authenticated credentials (username, password and CAPTCHA). So it was deduced that an ideal test for the login module would be one in which a user attempts to login to the system without valid details and empty details. The Figures 10 and 11 show the outputs of the test; an error message being displayed as the user attempts to access the system with invalid data (Figure 10) and user attempt to login with empty fields (Figure 11). Hence the login module is validated.



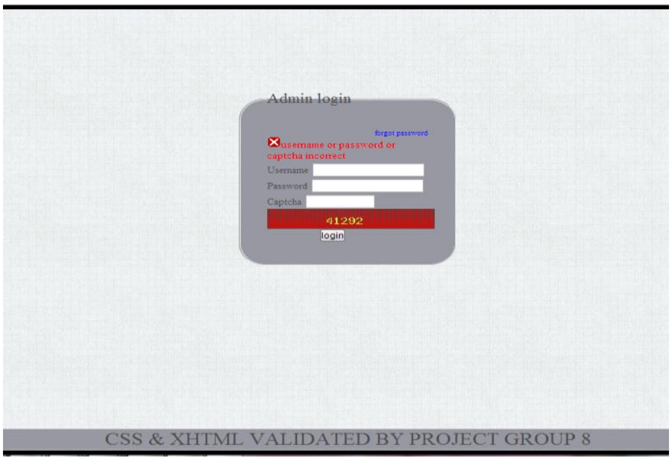
☒ Username cannot be missing

☒ Password cannot be missing

☒ Captcha cannot be missing

CSS & XHTML VALIDATED BY PROJECT GROUP 8

Figure 10: The Login Page Message (invalid data entry)



☒ username or password or captcha incorrect

CSS & XHTML VALIDATED BY PROJECT GROUP 8

Figure 11: The Login Page Message (empty fields' trial)

General Module Testing was done to ensure that the system will not permit any form of conflicting situation. The images below show the output of a test executed where, in the course adding new users with the same username, invalid email address the same email address for two people.

Figures 12 shows successful login page and Figure 13 shows the output of successful user registration.

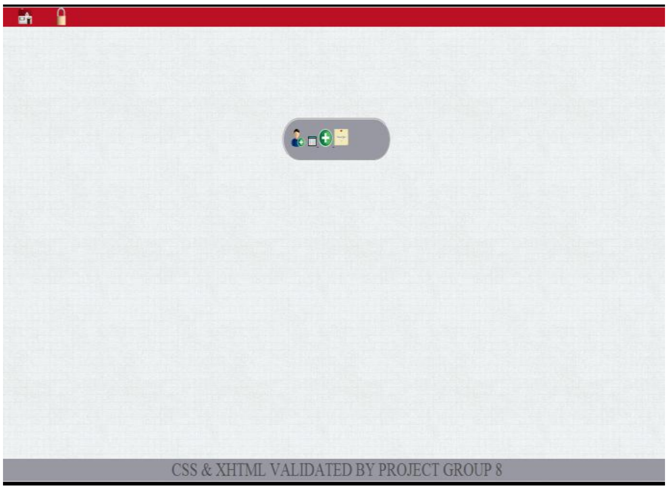
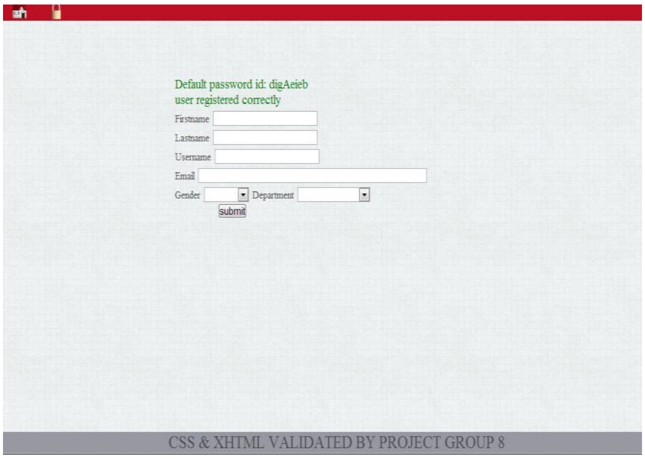


Figure 12: The Home Page (after a successful login)



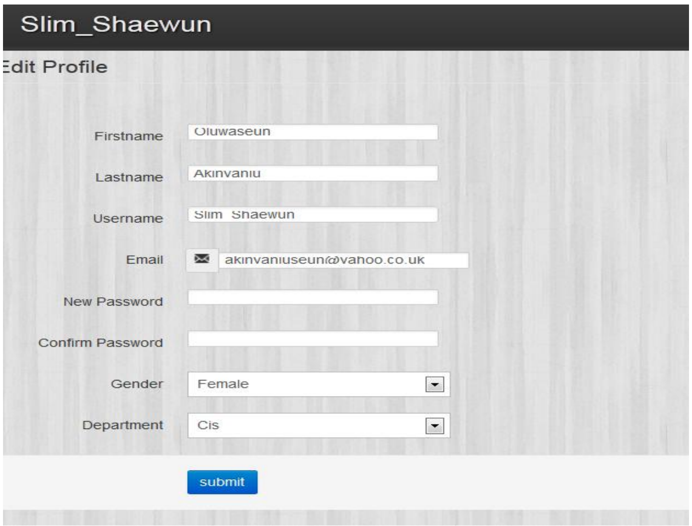
Default password id: digAeieb
user registered correctly

Firstname
Lastname
Username
Email
Gender Department
submit

Figure 13: Add a new user

2.3.2 Process Testing

The entire workflow process was tested avoiding repeating tests. The processes of changing password and sending broadcast messages were noted here. Every user is expected to change the default password. Figure 14 shows the Change Password and Edit Profile Page.



Slim_Shaewun

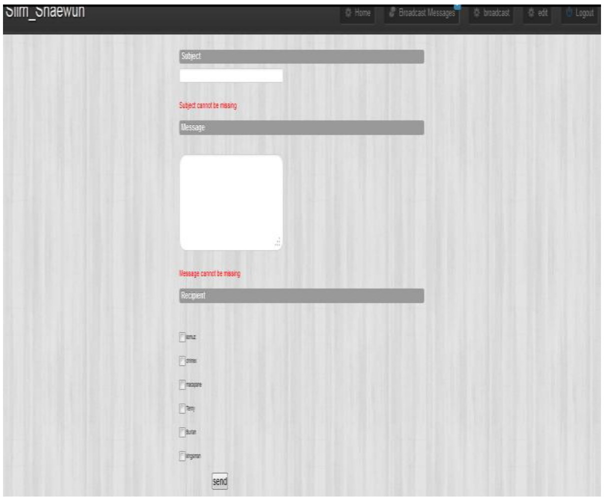
Edit Profile

Firstname	Oluwaseun
Lastname	Akinyanju
Username	Slim_Shaewun
Email	akinyanjuseun@yahoo.co.uk
New Password	
Confirm Password	
Gender	Female
Department	Cis

submit

Figure 14: Change Password and Edit Profile Page

The Send Broadcast test was done to test user friendly interface and also test the proper storage of inputted data of messages sent to others. Figure 15 shows the output of clicking Send without any input, and Figure 16 shows the window of Broadcast messages. Figure 17 shows the window where sent messages could be viewed, and Figure 18 shows the Chat Panel.



Subject

Subject cannot be missing

Message

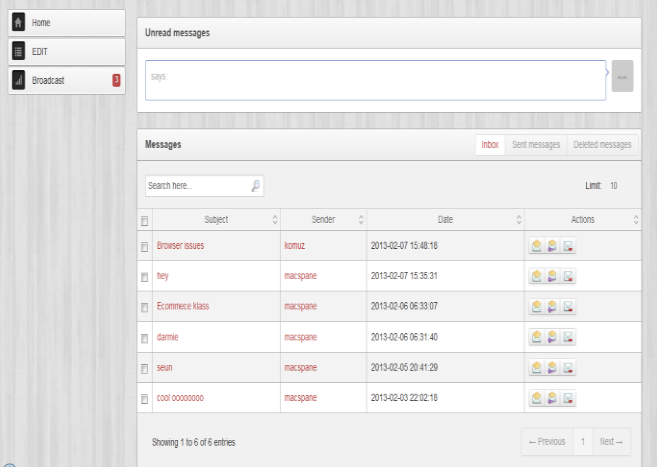
Message cannot be missing

Recipient

□ komuz
□ chimex
□ macspane
□ Tenny
□ Burlan
□ kingsman

send

Figure 15: Output of Clicking Send without any Input



Unread messages

says:

Messages

Inbox | Sent messages | Deleted messages

Search here...

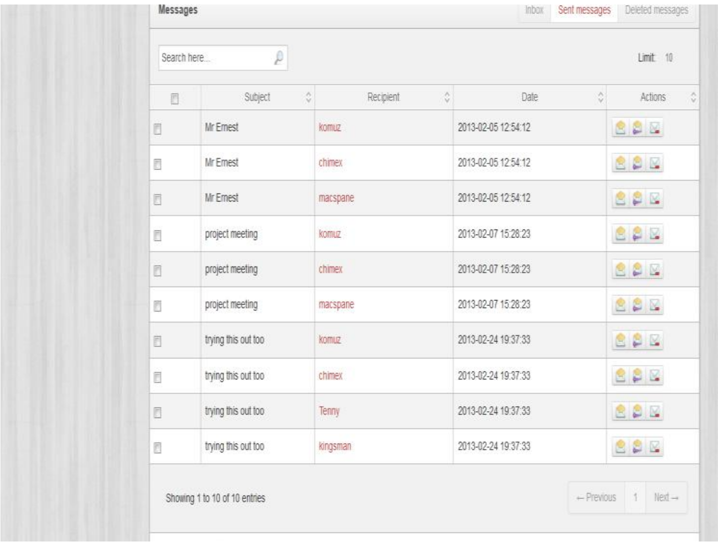
Limit: 10

□	Subject	Sender	Date	Actions
□	Browser issues	komuz	2013-02-07 15:48:18	
□	hey	macspane	2013-02-07 15:35:31	
□	Ecommece klass	macspane	2013-02-06 06:33:07	
□	darmie	macspane	2013-02-06 06:31:40	
□	seun	macspane	2013-02-05 20:41:29	
□	cool oooooooo	macspane	2013-02-03 22:02:18	

Showing 1 to 6 of 6 entries

← Previous 1 Next →

Figure 16: Reading Broadcast Messages



Messages

Inbox | Sent messages | Deleted messages

Search here...

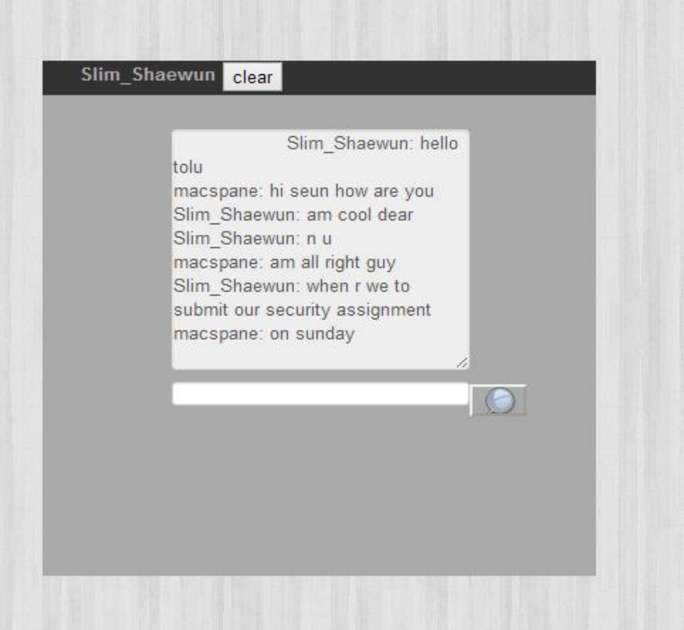
Limit: 10

□	Subject	Recipient	Date	Actions
□	Mr Ernest	komuz	2013-02-05 12:54:12	
□	Mr Ernest	chimex	2013-02-05 12:54:12	
□	Mr Ernest	macspane	2013-02-05 12:54:12	
□	project meeting	komuz	2013-02-07 15:28:23	
□	project meeting	chimex	2013-02-07 15:28:23	
□	project meeting	macspane	2013-02-07 15:28:23	
□	trying this out too	komuz	2013-02-24 19:37:33	
□	trying this out too	chimex	2013-02-24 19:37:33	
□	trying this out too	Tenny	2013-02-24 19:37:33	
□	trying this out too	kingsman	2013-02-24 19:37:33	

Showing 1 to 10 of 10 entries

← Previous 1 Next →

Figure 17: Viewing of sent messages



Slim_Shaewun: hello tolu
macspane: hi seun how are you
Slim_Shaewun: am cool dear
Slim_Shaewun: n u
macspane: am all right guy
Slim_Shaewun: when r we to submit our security assignment
macspane: on sunday

Figure 18: Chat Panel

2.4 Application Platform

The minimum hardware and the software requirements are: physical memory (RAM) of 512MB; Intel, Celeron or AMD Pentium 3 processor; hard disk capacity of 5 GB; Windows (98, 2000, ME, NT, XP, Vista, 7), Linux, or Mac OS; JavaScript-enabled web browsers (Mozilla Firefox (most suitable), Internet Explorer, Google Chrome, or Opera-mini), and Virtual server (Xampp version 1.7.1, Wamp server).

3.0 Related Works

One of the most important phases in the Internet chat history was the introduction of Internet Relay Chat (IRC) which was developed by Jarkko "WiZ" Oikarinen in 1988 at University of Oulu, Finland, where he was working at the Department of Information Processing Science. By the year 1989, there were already 40 servers that were provided with this service. Gradually, IRC started to gain popularity and was made available in almost sixty countries (Oikarinen & Reed, 1993). A number of other chatting programs were also developed in the attempt to replace IRC. IRC and Dalnet proved to be the most viable developments in the entire Internet chat history.

Groundbreaking innovations have further advanced the chatting methods and techniques. There are different types of Chat enabled today like Voice Chat and Video Chat. The chat systems voodoo, MushClient, GoogleTalk, Skype and facebook are discussed in succeeding paragraphs.

Voodoo Chat, opened in very late December 2001 is a text-based chat program with voice chat features created in the wake of Excite’s Virtual Places closure due to the company's bankruptcy following acquisition by the @Home Network (Parniak, 2009). Voodoo Chat uses two methods of security. The first is encryption. The second is the use of system hashes. It has the hub server which organizes other servers, login server which is a firewall and the chat server that controls the chat process. The limitations of Voodoo include that a user is restricted to a particular computer: once a user’s computer is registered, the user cannot use another computer to chat, and if the computer is stolen, there is an easy access to penetrate into the server.

Mushclient Chat System Characteristics include calling another player (having gotten the player’s TCP/IP address, and what port to use for chatting), receiving calls from one or more other players, and validating incoming calls and accepting or rejecting them. It enables splitting of users into groups, allows chats with the group members, and enables sending or receiving of files from other members. Its limitations include low security level: the chat system is prone to attack by penetration (Gammon, 2005). MUSHclient Chat System is a peer to peer chat system.

Google Talk is an instant messaging service that provides both text and voice communication. Google Talk applications are available for Microsoft Windows (XP, Server 2003, Vista, and Windows 7), Android, Blackberry, and Google Chrome OS operating systems. Because the Google Talk servers communicate with clients using an open protocol, XMPP, the service can also be accessed using any other client that supports XMPP. The connection between the Google Talk client and the Google Talk server is encrypted, except when using Gmail's chat over HTTP, a federated network that does not support encryption, or when using a proxy like IMLogic. End-to-end messages are unencrypted. Some XMPP clients natively support encryption with Google Talk's servers (Betabeat, 2012; Google Talk Beta, 2011; Google Talk Help, 2013). The technology used within the Google server network, however, is not publicly known. Its limitations include being prone to attack when chatting over HTTP or IMLogic.

Skype is a software program that uses the internet to make telephone calls. It also offers video, chat, SMS and presence awareness all in one application (Higginbotham, 2008). With the exception of logon servers, Skype has no central server to maintain the network. Instead, Skype uses peer to peer technology to decentralize the network and to help ensure a very high uptime percentage. Once you log in to Skype, your system becomes part of the network itself helping to decentralize the load of routing phone calls. Skype has an encryption system for security. This system cannot be turned on or off. Skype provides an uncontrolled registration system for users. People can use the system safely without revealing their real-life identity to other users of the system, but there is no way to know that the person they communicate with is the one they say they are. By using VoIP to make the calls, it is very cheap to call another phone. It does not cost money to call another Skype user. Every Skype user has a unique username which other users can use to talk with them (Wikipedia Skype, 2013). One of Skype limitations is lack of privacy (Skype has the keys to decrypt calls or sessions). Skype also makes it hard to enforce a (corporate) security policy. In addition, there are bugs and delays in the Linux version.

Facebook is the second largest social network on the web, behind only MySpace in terms of traffic (Yadav, 2006). Facebook has a tendency to share user information with third parties, including companies with which they have a relationship. Hence, privacy is not totally assured. Facebook also collects information about users from other sources, such as newspapers and instant messaging services. This information is gathered regardless of use of the website.

4.0 Conclusion

Faster communication through chatting with minimized risk of certain security breach and failure to receive the message because of no internet access could be implemented in any organisation. It was concluded, from the review of past works, that there existed a need to incorporate additional security measures into such a system besides the standard functionalities. This was implemented by the use of CAPTHA and encryption. Design tools that were appropriate were selected and utilized to bring about the functional Private Secure Chat System after thorough analysis of the objectives and requirements of the system. Different test were performed on the system to debug and fix flaws. The minimum system requirements were deduced from these tests.

The outcome provides secure private chat system that minimizes the breach risk of confidentiality through encryption, and grants user organisation possession of logged chat messages. It also provides basis for the design and implementation of similar systems. The created private chat system promises to contribute towards obtaining organisational competitive advantage. It is believed that the successful deployment of this system will aid faster and secure communication among users. Future work on the secure chat system could include features like voice chat, video chat, multicast chat and smiley. Further research in these aspects will improve the system.

References

Betabeat. (2012). Google Says GChat is not a word. Retrieved from: http://www.betabeat.com/2012/02/10/google-says-gchat-is-not-a-word/

Chamberglain, R., William, M. And Jarred, A. (2008). Practicum Chat Application: Requirements specification document. Retrieved from: http://simplechat.org/docs/project6_design_document.pdf

Date, C. J. (2003). *An Introduction to Database System.* Delhi: Pearson Education

Finkler, J. and Dao, D. (2001). Chat Application Requirements. Retrieved from: www.eveandersson.com/arsdigita/doc/requirements/chat_req_updated

Gammon, N. (2005). MUSHclient's Features. Retrieved from: www.gammon.com.au/mushclient/chat.htm

Higginbotham, B. (2008). Skype. Retrieved from: www.technologyevangelist.com/2008/05/a_bit_about how_skype.html

Google Talk Beta (2011). What's New on Google Talk? Retrieved from: http://www.google.com/talk/whatsnew_more.html

Google Talk Help (2013). Google Talk. Retrieved from: https://support.google.com/talk/?hl=en

Hughes, B. and Cotterell, M. (2009). *Software Project Management*. London: McGraw-Hill Education

Parniak, M. (2009). Voodoo Chat. Retrieved from: http://en.wikipedia.org/wiki/Voodoo_Chat

Oates, B. J. (2009). Researching Information Systems and Computing. London: SAGE

Oikarinen, J. And Reed, D. (1993). History of Chat System: retrieved from: http://daniel.haxx.se/irchistory.html

Whitten, J.L., Bently, L.D., and Dittman, K.D. (2001). *System Analysis and Design Methods.* New York: Irwin/McGraw-Hill. Pp 13, 479

Wikipedia Online Chat (2013). Online Chat. Retrieved from: http://en.wikipedia.org/wiki/Online_chat

Wikipedia Skype (2013). Skype. Retrieved from: https://simple.wikipedia.org/wiki/Skype

Yadav, S. (2006). Facebook Profile. Retrieved from: https://mashable.com/2006/08/25/facebook-profile/