A PROJECT REPORT

ON

End to End Encryption over chat

Submitted by

Vishal Gahlot (1/13/FET/BCS/2/080)
Bhaskar Mishra (1/13/FET/BCS/2/089)

Under the Guidance of

Dr. S.S Handa
(Professor)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Faculty of Engineering & Technology

Manav Rachna International University, Faridabad

November, 2016

APPENDIX 1

Declaration

We hereby declare that this project report entitled "**Text Translation End to End Encryption on chat**" *by* **Vishal Gahlot (1/13/FET/BCS/2/80, Bhaskar Mishra (1/13/FET/BCS/2/089),** being submitted in partial fulfillment of the requirements for the degree of Bachelor of Technology in **CSE** under Faculty of Engineering & Technology of Manav Rachna International University Faridabad, during the academic year **2013-2017**, is a bonafide record of our original work carried out under guidance and supervision of **Dr. S.S Handa, Professor , Computer science & Engineering** and has not been presented elsewhere.

1. Vishal Gahlot, (1/13/FET/BCS/2/080)
2. Rahul Dawra, (1/13/FET/BCS/2/082)
3. Bhaskar Mishra, (1/13/FET/BCS/2/089)

Manav Rachna International University, Faridabad

Faculty of Engineering & Technology

Department of Computer Science and Engineering

November, 2016

Certificate

This is to certify that this project report entitled "**Text Translation** E**nd to end chat encryption**" *by* **, Vishal Gahlot(1/13/FET/BCS/2/80), Rahul Dawra(1/13/FET/BCS/2/82) Bhaskar Mishra(1/13/FET/BCS/2/089),** submitted in partial fulfillment of the requirements for the degree of Bachelor of Technology in **Computer Science & Engineering** under Faculty of Engineering & Technology of Manav Rachna International University Faridabad, during the academic year 2013-2017, is a bonafide record of work carried out under my guidance and supervision. I hereby declare that the work has been carried out my supervision and has not been submitted elsewhere for any other purpose.

(Signature of Project Guide)
Dr S.S Handa
Professor
Department of Computer Science and Engineering
Faculty of Engineering & Technology
Manav Rachna International University, Faridabad

(Signature of HoD)
Dr.Suresh kumar
Head of Department
Department. of Computer Science and Engineering
Faculty of Engineering & Technology
Manav Rachna International University, Faridabad

Appendix 3

ACKNOWLEDGEMENT

The successful realization of project is an outgrowth of a consolidated effort of people from desperate fronts. We are thankful to **Dr. S.S Handa** (Professor) for their variable advice and Support extended to us without which we could not has been able to complete our project for a success it requires diligence, perseverance, and inspiration, motivation and innovation. Words cannot express our gratitude for all those people who helped us directly or indirectly in our Endeavour. I take this opportunity to express our sincere thanks to all staff members of CSE department for the valuable suggestion and also to our family and friends for their support.

We are thankful to **Dr. Krishan Kumar** (Project Coordinator), Associate Professor, CSE department for his guidance and support.

We express our deep thanks to **Dr. Suresh Kumar**, Head of Department (CSE) for warm hospitality and affection towards us. His constant encouragement has helped to widen the horizon of our knowledge and inculcate the spirit of dedication to the purpose.

We like to express sincere gratitude to our Executive Director, FET **Dr. M K Soni**, for their support and kind efforts.

Student name (Roll No.)

Vishal Gahlot (1/13/FET/BCS/2/080)

Rahul Dawra (1/13/FET/BCS/2/082)

Bhaskar Mishra (1/13/FET/BCS/2/089)

Appendix 4

List of Figure **Page No.**

Appendix 5

ABSTRACT

This project focused on creating a chatting application with communication environment. The objective of our project is to build a chatting system to facilitate the communication to obtain an effective channel among the Users themselves. The design of the system depends on socket concept where is a software endpoint that establishes bidirectional communication between a server program and one or more client programs while designing the database used for this system was made with Android Studio and normalization process through Enchat Algorithm. Languages that will be used for the development of this system: Java Development Kit (JDK): is a development environment for building applications and components using the Java programming language. Wamp Server: For creating a server database and MySQL: for searching various query through relational database and making it effective and with this a random library will be generated which will help clients avoiding the breach in security and avoid eavesdroppers from breaching in the security. It will also support the translate feature that helps in communicating through various languages with various clients across the globe.

Appendix 6

TABLE OF CONTENTS

4. Testing and Deployment

4.1 Testing
4.2 Verification
4.3 Validation: All Validation are done in php file
4.4 Evaluation

5. Conclusion and Future Enhancements

5.1 Summary of Work Done
5.1.1 Critical appraisal of work done
5.1.2 Proposal/scope of future enhancement

1. Introduction

1.1 Goals and Objectives: The Project is based on End to End Encryption on chat. End-to-end encryption (E2EE) is a system of communication where only the communicating users can read the messages. It will help in maintaining the privacy and security of the two client's interaction over the medium of communication. . Chat Server automates all the aspects stated above related to a communication in a highly secure environment. This project has been developed to receive instant and urgent messages and to provide total user satisfaction.

The entire process has been automated using JAVA technology and SQL SERVER to smoothen the flow of information in a highly secure environment across the network. The solution has been deployed, tested and validated thoroughly. While designing the system, care has been taken in efficiency, maintenance and reusability of the software for the present and future changes in the system.

1.2 Motivation: In principle, it prevents potential eavesdroppers – including telecom providers, Internet providers, and even the provider of the communication service – from being able to access the cryptographic keys needed to decrypt the conversation. It will also stop the interference of third party application for the use of encryption keys.

1.3 Method:

Enchat Algorithm: Developed by us with an add-on of RSA algorithm by integrating it with the android app & Wamp Servers

1. The user **Alice** wants to chat with **Bob**.
2. **Alice** generates a public/private key pair and send its public key to **Bob**.
3. Bob accepts the chat by generating a public/private key and sends the public part to **Alice**.
4. Now the server and the users know both public keys.
5. **Alice**, who is the user that initiated the chat, generate a symmetric key, encrypt it with **Bob** public key.
6. Since any other user could encrypt this key and send it with bob, **Alice** also send the encrypted message hash encrypted with its private key, so **Bob** can verify that it was **Alice** who generated the key.
7. **Alice** sends the message (symmetric key encrypted + digital signature) to **Bob**.
8. First, **Bob** decrypt the digital signature with **Alice** public key and compare to the encrypted message, if they don't check, the chat fails.
9. **Bob** decrypt the symmetric key with its private key.
10. Now **Bob** and **Alice** have the same symmetric key and **Bob** is sure that it was **Alice** who generated this key.

- **RSA Encryption is used.**

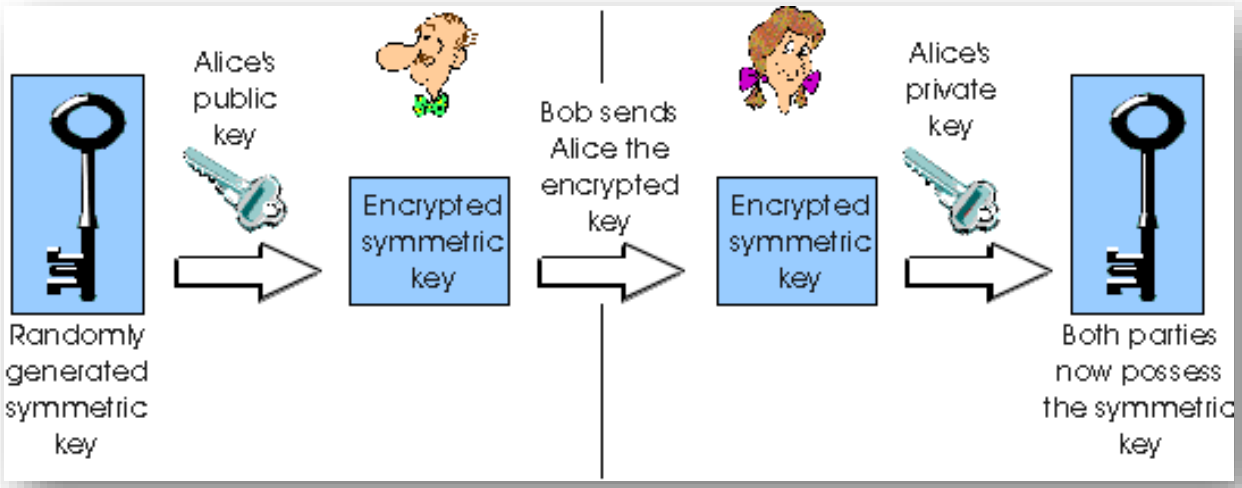


Fig 1.1

- **Added Text Translation to the system.**

```
import com.google.api.translate.Language;
import com.google.api.translate.Translate;
public class Main {
  public static void main(String[] args) throws Exception {

 String translatedText = Translate.DEFAULT.execute("Bonjour le monde",
Language.FRENCH, Language.ENGLISH);

  System.out.println(translatedText);
  }
}
```

Fig 1.2

1.4 Overview of the technical area –

- Server- Wamp Server
- FCM for Push Notification
- For Development- Windows
- For Testing- Android Device
- IDE used- Android Studio

1.5 Overview of the report- This App has been developed for avoiding cross scripting attack and D-Dos attack on various client chat system which works on the principle of random generation of Cipher text which is non-sharable by a third party applicants and is fully secured by various multiple attacks.

2. Literature Review

2.1 Introduction-The systems are designed to defeat any attempts at surveillance and/or Tampering because no third parties can decipher the data being communicated or stored. So, for avoiding such things we require a non-biased Encryption key which are randomly generated. It will also prevent the data from being attacked by cross multiple site attacked and D-DOS attack.

2.2 Survey- As through the survey we were able to conclude that end to end encryption used on multiple apps used nowadays is not that secure as there is use of third party application of data which compels to storing of crucial data which could be used at any later stages for monitoring of things when Required So to tackle such problem we have developed a new library encryption system which dissent require any use of a third party application Instead it draws data from a randomly generated library which is not known to any third party other than the two clients.

2.3 Conclusion-The apps which are used nowadays which are not as secured as they tell to be and contains a lot of loop holes in them so we have developed a new algorithm for securing the new loop hole.

Problem Definition and Requirement Analysis

2.4 Problem Definition-In this we had tried to focus on more securing the end to end communication of chat messaging As the biggest trending thing these days is instant messaging services and yet they are not secured as there is involvement of third party in such transaction. The problem today is that a lot of the Internet traffic is still unencrypted. This includes traffic such as email, chat, file transfers and video conversations. Gathering personal information from individuals around the world has become an industry in itself. This can easily be done by building a personal profile of users based on search history, email history etc. In this project I will not focus on all these problems, but rather focus on chat file transfers in a encrypted way, and look at how this can be made more secure.

2.5 Requirements- The system requirement can be classified into two categories:

Hardware Requirement:

- **a.** CPU 4 processor or higher.
- **b.** Memory 2 GB or higher.
- **c.** Hard disk 160 GB or higher.
- **d.** Network Access

Software Requirement:

- **a.** Window Operating System
- **b.** MySQL
- **c.** Android Studio

3. Design and Implementation

3.1 Introduction – In this app development we will use Enchat algorithm, FCM for chat security messaging, push notification for message transfer, Android Studio

Enchat Algorithm: The user **Alice** wants to chat with **Bob**

.

- **Alice** generates a public/private key pair and send its public key to **Bob**.
- Bob accepts the chat by generating a public/private key and sends the public part to **Alice**.
- Now the server and the users know both public keys.
- **Alice**, who is the user that initiated the chat, generate a symmetric key, encrypt it with **Bob** public key.
- Since any other user could encrypt this key and send it with bob, **Alice** also send the encrypted message hash encrypted with its private key, so **Bob** can verify that it was **Alice** who generated the key.
- **Alice** sends the message (symmetric key encrypted + digital signature) to **Bob**.
- First, **Bob** decrypt the digital signature with **Alice** public key and compare to the encrypted message, if they don't check, the chat fails.
- **Bob** decrypt the symmetric key with its private key.
- Now **Bob** and **Alice** have the same symmetric key and **Bob** is sure that it was **Alice** who generated this key.

3.2 Create Chat Application in Android using FCM Creating Android Studio Project

- Open android studio and create a new android project. Select empty activity and next->next->finish.
- Once your project is loaded, you need to add google-services.json in app directory, which you have generated while creating the app in google developer console. If you don't know what is it just go through the Android Push Notification Tutorial using FCM.

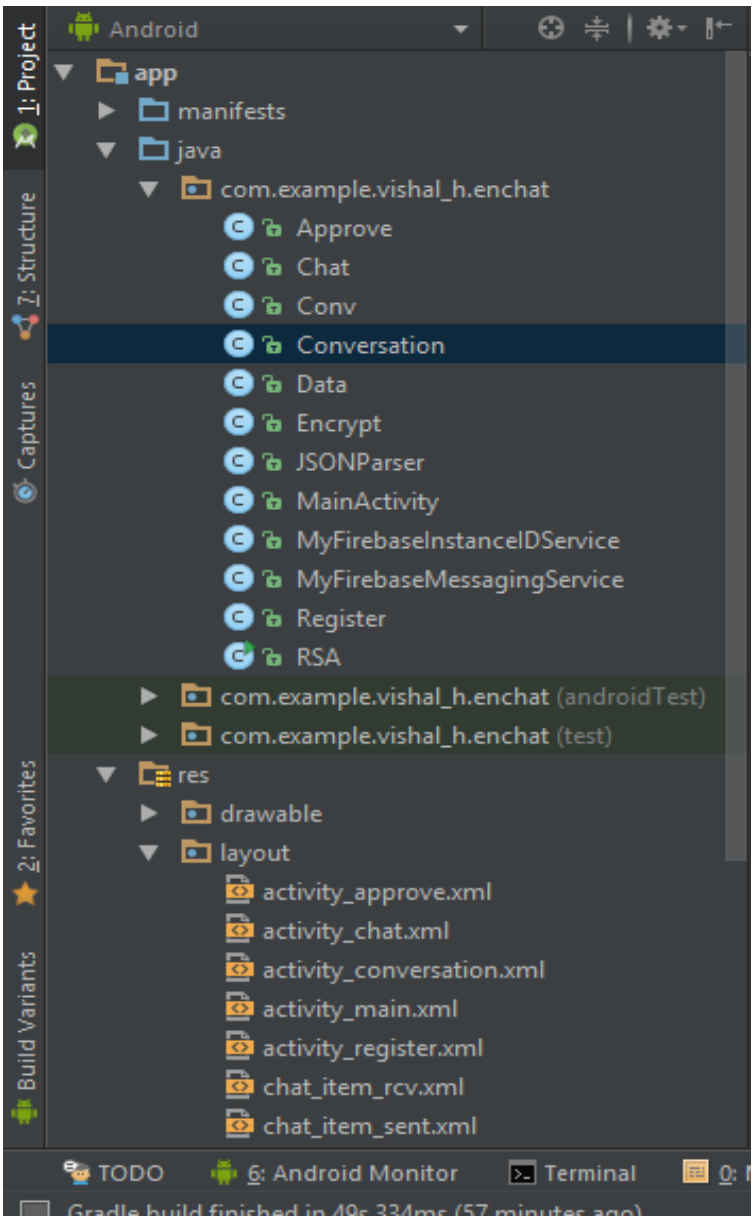


Fig 1.3

- We will be using many library dependencies for this project so this is my app level build.gradle file. You need to change your file according to this

```
compile 'com.android.support:design:23.4.0'
compile 'com.google.code.gson:gson:2.4'
compile 'com.android.Fig
support:support-v4:23.4.0'
compile 'com.google.android.gms:play-services-
appindexing:9.0.2'
compile 'com.google.firebase:firebase-messaging:9.0.0'
```

We need to add class path in the build gradle of project

classpath 'com.google.gms:google-services:3.0.0'

- **Creating Activities**

a. Register Activity: U ser can register itself. Individual need to enter his/her email and password.

Fig- 1.4

Code:-

```
package com.example.vishal_h.enchat;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.google.firebase.iid.FirebaseInstanceId;
import com.google.firebase.messaging.FirebaseMessaging;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.HashMap;

public class Register extends AppCompatActivity implements
View.OnClickListener {

    EditText ed3,ed4;
    Button bt4;
    String token;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);

        ed3=(EditText)findViewById(R.id.editText3);
        ed4=(EditText)findViewById(R.id.editText4);
        bt4= (Button)findViewById(R.id.button);

        assert bt4 != null;
        FirebaseMessaging.getInstance().subscribeToTopic("test");
        token= FirebaseInstanceId.getInstance().getToken();
        bt4.setOnClickListener(this);
    }


    @Override
    public void onClick(View v) {

        PostAsync b = new PostAsync();
        b.execute("register",ed4.getText().toString(),
ed3.getText().toString(),token);
    }
```

SQL QUERY:-

INSERT INTO user (email, password, token) VALUES ('$email', '$pass', '$token')

b. **Login Activity:** User can login to the application using their credentials.

Fig. 1.5

Code:-

```
@Override
public void onClick(View v) {
    PostAsync b = new PostAsync();
    b.execute("login", ed1.getText().toString(),
ed2.getText().toString(),token);
}
```

SQL Query:-

select * from user where email='$username' and password='$password'

c. **Send Invite Activity:** After login to the system, user need to send the invite first before starting the chat.

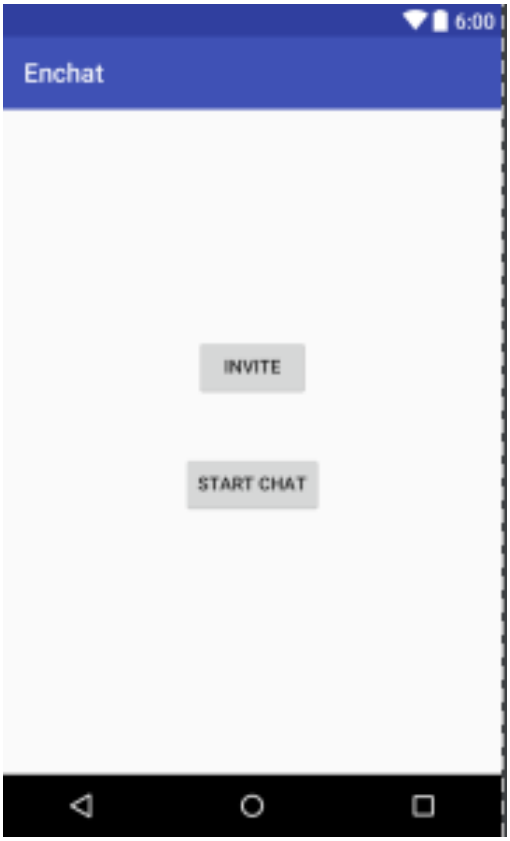


Fig. 1.6

Code:-

```
public void invite(View view) {

    SharedPreferences preferences =
PreferenceManager.getDefaultSharedPreferences(Chat.this);
    String a=preferences.getString("public", null);
    String n=preferences.getString("N", null);
    String token=preferences.getString("token", null);

    PostAsync b = new PostAsync();
    b.execute("sendkey",a,n,token);
}
```

Php Query:-

```
if(($_POST['tag'] === 'sendkey' ))
                {
```

```
                                        $public = $_POST['public'];
                                        $number = $_POST['number'];
                                        $mytok=$_POST['token'];

                                $arrr=array("key" =>
"success","public"=>$public,"number"
=>$number,"token"=>$mytok);
                                echo json_encode($arrr);
$sql = "Select token From user WHERE email='abc@gmail.com'";
        $result = mysqli_query($conn,$sql);
        $tokens = array();
        if(mysqli_num_rows($result) > 0 ){
                while ($row = mysqli_fetch_assoc($result)) {
                        $tokens[] = $row["token"];
                }
        }
echo json_encode($tokens);
        mysqli_close($conn);
        $message_status = send_notification($tokens, $arrr);
        echo $message_status;
```

d. Accept Request: Notification appear into the system to accept the request, user click on the notification and accept activity appear.

Fig. 1.7

e. **Chat Activity:** User can chat with each other.

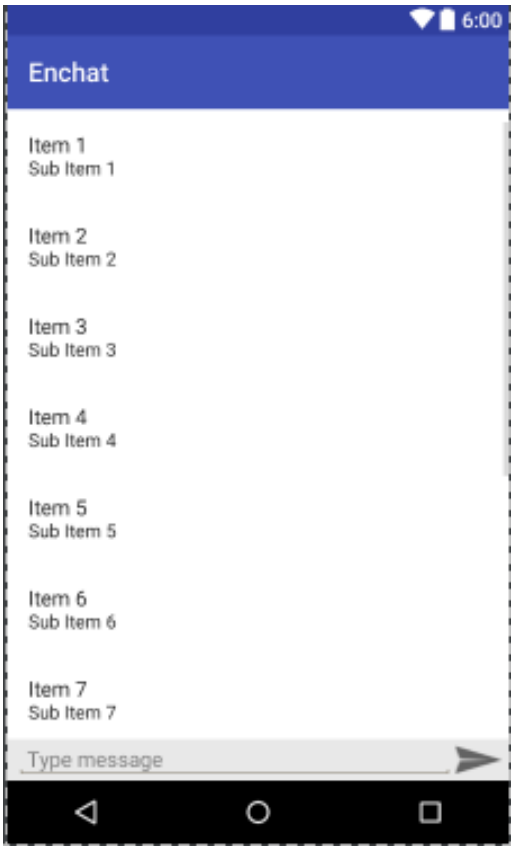


Fig. 1.8

Code:-

```
@Override
public View getView(int pos, View v, ViewGroup arg2)
{
    Conv c=getItem(pos);

    if (atoken.equals(atoken)) {

        v = getLayoutInflater().inflate(R.layout.chat_item_sent,
null);
    }
    else
        v = getLayoutInflater().inflate(R.layout.chat_item_rcv,
null);

    TextView lbl = (TextView) v.findViewById(R.id.lbl1);

lbl.setText(DateUtils.getRelativeDateTimeString(Conversation.this,
c
                    .getDate().getTime(),
DateUtils.SECOND_IN_MILLIS,
            DateUtils.DAY_IN_MILLIS, 0));

    lbl = (TextView) v.findViewById(R.id.lbl2);
```

```
    lbl.setText(c.getMsg());

    lbl = (TextView) v.findViewById(R.id.lbl3);
    if (atoken.equals(atoken))
    {
        if (c.getStatus() == Conv.STATUS_SENT)
            lbl.setText("Delivered");


        else if (c.getStatus() == Conv.STATUS_SENDING)
            lbl.setText("Sending...");
        else
            lbl.setText("Failed");

    }
    else
        lbl.setText("");

    return v;
}
```

f. Database: We have designed two tables: User and Message

User Table

	id	email	password	token
Edit Copy Delete	1	vishal@gmail.com	gffjkk	dhfhdfhfhjfhfjDDHGHjfjwefDDFH988746bgy2bhhGGYHj
Edit Copy Delete	2	bhaskar@gmail.com	gjhyadh	eTgC1oeKaBY:APA91bEHzkN9oMRYTyYWQbxj4MYxSJfP1IP3Xh...
Edit Copy Delete	3	123456	techbjunks@gmail.com	frN4Y6hd2ws:APA91bHMqwJBgAjsY_Ez20Bwt3urJsnkTuLMID...
Edit Copy Delete	4	techbjunks@gmail.com	abc	frN4Y6hd2ws:APA91bHMqwJBgAjsY_Ez20Bwt3urJsnkTuLMID...
Edit Copy Delete	5	abc@gmail.com	abc	cBNsbnu2OJM:APA91bEHI4sI4L4uJMbDtmaxyIbxRsSTbSIr0z...
Edit Copy Delete	6	abc1@gmail.com	abc	fDV3KJe28Hk:APA91bFkNPcS0avOdnj3WfmyrD0Y24ce30fS65...
Edit Copy Delete	7	vishal@g.c	hshdhdb	eBtFDYAktnU:APA91bFquKKsG-K4UGLlzzWSE8NEdQGEuCk2s-...
Edit Copy Delete	8	abc2@gmail.com	abc2	ejQ4Ux8Y-ug:APA91bEC9MdQwssh75QHN5PM1iNjubJOe5RmyH...

Fig. 1.9

Message table

	message_id	from	message	to	time
Edit Copy Delete	101	ctajeO1B1-k:APA91bElza250NW_VeGyU4sqo_\	15256692296380	cBNsbnu2OJM:APA91bEHI4sI4L4uJMI	19 Dec 2016 9:39:26 a.m.
Edit Copy Delete	102	cBNsbnu2OJM:APA91bEHI4sI4L4uJ	16166102270242	ctajeO1B1-k:APA91bElza250NW_VeGyU4sqo_W:	19 Dec 2016 9:49:00 am
Edit Copy Delete	103	ctajeO1B1-k:APA91bElza250NW_VeGyU4sqo_\	14403967629130	cBNsbnu2OJM:APA91bEHI4sI4L4uJMI	19 Dec 2016 9:49:14 a.m.
Edit Copy Delete	104	cBNsbnu2OJM:APA91bEHI4sI4L4uJ	17047791501623	ctajeO1B1-k:APA91bElza250NW_VeGyU4sqo_W:	19 Dec 2016 11:25:24 am
Edit Copy Delete	105	ctajeO1B1-k:APA91bElza250NW_VeGyU4sqo_\	14109725942390	cBNsbnu2OJM:APA91bEHI4sI4L4uJMI	19 Dec 2016 11:25:41 a.m.
Edit Copy Delete	106	ejQ4Ux8Y-ug:APA91bEC9MdQwssh75QHN5PM	17674282019160	ejQ4Ux8Y-ug:APA91bEC9MdQwssh75QHN5PM1i	Dec 20, 2016 14:05:54
Edit Copy Delete	107	ejQ4Ux8Y-ug:APA91bEC9MdQwssh75QHN5PM	15509243031205	ejQ4Ux8Y-ug:APA91bEC9MdQwssh75QHN5PM1i	Dec 20, 2016 14:06:25
Edit Copy Delete	108	ejQ4Ux8Y-ug:APA91bEC9MdQwssh75QHN5PM	14187439693111	ejQ4Ux8Y-ug:APA91bEC9MdQwssh75QHN5PM1i	Dec 20, 2016 14:07:54
Edit Copy Delete	109	ejQ4Ux8Y-ug:APA91bEC9MdQwssh75QHN5PM	17469053669590	ejQ4Ux8Y-ug:APA91bEC9MdQwssh75QHN5PM1i	Dec 20, 2016 14:09:05
Edit Copy Delete	110	ejQ4Ux8Y-ug:APA91bEC9MdQwssh75QHN5PM	15692952649266	ejQ4Ux8Y-ug:APA91bEC9MdQwssh75QHN5PM1i	Dec 20, 2016 14:10:54

Fig. 2

3.3 Functional Decompositions-

- **ER-Diagram**

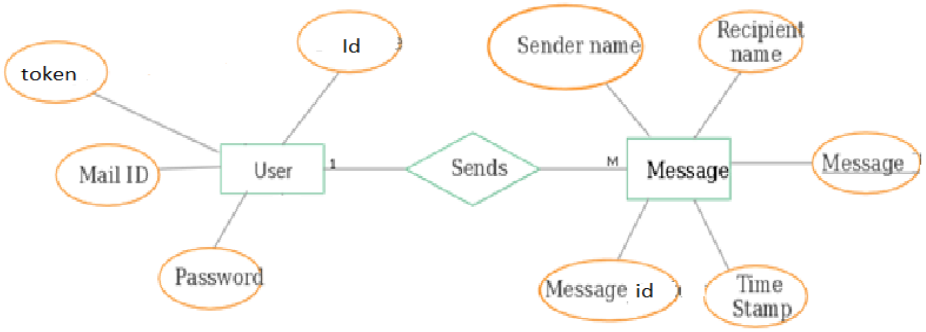


Fig. 2.1

- **Activity Diagram For Chatting:**

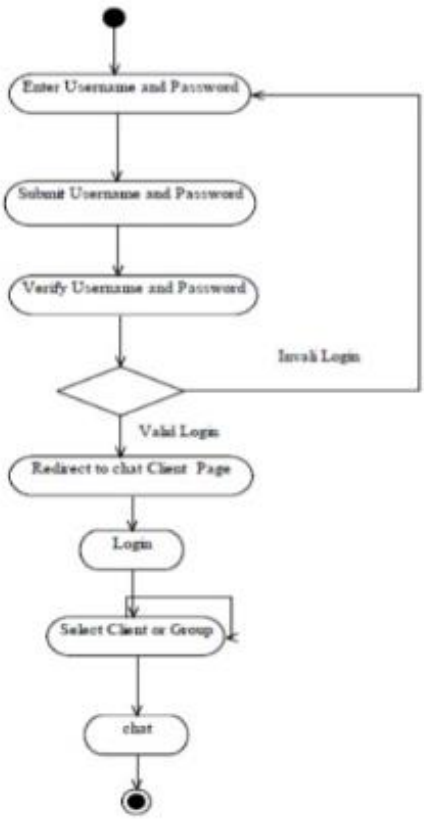


Fig. 2.2

4. Testing and Deployment

4.1 TESTING

System testing is the process of performing a variety of tests on a system to explore functionality or to identify problems. System testing is usually required before and after a system is put in place. A series of systematic procedures are referred to while testing is being performed. These procedures tell the tester how the system should perform and where common mistakes may be found. Testers usually try to "break the system" by entering data that may cause the system to malfunction or return incorrect information.

So, this could mean two things depending on an SDLC model. The first type of testing is the actual testing by users. This is usually done in models wherein implementation does not go with pre-testing with users. On the other hand, there are also testing that uses professionals in the field. This testing is aimed in cleaning the software of all the bugs altogether. For software that is set for public release, the software is first tested by other developers who were not in charge in creating the software. They will weed out the bugs and suggest fixes if every they find one. Once this stage is completed, it is time to test the software not just to the developers but to actual users.

The following steps are important to perform System Testing:

Step 1: Create a System Test Plan.

Step 2: Create Test Cases.

Step 3: Carefully Build Data used as Input for System Testing.

Step 4: If applicable create scripts to Build environment and To automate Execution of test cases.

Step 5: Execute the test cases.

Step 6: Fix the bugs if any and re test the code.

Step 7: Repeat the test cycle as necessary

4.2 Verification

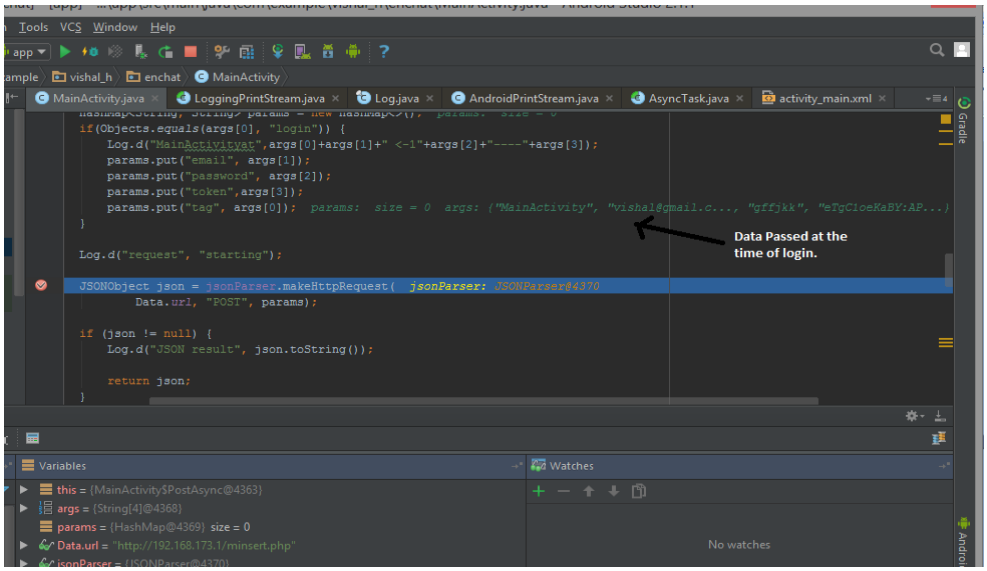
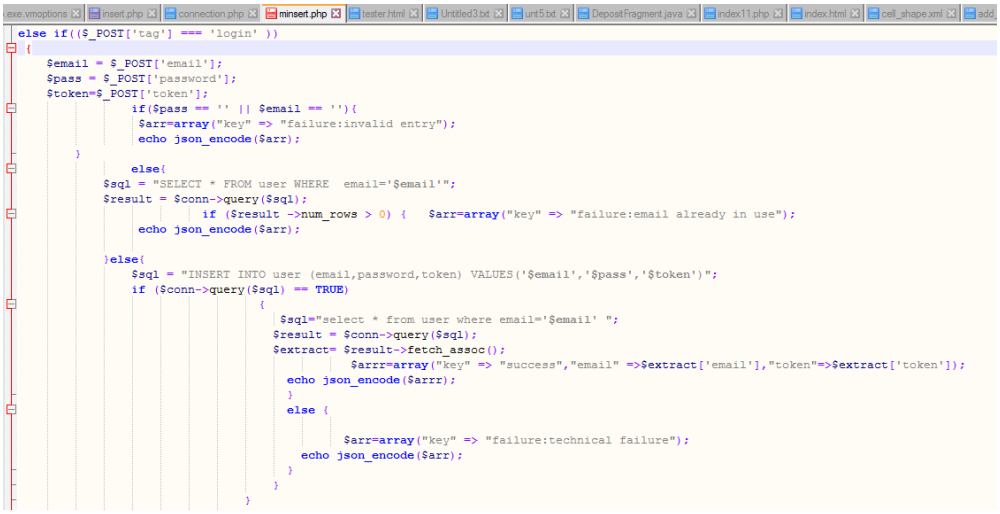


Fig 2.4

4.3 Validation: All Validation are done in php file.



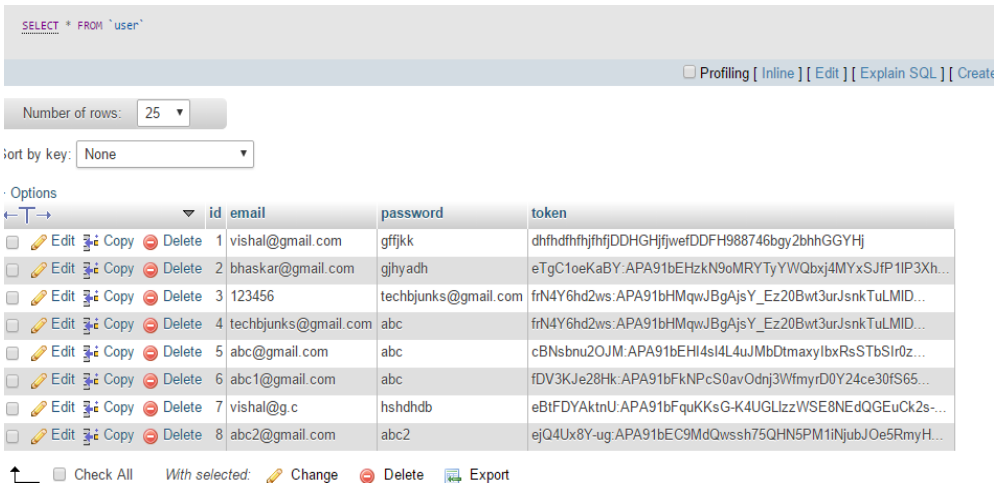
```
else if(($_POST['tag'] === 'login' ))
{
    $email = $_POST['email'];
    $pass = $_POST['password'];
    $token=$_POST['token'];
                if($pass == '' || $email == ''){
                 $arr=array("key" => "failure:invalid entry");
                 echo json_encode($arr);
        }
                else{
            $sql = "SELECT * FROM user WHERE  email='$email'";
            $result = $conn->query($sql);
                        if ($result ->num_rows > 0) {   $arr=array("key" => "failure:email already in use");
                 echo json_encode($arr);

            }else{
                $sql = "INSERT INTO user (email,password,token) VALUES('$email','$pass','$token')";
                if ($conn->query($sql) == TRUE)
                                {
                                  $sql="select * from user where email='$email' ";
                                $result = $conn->query($sql);
                                $extract= $result->fetch_assoc();
                                        $arrr=array("key" => "success","email" =>$extract['email'],"token"=>$extract['token']);
                                  echo json_encode($arrr);
                                  }
                                  else {

                                        $arr=array("key" => "failure:technical failure");
                                    echo json_encode($arr);
                                  }
                                }
                            }
```

Fig 2.5

4.4 Evaluation



```
SELECT * FROM `user`
```

Profiling [Inline] [Edit] [Explain SQL] [Create

Number of rows: 25

Sort by key: None

Options

	id	email	password	token
Edit Copy Delete	1	vishal@gmail.com	gffjkk	dhfhdfhfhjfhfjDDHGHjfjwefDDFH988746bgy2bhhGGYHj
Edit Copy Delete	2	bhaskar@gmail.com	gjhyadh	eTgC1oeKaBY:APA91bEHzkN9oMRYTyYWQbxj4MYxSJfP1IP3Xh...
Edit Copy Delete	3	123456	techbjunks@gmail.com	frN4Y6hd2ws:APA91bHMqwJBgAjsY_Ez20Bwt3urJsnkTuLMID...
Edit Copy Delete	4	techbjunks@gmail.com	abc	frN4Y6hd2ws:APA91bHMqwJBgAjsY_Ez20Bwt3urJsnkTuLMID...
Edit Copy Delete	5	abc@gmail.com	abc	cBNsbnu2OJM:APA91bEHI4sI4L4uJMbDtmaxyIbxRsSTbSIr0z...
Edit Copy Delete	6	abc1@gmail.com	abc	fDV3KJe28Hk:APA91bFkNPcS0avOdnj3WfmyrD0Y24ce30fS65...
Edit Copy Delete	7	vishal@g.c	hshdhdb	eBtFDYAktnU:APA91bFquKKsG-K4UGLlzzWSE8NEdQGEuCk2s-...
Edit Copy Delete	8	abc2@gmail.com	abc2	ejQ4Ux8Y-ug:APA91bEC9MdQwssh75QHN5PM1iNjubJOe5RmyH...

Check All With selected: Change Delete Export

Fig 2.6

5. Conclusion and Future Enhancements

Summary of work done : In this We created our own Enchat Encryption algorithm for not using the third party application for compromising with the security of the system and we use a random generated library so that there is no breaching in the system with the help of FCM push notification and wamp server. After compiling the message and creating a TCP between different ports the translate algorithm works for converting various languages.

Team Perspective: Instead of selecting a particular task done individually we divided each task in sub parts so that we all can work on the same phase such that no member has to wait to show his capabilities.

1.) Critical appraisal of work done-

- The system firewall has to be disabled for intra network communication.
- The system is dependent on single algorithm Enchat

2.) Proposal/scope of future enhancement

- File transfer: It will enable the users to sends the file.
- Video Conferencing : Attaching multiple clients on same chat room
- Integrating most languages : Will try to improve and integrate various languages

References/Bibliography

- Text translation - https://cloud.google.com/translate/docs/ http://itools.com/tool/google-translate-text-translator.
- End to end encryption - https://en.wikipedia.org/wiki/End-to-end_encryption , http://security.stackexchange.com/questions/56243/instant-messaging-end-to-end . https://www.comodo.com/resources/small-business/digital-certificates2.php.
- RSA Algo - https://www.youtube.com/watch?v=QjmHSVjmvGo
- Cryptography - Theory and Practice by Douglas Stinson
  CRC Press
  Boca Raton, 1995
- Applied Cryptography by Bruce Schneier
  Second Edition
  John Wiley & Sons, Inc.
  New York, c. 1996
- Handbook of Applied Cryptography by Alfred J. Menezes and others, Available freely on the web
- RSA Laboratories' Frequently Asked Questions About Today's Cryptography, Version 4.1
  RSA Laboratories, 2000
  RSA Security Inc.
  Available at http://www.rsadsi.com
- Internet Cryptography by Richard E. Smith
  Low Priced Edition, Pearson Education Asia
  Addison Wesley Longman 1997