



Green University of Bangladesh
Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Semester: (Spring, Year:2021), B.Sc. in CSE (Day/Eve)

Course Title: Algorithm Lab
Course Code: CSE 206 **Section: 201DK**

Lab Project Name: Secret-chat system by Huffman Algorithm

Student Details

Name		ID
1.	Prattay Dhar Logna	201002133

Submission Date : 02/01/22
Course Teacher's Name : Md. Gulzar Hussain

[For Teachers use only: **Don't Write Anything inside this box**]

Lab Project Status

Marks:

Signature:

Comments:

Date:

Table of Contents

Chapter 1 Introduction	3
1.1 Introduction.....	3
1.2 Objective.....	3
Chapter 2 Design/Development/Implementation of the Project.....	4
2.1 Design Secret-chat System.....	4
2.2 Implementation	5
Chapter 3 Performance Evaluation	8
3.1 Simulation Procedure	8
3.2 Results and Discussions	9
Chapter 4 Conclusion	10
4.1 Introduction.....	10
4.1 Practical Implications	10
4.2 Scope of Future Work.....	10
References	10

Chapter 1

Introduction

1.1 Introduction

Secret-chat system is a secret messaging concept. There are two parts in this messaging system. Which is sender part and receiver part. And sender and receiver part starting with the log in option where sender or receiver both need to log in with correct password. First of all sender log in and then sent any type of message. Then comes receiver option, In this part receiver need to log correct password. And receiver see message in decoding style its will be binary number. I make it with help java programming language and Huffman coding algorithm. Huffman coding is a lossless data compression algorithm. The idea is to assign variable-length codes to input characters, lengths of the assigned codes are based on the frequencies of corresponding characters. The most frequent character gets the smallest code and the least frequent character gets the largest code.

1.2 Objective

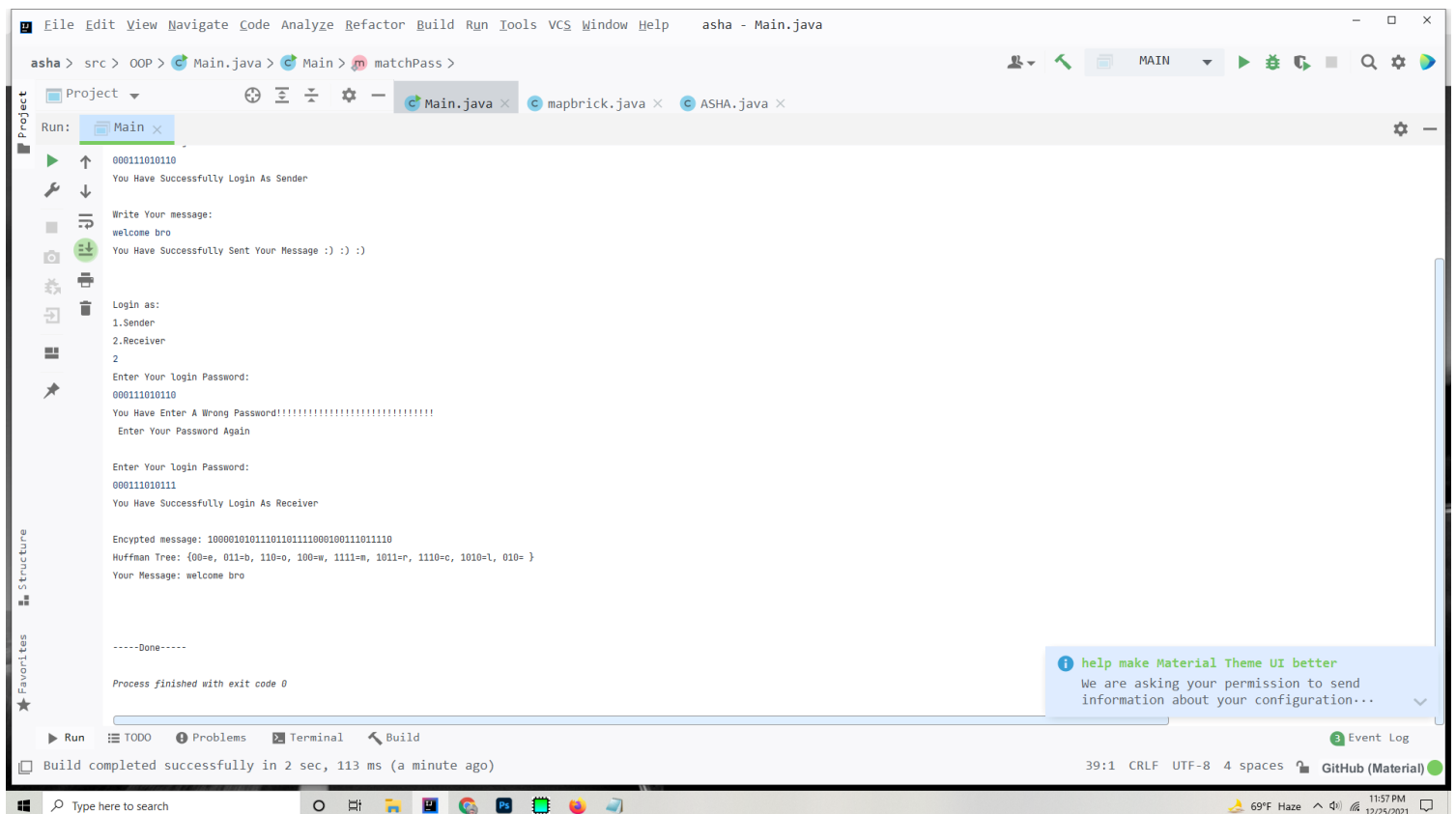
- To build a secret chatting system which is specially use for secret messaging.
- Sender and receiver chatting secretly.
- Learn more about Java programming language and Huffman coding

Chapter 2

Design/Development/Implementation of the Project

2.1 Design Secret-chat System

There are two parts in this messaging system. Which is sender part and receiver part. And sender and receiver part starting with the log in option where sender or receiver both need to log in with correct password. First of all sender log in and then sent any type of message. Then comes receiver option, in this part receiver need to log correct password. And receiver see message in decoding style its will be binary number.



```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help asha - Main.java
asha > src > OOP > Main.java > Main > matchPass >
Project Run: Main x
000111010110
You Have Successfully Login As Sender
Write Your message:
welcome bro
You Have Successfully Sent Your Message :) :) :)
Login as:
1.Sender
2.Receiver
2
Enter Your login Password:
000111010110
You Have Enter A Wrong Password!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Enter Your Password Again
Enter Your login Password:
000111010111
You Have Successfully Login As Receiver
Encrypted message: 1000010101110110111000100111011110
Huffman Tree: {00=e, 011=b, 110=o, 100=w, 1111=n, 1011=r, 1110=c, 1010=l, 010= }
Your Message: welcome bro
-----Done-----
Process finished with exit code 0
help make Material Theme UI better
We are asking your permission to send information about your configuration...
Run TODO Problems Terminal Build
Build completed successfully in 2 sec, 113 ms (a minute ago)
39:1 CRLF UTF-8 4 spaces GitHub (Material)
```

Figure 2.1: Output

2.2 Implementation

Now we will discuss about some important part of our project code

```
public static void main(String[] args) {
    System.out.println("*****Wellcome To Sectrect Chat*****\n");

    while(true)
    {
        Scanner input = new Scanner(System.in);

        System.out.println("Login as: \n1.Sender \n2.Receiver");
        int option = input.nextInt();
        Scanner input2 = new Scanner(System.in);
        while(true)
        {
            System.out.println("Enter Your login Password: ");
            String password = input2.nextLine();
            dicission = login(password, option);
            if(dicission == true && option == 1)
            {
                System.out.println("You Have Successfully Login As Sender\n");
                break;
            }
            else if(dicission == true && option == 2)
            {
                System.out.println("You Have Successfully Login As Receiver\n");
                break;
            }
            else
            {
                System.out.println("You Have Enter A Wrong Password!!!!!!!!!!!!!!!!!!!!!! \n Enter Your Password Again\n");
            }
        }

        switch (option)
        {
            case 1:
                if(dicission == true)
                {
                    System.out.println("Write Your message: ");
                    Scanner input3 = new Scanner(System.in);
                    String message = input3.nextLine();
                    String encrypted = encodder(message);
                    System.out.println("You Have Successfully Sent Your Message :) :) :) \n\n ");
                }
            }
        }
    }
}
```

Here set user option. Here show sender or receiver choosing option .If User want to login as a sender need to press 1 and if want to login as a receiver need to press 2.

```

public static boolean matchPass(String pass, int option) {
    if ("111011101000".equals(pass) && option == 1) {
        return true;
    } else return "000111010111".equals(pass) && option == 2;
}

public static boolean login(String password, int select_option) {
    while (true) {
        switch (select_option) {
            case 1: {
                Huffman huff = new Huffman(password);
                String pass = huff.encode();

                return matchPass(pass, select_option);
            }
            case 2: {
                Huffman huff = new Huffman(password);
                String pass = huff.encode();
                boolean decision = matchPass(pass, select_option);
                return decision;
            }
        }
    }
}

```

This is the Password section part. Here we set two binary type passwords for sender and receiver login option.

```

}

private void buildTree() {
    buildMinHeap();
    node left, right;
    while (!pq.isEmpty()) {
        left = pq.poll();
        treeSize++;
        if (pq.peek() != null) {
            right = pq.poll();
            treeSize++;
            root = new node('\0', left.weight + right.weight, left, right);
        }
        else { // only left child. right=null
            root = new node('\0', left.weight, left, null);
        }

        if (pq.peek() != null) {
            pq.offer(root);
        }
        else { // = Top root. Finished building the tree.
            treeSize++;
            break;
        }
    }
}

public void buildMinHeap() {
    for (Map.Entry<Character, Integer> entry: hmapWC.entrySet()) {
        Character ch = entry.getKey();
        Integer weight = entry.getValue();
        node n = new node(ch, weight, null, null);
        pq.offer(n);
    }
}

private void countWord() {
    Character ch;
    Integer weight;
    for (int i=0; i<orgStr.length(); i++) {
        ch = orgStr.charAt(i);
        if (hmapWC.containsKey(ch) == false)
            weight = 1;
        else
            weight = hmapWC.get(ch) + 1;
        hmapWC.put(ch, weight);
    }
}

```

Here we count the word and encoding using java HashMap and Huffman coding algorithm. Huffman coding is a lossless data compression algorithm. The idea is to assign variable-length codes to input characters, lengths of the assigned codes are based on the frequencies of corresponding characters. The most frequent character gets the smallest code and the least frequent character gets the largest code.

```
package algo.project;

import java.util.HashMap;

public class Decoder {

    public String decodedStr;
    public final HashMap<String, Character> hashMap;
    public HashMap<String, Character> hmapCodeR;

    public Decoder(String decodedStr, HashMap<String, Character> huffmanTree) {
        this.decodedStr = decodedStr;
        this.hashMap = huffmanTree;
    }

    public String decode() {
        StringBuilder sb = new StringBuilder();
        String t = "";

        for (int i = 0; i < decodedStr.length(); i++) {
            t += decodedStr.charAt(i);
            if (hashMap.containsKey(t)) {
                sb.append(hashMap.get(t));
                t = "";
            }
        }
        decodedStr = sb.toString();
        return decodedStr;
    }

}
```

Here Decoding the message. I use this for decoding my message. Here I use HashMap and decoding function.

Chapter 3

Performance Evaluation

3.1 Simulation Procedure

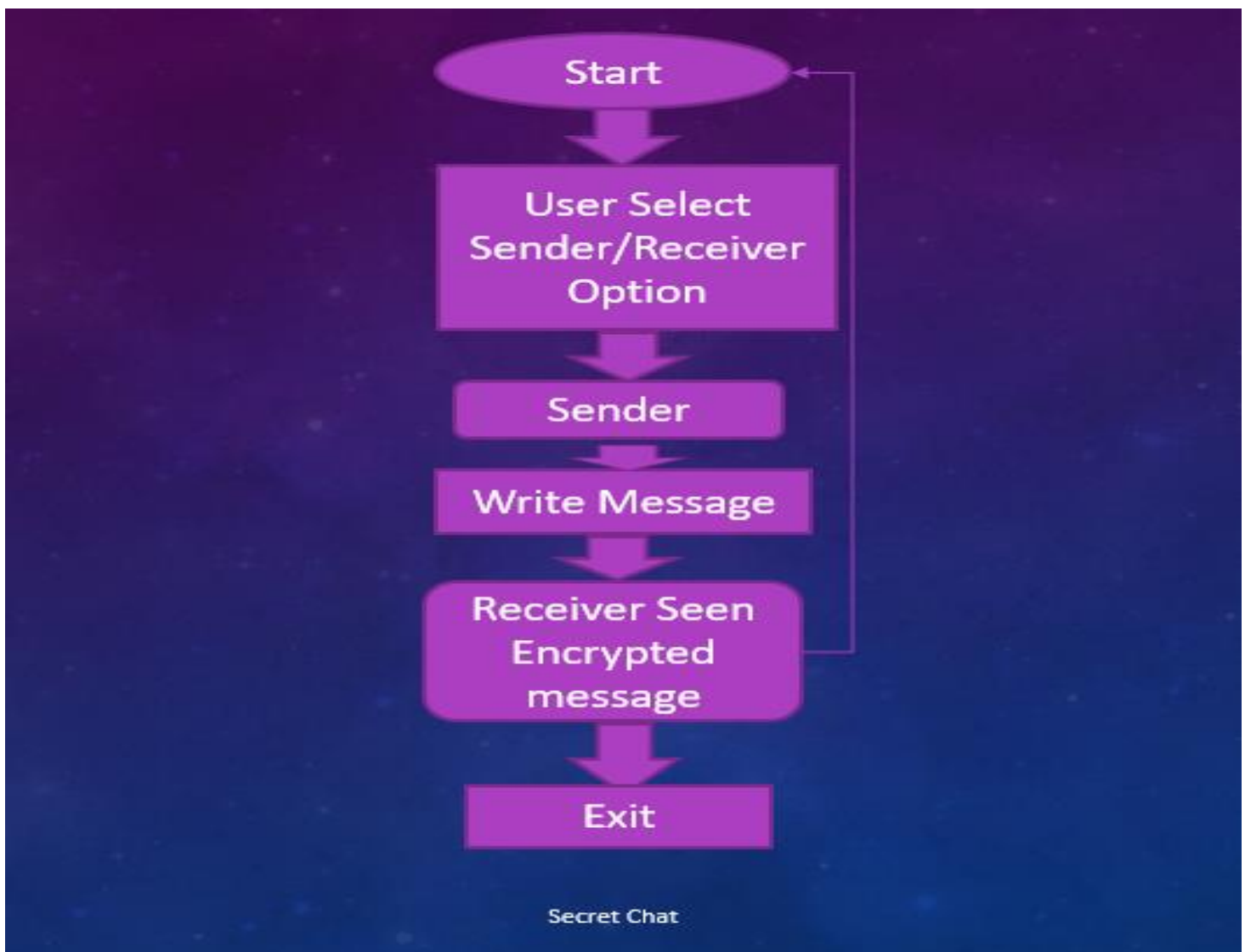
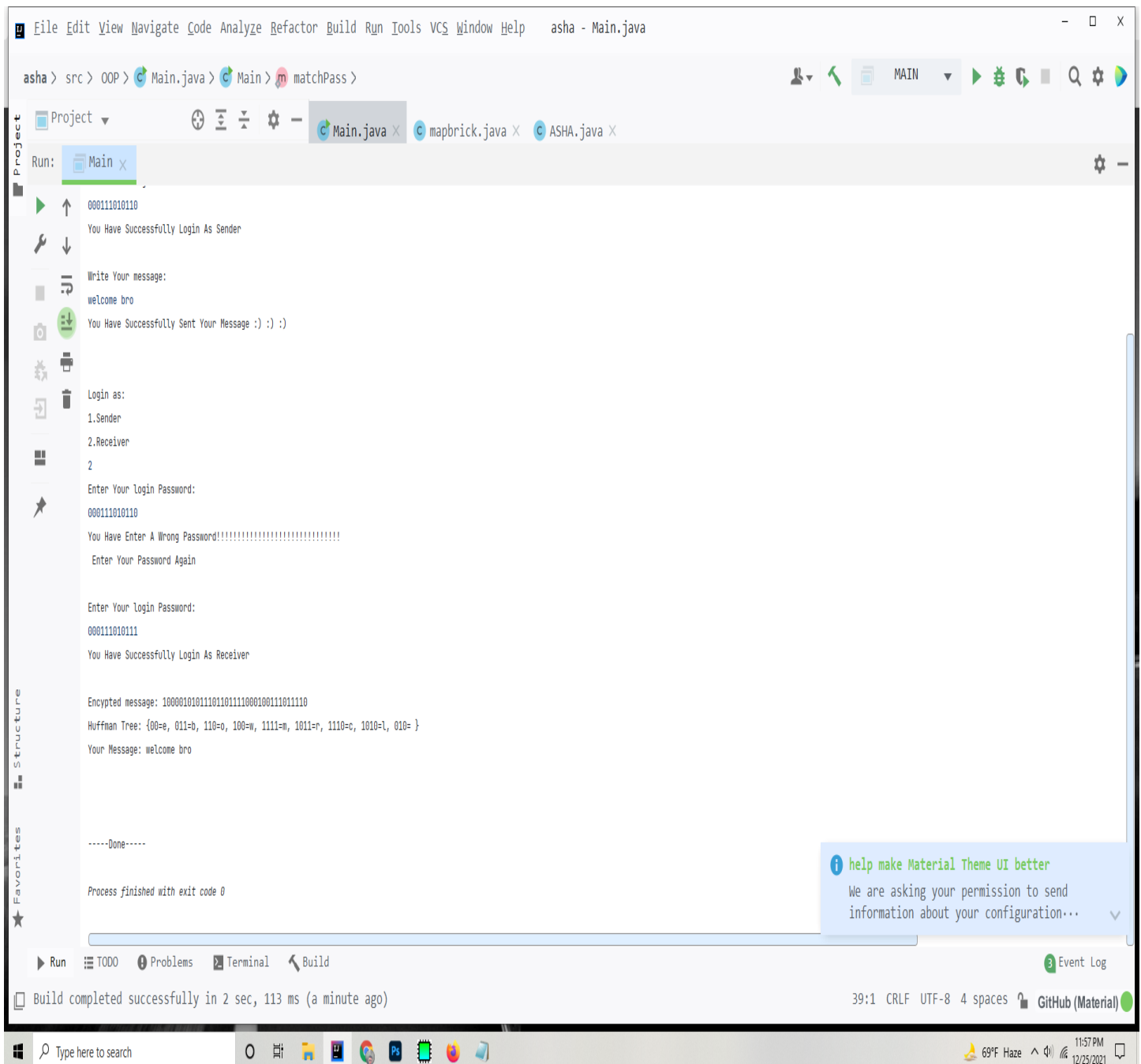


Figure 3.1: Simulation Procedure

3.2 Results and Discussions

Secret-chat system is a secret messaging concept. There are two parts in this messaging system. Which is sender part and receiver part. And sender and receiver part starting with the log in option where sender or receiver both need to log in with correct password. First of all sender log in and then send any type of message. Then comes receiver option, In this part receiver need to log correct password. And receiver see message in decoding style its will be binary number.

3.2.1 Results



```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help asha - Main.java
asha > src > OOP > Main.java > Main > matchPass >
Project Run: Main x
00011010110
You Have Successfully Login As Sender
Write Your message:
welcome bro
You Have Successfully Sent Your Message :) :)
Login as:
1.Sender
2.Receiver
2
Enter Your login Password:
00011010110
You Have Enter A Wrong Password!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Enter Your Password Again
Enter Your login Password:
00011010111
You Have Successfully Login As Receiver
Encrypted message: 1000010101101101110001001101110
Huffman Tree: {00=e, 011=b, 110=o, 100=w, 1111=n, 1011=r, 1110=c, 1010=l, 010= }
Your Message: welcome bro
-----Done-----
Process finished with exit code 0
Build completed successfully in 2 sec, 113 ms (a minute ago)
```

Chapter 4

Conclusion

4.1 Introduction

Secret-chat system is a secret messaging concept. Anyone can send message secretly. It's a very helpful to chatting secretly. People will be able to keep their personal message safe. This will protect messages and protect information from being leaked.

4.1 Practical Implications

We use it everywhere. Its a message system. 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.

4.2 Scope of Future Work

- Make more user-friendly
- Give good graphical interface
- More easy to use.

References

- [1] <https://www.youtube.com/watch?v=H62Jfv1DJIU>
- [2] <https://www.youtube.com/watch?v=8wIE6DgOWBs>
- [3] <https://www.youtube.com/watch?v=AUdLXnV0JHo>
- [4] <https://www.youtube.com/watch?v=HmBH30NrM7c>