KHULNA UNIVERSITY OF ENGINEERING & TECHNOLOGY

Building Real Time Chat Application and Implementing In Campus Area Network for Improving Transmission Rate, Balancing Load and Cost Reduction

Presented By:

Showrin Barua (1509048, Dept. of ECE, KUET)

Nafisa Tasneem (1509032, Dept. of ECE, KUET)

Fahmida Yeasmin Chowdhury (1509012, Dept. of ECE, KUET)

Contents

Objectives:

Objectives of this project are:-

1. To build a real time chat application which offers

   - Real time individual messaging feature

   - Real time group messaging feature

2. To implement real time chat application in Campus Area Network

3. To reduce Transmission Delay, Bandwidth consumption in instant messaging

Introduction:

Now a days we are very familiar to use different real time chat applications. *Real-time chat is virtually any online communication that provides a real-time or live transmission of text messages from sender to receiver*. We've become very much dependent on various social platforms like Facebook, WhatsApp, messenger, viber, imo etc. We used these technologies mainly for communicate for example: messaging, calling etc. Because they are cost efficient. Cost for making a video call using Facebook messenger, WhatsApp or others is very much less than the cost when the call is made using mobile SIM. Since internet is easily accessible in most places, maximum people make call with these platforms. All of these technologies have a common network structure. The structure is given below:

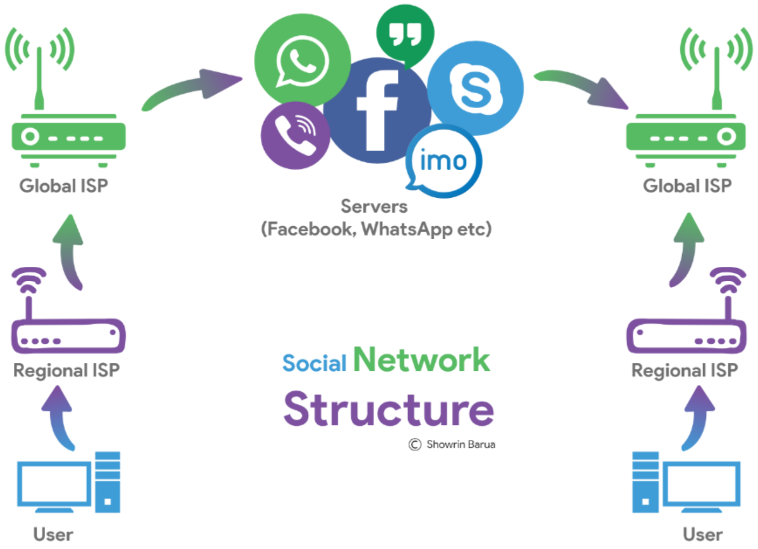


Fig 1: Diagram of content exchange

In that structure, a main server is setup in a suitable area of any suitable country. When a message or call request is made, the request will be directed to the main server if the source and destination devices are in the same region. The scheme is illustrated in Fig: 1.The request from source is sent to the server via regional ISP, Global ISP. And then the server makes a connection with the destination devices (If the device is in online).

In this project, we have created a real time chat application using Node.js, Express, Socket.io, and MongoDB etc. It has the features of real time private message (person to person individual chat) and group message.

Tools Used:

- Node.js (Server Side Programing)

- Socket.IO (For real time messaging)

- HTML, CSS (For client side interface)

- JavaScript (As core Programing Language)

- Express.js (Node.js web application framework)

- MongoDB (For data storing)

- PUG (An HTML template engine)

Proposed Network Structure:

Our proposed structure is given below:

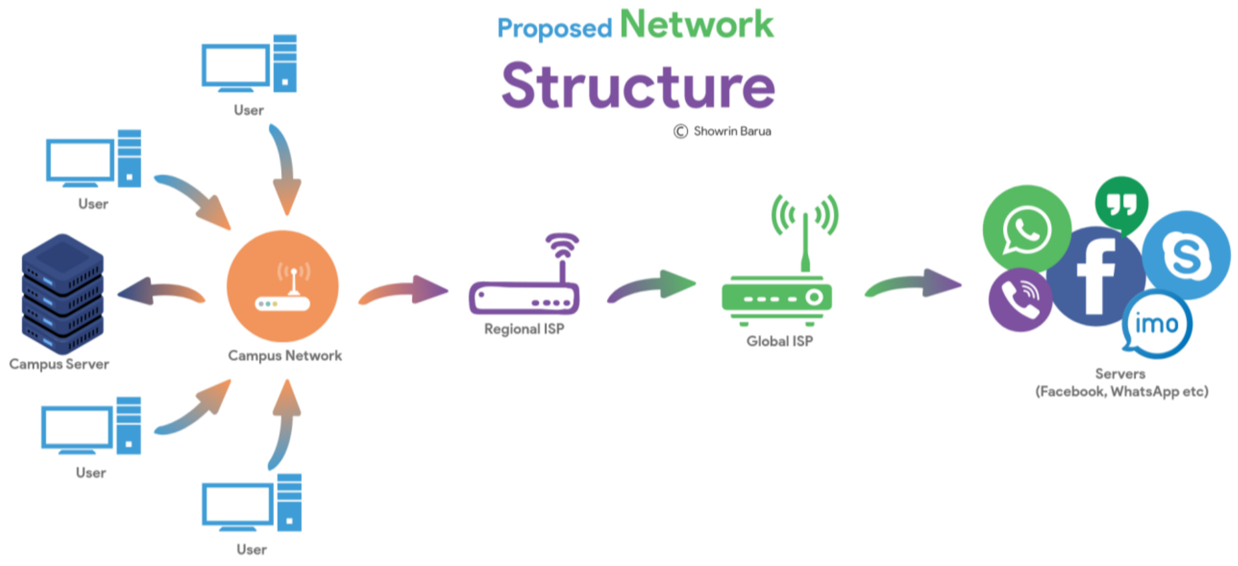


Fig 2: Proposed Structure

In the new structure, we have added a new portion, a local server which plays the main role of our proposed system.

System Model:

The operation of the proposed system is divided in four parts:

1. Local Server Setup (In-Campus)
2. Client-Server Socket Creation
3. Checking Destination IP (In Server)
4. Selecting The Communication Route

Operations:

1. **Local Server Setup (In-Campus):** At first a local server has to be set up. The server has to be HTTP Server and Web socket protocol enabled. And the local network should be made with one main router and a number of network extenders. If more router is used, there has to change a little in their programmed code.

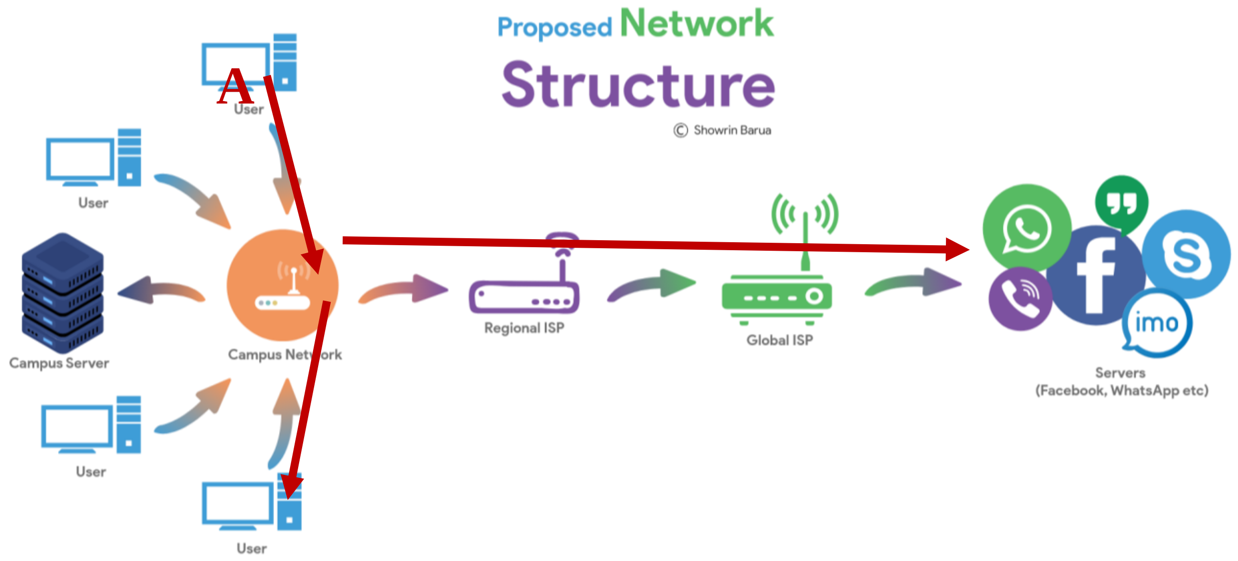


Fig 3: Operation flow

2. **Client-Server Socket Creation:** When a client makes a message request, a socket will be created between the server and client.

3. **Checking Destination IP (In Server):** Then the server checks the destination IP to ensure that whether it is inside the local network or not.

4. **Selecting the Communication Route:** If it is inside the network, the server bypasses the request towards the destination otherwise sends it to the main server.

Outcome:

After entering to the address, the signup and login form page arrives.

After signing up (creating new account) or logging to previously created account, the chat application interface is shown.

At the top, the user name is shown and there is Logout option.

At the left side, there is the available user names. For individual messaging, one user can choose any of the available users, and chat with them privately. The message is sent to the other user instantly.

At right side, there is the messaging option. The sender's message is sent in ash color, the receiver at the other side sends message in violate background.

After logout, the webpage again shows the signup and login form.

The user information is stored in the MongoDB database. The individual person to person message are also stored in the database.

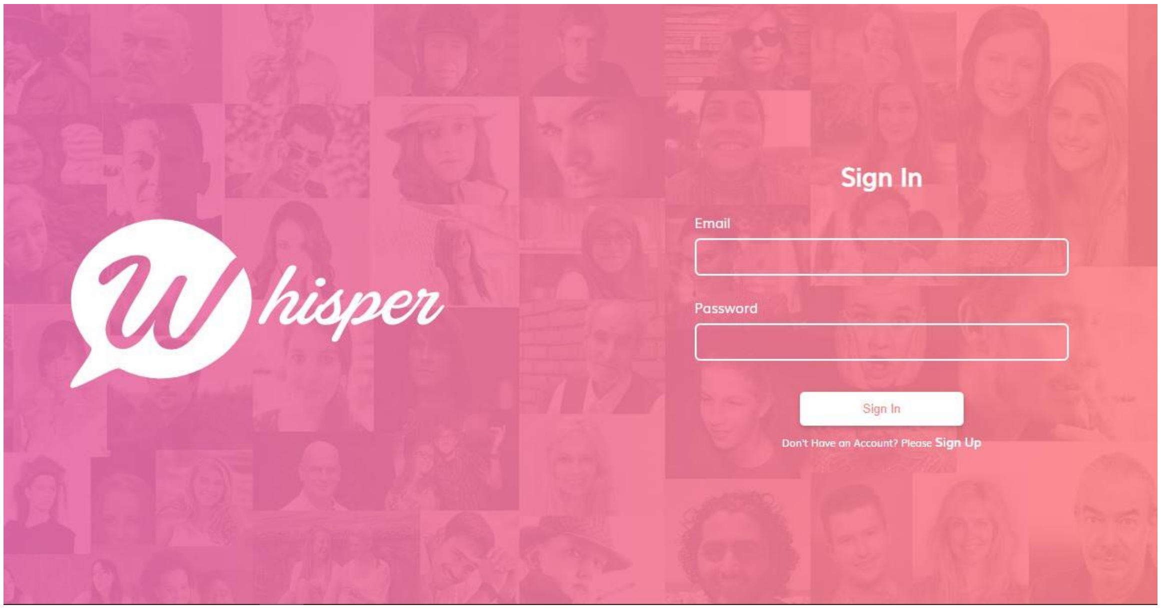


Figure 4: Basic Sign In form for initialization

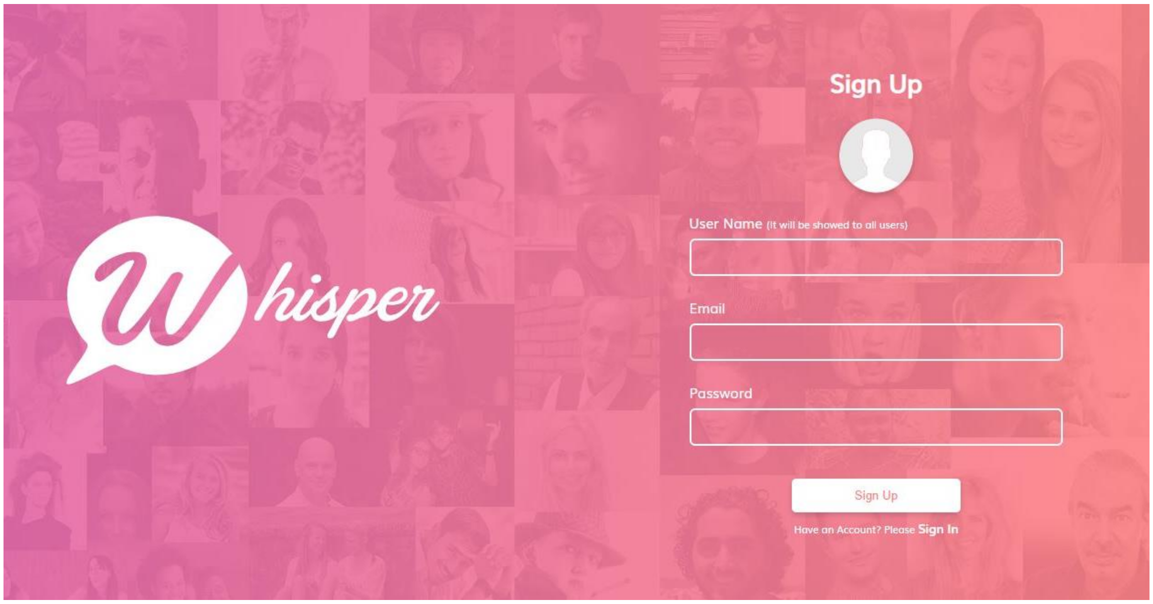
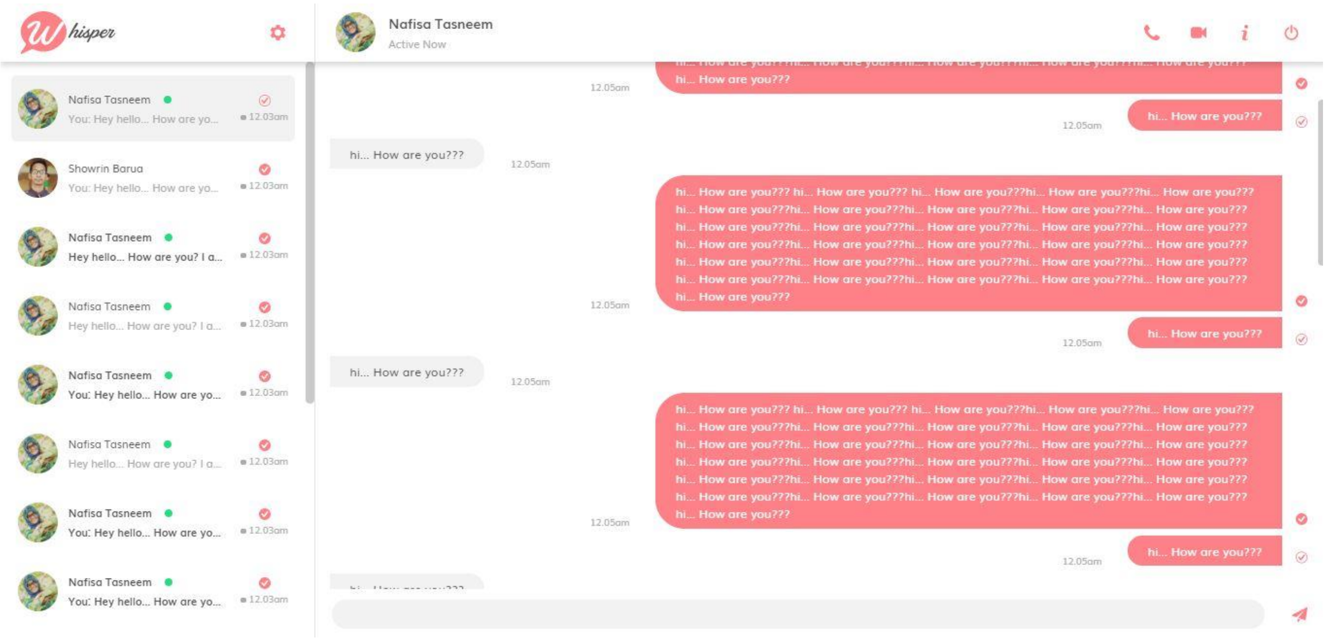

Whisper
Sign Up
User Name (It will be showed to all users)
Email
Password
Sign Up
Have an Account? Please Sign In

Figure 5: Basic Sign Up form for initialization



Nafisa Tasneem — You: Hey hello... How are yo... — 12.03am

Showrin Barua — You: Hey hello... How are yo... — 12.03am

Nafisa Tasneem — Hey hello... How are you? I a... — 12.03am

Nafisa Tasneem — Hey hello... How are you? I a... — 12.03am

Nafisa Tasneem — You: Hey hello... How are yo... — 12.03am

Nafisa Tasneem — Hey hello... How are you? I a... — 12.03am

Nafisa Tasneem — You: Hey hello... How are yo... — 12.03am

Nafisa Tasneem — You: Hey hello... How are yo... — 12.03am

hi... How are you??? — 12.05am

hi... How are you??? — 12.05am

hi... How are you??? — 12.05am

hi... How are you??? hi... How are you??? hi... How are you???hi... How are you???hi... How are you??? hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you??? hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you??? hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you??? hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you??? hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you??? hi... How are you??? — 12.05am

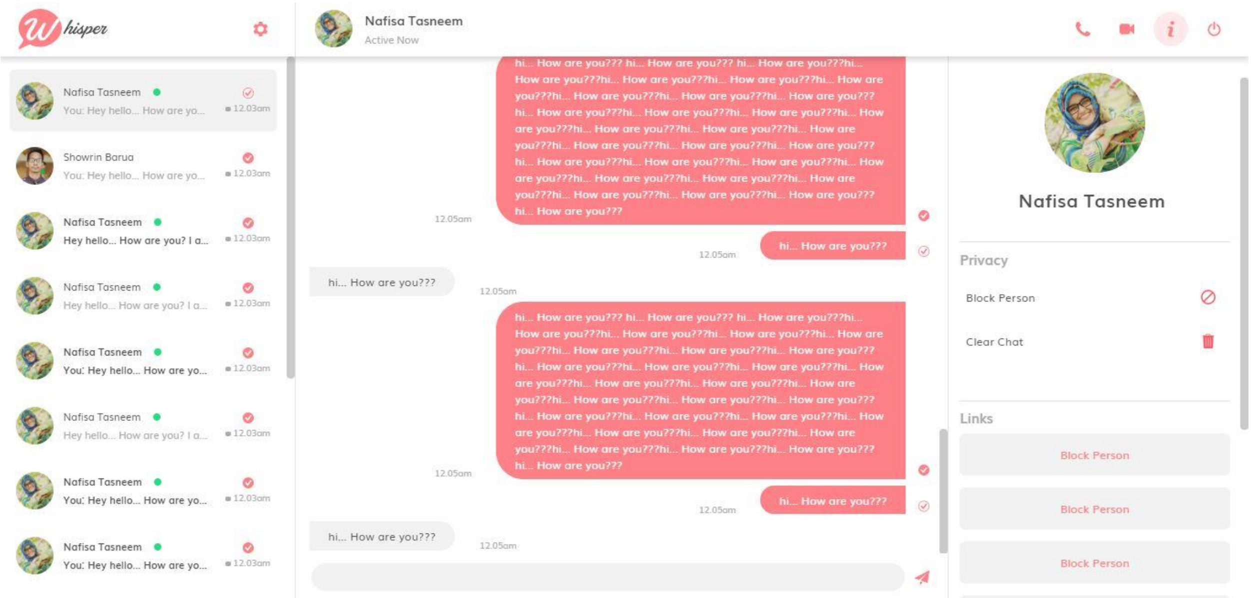
hi... How are you??? — 12.05am

hi... How are you??? — 12.05am

hi... How are you??? hi... How are you??? hi... How are you???hi... How are you???hi... How are you??? hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you??? hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you??? hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you??? hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you??? hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you??? hi... How are you??? — 12.05am

hi... How are you??? — 12.05am

Figure 6: Chat application interface for Private message



Nafisa Tasneem — You: Hey hello... How are yo... — 12.03am

Showrin Barua — You: Hey hello... How are yo... — 12.03am

Nafisa Tasneem — Hey hello... How are you? I a... — 12.03am

Nafisa Tasneem — Hey hello... How are you? I a... — 12.03am

Nafisa Tasneem — You: Hey hello... How are yo... — 12.03am

Nafisa Tasneem — Hey hello... How are you? I a... — 12.03am

Nafisa Tasneem — You: Hey hello... How are yo... — 12.03am

Nafisa Tasneem — You: Hey hello... How are yo... — 12.03am

hi... How are you??? hi... How are you??? hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you??? hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you??? hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you??? hi... How are you??? — 12.05am

hi... How are you??? — 12.05am

hi... How are you??? — 12.05am

hi... How are you??? hi... How are you??? hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you??? hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you??? hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you???hi... How are you??? hi... How are you??? — 12.05am

hi... How are you??? — 12.05am

hi... How are you??? — 12.05am

Figure 7: Chat application interface with other features

Analysis of Current Scenario:

According to cell phone usage data from Pew Research Center, 18-24 year old sent and received about 128 text messages per day. There is approximately 4500 current students and 500+ staff members in KUET. (128*5000) =640000 messages are sent every day. If we assume that, internal LAN is used in 60% of message transfer, then (0.6*640000) =384000 messages will be sent via internal LAN, per day. Which is (384000*30) =11520000 messages per month. Thus we will have to buy less amount of data from the ISPs, also there will be less bandwidth consumption in the public network, and thus overall system will be improved.

Impact:

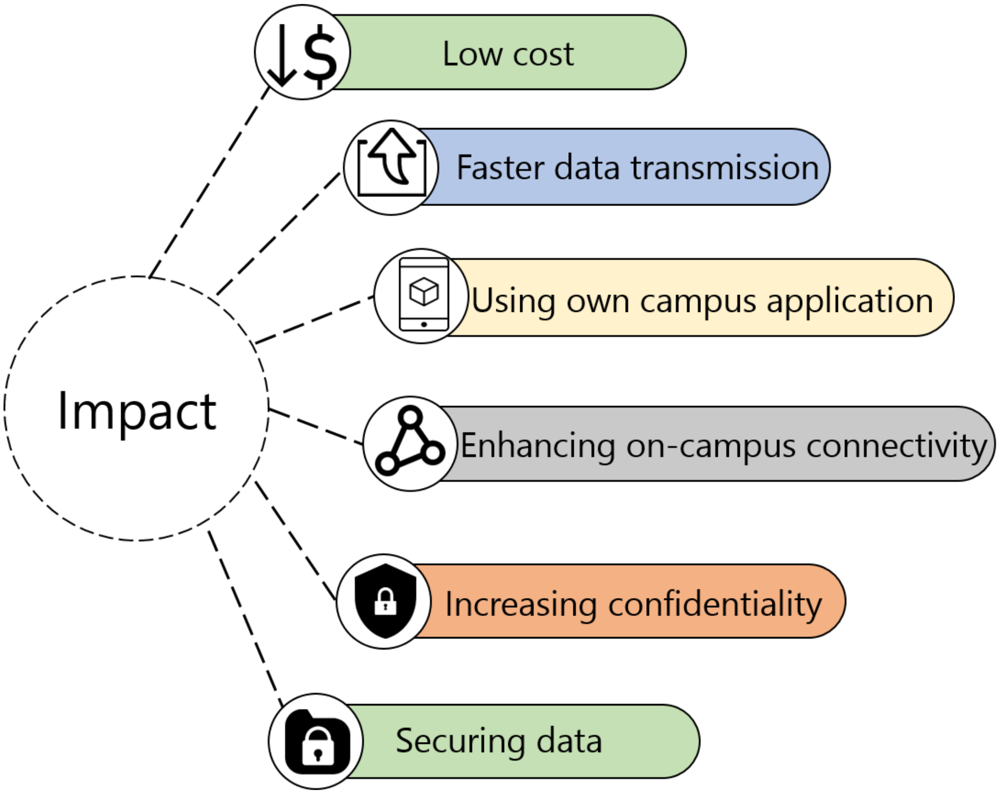

Impact
Low cost
Faster data transmission
Using own campus application
Enhancing on-campus connectivity
Increasing confidentiality
Securing data

Figure 8: Infographic showing the impact of using proposed Chat application

Conclusion:

In this project, we made real time chat application. Different user can access it if they are in the same network. The messages are stored in database.

However, we faced some problem while doing this project. There is problem in loading the JQuery file if the internet connection is slow. The server sometimes have problem in connecting, at first the browser has to made refresh, then the server gets connected.

We can use this app when we are in same network. So we can use it in KUET, in private friend community. In future, we are planning to make it more user friendly and more accessible. This project helped us to build our concept about socket programming and real time chat app design.