Initial Report

Written by : GoChat Team

February 7, 2017

1 Introduction

Nowadays, with the rapid development of the digital technologies, distributed chat systems play an increasingly indispensable role in people's daily life. They provide a way for people with geographical barriers to communicate, to study, to work together. Our project, go-chat, is an example of distributed chat systems, which allows users on two clients to enjoy essential features (add their friends, send messages, make group chats, etc.) and additional features (make notes, send emotion, download history, etc.) without compromising their privacy. It consists of two clients, Windows desktop application and web page, and one server. C# and PHP are chosen as the programming languages for the Windows desktop application and web page respectively. And WebSocket Protocol is used to ensure the communication between the clients and the server.

2 Project Description

2.1 Project Structure

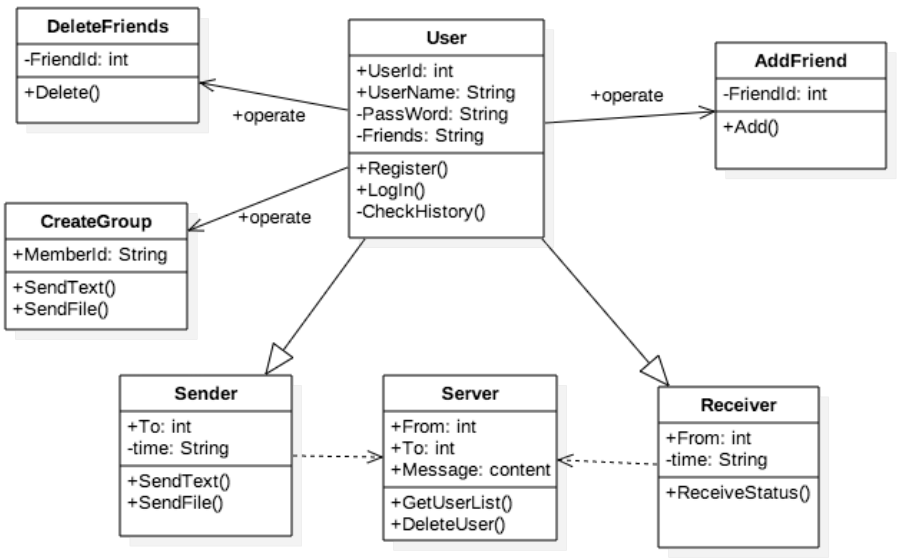

Figure 1: UML Class Diagram

2.2 Project's Aims and Strategy to Achieve Our Aims

Our aim for this project is to build a reliable, secure, and attractive chat system. Reliable in the term that every message sent by the client will receive in a relatively real-time. The client would not have to concern about lost message even in a slow network connection or

offline case. Meanwhile, secure in this case means that there is no unauthorized principal can read a message sent from one client to another client.

In order to achieve those goals, we divided our project into three main level, Basic Chatting Program, Security, and Other Additional Functions.

2.2.1 Level 1: Basic Functionalities

In this level, basic infrastructure will be built for the server and two clients, the web page and the Windows desktop application, to achieve basic functionalities for the system.

- Server
    1. The database is used to store users' names, passwords, personal information, friends' information and chatting information in the server.
    2. Apache server is used to make a connection between server and clients.
- Client
    1. Clients could set up the connection with server (Figure 2).
    2. Client 1 sends message to server
    3. Client 2 receive the message from the server which from the client 1.
    4. If the message is not a plain text, client 2 send a request to the server to receive data.
    5. Client 2 receive data which as picture and document.
- User
    1. Users on the clients could sign up creating their usernames and passwords and also filling in their personal details.
    2. Users on the clients could add their friends and send messages (text, files and pictures) to them.
    3. Users on the clients could check whether their friends are online or not.
    4. Users on the clients could create groups with adding their friends to the group and begin group chats.

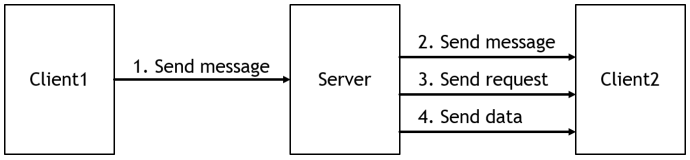

Figure 2: Server/Client Model

2.2.2 Level 2: Security

Chatting program is a private communication. Confidentiality, Privacy, and Integrity are common issues at the chatting program. It contains personal information, photos, business secret, and so on. This kind of information must be encrypted to avoid leakage. For encryption, many solutions are available such as trusted third party protocol or asymmetric keys encryption.

While level 1 provides basic functionalities without considering the security, the aim of the go-chat level 2 is to keep everything secure. In level 1, program store unencrypted history at the server. Meanwhile, in level 2, it will be encrypted by security protocols, so that only those who participate in the chatting can read their chatting history. In this level, the user also should be able to delete their chatting history to prevent data theft. In addition, user can customise message expire time. For example, if user set the expire time to one day, the message will be automatically deleted after 24 hours. The message will be deleted even if the receiver has not seen the message.

In term of security aspect, we will focus on two parts, communication security and security of data storage. For communication security, WebSocket protocol doesn't handle the authorization or authentication. However, just like the HTTPs which is HTTP over TLS, using the WSS (WebSocket over TLS/SSL) to encrypt our connection can make sure our information transfer security. In this case, we just need to configure TLS encryption for WebSocket and self-sign the certificate. For database security, we choose to use SALT and hash function to encrypt the username and password to avoid dictionary attacks. Using different encryption algorithm is for other data. In this way, we can prevent the SQL injection attack.

2.2.3 Level 3: Other Additional Functionalities

Chatting program has to be attractive. It is not only for sending plaintext messages to friends. It has to be possible to express emotion of people. That's why people use emojis and photos. In level 3 period, go-chat program contain attractive function including using emojis, gif, drawing picture, recommending friend, offline message, printing history, recalling the message, and image compose.

2.3 Time Management

Table 1: Task time table

No.	Task name	01	02	03	04	05	06	07	08	09	10
1	UML design										
2	Database design										
3	Intermediate report & presentation										
4	Developing environment setting										
5	Level 1: basic chatting program										
6	Level 2: security										
7	Level 3: Other additional function										
8	Debug										
9	Documentation										
10	Final report & presentation										

2.4 Initial Progress

We already design the project structure (Figure 1), the database (Figure 3) and make the project plan(timetable). The timetable can see in Table 1.

Since storing chat messages and retrieving those messages at a very fast rate is much needed for a chatting software, the MYSQL is used as our database management system, and four tables will be created to store different sorts of data.

- Table 'user' would contain the basic information of all users including userID, password, username, etc.

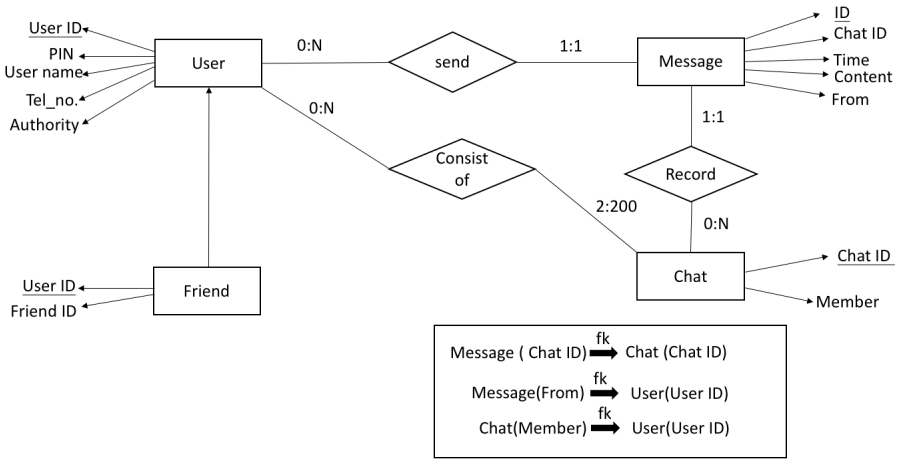

Figure 3: Entity Relation Diagram

- Table 'friend' is used to store the userID and the friendID of their friends'.
- Table 'message' will list the detail of every single message including chatID, sending time, a content of the message, and the sender's userID.
- Table 'chat' is used to record the member of chatting and their chatID.
- Key constraint:

  Primary key: user(userID), message(ID), friend(userID), chat(chatIÅD).

  Foreign key: message(chatID), message(from), chat(member).

3 Project Organization

3.1 Roles of Team Members

We divided our team member into two small teams, web application team and window application team. The first team will build the web client, while the other will build the window client. We also decided to build the server together.

3.2 How to Collaborate with Each Other (Tools)

We decided to hold a meeting every Thursday at 19.30 at Guy's Campus Library. In this meeting, we will discuss our progress for the last seven days and our target for the next week. We will also discuss the problems that might be arisen here. We will use WhatsApp application as our main way of communication.

3.3 Peer Assessment

We believe that we as the member of the group have the same responsibilities for the success of our project. We firmly believe that each of our members will work as hard as they can to ensure this project is a success. That is why we decided to give each of us a 15 mark for our effort. On the other hand, We will divide the rest of 10 marks based on our member's creativity, ability to handle stress and pressure, and capacity to keep our group together.

3.4 Conflict Handling

Conflict related to the development of the system will be discussed every week. We will discuss the problem and narrow down the possible solutions. We will also consider the advantages and disadvantages of every option before deciding the best choice.