

SYSC 4907

Come and Chat (C&C): Fourth Year Engineering Project Proposal

Carleton University

Group Members:

Yunzhou Liu 101027110
Shizhong Shang 101115304
Zirui Qiao 101100225

Supervisor:

Lynn Marshall

19 October 2022

Contents

1. Introduction.....	3
1.1 Purpose.....	3
1.2 Scope.....	3
2. Background.....	3
2.1 Discord.....	3
2.2 Database.....	3
2.3 Whatsapp.....	4
2.4 Facebook Messenger.....	5
3. Objectives.....	6
3.1 Functional Requirements.....	6
3.2 Page Requirements.....	7
3.3 Design Constraints.....	7
3.4 Software Quality Attributes.....	7
3.5 Other Requirements.....	7
3.6 Measurability.....	8
3.7 Sample Screens.....	8
4. Individual Mission Statements.....	9
4.1 Yunzhou Liu.....	9
4.2 Shizhong Shang.....	9
4.3 Zirui Qiao.....	10
5. Relation to Degree Program.....	10
5.1 Yunzhou Liu.....	10
5.2 Shizhong Shang.....	10
5.3 Zirui Qiao.....	11
6. Individual Contributions.....	11
7. Plan and Timetable.....	12
8. Risks and Mitigation Strategies.....	12
9. Conclusion.....	12
10. References.....	14
Appendix 1. Definition, Acronyms, and Abbreviations.....	16

1. Introduction

1.1 Purpose

Instant messaging is an efficient way to deliver information, and today, instant messaging software has become an integral part of our lives. For some of the most famous instant messaging software, such as Discord, WhatsApp and Facebook Messenger, we enjoy the convenience they offer while occasionally complaining about their shortcomings. This has led us to take a keen interest in such software.

Based on the instant messaging software that we have studied, in this project, we will build an imitated, web-based instant messaging software called Come and Chat (C&C). This proposal will outline the proposed plan and objectives to complete a working prototype of the project in early January 2023.

1.2 Scope

The C&C will provide mechanisms for signing up/in users, searching users, adding/deleting friends, sending/receiving messages to/from friends, building group chats, recording chat history, and allowing users to post messages as their status, allowing users' friend to see users' status.

More functions could be added after the functions above are implemented. Customized parts are always considered after the procedures above.

2. Background

2.1 Discord

To study the functions and implementation methods that we will use in our project, we first looked at Discord [1].

Discord is a chat software. Discord started from game voice and IM tool services, then turned to live broadcast platforms, and then opened a community platform for game stores, becoming the preferred tool for players to communicate and collaborate in games.

According to the Discord development log, Discord's internal resources include Application, Audit Logs, Auto Modification, Channels, Emoji, Guild, Guild Scheduled, Guild Template, Invite, Stage Instance, Sticker, Users, Voice, and Webhook. Each instance corresponds to the user's operation. For example, a channel. You can create a new channel. Channel settings, such as name and type, are determined by channel attributes. The user enters the information on the front-end page. After the information is transferred to the back-end, the back-end creates a channel instance to contain the input information; and then operates the database through the instance.

Our project also uses chat technology. Some internal resources of Discord are helpful to our project. The user object in Discord contains a variety of attributes, including id, username,

avatar, locale, email and other attributes that we can apply to our projects. The services for corresponding users include acquiring users, acquiring current users, modifying user information, and acquiring current users' Guilds. Guild object has id, name, owner_ID, roles, etc. The corresponding service methods include creating Guilds, obtaining Guilds, modifying Guild's information, deleting Guilds, etc. Object is associated with a database structure [1].

This information describes the functions and specific implementation methods of Discord, and helps us define the basic function we need to have when designing projects.

2.2 Database

A database is a warehouse that organizes, stores and manages data according to the data structure. Each database has different APIs for creating, accessing, managing, searching, and copying saved data.

The database is implemented by MySQL technology. MySQL is developed by MySQL AB of Sweden and belongs to Oracle. MySQL is a relational database management system. Relational databases store data in different tables instead of putting all data in a large warehouse, which increases speed and flexibility [2].

2.3 WhatsApp

We have investigated WhatsApp in detail.

WhatsApp is a famous IM software focusing on simple, secure, reliable messaging. Its features include text messaging, voice and video calls, photo and video sharing, document sharing, voice sharing, group chat, end-to-end encryption, feasible on the web, desktop and mobile [3] and many more. These features of WhatsApp have inspired our team, and we will use some of them as a C&C requirement.

Another document about WhatsApp has also been investigated: WhatsApp Business Platform. From WhatsApp Business Platform's official website: "*The WhatsApp Business Platform gives medium to large businesses the ability to connect with customers at scale. You can start conversations with customers in minutes, send customer care notifications or purchase updates, offer your customers a level of personalized service and provide support in the channel that your customers prefer to be reached on.*" [4]. It is certainly possible to implement C&C with the help of WhatsApp Business API, but that would be too boring, and we would learn very little from it.

Instead of using the API directly, our team decided to learn from the APIs and design our APIs. One exciting thing found during the investigation is called Authentication; the WhatsApp service will authenticate the sender's identity each time a request is received. The request is responded to only when the identity has the power. Only a few operations need no authentication, such as login and register. The authentication operation is done by verifying the token in the HTTP request header, and the token should follow a specific format. The token will be generated after login and expire in 7 days, or after logout [5].

WhatsApp sends all requests in RESTful API form and only in 5 common methods: post, get, delete, patch and put. "A REST API (also known as RESTful API) is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows

for interaction with RESTful web services. REST stands for representational state transfer and was created by computer scientist Roy Fielding.” [6]. This technology is very helpful for the C&C project in HTTP request implementation.

WhatsApp Business document published their API response format which is made up of 4 components:

1. HTTP Status Code: Status codes are issued by the server in response to a request made to the server. Represent the status of the response (e.g., error, success ...)
2. Payload: Actual data, only returns when the request is successful.
3. Meta: Some version data (e.g., API client version)
4. Errors: Errors' detail, only returns when errors occurred.

This is very helpful for our team to design the response data form. Except for Meta, C&C will need all the other three components in responses [7].

Capacity Rate Limits are also provided for each request, the sender cannot send requests higher than a specific rate. Our team believes this is a means of combating DDoS attacks and will not be used in this project as C&C will not be very large.

In general, WhatsApp is excellent as an IM software. Not only is it efficient and secure, but it is also prepared to deal with many unpredictable events. Our team's investigation also closely examined its client architecture [8], security [9], API reference [10] and current research [11]. Still, they are all less relevant to this project for various reasons, so no more details will be provided here.

2.4 Facebook Messenger

Messenger is an instant messaging service owned by Facebook. But at the same time Messenger can be used without a Facebook account and it can be used on Android and iOS devices (mobile application), works as a Web-app, and it also has a desktop version for Windows and iOS. So Messenger is a successful app that not only keeps users from Facebook, but also gathers more users whose not using Facebook and makes them know the Facebook app from using Messenger, it is a perfect example of expanding the user amount of a single app [12].

As a messaging app, it can do the basic function which allows users to send messages, short audio, short video, and pictures and it also can be used to make phone calls and video calls. After those basic functions, this app can also connect with other apps, such as sending locations from Map, sending pictures or videos from the local album, texting friends from both Facebook and Instagram, and adding bank cards and PayPal to transfer money between friends [13].

Messenger also has more functions, making this app more usable and fancier. Users can create chat groups and send messages, make group phone calls and video calls as well, then the group members can play games and do the poll in the group chat, when the user thinks the chat or the group is too annoying, they can mute them for the time length they chose. On the chat page, users can change the theme of the app, to make the app unique and more personal,

when the user gets a message, they can simply replay it with emojis. Messages also support the users to share their life and make them closer to their friends, there is a function called watch together, it allows chat friends to watch shows, games, films and TVs together over video chat, so people can talk about the same video anywhere anytime. To make this app more secure, whenever the user gets back to Messenger for other apps on their phone, their device will ask for the Face ID or the fingerprint to let the user get into the app [14]. These features increase the playability of Messenger.

Messenger is one of the most typical instant message-sending apps for the public, almost all features in the app are free. This app makes life became more convenient. It allows users to connect anywhere anytime as long as there is data from the service company or WiFi. At some point, it helps people save money by not calling others through the phone and gives the users a better life. Messenger provides our team with many new ideas for developing a new useable web app, to make our app better, some deeper research and more rigorous analysis will be needed.

3. Objectives

All requirements listed in each subsection of section 3 are in priority order, which means the requirement with the highest priority order in 3.1.1 is shown as the first requirement in section 3.1.1.

3.1 Functional Requirements

3.1.1 Identity and Relationship Functions

1. Users provide phone number/email address, nickname, and password to sign up for the system.
2. Users provide their phone number/email address and password to sign in.
3. Users can modify their profiles by changing personal information.
4. Users search for other users by phone number/email address/username.
5. A user can apply to be a friend of another user.
6. A user can approve/reject a friend request from another user.
7. A user can delete a friend.
8. A user can create a group chat by inviting friends as group members.
9. A user can leave a group chat.
10. A user, as a group member of a group chat, can invite their friends to join the group chat.

3.1.2 Communication Functions

1. Users can only communicate with their friends.
2. Only the user themselves and the friend can see messages in a chat.
3. All users in a group chat can see messages from all users in the group chat.
4. A user can view the chatting histories of chats and group chats.
5. Users can send text, image, audio, video, link, or hybrid-type messages in chats.
6. Users can only see their own and their friends' status.
7. Users can write text as comments to status.
8. Users can post text, image, audio, video, link, or hybrid-type messages as their

status.

3.1.4 System Scheduling

1. The system shall use the MVC model.

3.2 Page Requirements

1. The list shows all friends' chats, and group chats should exist the whole time after signing in.
2. Signing in/up page should be a separate page from the chatting pages.
3. The avatar list of all users in a group chat should be shown when users are in a group chat window.
4. All messages sent by a user should appear with their avatar simultaneously.

3.3 Design Constraints

1. Performance should be reasonably fast.
2. The system shall be usable at least on Chrome.
3. The system shall support PC and mobile.
4. The system must comply with the relevant privacy legislation.

3.4 Software Quality Attributes

1. Users can only see messages in chats between themselves and their friends and the group chat they joined.
2. Users can only view the history of messages that they can see.
3. The system can be remotely accessible to all users.
4. The system shall protect all users' information.

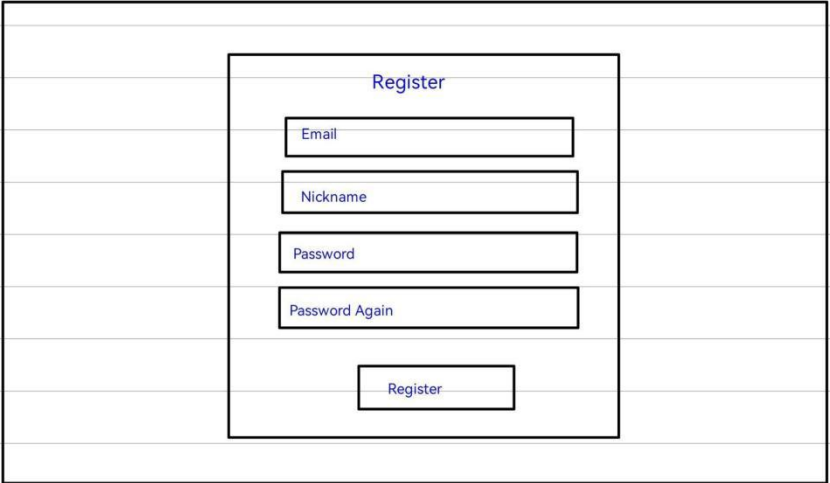
3.5 Other Requirements

1. The time plan can be divided into 2 parts, version 1 and version 2. Version 1 will be completed before Jan 22, 2023, and version 2 will be completed before Mar 14, 2023.
2. The hard deadline for the system is the end of April of next year for project completion.
3. One or many servers are required for this project. Specific requirements are still in process of investigation.

3.6 Measurability

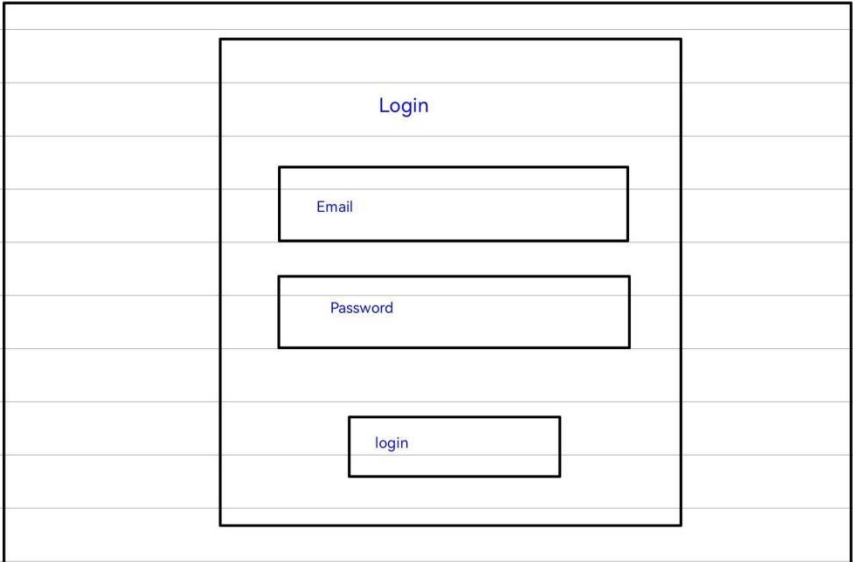
When all 3.1 (Functional requirements), 3.2 (Page requirements) and 3.6 (Other requirements) are met, the project is primed for completion, which could be seen as a "B" project. When 3.3 (Performance requirements), 3.4 (Design Constraints) and 3.5 (Software Quality Attributes) have been completed, the project has been fully completed, which could be seen as an "A" project. If extra functions are added the project is considered to be over-completed, and the project can then be rated as an "A+" project.

3.7 Sample Screens



The Register Page UI mockup consists of a large outer rectangle representing the browser window. Inside, centered, is a smaller rectangle representing the registration form. At the top of this form is the title "Register". Below the title are four input fields stacked vertically, each with a label inside: "Email", "Nickname", "Password", and "Password Again". At the bottom of the form is a button labeled "Register".

Figure 1.0: Register Page



The Login Page UI mockup consists of a large outer rectangle representing the browser window. Inside, centered, is a smaller rectangle representing the login form. At the top of this form is the title "Login". Below the title are two input fields stacked vertically, each with a label inside: "Email" and "Password". At the bottom of the form is a button labeled "login".

Figure 1.1: Login Page

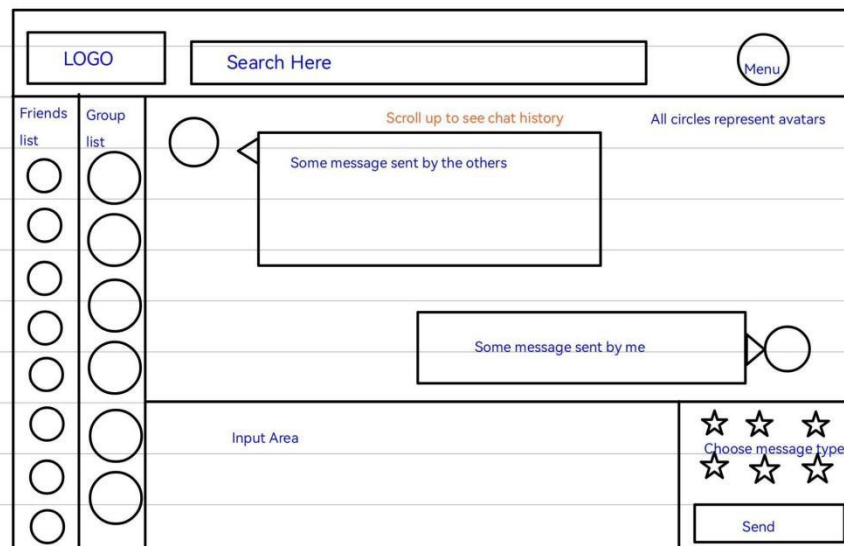


Figure 1.3: Chat Page

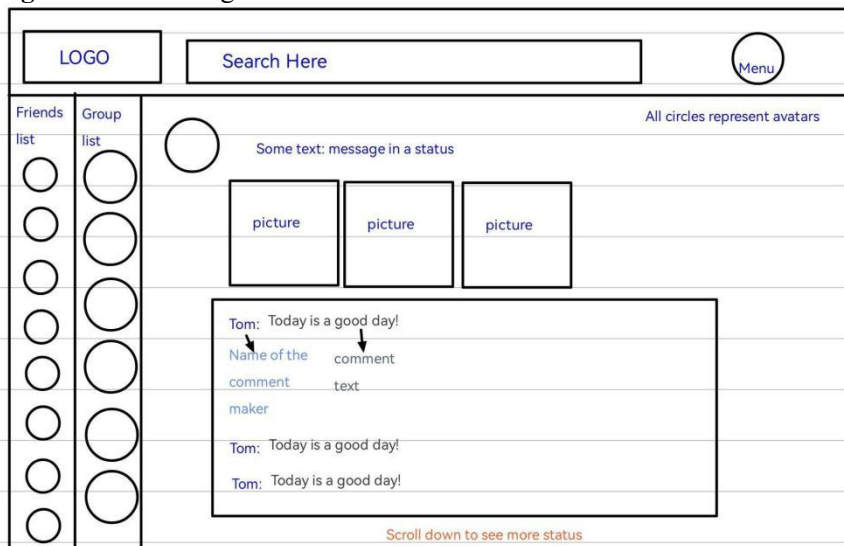


Figure 1.4: Status and Comments

4. Individual Mission Statements

The title of the courses are described in Appendix 1.

4.1 Yunzhou Liu

In this graduation project, I am mainly responsible for the back-end part of the project. The back-end part includes the creation and maintenance of databases. For the instance associated with the database, the instance properties need to correspond to the database fields in order to retrieve the database records. The back end also includes the service layer, whose main function is to integrate database operations, such as adding, modifying, deleting and searching data.

4.2 Shizhong Shang

In this project, I will focus on the front end. The front end is supported by the back end and contains the transfer data from the database, the design of the site layers, and UI design. And the most important part of my job is making our function easy enough to use, understandable for all the users and looks neat and peaceful.

4.3 Zirui Qiao

In designing, my job is to lead our team to think and establish the general structure of our project. In development, I will be the connecting hub between the front end and back end. I will help my teammates with their work if any of them are too busy to do their part of the project development. In testing, my responsibility is to create the performance tests and record the results.

5. Relation to Degree Program

5.1 Yunzhou Liu

My major is computer system engineering. I learned the basic programming languages Java, python and c. The back-end part I am responsible for in the project is written in the java language. I also learned about database structure, which can be applied to the creation of project databases.

I learned SYSC 3020, which introduces the structure of software engineering, and helps me build the overall structure of the project. The various drawing techniques I used in the project, such as UML diagrams, and the hierarchical knowledge required for a complete project come from this course. In the two courses SYSC 3303 and SYSC 3310, I learned about real-time systems, which can help me solve the concurrent problem of multiple people sending messages at the same time in the project. SYSC 4801 taught about security issues in information transmission. I will use this knowledge to ensure the safety of information transmission in the project.

ECOR 2050 and ECOR 4995 have taught me the processes and specifications to pay attention to when making projects.

5.2 Shizhong Shang

I am majoring in Software Engineering, which mainly studies program designing, programming and communication skills. In the final project, we will use HTML [15], CSS [16] and JS [17] to build the front-end of this web app, and the back-end is java based, the database is built with MYSQL and Redis.

I have learned Java code from SYSC 2004, SYSC 3110 and SYSC 4806, these three courses teach java code. In SYSC 3110, we are divided into groups to do a game project, which will help with scheduling and team working. In SYSC 4806, I chose to make an online server as the project, it will help us with the use of database, HTML, CSS and JS.

SYSC 3120 and SYSC 4106 will help us with the beginning of the designing of this web app, such as the structure, use cases and diagrams. And by digging into the project, we will get a better understanding of how to organize this project and risk management. CCDP 2100 has trained us on how to communicate with each other, especially when the team members are from different backgrounds.

5.3 Zirui Qiao

My major degree is Software Engineering which taught me about programming skills, data structure and, most important - project management. In this project, we will build the front-end part with Vue3 [18], the back-end part with Java [19] and the database part with MySQL [20] and Redis [21].

I learned about Java and used it as a tool to build projects in SYSC2004, SYSC3110 and SYSC3303. These courses helped me build a firm base for Java which would be very helpful in this project. SYSC2100 taught me about data structure and algorithms, and COMP3005 taught me about the operation of Databases; these two courses will reduce the hindrance in database development. The design and management of a project are always essential, SYSC3120 and SYSC4106 are great courses, and I will practice the knowledge in this project. This semester, SYSC4101 is a course on software validation, and I believe I will use what I have learned in this class in this project.

Communication skills can be easily overlooked, but with the help of CCDP2100 and ECOR4995, I am aware of the importance of communication skills and will apply them in this project.

6. Individual Contributions

Name	Report Contribution
Yunzhou Liu	2.1 Background - Discord 2.2 Background - Database 4.1 Individual Mission Statements - Yunzhou Liu 5.1 Relation to Degree Program - Yunzhou Liu
Shizhong Shang	2.4 Background - Facebook Messenger 4.2 Individual Mission Statement - Shizhong Shang 5.2 Relation to Degree Program - Shizhong Shang 7 Plan and Timetable (until Github Link) 8 Risk and Mitigation 9 Conclusion
Zirui Qiao	1 Introduction 2.3 Background - Whatsapp 3 Objectives 4.3 Individual Mission Statement - Zirui Qiao 5.3 Relation to Degree Program - Zirui Qiao 7 Plan and Timetable (from Github Link to the end of section)
Together	6 Individual Contributions 11 Reference Appendix 1. Definition, Acronyms, and Abbreviations

Table 1.0 Individual Contribution in Proposal

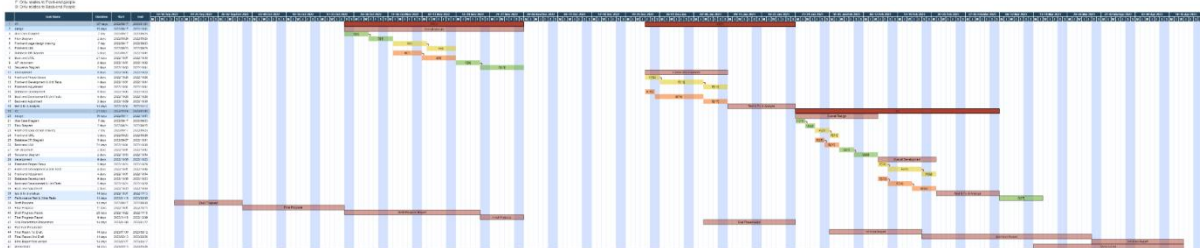
7. Plan and Timetable

The project plan is made according to the group members' ability and school schedule, the schedule is divided into two blocks, the first semester and the second semester. The first semester is mainly focused on designing and some parts of the coding, and the second semester will be used for coding, improving the web app design and writing the report. Besides that, there are some gap times in the timetable, which are based on team members' midterm and the final exam, which means to left more space for the team members to study. For more details, refer to section 7 - Timetable.

For this project, we are making an instant communication web app, so the database is needed, depending on if the system will be tested by a large number of testers, the budget for the purchase of the database should be kept.

We use GitHub to record the code required by the project. Link is <https://github.com/WEB-NART/NART.git> [22]

The timetable can be divided into 2 parts, version 1 and version 2. Version 1 will be completed before Jan 22, 2023, and version 2 will be completed before Mar 14, 2023. Timetable link: https://github.com/WEB-NART/NART/blob/main/doc/time_table.png?raw=true [23]



8. Risks and Mitigation

As a graduate project, the system will be built by our group, so one of the biggest risks will be information security. Because of this, anything that might have a connection with the money or the bank will not be included in the project. Such as money transfers, sending gift cards and in-app purchases. And some other security coding methods will be added to the system to protect the personal information provided during the registration.

To avoid the system that might be overwhelmed by the amount of data, a backup plan is needed, for example, a bigger and higher functional database is to be found. By using a better database, this problem will be solved for now.

As a backup plan, if we are not able to follow the proposal to finish this Web app, in our timetable, we have some blank space that is used to let the team member have some break. If we realized that we are not able to finish this project, we will use this time to catch up. If after that we are still not able to finish the project, we will choose to give up some of the add-on functions, and keeps the basic functions only to make sure this Web-app can be used by the public as a messaging application.

9. Conclusion

C&C is aimed to build a public, fun, peaceful and attractive web communication tool for everybody. People can upload everything with no worries, and talk to friends or strangers to have a good time. To achieve this goal, we will build C&C from basic functions: texting, grouping, muting people, deleting people and sending pictures. After that, we will find some way to make strangers able to talk to each other, so people can have a chance to know more people. C&C might also add some functions that can let users be able to do their diaries and show out their life.

10. References

- [1]“Discord Developer Portal — API Docs for Bots and Developers,” *Discord Developer Portal*, 2022. <https://discord.com/developers/docs/intro> (accessed Oct. 11, 2022).
- [2]“MySQL :: MySQL 8.0 Reference Manual :: 3 Tutorial,” *Mysql.com*, 2022. <https://dev.mysql.com/doc/refman/8.0/en/tutorial.html> (accessed Oct. 11, 2022).
- [3] “WhatsApp Features,” *WhatsApp.com*, 2022. <https://www.whatsapp.com/features> (accessed Oct. 11, 2022).
- [4]“WhatsApp Business Platform - Documentation - Facebook for Developers,” *Facebook.com*, 2022. <https://developers.facebook.com/docs/whatsapp/> (accessed Oct. 11, 2022).
- [5]“Authentication - Marketing API - Documentation - Facebook for Developers,” *Facebook.com*, 2022. <https://developers.facebook.com/docs/marketing-apis/overview/authentication> (accessed Oct. 11, 2022).
- [6]“What is a REST API?,” *Redhat.com*, 2022. <https://www.redhat.com/en/topics/api/what-is-a-rest-api> (accessed Oct. 11, 2022).
- [7]“Overview - WhatsApp Business Platform On-Premises API - Documentation - Facebook for Developers,” *Facebook.com*, 2022. <https://developers.facebook.com/docs/whatsapp/on-premises/overview> (accessed Oct. 11, 2022).
- [8]“Client Architecture - WhatsApp Business Platform On-Premises API - Documentation - Facebook for Developers,” *Facebook.com*, 2022. <https://developers.facebook.com/docs/whatsapp/on-premises/overview/client-architecture> (accessed Oct. 11, 2022).
- [9]“Security - WhatsApp Business Platform On-Premises API - Documentation - Facebook for Developers,” *Facebook.com*, 2022. <https://developers.facebook.com/docs/whatsapp/on-premises/overview/security> (accessed Oct. 11, 2022).
- [10]“Reference - WhatsApp Business Platform On-Premises API - Documentation - Facebook for Developers,” *Facebook.com*, 2022. <https://developers.facebook.com/docs/whatsapp/on-premises/reference> (accessed Oct. 11, 2022).
- [11]WEB-NART, “NART/whatsapp.md at main · WEB-NART/NART,” *GitHub*, Oct. 09, 2022. <https://github.com/WEB-NART/NART/blob/main/doc/whatsapp.md> (accessed Oct. 11, 2022).
- [12]“Messenger – Apps on Google Play,” *Google*. [Online]. Available: https://play.google.com/store/apps/details?id=com.facebook.orca&hl=en_CA&gl=US. [Accessed: 11-Oct-2022].
- [13]E. Moreau, “Facebook Messenger: Everything you need to know,” *Lifewire*, 30-Dec-

2021. [Online]. Available: <https://www.lifewire.com/facebook-messenger-4103719>. [Accessed: 11-Oct-2022].

[14]“Messenger,” *Facebook*. [Online]. Available: https://www.messenger.com/features?locale=en_GB. [Accessed: 11-Oct-2022].

[15]“HTML introduction,” *Introduction to HTML*. [Online]. Available: https://www.w3schools.com/html/html_intro.asp. [Accessed: 19-Oct-2022].

[16]“What is CSS? - learn web development: MDN,” *Learn web development | MDN*. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS. [Accessed: 19-Oct-2022].

[17]“What is javascript? - learn web development: MDN,” *Learn web development | MDN*. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript. [Accessed: 19-Oct-2022].

[18]“Introduction | Vue.js,” *Vuejs.org*, 2022. <https://vuejs.org/guide/introduction.html> (accessed Oct. 18, 2022).

[19]*Java.com*, 2022. https://www.java.com/en/download/help/whatis_java.html (accessed Oct. 18, 2022).

[20]“MySQL,” *Mysql.com*, 2022. <https://www.mysql.com/> (accessed Oct. 18, 2022).

[21]Redis, “Redis,” *Redis*, 2022. <https://redis.io/> (accessed Oct. 18, 2022).

[22] “Where the world builds software,” *GitHub*. [Online]. Available: <https://github.com/>. [Accessed: 18-Oct-2022].

[23]Web-Nart, “Web-Nart/NART,” *GitHub*. [Online]. Available: <https://github.com/WEB-NART/NART>. [Accessed: 19-Oct-2022].

Appendix 1. Definition, Acronyms, and Abbreviations

C&C --- The Come And Chat Instant Messaging Software System.

Message --- Information in text, image, audio, video, link, or hybrid.

Friend --- A relationship between users that gives both parties a quick link to each other.

Status --- A message posted by a user that the user's friends can only see.

Chat --- A connection tube between two users.

Group Chat --- A connection tube among three or more users.

The system --- represents the C&C Instant Messaging Software system.

IM --- represents Instant Messaging.

API --- application programming interface, which is a software intermediary that allows two applications to talk to each other.

HTTP --- The Hypertext Transfer Protocol is an application layer protocol in the Internet protocol suite model for distributed, collaborative, hypermedia information systems.

DDoS --- A distributed denial-of-service (DDoS) attack is a malicious attempt to disrupt the normal traffic of a targeted server, service or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic.

Guild --- A server that can accommodate multiple users to communicate at the same time

Instance --- A format used by the back end to contain the data transmitted by the front end.

Usually, instances can also be added, deleted, modified, and searched for database records.

SYSC 2004 --- Object-Oriented Software Development (BlueJ Java)

SYSC 3020 --- Intro. to Software Engineering

SYSC 3110 --- Software Development Project

SYSC 3303 --- Real-Time Concurrent Systems

SYSC 3310 --- Introduction to Real-time Systems

SYSC 2100 --- Algorithm and Data Structures

COMP 3005 --- Database Management Systems

SYSC 3120 --- Software Requirements Engineering

SYSC 4106 --- The Software Economy and Project Management

SYSC 4101 --- Software Validation

SYSC 4801 --- Intro. to Network and Software Security

ECOR 2050 --- Design & Analysis of Engineering Experiments

ECOR 4995 --- Professional Practice

CCDP 2100 --- Communication Skills for Engineering Students

SYSC 4806 --- Software Engineering Lab

SYSC 4120 --- Software Architecture & Design