



CUChat

High-Level Design Document

Department of Computer Science and Engineering

The Chinese University of Hong Kong

February 3, 2023

Group E4

An Keyu, 1155174008

Wang Zixin, 1155173944

Huang Yanwen, 1155157280

Yuan Lin, 1155141399

Yao Shihurong, 1155157155

Contents

1. Introduction

1.1 Project Overview

1.2 System Features

2. System Architecture

2.1 Technologies

2.2 System Components

2.3 Architecture Diagram

1. Introduction

1.1 Project Overview

With the pandemic subsiding, people are going out more frequently. The demand for social networking apps is also gradually increasing. CUChat is an online social platform where users can express their emotions, showcase their lifestyles, and freely communicate with "CUChat friends", which can satisfy people's social needs today. Users can make their posts visible to everyone, restrict visibility to specific groups, or keep it private for themselves. Additionally, users can search for posts, like, comment, and chat with the poster by accessing their profile. It also recommends some posts that the user may be interested in. Users can also like posts privately if they don't want to show their information. To protect privacy, CUChat only allows chatting between mutually followed users. CUChat offers slightly different interfaces and functionalities for guests, users, and administrators. In conclusion, CUChat is a chat and sharing platform with significant potential for CUHK students.

1.2 System Features

1.2.1 Post Features

The posting functions are only accessible to signed-in users in CUChat. Guest users are required to sign up or log in before they can participate in the posting feature. The platform supports text-based, image-based and video-based posts.

- **Post anonymously**

Users can post content anonymously or provide their identity optionally. When users choose to post anonymously, they can only give anonymous comments within their own anonymous posts.

- **Post with tag**

Users can add different tags for their posts. CUChat offers a wide range of tags, including academic, job advert, sentimental topics, vote and so on. Users can select a specific tag to view posts only in that category.

- **Comment a post**

Users can decide whether they want to comment anonymously or show their identity. The comment history in a post is organized chronologically. Currently, CUChat only supports text-based comments.

- **Retweet posts**

In CUChat, users can retweet others' posts along with their own comments. Comments in a retweet action are text-only.

- **Like/Dislike a post**

Every post in CUChat comes with an associated like-counter. Users can express their preference by using the like/dislike buttons, thus influencing the post's like-counter accordingly.

- **Favorite a post**

Users can add a post to their favorites by clicking the "favorite" button. They can later access these saved posts in their favorite posts bank.

- **Post showing**

While browsing the homepage, users can explore either unfollowed users' posts in the “Explore” page, or posts from users they are already following in the “Follow” page. The posts are organized in chronological order for a seamless viewing experience.

1.2.2 Account Features

- Upon registration, each user is given a permanent and unchangeable ID in CUChat. During registration, the platform will prompt users to provide a username, profile photo, and signature, all of which can be modified in their self profile page later on.
- Users can view other users' profiles through posts and comments by clicking on the user's head portrait. Within the profile page of other users, there are options to follow or unfollow them, as well as initiate a private chat.

1.2.3 Search Features

- **Post searching**

Users can search for posts by entering specific words in the search box. If a post contains the entered keywords, it will be shown in the search results.

- **User searching**

Users can search for other users by using either their username or user ID. Given the possibility of multiple users may have the same username, it is recommended that users use the unique ID for more accurate search.

1.2.4 Authentication Feature

- **Sign in/out**

To sign in the account, the user needs to input their username and password correctly for authentication check. Successful sign-in will lead to the redirection to the homepage for normal users, and to the administration interface for system admin. The signed-in user can sign out and then be redirected to the sign-in page.

- **Sign up**

The user can sign up for a new normal user account by providing a user name as an unique identifier and a password. Validity will be checked by the system before successful sign-up.

- **Visitor**

Without signing in, the user can get limited access to the homepage as a visitor.

1.2.5 Administration Features

The administration feature of CUChat is exclusively provided to the system admin. After signing in, the admin user is provided an user interface to view all users and post information and perform CRUD operation.

- **Create a user**

The admin can create a normal user by providing a valid user name and a valid password.

- **Read user information**

The admin can view all normal users' related information, including their posts, profile, comments, likes, and chats.

- **Update a user**

The admin can modify the name and/or password of a normal user to a new valid name and/or password.

- **Delete a user**

The admin can delete a normal user. When a user is deleted, all their posts, profile, comments, and chats will be removed from the database.

- **Delete a post**

The admin can delete a post. When a post is deleted, the comments/likes under it and the repost relying on it will also be deleted.

1.2.6 Home Features

The home page of CUChat includes two modes called Explore and Browse and two functions called Reminder and New Post.

- **Explore**

The Explore mode is provided to both signed-in normal users and visitors and will be displayed after redirecting from the sign-in page. It will display posts that are set to be visible to the current normal user following a recommendation algorithm based on the frequency of user's previously used post tags and posting time of the post. Signed-in normal users can click displayed elements, use the functions and redirect to other components for interaction, while visitors will be redirected to the sign-in page after clicking any interactable element.

- **Browse**

Signed-in normal users can switch to the Browse mode, and it will display posts of other users that the current user is following by the order of posting time and allow the user to interact.

- **Reminder**

Reminder function is provided to signed-in normal users under both Explore and Browse mode to remind them of unread notification and serve as an interactable element for evoking redirection to notification page.

- **New Post**

New Post function is provided to signed-in normal users under both Explore and Browse mode. After evoking, a window will pop out for the user to put in the content of the post and select visibility and tags for the post.

1.2.7 Notification Feature

The notification feature is provided to the signed-in normal users. It will display interaction involving the current user, including getting “like”, “repost”, or “comment” of the user’s posts, getting new followers and chat messages from their followers.

1.2.8 Chat Feature

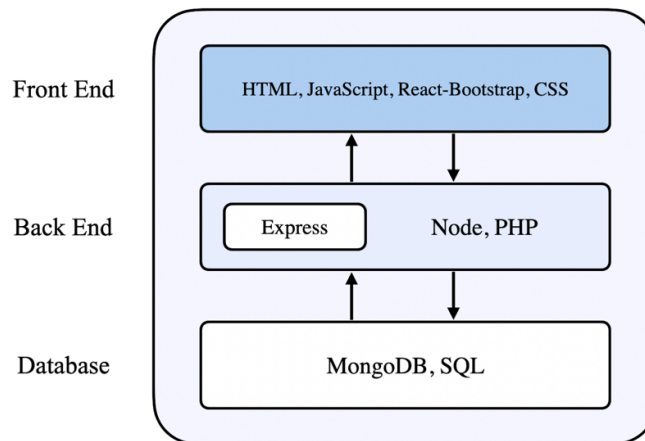
The chat feature is allowed between two normal users when they are following each other, whose content can include text, images and videos.

2. System Architecture

2.1 Technologies

The adopted technologies in this program include HTML, PHP, CSS, JavaScript, React-Bootstrap, CSS, Node, Express, MongoDB, SQL, Git and npm.

For the front end of the project, we utilize HTML and JavaScript to build the basic web structure. As for the UI implementation, React-Bootstrap and CSS is mainly used to establish a responsive and interactive user interface. The backend of the project will be built with Node and PHP. In the node server, we use Express to handle the requests and responses. The database is built with MongoDB and SQL to store users’ information. The full-stack solution is described in the following figure. The whole project is managed using Git to store code, track revision history, merge code changes, and revert to earlier code versions if needed. As for all the packages needed in the software development, we install them using npm.



2.2 System Components

For the front end, there are seven main pages, as shown in blue rectangular, each with corresponding user operations outlined in white rectangles.

Authentication Interface:

- Users can sign in, sign up, and visit as guests. The Authentication module on the backend collects user information.

Administration Interface

- Administrators can manage posts and the accounts of users. The Post Management from the backend is responsible for deleting the post. The Account Management from the backend is responsible for adding the account and deleting the account.

Post Interface

- Users can change the visibility of post content, change the tag, delete the post, and delete the comment. The Post Management from the backend supports all the above functions.

Search Interface

- Users can search for posts in CUChat by entering specific keywords or search for other users in CUChat using either their username or unique ID.

Notification Interface

- Signed-in normal users will be notified if there is any interaction regarding himself or herself, including getting “like”, “repost”, or “comment” of his or her posts, getting new followers and chat messages from their followers.

Profile Interface (Self)

- Users can customize their profile by changing their profile picture, updating their introduction. Users can also manage their followers. The Profile Management from backend supports these modifications, while the Follower Management handles all follower-related information.

Profile Interface (Others)

- Users can view other users' profiles and perform follow/unfollow actions. The Follower Management from the backend is responsible for the follow/unfollow operations.

Chat Interface

- Users can chat with people who are following each other. The Chat System for the backend supports the chat feature.

2.3 Architecture Diagram

