****

**CUChat: Data Flow Diagram Document**

Department of Computer Science and Engineering

 The Chinese University of Hong Kong

February 24, 2024

**Group E4**

An Keyu, 1155174008

Wang Zixin, 1155173944

Huang Yanwen, 1155157280

Yuan Lin, 1155141399

Yao Shihurong, 1155157155

1. **High Level Diagram**
2. **System Diagram**
   1. DFD Diagram
   2. Description
3. **System Features**
   1. Post Features’ description and DFD
   2. Account Features’ description and DFD
   3. Search Engine’s description and DFD
   4. Authentication Features’ description and DFD
   5. Administration Features’ description and DFD
   6. Home Features’ description and DFD
   7. Notification System’s description and DFD
   8. Chat Features’ description and DFD
4. **High Level Diagram**

**图示

描述已自动生成**

In our **CUChat** system, we'll have two types of operators: Administrators (*Admin*) and Normal Users (*User*). Our design includes five data stores: *Admin Data, Post Data, Chat Data, Comment Data, and User Data*. Once logged in, Admin users can access admin functions to manage user accounts and post information, while Normal Users can engage with the platform by posting or reading threads, interacting with posts through likes/dislikes or comments, and chatting with friends.

For simplicity, the diagram shows multiple data flows along the same path and some data flows are not elaborated, but it's important to note that not all data are used simultaneously, and there's no specific order to the data flow. For details on what data each function requires, please refer to the detailed descriptions for each feature.

**2. System Diagram**

**2.1 DFD (level 1)**

**图示, 示意图

描述已自动生成**

**2.2 Description**

All users and administrators should first correctly enter their username, user ID and password to log into the application to use the system with all functions. Users can also use arbitrary explore functions without sign in. To confirm the entered username and password, the ***Admin Sign-in System*** and the ***Authentication Function*** retrieve the relevant user information from the admin data and user data databases, respectively. Users who do not have their username and password for login have to visit as guests or sign up through ***Authentication Function*** to create their account with username and password through the system.

After a successful login, an authentication token will be recorded. If the authentication token is the same as this user, then the user can use ***Post Management***, ***Home Management***, ***Search Engine***, ***Notification Management*** and ***Chat Function***. If the authentication token is the same as this admin, then the admin can use all functions.

The admin can use ***Admin Funciton*** to add, update and delete user account information from the *Chat data, User data, Comment data* and *Post data* database, according to the target user ID. Meanwhile, the admin can also use ***Admin Funciton*** to update and delete the unproperly content in the post and the comment from *Post data* and *Comment data* database respectively, according to the target post ID.

The ***Post Management*** procedure enables users to retweet, like/dislike, and favor/unfavor posts in the *Post data* database. Users can also create and delete posts using the *Comment data* database. Finally, write the updated information to the *Post data* database and *Comment data* database respectively.

The ***Home Management*** procedure allows a user to explore and browse posts with the *Post data* database. The post is related to the user’s following and like data in *User data* database.

The ***Search Engine*** procedure allows a user to search the post and user with the *Post data* database and *User data* database, according to the target post ID and target user ID.

The ***Notification Management*** procedure allows a user to be informed when the user has new followers, new likes, new comments and new chat messages. The procedure iteratively fetches information from the *User data*, *Post data*, *Comment data*and *Chat data* database, and writes the data to the ***Home Management*** for display.

The ***Chat Function*** enables two users who follow each other to have a private chat. The sender of the message provides the recipient and content of the chat message. The process then saves the chat information in an Instant chat data database and regularly retrieves the previous chat history between the sender and recipient from the same database for display.

1. **System Features**

**3.1 Post Features**

**3.1.1 Description**

To release a post, the user passes his user ID, post content, tag, choice of availability (public or private) and choice of displayed identity (his user name or anonymous) to the ***release\_a\_post*** function, which records the author’s user ID, post content, post time, post tag, post availability and author’s displayed identity in the *post data* database.

To retweet a post, the user passes his user ID, the ID of the retweeted post, retweet content, tag, choice of availability (public or private) and choice of displayed identity to the ***retweet\_a\_post*** function, which records the author’s user ID, the ID of the retweeted post, retweet content, time, tag, availability and author’s displayed identity in the *post data* database.

To like/dislike a post, the user passes his user ID and the post ID to ***like/dislike\_a\_post*** function, and the user ID and the corresponding post ID will be stored in the *post data* database.

To favor/unfavor a post, the user passes his user ID and the post ID to ***favorite/unfavorite\_a\_post*** function, and the user ID and the corresponding post ID will be stored in the *post data* database.

To delete a post of his own, the user passes his user ID and the post ID to ***delete\_a\_post*** function and the post data will be deleted from the *post data* database.

To delete a comment of his own, the user passes his user ID and the comment ID to ***delete\_a\_comment*** function and the comment data will be deleted from the *comment data* database.

To make a comment, the user passes his user ID, the ID of post that he wants to comment, the comment content and his displayed identity to the ***comment*** function and the user ID, post ID, displayed identity, comment content, comment time, comment ID will be stored in the *comment data* database.

The ***show\_post\_details*** function fetches the *post data* and *comment data* asynchronously to get data consisting the displayed post including author displayed identity, post time, post ID, related post ID (for retweet post), post content, post tag, comments (each with its comment ID, comment displayed identity, comment content), # of likes of the post, current user’s like status to the post and current user’s like status to the post, and shows it to the user.

**3.1.2 DFD**

**图示

描述已自动生成**

**3.2 Account Features**

**3.2.1 Description**

To update oneself’s profile, the user passes to the ***update\_self\_profile*** function his user ID and the personal information that he wants to update, such as new avatar, new self-introduction, new user name or new password. After checking the validity, the user personal information will be updated in the *user* database.

The ***remove\_self\_follower*** and ***remove\_self\_following*** function help the user remove others who is following him or followed by him. The user passes in his own user ID and the ID of other users that he wants to delete from his following/follower list, and his list will be updated in the *user* database.

The user can follow/unfollow other users in CUChat through ***follow/unfollow\_others*** function by providing his own user ID and the other’s user ID, and the follow relationship will be updated in the *user* database.

The user can view his own follower and following list through ***display\_self\_follower*** and ***display\_self\_following*** functions. The *user* database will pass in the avatar and username of his followers/following users and display as a list of pairs to the user.

To display a user’s profile (either self or others), the *user* database will pass in the user ID, his avatar, introduction, user name, followers, following users and the *post* database will pass in the user’s post and his favorite post data to the ***display\_user\_profile*** function and, the looked-up user’s ID, avatar, introduction, user name, # of followers, # of followings, # of released posts and # of favorite posts will be showed to the query user.

To display one’s own favorite or released post through the ***display\_self\_favorite\_post*** or ***display\_self\_post*** function, the user ID and the ID of his favorite or released posts will be passed from *user* database and the post data will be passed from *post* database. Then the favorite or released posts of him will be showed to the user in chronological order.

**3.2.2 DFD**

图示

描述已自动生成

* 1. **Search Engine**

**3.3.1 Description**

1. ***search\_post*** function: Users can search the post by entering partial post content or directly searching by post ID, then the entered search content will be processed by the function and get the matched result from the *post* database.
2. ***search\_user*** function: Users can search the user by entering partial username or directly searching by user ID, then the entered search content will be processed by the function and get the matched result from the *user* database.

**3.3.2 DFD**

**图示

描述已自动生成**

**3.4 Authentication Feature**

**3.4.1 Description**

***user\_sign\_up*** function: Users can create a new account by providing username and password, then the function will pass the username, password and unique ID generated to the User database.

***user\_sign\_in*** function: Users who already have an account can sign in by entering their username and password. The function will acquire the information from the user database and verify whether the user's input is correct. If verification fails, the system will display a user login failed message. If correct, then the system will generate authentication tokens for later use.

***user\_sign\_ou***t function: For users who have already logged in, they can choose to logout by sending a logout request, then the system will display the sign in page.

**3.4.1 DFD**

**图片包含 图形用户界面

描述已自动生成**

****

**3.5 Administration Features**

**3.5.1 Admin Sign in**

**3.5.1.1 Description**

***admin\_sign\_in*** function: The first step of this function is that admin users first enter admin username and password, then the system will acquire admin username and password information from Admin Database and do verification. If the information unmatch, the system will display a login failed message. If the information matches, authentication token will be created and used in later actions.

**3.5.1.2 DFD**

图形用户界面, 应用程序

描述已自动生成

**3.5.2 User Management by admin user**

**3.5.2.1 Description**

***admin\_create\_a\_user*** function: After passing the authentication token, the admin user can create a new user by providing username and password to the function, and the function will pass username, password and the unique user ID to the User database.

***admin\_read\_user\_information*** function: After passing the authentication token and providing the user ID, the admin user can acquire users’ information, including posts, profile, comments, likes and chats information by retriving each database.

***admin\_update\_user*** function: After passing the authentication token, the admin user can update users’ username and password by providing the user ID to the function, and the function will rewrite them into the User database.

**3.5.2.2 DFD**

图形用户界面

描述已自动生成

**3.5.3 User Deletion**

**3.5.3.1 Description**

***admin\_delete\_a\_user*** function: After passing the authentication token, the admin user can delete a user. Once deleted, all information will be deleted accordingly, including posts, profile, comments, likes, and chat information.

**3.5.3.2 DFD**

图形用户界面, 应用程序

描述已自动生成

**3.5.4 Post Deletion**

**3.5.4.1 Description**

***admin\_delete\_a\_post*** function: After passing the authentication token, the admin user can delete users’ posts by providing post ID. When a post is deleted, all comments and likes under this will be deleted as well.

**3.5.4.2 DFD**

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

**3.6 Home Features**

**3.6.1 Explore**

**3.6.1.1 Description**

***explore*** function: For both signed-in and unsigned-in users, they will be recommended with some posts to explore on the home page. For signed-in users, the posts will be personalized-designed. The poster-selector will first have access to the post tag and posting time of his or her previous posts. And it will select others’ posts whose tags and posting time are more similar to user’s. Then such posts will be recommended by the post recommender and displayed to the user. As for unsigned-in users, the recommended posts will be arbitrarily selected by the post selector and then displayed to them.

**3.6.2.2 DFD**

**图形用户界面, 应用程序

描述已自动生成**

**3.6.2 Browse**

**3.6.2.1 Description**

***browse*** function: After signing in, the users can browse the posts of the people he or she is following according to the posting time. The function will access the user and post database with user ID, username, followers’ ID and followers’ name. Then it will display the post.

**3.6.2.2 DFD**

图形用户界面, 应用程序

描述已自动生成

**3.7 Notification System**

**3.7.1 Description**

1. ***new\_followers\_notification*** function: The function will access the *user* database with specific frequency to get the latest user-follower relation to detect the new followers. Then it will display the new followers notification with the new follower’s username to the target user.
2. ***new\_like\_notification*** function: The function will access the *post* database with specific frequency to get the latest like/dislike information to detect the new like/dislikes. Then it will display the new like/dislike notification with the username of the liker/disliker and the related post ID to the target user.
3. ***new\_comment\_notification*** function: The function will access the *comment* database with specific frequency to get the latest comment information to detect the new comments. Then it will display the new comment content and related post ID and commenter username to the target user.
4. ***new\_chat\_notification*** function: The function will access the *chat* database with specific frequency to get the latest chat information to detect the new likes/dislikes. Then it will display the new chat notification with chat content and sender username to the target user.

**3.7.2 DFD**

**图示

描述已自动生成**

**3.8 Chat Feature**

**3.8.1 Description**

Users can only chat with users who are both followers of each other. When chatting, a user has to open a chatroom for himself and his counterparty by passing his username and the receiver’s username to the open chatroom function. Once the chatroom is open, the retrieve chat function will have access to sender’s username and receiver’s username and retrieve their chat history from the *chat history data* database. When texting, the messages they send to each other will be passed through send message function and store new message function and be finally stored in the *chat history data* database, which can be loaded when the users open a chatroom and retrieve the history chat data.

**3.8.2 DFD**

**图形用户界面, 应用程序

描述已自动生成**