DFD Specification Document

C^3U

Group A1

Version 1.0.0

Chen Hing Chin 1155174447

Du Fangzhou 1155173892

Gao Jiaqi 1155157043

Li Peiran 1155174020

Zhou Jieya 1155173741

Printing Date: 20/02/2024

CSCI3100

Department of Computer Science and Engineering

The Chinese University of Hong Kong

Table of Contents

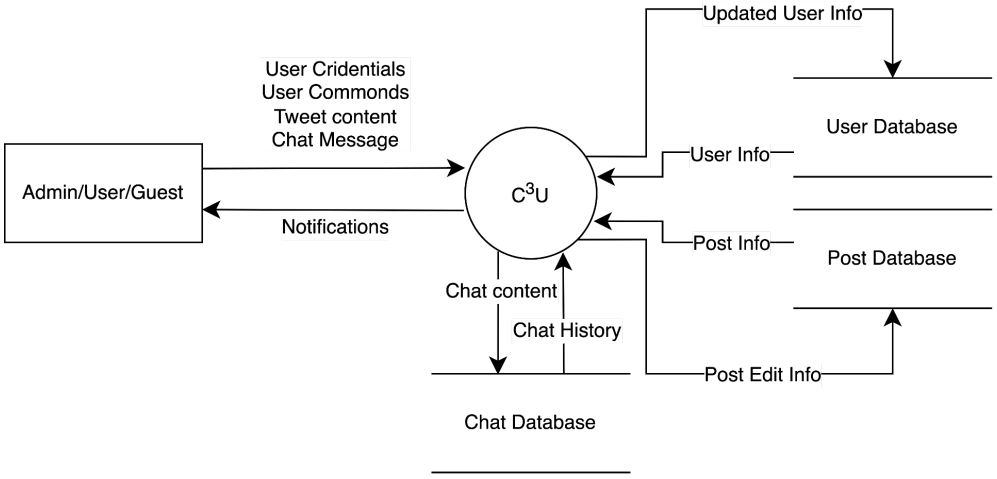
1. High-Level Context Diagram

1.1 Level 0

1.1.1 Description

The Level 0 DFD shows the general data flow of the C^{3}U between input devices, systems and databases. As shown above, the Admin or User or Guest sends user credentials, user commands, post content and chat messages to the systems and receives notifications sent by the systems. The C^{3}U systems also interact with User Database, Post Database and Chat Database retrieving and updating the user information, post contents and chat contents in the databases.

1.1.2 DFD Diagram



1.2 Level 1

1.2.1 Description

In C^{3}U, Admins and Users start their trip with corresponding login systems, which are **Admin Login** and **Login** respectively, secure login will be carried out after providing their unique IDs and passwords. An authentication token is then generated, and assists users to access diverse functions in C^{3}U. Newcomers can create their accounts with zero threshold by **Register**. In the login page, there is also a **Reset Password** function allowing users to reset password after verifying their email.
Once logged in, users can begin personalizing their **User Profiles**. They are encouraged to make their first post, enriched with multimedia content, via the **Post**, and to engage with the community by replying to comments. Additionally, navigating through their homepage could lead to an enhancement in the accuracy of personalized content recommendations. Our **Recommendation System** is dedicated to aligning with users' interests. Notice that any unpeaceful comments or post will be deleted under the supervision by Admin, Admin has a root access to editing post, comments, and has right to forbid unwelcome users through **Post** and **User Management System**.
The **Admin User** system consists of two sub-functions: **Post Related** Function and **User Related** Function. The admin user can view and delete posts, as well as view, delete, add, block, and unblock users by inputting the corresponding post or user information. Any updates made will be stored in the post and user databases.
The **Post system** encompasses the functionalities of posting, commenting, interacting, and editing posts. It receives user inputs, including post content, comment content, and other relevant information,

and interacts with the post database to store and retrieve these contents. Simultaneously, the **post system** records the user's post-related activities in the user database for future reference.

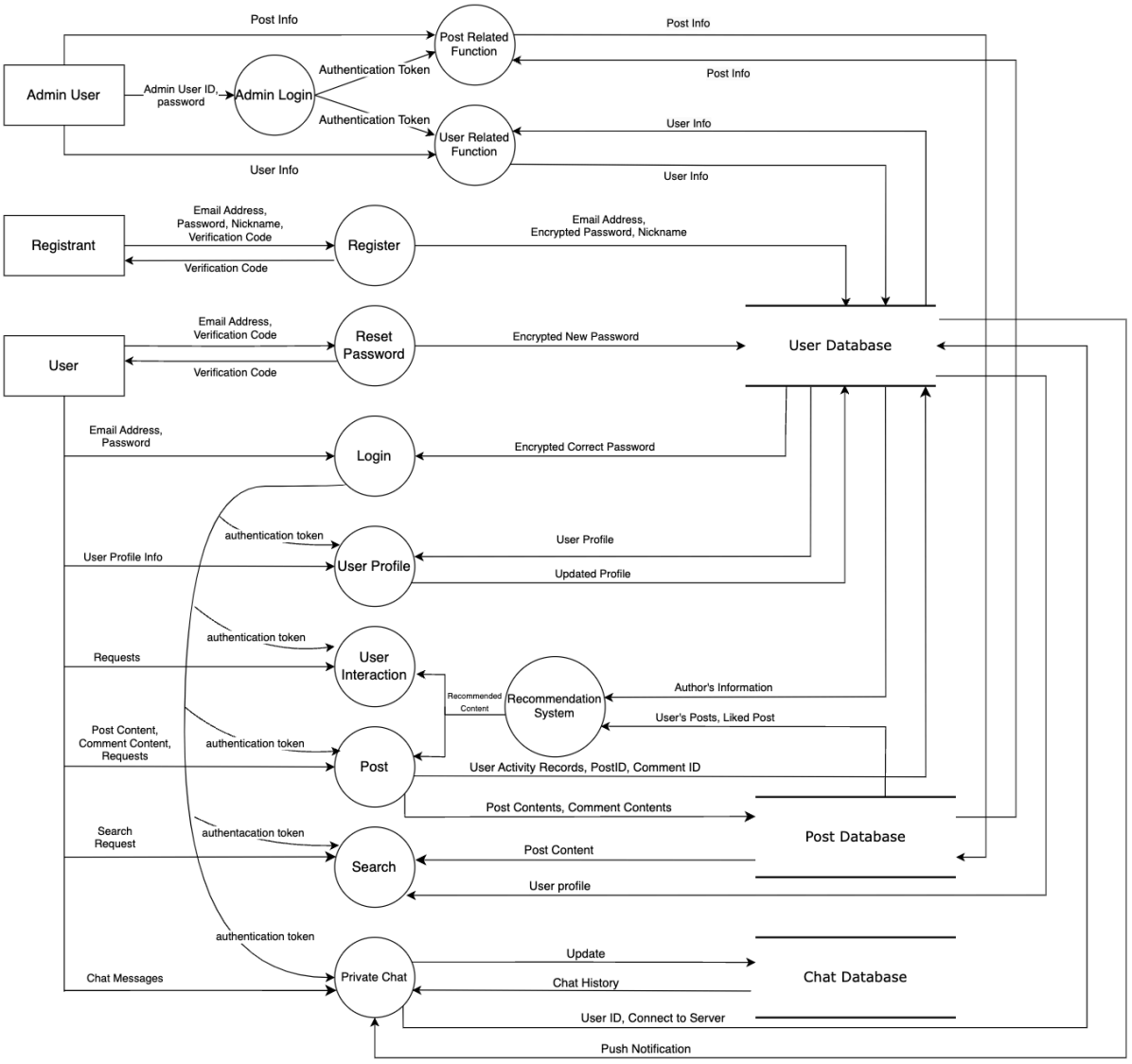
The C3U society is further enriched with a dynamic **User Interaction System**. Users could follow others for receiving their latest Activities **Notifications**. Furthermore, the platform includes a **Private Chat System**, supports personal and immediate interaction between users, allowing for both real-time and delayed communication.

The C^3U **User Interaction History** and **User Interaction Analytics** provide a snapshot of individual engagement and platform-wide trends. This report tracks user interactions, from posts to messages, offering valuable personal reflection. With clear visualizations and key insights, it enables users to refine their social presence and allows admins to steer the community effectively.

The **Search System** first receives the user's search request and then retrieves and displays relevant content from the database, including user-generated content and posts, based on the user's input.

The **Recommendation System** initially gathers the user's preferences by analyzing their own posts and liked posts records from the post database. It then proceeds to recommend posts with similar labels and suggests the authors of those posts to the user.

1.2.2 DFD Diagram



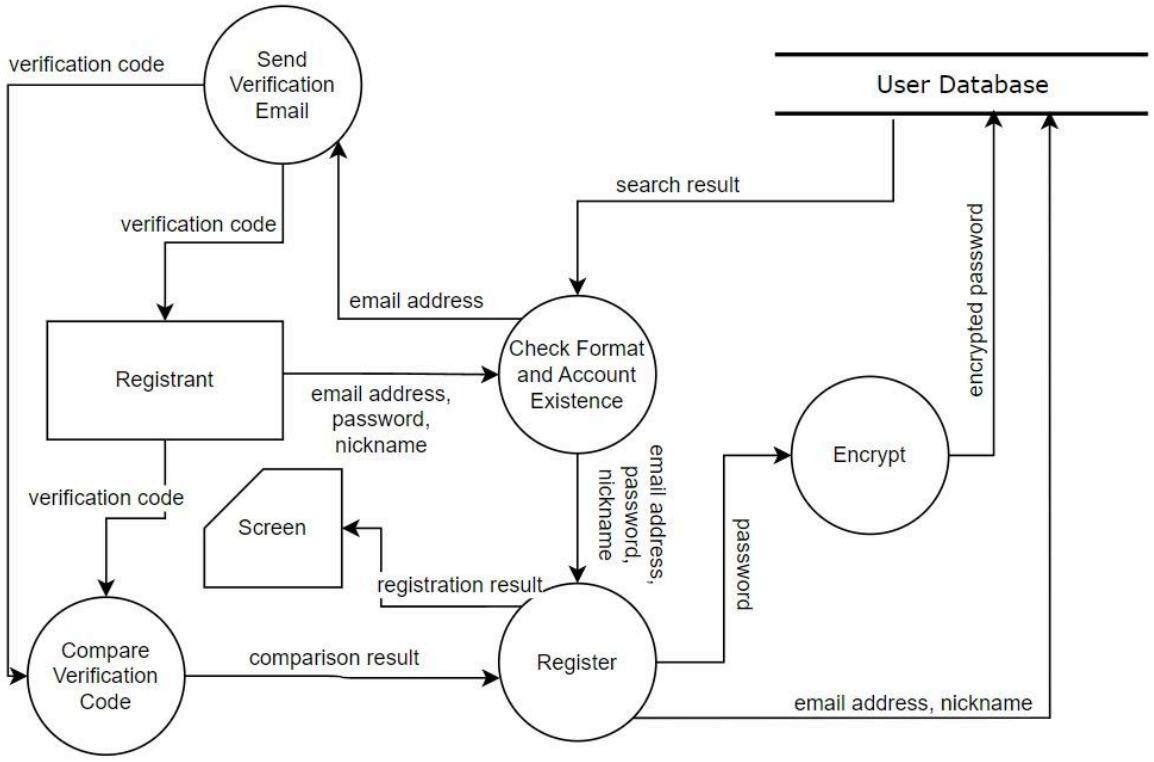
2. Feature Diagrams

2.1 Registration System

2.1.1 Description

The registrant inputs an email address, a password, and a nickname to the system, which will first be checked for format validation. In addition, the **Check Format and Account Existence** function will query the **User Database** to check whether the email address already exists in the database. If the format of the input is valid and the email address has not been registered before, the function will pass the email address data to the **Send Verification Email** function, which will send a verification code to the user's email address. On receiving the verification code, the user will input the code, which will be compared with the code generated by the system to check whether the user can access the email address. On successful verification, the **Register** function will store the email address, password, and nickname in the **User Database**, where the password should be encrypted before storage. Then, the **Register** function will present the registration result on the screen.

2.1.2 DFD



2.2 Login and Logout System

2.2.1 Description

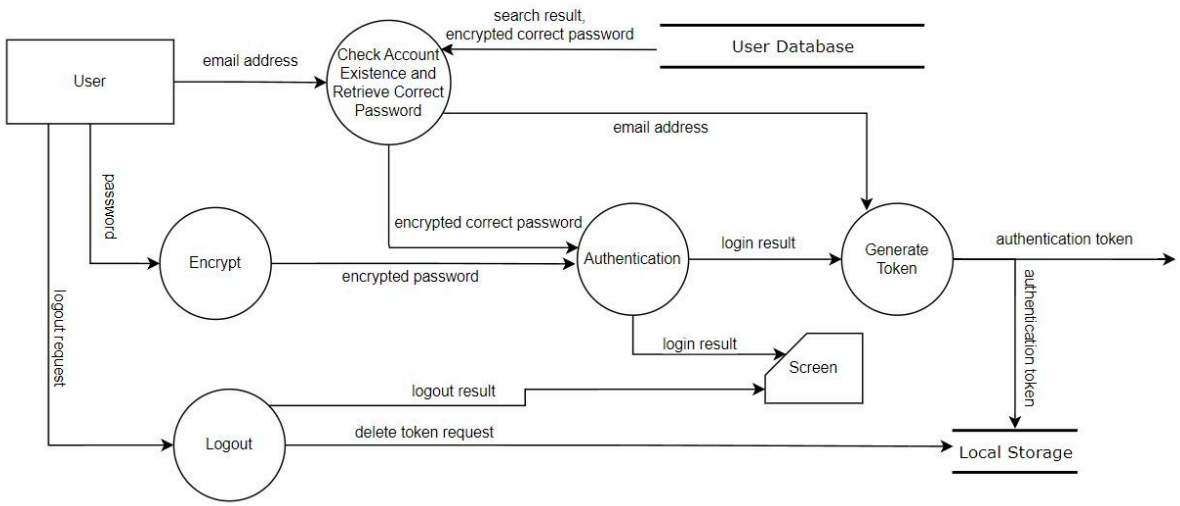
For login, the input email address will be handled by the **Check Account Existence and Retrieve Correct Password** function, which uses the email address as the search keyword to check whether the account exists and retrieve the corresponding encrypted correct password from the database. If the

account exists, the encrypted correct password will be compared with the encrypted input password in the **Authentication** function. If the two passwords are identical, the **Generate Token** function will generate a token according to the email address of the user, which will be used for authentication in the following operations to indicate the logged-in state.

For logout, the **Logout** function will delete the authentication token in **Local Storage** and print the logout result on the screen.

The login and logout process of admin users are similar to those of normal users, except that the databases of user information are different and the admin users will be redirected to the admin function page after login. The Data Flow Diagram structure of normal user login and admin user login are similar. Therefore, to avoid duplication, only one diagram is shown in the document.

2.2.2 DFD

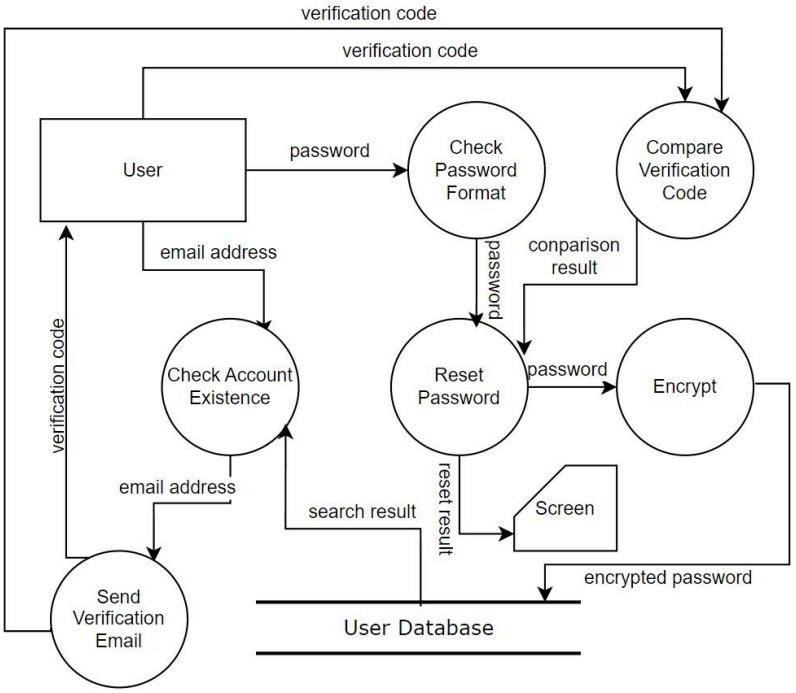


2.3 Reset Password

2.3.1 Description

When a user requests a password reset, the **Check Account Existence** function will search the **User Database** to check whether the account exists. If so, the **Send Verification Email** function will send an email to the user with a verification code. After the user inputs the verification code, the **Compare Verification Code** function will compare the input code with the code sent to verify the user's identification. At the same time, the input password will be checked for format validation in the **Check Password Format** function. If both the verification code and the password format are correct, the new password will be encrypted and stored in the **User Database**. The result of the password reset operation will be shown on the screen.

2.3.2 DFD

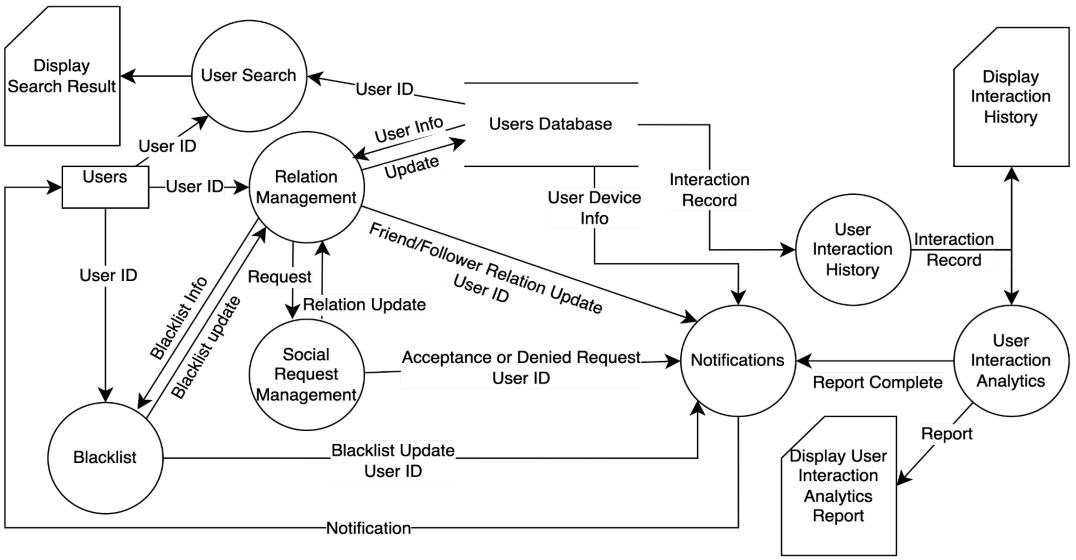


2.4 User Interaction System

2.4.1 Description

The User Interaction component is responsible for the social connection in C^{3}U. The process starts with **User Search**, where users can look up others by their User ID, the search results are then displayed. In **Relation Management**, users can delete friend or follower directly here and manage connections through friend or follower requests, which are processed in **Social Request Management**, responses to these requests update the Friend/Follower Relation. Users also have the option to blacklist others to prevent unwanted interactions, with updates reflected in the **Blacklist**. **Interaction History** is maintained, which logs the data and makes it accessible for users to review. **Notifications** are sent out to users based on certain information like updates or interactions. Lastly, **User Interaction Analytics** compiles data from interaction records, generating comprehensive reports that provide insights into user behaviors and trends, which are then displayed to the users.

2.4.2 DFD



2.5 Post System

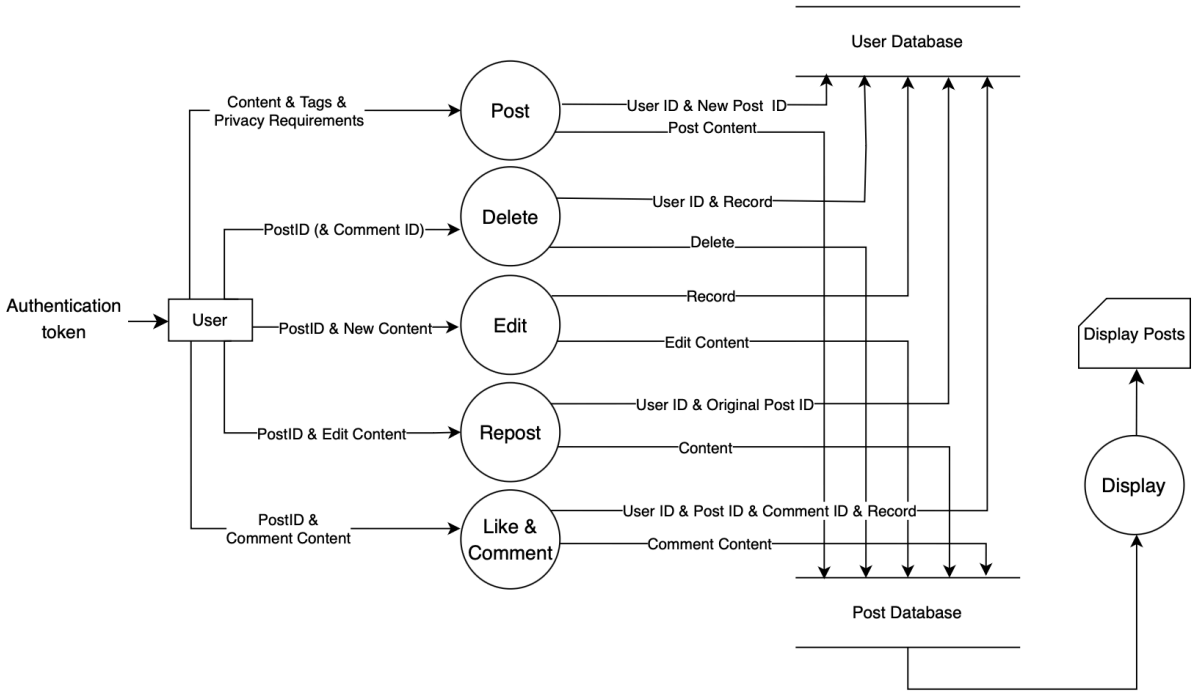
2.5.1 Description

Posting is one of the main functions of C^{3}U. The five functions related to posts include: posting posts, deleting posts or comments, editing previously published posts, reposting other users' posts, and liking and commenting on posts.

As soon as a user posts a new post, a PostID will be given and stored in the user database, which will supply information for later retrieval and recommendation. Simultaneously, the post's tags and content will be stored in specific areas of the post database based on the user's privacy requirements. When a user deletes a post or comment, the action will be recorded in the user database, and instructions will be sent to delete the corresponding content from the post database. When a user edits a previously posted post, the modification record will be transmitted to the user database, and the desired changes will be sent to the post database to update the post content. When a user reposts another user's post, the PostID of the original post will be sent to the user database. It doesn't create a new post, so if the original poster deletes the post, the reposted post will vanish as well. Users may write some comments when reposting, which will be saved in the post database. When a user comments on a post, an ID will be assigned to the new comment, which is then sent to the user database along with the corresponding post's PostID. And the content of the comment will be recorded in the post database. If the user does not make a text comment, but simply "likes" the post, this signal will be recorded in the user database as a user preference, and the data will be passed to the post database to update the number of times the post has been "liked".

In the end, the contents in the post database will be displayed to users, while the information in the user database will be used for other functions.

2.5.2 DFD

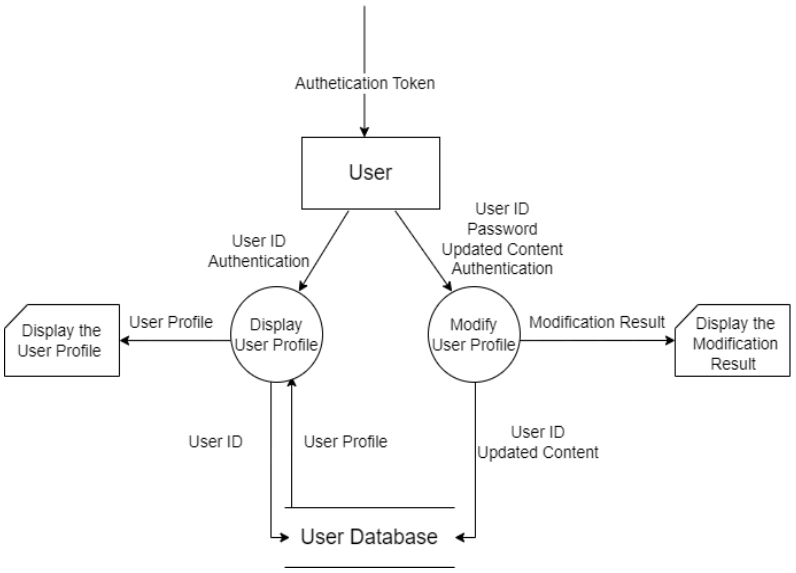


2.6 User Profile System

2.6.1 Description

Users can modify their profile by inputting the password and content to be updated. Once the authentication is received, the **Modify User Profile** function will pass the user ID and updated data to **User Database**, and then display the modification result to the user. Each user has a page showing their profile, which includes nickname, avatar, bio etc. Once the **Display User Profile** function receives the authentication and user ID, it fetches the date from the **User Database** and displays the profile to the user.

2.6.2 DFD



2.7 Admin User System

2.7.1 Post Related Function Description

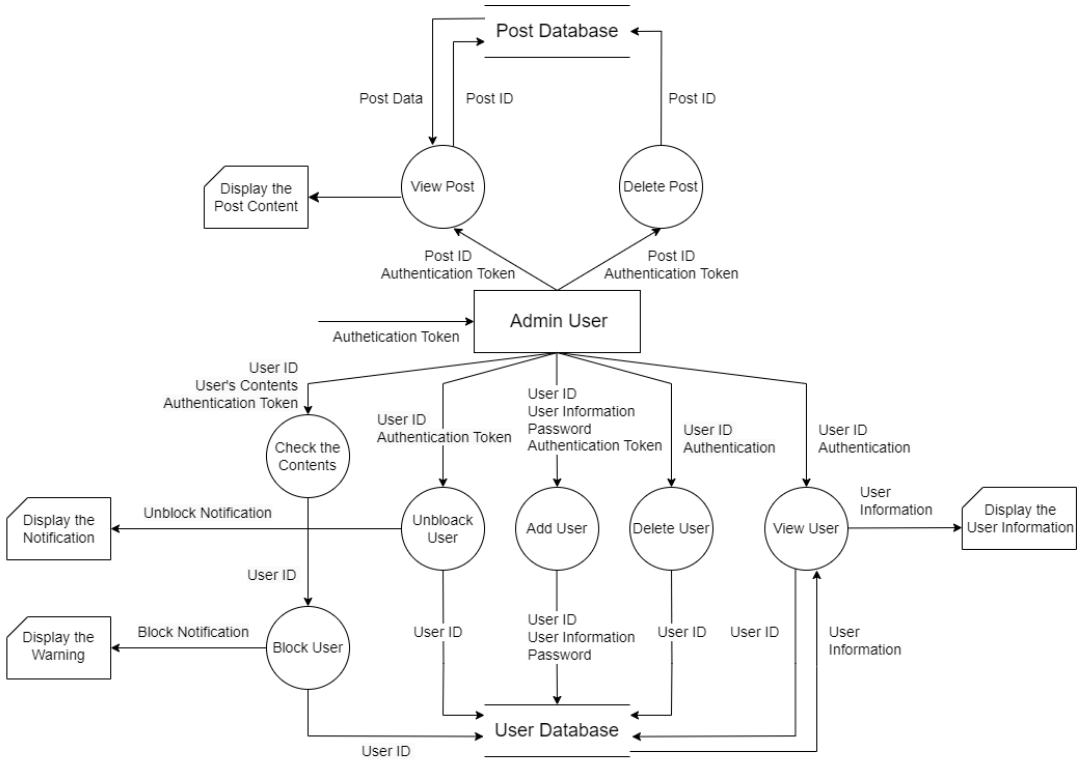
To view the post, the admin user needs to pass the authentication and keywords (such as Poster name/ID, specific words, etc) to the **View Post** function, which retrieves the post from **Post Database** and displays the content to the admin user. If it needs to delete posts, the admin user passes the authentication and PostID to **Delete Post** function, which will pass the PostID to **Post Database**, and further delete the posts.

2.7.2 User Related Function Description

The admin user can Add user by passing the authentication and user information (including user name/ID and password) to **Add User** function. After the Add User function passes the userID and information to **User Database**, it will add the new user. To delete users, the admin user passes the authentication and UserID to **Delete User** function, which will pass the UserID to **User Database**. And then delete the user. Besides, the admin user passes the authentication and UserID to **Check the Content** function. Once the contents (posts and chat) are offending the UserGuide, the UserID will be passed to **Block User** function, which will pass the UserID to **User Database**. Then it blocks the user's account and displays the warning to the user. Also, when the user's account ends the block

time, the admin user will pass the UseID to **Unblock user** function, which will pass the UserID to **User Database**. And then unblock the user's account and display the notification to the user. What's more, the admin user can view all user information by inputting UserID to **View User** function. When it receives the information and authentication, the **View User** function passes the UserID to **User Database**, and then fetches the users' information.

2.7.3 DFD



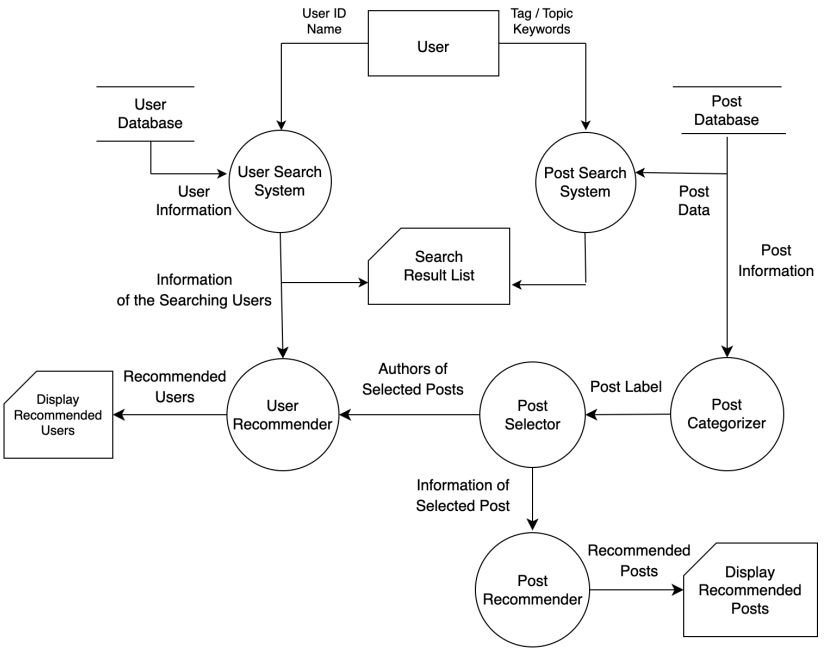
2.8 Search System

2.8.1 Description

Users can search for posts and users by typing the key information and pressing the "search" button. For posts, when a user searches according to its tag, topic and its keywords, a query will be sent to the Post Search System, the system will find posts with that key information from the Post Database and output the results on the Search Result List. For user accounts, users can search by the user ID and username. After receiving the key information, the User Search System will retrieve the information of relative users from the User Database and output the result on the Search Result List.

The system recommends users and posts to a user based on his preference. The user's preference obtained from the content of his post and the post he liked or disliked. The Post Categorizer fetches post information from the post database and returns its label to the Post Selector. The post selector selects posts that share the same label with most of the posts created and liked by the Post Recommender. The authors of the selected posts will become the input of the User Recommender. As a result, User recommender and post recommender will display relevant results to the user.

2.8.2 DFD



2.9 Private Chat System

2.9.1 Description

Users begin by authenticating into **Chat Service**, which can let users send and receive messages securely. Then they can start sending messages to specific target IDs. The chat service processes these messages and updates the **Chat Database** in real time. Users can retrieve and view their message history. When a message is sent, the **Notification Service** is activated and sends a push notification to recipients, user's device information is retrieved from the **user database**. The **Web Service** handles these interactions by managing user sessions and supporting the chat service with necessary user data. Overall, the system provides a seamless communication experience, with real-time messaging and notifications keeping users engaged and informed.

2.9.2 DFD