Server

Now the mainstream web server software is mainly composed of IIS or Apache. IIS supports ASP and can only run on Windows platform. Apache supports PHP, CGI, JSP and can run on multiple platforms and Apache is the world's number one web server platform. Although Windows is easier to use, considering that our front end wants to use thinkphp, we finally chose the Apache server.

LAN interaction  
We initially thought about putting data into the cloud, so that we can achieve synchronous uploading and other functions under different network platforms. However, we need to rent a server and take into account the cost of test development, we finally put it in the LAN. This avoid network transmission problems and are more convenient to collaborate on development.  
Preliminary design of the database

First, we identified several basic features we needed and customized the sketches. The most basic feature we want to solve is that multiple users upload and edit files at the same time. For this purpose we first need a user table for storing the user's UID and name, password, etc. Secondly we need a file table for storing file IDs, file names, creators, etc. Next, set up a authority table for viewing file permissions. Here, the file ID is used as the primary key. In consideration of possible problems during the operation of the program, in order to better understand our project, we also designed an error log table for the synchronization error occurred in the upload operation for checking.

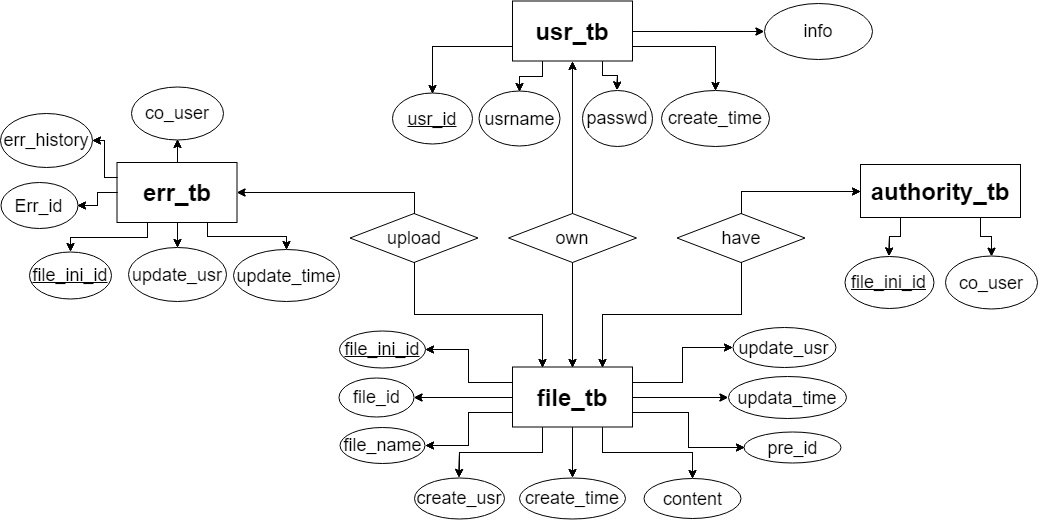
File upload conflict

After group discussion, our solution is that when the system determines that two users have performed the download operation, we will first mark the version of the document, and the version will be automatically added when uploading. When the uploading detects that the existing version of the database is greater than or equal to the uploaded version, a conflict is found. The second uploader is told and asked that another user edits the file while he is editing. The second uploader then selects the version and then overwrites the database version. The uploader and upload time will be recorded in the version. The next user will see all historical versions when searching for this file and can open or choose to download them.

Some other details

After completing the login screen, we first added the registration feature to join the new user. After the file modification function is completed, we have added the file renaming function in consideration of the file name that may need to be modified. Since we have considered that the file sync upload may overwrite the previous user's article, or if the user wants to view the old version and edit it, we have added the history section to view the old version. On this basis, we have adjusted the database again.

The final ER\_design for database



Transmission protocol

When choosing HTTP and HTTPS protocols, we originally wanted to choose HTTPS protocol. HTTPS protocol is a network protocol built by SSL+HTTP protocol for encrypted transmission and identity authentication. It is better than http protocol security and can better protect users’privacy. But considering that https needs to apply for a certificate, and there are very few free certificates. Finally, we still chose the HTTP protocol, which is relatively simple and convenient.

UI design

After designing the first draft of the database, we have roughly identified some of the function keys for our UI interface design. The first is the welcome screen, there will be two options for registration and login. After logging in, you will be taken to the user's main page. You can see the user documentation and user information, in this page user can also upload and create new files. Click to enter the file page, the user can edit and rename the text, and the user can also add or delete the permissions of other users to this file. When there is a conflict in the uploaded file, there will be a file comparison page for the user to select the file they want to save.