Deign Document: Manager Layer – User Info Manager

Hang Yuan ([hyuan211@gmail.com](mailto:hyuan211@gmail.com))

Version: 0.2 (09/05/19)

**0 Version Changes**

*Since 0.1*

This version removed all designs for client side due to changes to system structure*.*

**1 Goals**

This module will manage the information of each user in remote server system (Note: based on our design, there are no User Info Manager on client side). This design document will introduce all functions needed for user info manager.

*This design doc is only designed for server side.*

**2 Design**

The design for User Info Manager module includes two parts: (1) system design; (2) file type and format.

**2.1 System Design**

The module will manipulate the *user.data* on remote server to manage all the users’ basic information by reading, updating, creating or deleting user information or user account.

**2.1.1 Functions differ depending on sides**

Basic functions will be:

CreateUser (const ServerUserInfo &userInfo)

CloseUser (const string &userAccount)

ReadUser (const string &userAccount, ServerUserInfo &userInfo)

UpdateUser (const ServerUserInfo &userInfo)

Login (const ServerUserInfo &userInfo)

Login (const string &userAccount)

|  |  |
| --- | --- |
| **Function Name** | **Aim** |
| CreateUser | Add the new user info into the specified *user.data* file |
| CloseUser | Delete relevant user information from the *user.data* file |
| ReadUser | Read and return all information of this user from its system file |
| UpdateUser | Update relevant proportion of user information. If the user info doesn’t exist, create it first |
| Login | Verify account and password with the given information |
| Logout | Update logout time |

**2.1.2 UserInfo structure**

Based on the requirement of organizing data at server’s database, two data structures are needed.

Struct UserInfoHeader {

Unsigned totalUserNumber;

};

Struct UserInfo {

char username[15]; // Example: user

char domainName[15]; // Example: example.com

char password[16]; // Example: 1a2b3c4d

time\_t lastLoginTime;

time\_t lastLogoutTime;

bool changeStatus; // True: changed, False: no change

};

**2.2 File Type and Format**

**2.2.1 File types**

Because the user info data will be accessed multiple times, the user info files adopts *.data* as the file type.

**2.2.2 File format**

Users will be organized by its domain names in different folders. And each folder will have a user.data file to store all the user info data.

All files will contain a file header to indicate the total number of users in the file. User info data entries will follow the header one by one.