Lab 1 - Manage User Accounts

2/24/2024

0/40 Points

Attempt 1 | Review Feedback | Offline Score: **0/40** | Add Comment

Anonymous Grading: **no**

Unlimited Attempts Allowed
1/29/2024

Details

Create an object-oriented programming (OOP) program that manages user profiles. Your program should run continuously until the user selects exit.

*** We will code it in class. Please bring your laptop to class.**

Requirements:
1) Create a base class Time and all mutator/accessor methods, constructor, and destructor

private:
int hour;
int minute;
int second

2) Create a derived class DateTime : public Time and all mutator/accessor methods, constructor, and destructor

private:
int month;
int day;
int year;

3) Create a class user.h and user.cpp that contains username, password, sign-in datetime, and sign out datetime.

// See **users_data.csv** for the data requirements and create the member variables:

private:

int userId;
string username;
string password;

DateTime loginDateTime;
DateTime logoutDateTime;

Your program allows:

* As discussed in class today, below is the suggested format. You may use any DateTime format for display, input, etc.

- Save to disk **users_data.csv (https://ohlone.instructure.com/courses/29469/files/5109920?wrap=1)** **(https://ohlone.instructure.com/courses/29469/files/5109920/download?download_frd=1)** datetime format: dd/mm/yyyy hh:mm:ss
- Display format: Monday 01/29/2024
- Users input from system datetime format - you may use the provided utils.cpp

4) Create a base class Menu
5) Create a derived class UserMenu

- Read from file **users_data.csv (https://ohlone.instructure.com/courses/29469/files/5109920?wrap=1)** **(https://ohlone.instructure.com/courses/29469/files/5109920/download?download_frd=1)** and populate users to vector<User>
- and prompt the login screen menu, like:

6)Your Manage User Accounts, including:

User Menu
==========
1)Sign-in
- Enter email or mobile phone number
- Enter password
- Update sign in datetime
2) Sign Out
- User must be already sign-in
- Update sign out datetime
- Save new sign-in and sign-out datetime to users_data.csv
3) Reset Password
- User must be already sign-in
- Enter old password
- Enter new password
- Save new password to users_data.csv
4)Create account
- Enter first and last name
- Enter mobile number or email
- Enter password
- Enter re-enter password
- Forgot your password? Only role=admin can reset your password.
- Save new sign-in, sign-out datetime, and data to users_data.csv
5)Manage Profiles
- User must be already sign-in
- Allow user to change (update or delete) personal information, name, role, address, phone, email
- Save new data to users_data.csv

x) Quit
- Enter char 'x' to exit Sign Out and update sign out datetime.

7) Create a class Person (person.h for the class definition and person.cpp for the class implementation)

Example userMenu.h:

const string USERS_DATA = "**users_data.csv (https://ohlone.instructure.com/courses/29469/files/5109920?wrap=1)** **(https://ohlone.instructure.com/courses/29469/files/5109920/download?download_frd=1)** "

Your data requirements:

Id	Username	Password	Sign-in datetime	Sign out datetime	Address	City	State	Zip	Phone	Email
1001	guess	welcome	1/29/2024 18:30:11	2/10/2024 20:30:14	39399 Cherry St	Fremont	CA	94560	510-7422300	guess@ohlone.edu
1002	admin	welcome1	1/29/2024 17:10:10	2/10/2024 20:35:45	43600 Mission Blvd	Fremont	CA	94539	510-659-6000	admin@ohlone.edu
1003	john	doe	1/29/2024 18:30:01	2/10/2024 20:45:50	43600 Mission Blvd	Fremont	CA	94539	510-659-6000	jdoe@ohlone.student.edu
1004	jpham	admin	1/29/2024 18:30:01	2/10/2024 20:45:50	43600 Mission Blvd	Fremont	CA	94539	510-659-6000	jpham@ohlone.edu

```
enum USER_MENU_OPTION {
  USER_MENU_SIGNIN = '1',
  USER_MENU_CREATE = '2',
  USER_MENU_CREATE = '3',
  USER_MENU_REMOVE = '4',
  USERMENU_RESET = '5',
  USERMENU_MANAGE_PROFILE = '6',
  USER_MENU_EXIT = 'x'
};

class UserMenu : public Menu
```

private:

// Member variables and any other variables if necessary
fstream inFile;
User user;
vector<User> users;

- Prompt username and password, and authentication

- Allow 3 retries if the user enters an invalid username and password

- If the user enters a valid username and password, output a message to indicate successful login or error

8) Add to class UserMenu the following member methods:

private:
void initUserData(); // Initialize and read from users_data.csv; and populate the list (vector<User> users;)

public:
bool signIn(); // Add a member method LoginMenu::signIn definition and implementation

public:
bool create(); // Add a member method create user definition and implementation

public:
bool remove(); // Add a member method remove user definition and implementation

public:
bool reset(); // Add a member method reset user definition and implementation

public:
bool manageProfiles(); // Add a member method to manage user definition and implementation

public:
void exit(); // Enter char 'x' to exit Sign Out and update sign out datetime.

example

public:
bool signOut(); // Save loginDateTime and logoutDateTime data to file users_data.csv

9) Create a main.cpp and implement the above requirements

int main() {

// TODO implement Login here!!!
return 0;
}

10) Your program should follow the C++ Programming guidelines

Submission: Please submit the midterm programming like your lab assignment. Ex. **lab1-<your name>.zip** and **the screenshots of the output**.

Importantly: Your lab assignment is to be done individually. You may not copy someone else's work.

Choose a submission type

Upload | Canvas Studio | More

Choose a file to upload

or

Webcam Photo

Canvas Files

(https://ohlone.instructure.com/courses/29469/modules/items/1472583)

Assignment

(https://ohlone.instructure.com/courses/29469/modules/items/