User Account Management System Documentation

Objective

Design and implement a User Account Management System (UAMS) for our application to allow users to access the system. The system allows for the management of user accounts, with different functionalities for users and administrators.

Background

The application requires a user account management capability to enable users to access the system. Administrators should be able to perform various tasks related to user account management.

The System

This UAMS Project [1] is developed using the Java programming language which is known to provide great support for object-oriented programming (OOP) principles, making it an ideal choice for implementing the system.

The project is structured into the following Java classes:

- **User.java**: Defines attributes and methods common to all users.
- **Admin.java**: Inherits from the User class, adding admin-specific methods.
- **UAMS.java**: Manages user interactions and method invocations.
- **App.java**: Contains the main method for user input and system operation.

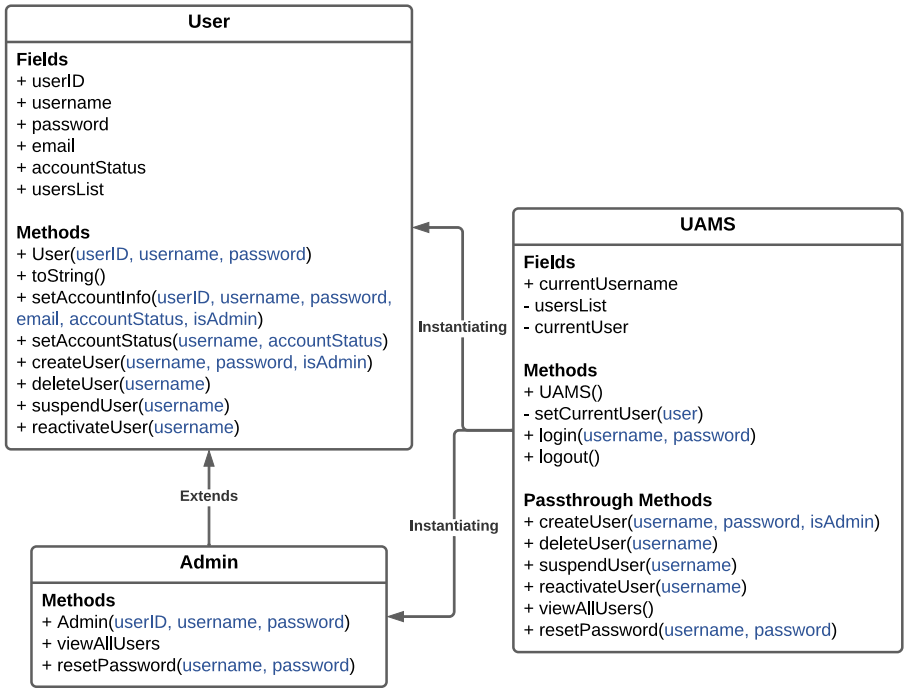

Figure 1. A high-level overview of the architecture

The system operates in two distinct stages. Stage 1 is dedicated to the login process, while Stage 2 encompasses a wide array of user actions and essential commands.

To interact with the system, simply type the command without trailing information. If required, the system should prompt you to input extra information for the task.

For an overview of the system operations, please refer to Figure 2.

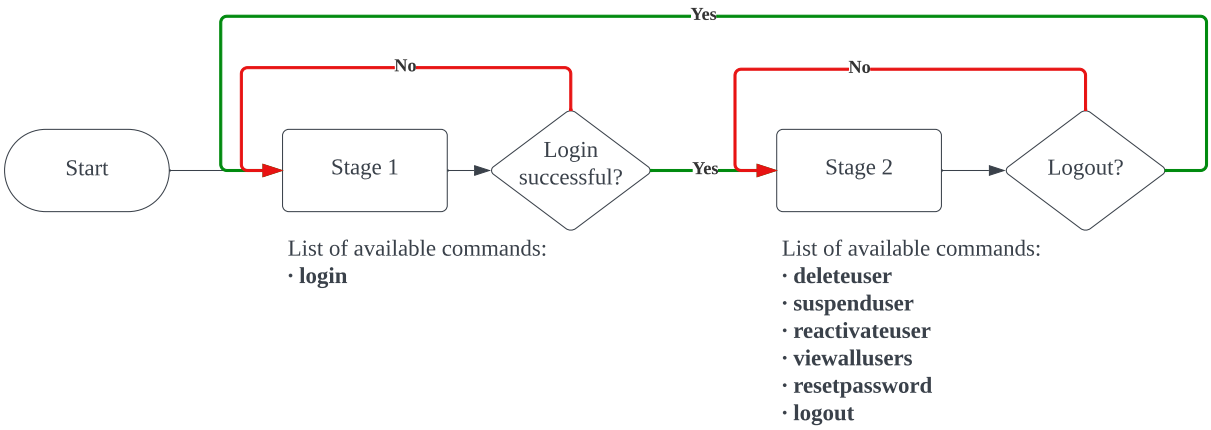

Figure 2. A brief introduction to system operation

User permissions in the system dictate the actions users and administrators can take within the system. These permissions shape user control over accounts, ensuring a secure and organized environment through meticulous assignment.

For further details about available user permissions, please refer to Table 1.

Table 1. Permission Table for Users and Admins

A User can/cannot modify…

Command	User	Admin
createuser	✔	✖
deleteuser	self-only	✖
suspenduser	✖	✖
reactivateuser	✖	✖
viewallusers	✖	✖
resetpassword	self-only	✖
logout	self-only	n/a

An Admin can/cannot modify…

Command	User	Admin
createuser	✔	✔
deleteuser	✔	✔
suspenduser	✔	✔
reactivateuser	✔	✔
viewallusers	✔	✔
resetpassword	✔	✔
logout	self-only	n/a

Design Choices

1. System Inspiration: The design of our User Account Management System draws inspiration from Microsoft Windows' UAMS, a well-established and dependable system. The familiarity and reliability of this model influence the approach taken in our implementation.

2. Initial Administrator: The system begins with an initial administrator account, responsible for creating new user profiles. This approach ensures the controlled creation of accounts and aligns with the principle of granting administrative privileges to trusted users.

3. Unique User IDs: The User Account Management System employs a unique User ID (userID) for each user. This design choice prevents the reuse of user IDs, ensuring that each user's identity remains distinct. This approach adds an extra layer of security and traceability to the system.

4. Case-Insensitive Username: A notable feature of the system is the case-insensitivity of usernames during the login process. This feature serves two purposes:

- **Aesthetic Display:** The case-insensitive nature of usernames ensures that they are displayed uniformly, contributing to a visually pleasing interface.
- **User Convenience:** Users can enter their usernames without worrying about case sensitivity, streamlining the login experience.

5. User and Administrator Functionalities: The system is designed to offer distinct functionalities to regular users and administrators. While standard users primarily interact with the application, administrators possess elevated privileges to manage user accounts effectively.

6. Error Codes and Help Messages: The system employs a comprehensive set of error codes and associated messages for most actions. Users receive intuitive feedback for the outcome of each command, enhancing their interaction experience.

7. Security Measures: In order to enhance the security of the system, the following security measures have been implemented:

- **Confined Object Access:** User and Admin objects are confined within the UAMS class, bolstering security by preventing external access.
- **Password Security:** Passwords are securely handled, and plain text storage is avoided to uphold data privacy.

8. Reliability and Proven Model: By emulating the reliable structure of Microsoft Windows' UAMS, we create a system that adheres to established design practices. This decision contributes to the robustness and effectiveness of our User Account Management System.

Conclusion

The design and implementation of the User Account Management System (UAMS) stem from a blend of established practices and user-centered considerations. By drawing inspiration from a proven model, utilizing password hashing, and embracing a case-insensitive username approach, the system provides an effective, user-friendly, and secure experience for both regular users and administrators.

References

[1] E. Huang, "UAMS: A simulation of User Account Management System," 2023. [Online]. Available: https://github.com/erichuang1/UAMS.