Task Documentation: User Authentication and Management System

1. Introduction

This document presents the implementation steps for a User Authentication and Management System. The system comprises user registration, login, password update, and account deletion functionalities. Each step is meticulously documented to ensure clarity and completeness.

2. Requirements Analysis

The system requirements are as follows:

User Registration:

Implement the register_user function.
Check for existing usernames.
Store user information securely.
Validate email format during registration.

User Login:

Implement the login_user function.
Check for logged-in users.
Verify credentials.
Enforce password complexity.

Password Update:

Implement the update_password function.
Check if the user is logged in.
Verify old password.
Enforce new password complexity.

Account Deletion:

Implement the delete_account function.
Check if the user is logged in.
Verify credentials.

Testing:

Create scenarios for testing.
Handle expected exceptions.
Validate security measures.

3. Implementation

a. Database Setup:

The SQLite database users.db is used to store user information.
Two tables are created:
register_user: Stores username, email, and password.
login_user: Tracks logged-in users.

b. Custom Exceptions:

Custom exceptions are defined to handle specific error cases:
UsernameExistsError: Raised when attempting to register with an existing username.
InvalidCredentialsError: Indicates invalid login credentials or password update details.
UserNotLoggedInError: Raised when attempting password update or account deletion without being logged in.
UserAlreadyLoggedInError: Indicates attempting to log in while already logged in.

c. Function Implementations:

User Registration:

Function: register_user(username, email, password)
Checks for existing usernames.
Validates email format.
Securely hashes the password before storing.

User Login:

Function: login_user(username, password)
Checks for existing login sessions.
Verifies credentials against stored information.
Securely hashes the entered password for comparison.

Password Update:

Function: update_password(username, old_password, new_password)
Validates the user's login status.
Verifies the old password.
Updates the password after ensuring complexity requirements are met.

Account Deletion:

Function: delete_account(username, password)
Checks if the user is logged in.
Verifies credentials before deleting the account.

d. Main Function:

The main function provides a user interface to interact with the system.
It allows users to register, log in, update their password, and delete their account.

4. Conclusion

The User Authentication and Management System have been successfully implemented according to the specified requirements. Thorough testing has been conducted to ensure functionality and security.

Task Completed By: Santosh Shrestha

Date: 2024/03/05