

Abstract Entity

User

Parent Class:

Child Class:

Responsibilities:

- Contain private instance variables:
userName(unique), password, isAdmin(0 stands for false, 1 stands for true), loginHistory, bannedUntil(stores the date ban will be lifted), isSignedIn(True if yes, False if no)
- Getters and setters for username, password, loginHistory, bannedUntil, isSignedIn, isAdmin
- Append to loginHistory
- Getter for loginHistory

This class does not depend on other classes.

Use case

UserManager

Parent Class:

Child Class:

Responsibilities:

- Log in a User
- Stores a map of all Users (username to user)
- Check if a User has logged in yet by username
- Temporally Ban a non-admin user.
- Delete a non-admin user by username
- Return the login history of a user by username.
- Create a admin/normal user
- Read user data into map.
- Save user data to local file.

User
IGateway

Use case

UserController

Parent Class:

Child Class:

Responsibilities:

- Stores an instance of UserManager
- Calls methods in UserManager(login, delete, ban, create account, check login history) when needed.

Usermanager

Controller

LoginController

Parent Class:

Child Class:

Responsibilities:

- Takes user input and decides to send message to other two controllers accordingly.
- Prompts users to enter their instructions in the terminal,
- Stores an instance of AdminController
- Stores an instance of NormalUserController
- Reads in current user data and save them appropriately.

AdminController

NormalUserController

Interface

IGateway

Parent Class:

Child Class:

Responsibilities:

- Make sure the class implementing it has methods read, save.

Gateway

Gateway

Parent Class:

Child Class:

Responsibilities:

- Has access for saving and loading of users data to and from a local file.

IGateway