Class User

- _firstName (string)

- _lastName (string)

- _id (int)

- _password (string)

- _canCreateUsers (bool)

- _canAssignResponsibilities (bool)

- _responsibilities (Responsibilities[])

+constructor()

*constructor(firstName, lastName, id, password)

*SetCanCreateUsers (bool)

*GetCanCreateUsers (bool)

*SetCanAssignResponsibilities (bool)

*GetCanAssignResponsibilities (bool)

*CreateUser(user : User)

*AssignResponsibilities(user : User, responsabilities : Responsibilities[])

Class SuperUser : User

+constructor()

*constructor(firstName, lastName, id, password)

*TurnIntoSuperUser(user: User)

*TurnIntoNormalUser(user: User)

Enum Responsibilities

Check server log
Fix responsive layout
Update project hierarchy

To prevent redundancy getters and setters for the example attributes are not presented

In the User class the constructor initializes the canCreateUsers and canAssignResponsibilities variables as false, and in the SuperUser class it initializes them as true

CreateUser and AssignResponsibilities methods validate their respective variables (canCreateUsers or canAssignResponsibilities) before doing anything to ensure the user has those priviligies

The SuperUser class has two additional methods: TurnIntoSuperUser and TurnIntoNormalUser. They add all priviligies to the user and remove them respectively