

```

import pickle #used to store objects

'''import exceptions classes:'''
from auth import (AuthException, UserAlreadyLoggedIn, InvalidCredentials,
UsernameAlreadyExists, PasswordTooShort, PermissionExistsError, UserUnknown,
PermissionError)

'''import classes:'''
from auth import User, AccessControlListEntry, Authenticator, Authorizer

'''import auxiliary functions:'''
from auth import screen_clear

#-----
# Storage Locations
#-----

USER_PICKLE = 'user_pickle.dat' #storage location of user data
ACL_PICKLE = 'acl_pickle.dat' #storage location of access control list ACL

# -----
# init & Welcome screen
# -----

authenticator = Authenticator() #start up a authenticator object
authorizer = Authorizer() # start up a authorizer object
screen_clear() #clear the screen
print('Welcome to ASMIS V0.9')
print(''#-----#''')
print(''# Check README.md for login credentials #''')
print(''#-----#''')
print()

#-----
# Functions used within this module
#-----
def terminate_asmis():
    ''' End ASMIS. Note: pickle files don't need to be closed sind the 'with
    open' statement is used'''
    print()
    print('ASMIS terminated')
    exit()

#-----
# Login
#-----

username_typed = input('Please enter your username: ' )
password_typed = input('Please enter your password: ' )
screen_clear() #clear the screen

```

```

try:
    authenticator.login(username_typed,password_typed)
except UserAlreadyLoggedIn:
    print ('You are already logged in')
    terminate_asmis()
except InvalidCredentials:
    print ('Username and / or password incorrect')
    terminate_asmis()
except:
    print ('Something went wrong')
    terminate_asmis()
else:
    print ('Login successfull')
    print ()

#-----
# Menu & program flow
#-----

while True:

    print('You are logged in as', authenticator.logged_in_users[0])
    print()
    print('ASMIS menu')
    print('1 - See your medical history')
    print('2 - See other patients medical history')
    print('3 - Add new user')
    print('4 - Display permission of a certain user')
    print('5 - Change / Assign permissions to a certain user')
    print('6 - Add new global permissions')
    print()
    print('9 - Terminate ASMIS')
    print()

    user_selection = input('Please enter a number and press enter: ')
    screen_clear()

    if user_selection == '1':
        '''See your medical history'''
        if authorizer.is_authorized(authenticator.logged_in_users[0],
            'read_own_medical_history'):
            users = []
            with open(USER_PICKLE, 'rb') as f:
                users = (pickle.load(f)) #load objects from pickle
            for user in users:

```

```

        if user.username == authenticator.logged_in_users[0]:
            user.read_own_medical_history()
    else:
        print("You do not have permission to do this task")

elif user_selection == '2':
    '''See other patients medical history'''
    if authorizer.is_authorized(authenticator.logged_in_users[0],
    'read_other_patients_medical_history'):
        users = []
        with open(USER_PICKLE, 'rb') as f:
            users = (pickle.load(f)) #load objects from pickle
        username_typed = input('Please enter the username of the user, where'
        'you want to see his / her medical records: ')
        for user in users:
            if user.username == authenticator.logged_in_users[0]:
                try:
                    user.read_other_patients_medical_history(username_typed)
                except UserUnknown:
                    print('Username unkown - try entering a valid username of'
                    'the ASMIS')
                except:
                    print('Something went wrong')
    else:
        print('You do not have permission to do this task')

''' Add new user'''
elif user_selection == '3':
    if authorizer.is_authorized(authenticator.logged_in_users[0], "add_user"):
        username_typed = input('Please enter the username of the user to be'
        'created: ')
        password_typed = input('Please enter the password of the user to be'
        'created: ')
        try:
            authenticator.add_user(username_typed, password_typed)
        except UsernameAlreadyExists:
            print('Username already exists. Try logging in to your account, '
            'or if you have no account, pick another name')
        except PasswordTooShort:
            print('password must be min. 8 characters')
        #except:
        #    print('Something went wrong')
        else:
            print('User added')
    else:
        print('You do not have permission to do this task')

```

```

''' Display permission of a certain user'''
elif user_selection == '4':
    if authorizer.is_authorized(authenticator.logged_in_users[0],
        "print_user_permissions"):
        username_typed = input('Please enter a username to view his / her '
            'permission(s): ')
        try:
            authorizer.print_user_permissions(username_typed)
        except UserUnknown:
            print('Username unknown - try entering a valid username of the '
                'ASMIS')
        except:
            print('Something went wrong')
    else:
        print('You do not have permission to do this task')

elif user_selection == '5':
    '''Change / Assign permissions to a certain user'''
    if authorizer.is_authorized(authenticator.logged_in_users[0],
        "change_user_permissions"):
        username_typed = input('Please enter a username to view his / her '
            'permission(s): ')
        try:
            authorizer.print_user_permissions(username_typed)
        except UserUnknown:
            print('Username unknown - try entering a valid username of the '
                'ASMIS')
        except:
            print('Something went wrong')
        print()
        print('These are all current global permissions:')
        for entry in authorizer.global_permissions:
            print(entry)
        print()
        while True:
            action = input('Do you want to (1) delete or (2) add a permission '
                '? Please enter a number to select: ')
            if action == '1' or action == '2':
                break
            else:
                print('Please enter number 1 or 2')
        perm_name = input('Please enter the name of the permission you want '
            'to delete or add: ')

        try:
            authorizer.change_user_permissions(perm_name,username_typed,action)
        except PermissionError:
            print()

```

```

        print('This global permission does not exist.')
        print('No changes were made')
    #except:
        #print('Something went wrong')
else:
    print('You do not have permission to do this task')

''' Add new global permissions'''
elif user_selection == '6':
    if authorizer.is_authorized(authenticator.logged_in_users[0],
    "add_new_global_permission"):
        print('(!) Global permissions are not stored at the backend with '
        'this test version of the ASMIS.')
        print ('(!) Changes to the global permissions are'
        'therefore lost after a restart.')
        print()
        print()
        print('These are the current global permissions')
        print(authorizer.global_permissions)
        print()
        permission_typed = input('Please enter a new global permission you '
        'want to add: ')
        screen_clear()
        try:
            authorizer.add_new_global_permission(permission_typed)
        except PermissionExistsError:
            print ('This global permission is already in the list. Try '
            'another one')
        except:
            print('Something went wrong')
        else:
            print('new global permission successfully added')
    else:
        print('You do not have permission to do this task')

elif user_selection == '9':
    terminate_asmis()

else:
    '''catch any nonsense user menu input'''
    print('invalid choice')

print() #new line before printing the menu again

```

