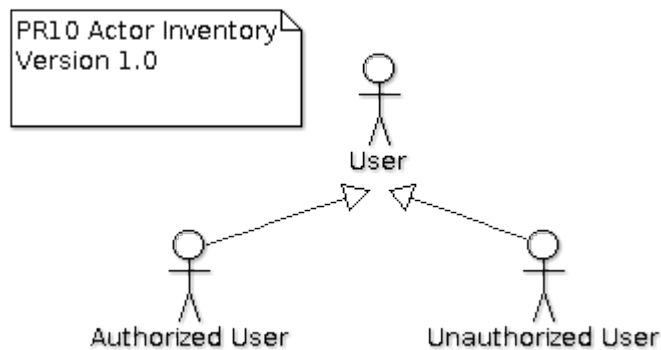


Problem Domain:	Login
Mission Summary:	Create a "Login Loop"
Prerequisite:	Completion of "Python 1000" (available at Udemy.com)
Your Script Name:	PR10_PasswordCounter.py
Solution Name:	PR10S_PasswordCounter.py
Version:	1.0

## Synopsis

Application security requires the validation of credentials. In this exercise, we will create a program that will "loop" until a proper password has been entered.



System actors supported include a "User" as well as the "Authorized" and "Unauthorized" user types.

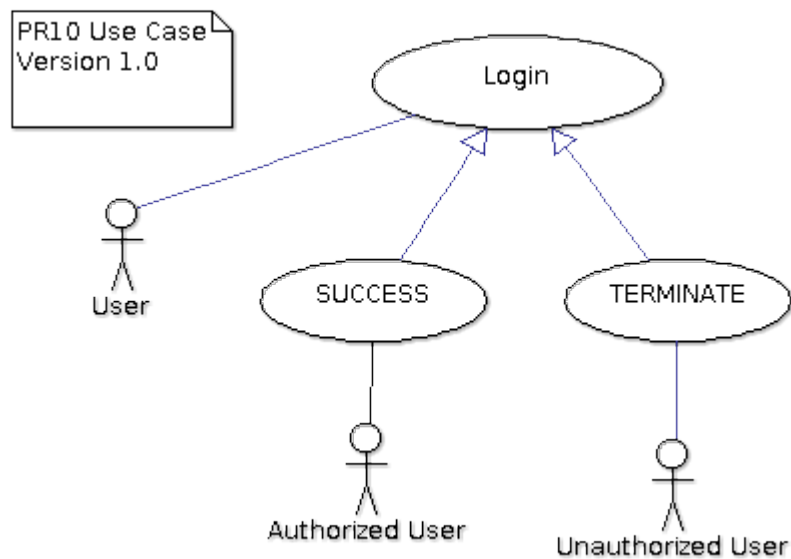
## Requirements

1. Create a script named PR10\_PasswordCounter.py
2. Define a "User ID" and a "Password"
3. USER\_LOOP: Create a "While Loop"
  1. Ask for User Name
  2. Greet the user by saying "Hello" for the name given
  3. PASSWORD\_LOOP: Create a "Nested Loop"
    1. Ask for the Password
      1. SUCCESS: Loop until the "hard coded" password is entered
        1. When Password is Entered Correctly:
        2. Display "Login Success"
        3. Display "Welcome" for the user name provided
      2. TERMINATE: After three (3) unsuccessful attempts
        1. Display "Too Many Unsuccessful Attempts"
        2. Display "Program Termination"
        3. Use exit() to terminate the program
4. BONUS
  1. Place the USER\_LOOP into a Function called "Login"
  2. "Encapsulate" (hide) the Password inside of the function
  3. Provide the number of PASSWORD\_LOOP attempts as an input parameter
  4. Return the "User ID" entered

## Related Diagrams

The following is a graphical requirement overview.

1.) The top-level Use Case defines three (3) main scenarios:



2.) The main Activity Diagram depicts a graphical summary of the requirements, as well as a reasonably demonstrative operational overview:

