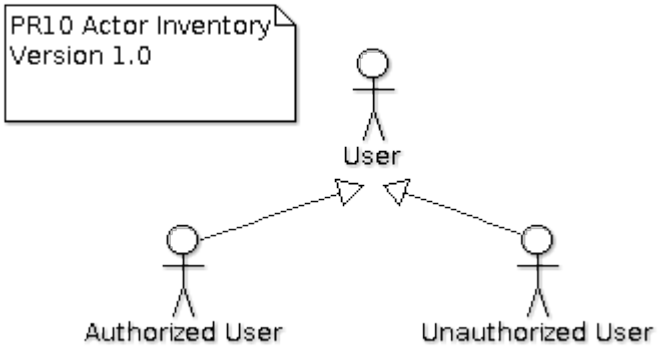
Problem Domain:	Login
Mission Summary:	Create a “Login Loop”
Prerequisite:	Completion of “Python 1000” (available at Udemy.com)
Your Script Name:	PR10_PasswordCounter.py
Solution Name:	PR10S_PasswordCounter.py
Version:	1.0

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.

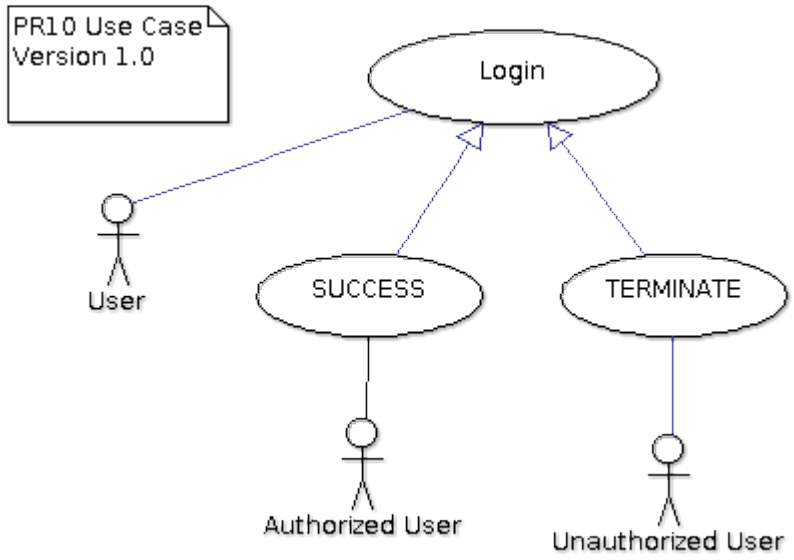
Requirements

1) Create as script named PR10_PasswordCounter.py
2) Define a “User ID” and a “Password”
3) USER_LOOP: Create a “While Loop”
   a) Ask for User Name
   b) Greet the user by saying “Hello” for the name given
   c) PASSWORD_LOOP: Create a “Nested Loop”
      i) Ask for the Password
         (1) SUCCESS: Loop until the “hard coded” password is entered
            (a) When Password is Entered Correctly:
            (b) Display “Login Success”
            (c) Display “Welcome” for the user name provided
         (2) TERMINATE: After three (3) unsuccessful attempts
            (a) Display “Too Many Unsuccessful Attempts”
            (b) Display “Program Termination”
            (c) Use exit() to terminate the program
4) BONUS
   a) Place the USER_LOOP into a Function called “Login”
   b) “Encapsulate” (hide) the Password inside of the function
   c) Provide the number of PASSWORD_LOOP attempts as an input parameter
   d) 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: