# **CPS 474/575 - Secure Application Development, Spring 2020**

Dr. Phu Phung

Department of Computer Science,  
University of Dayton

Programming Assignment - A Secure and Robust Chat System in Golang and Node.js

Announcement: Lecture 10 (2/13/2020)

Instruction released: 11:55 pm 2/18/2020

Deadline: TBD

## Introduction

This programming assignment is an extension of Lab 3 - Concurrent Programming in Java, Golang, and Node.js. In this assignment, you will develop a simple yet secure, and robust concurrent chat system with the following requirements:

* Extend the BroadcastEchoServer.go (From Lab 3) program to become a Chat Server:
  + A user must be authenticated with username/password (can be hard-coded)
  + Serve the clients with the following use cases:
    - Return the list of logged-in users
    - Send a message to all users (public chat)
    - Send a message to a specific user (private chat)
* Extend the telnet.js (From Lab 3) program to become a Chat Client:
  + Ask username/password from the user to send to the server for authentication
  + Print a simple menu for the user to
    - Execute the functions provided by the server or exit the chat program.

## Requirements and Grade Distribution

Design (10p) and implement your applications (client and server - 90p) with the minimal requirements/scenarios below:

* *(5p) A user can log in with the correct username/password* 
  + *(5p) Password must be masked*
* *(5p) Login failed due to incorrect username/password or invalid data*
* *(5p) Get a list of logged-in users*
* *(50p) Public messages only sent to logged-in users*
* *(5p) Private messages sent correctly to the receiver*
* *A user can log in from two different terminals*
  + *(5p) can still receive private messages*
  + *(5p) If one terminal is closed, the user is still in the logged-in user list*
* *(5p) A user logout to the system and the user list is updated.*

## Report

Write your report (recommended written in Google Docs) and export it to **PDF** and submit on Isidore by the deadline. The report must contain your name, ID#, email, course number and name, instructor, and the assignment title with the following sections:

1. Introduction

*Overview of your assignment development and your achievement. Remember to push all your code to your private repository and* ***provide the link in this section (-5% for missing the link, and -5% if the code is not in your repo)****.*

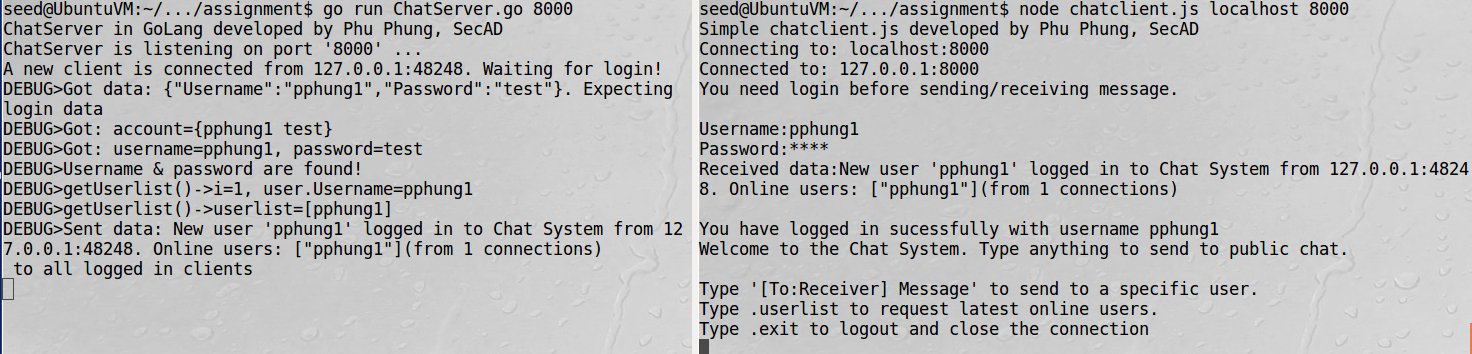
1. Design

*Present here how you design the protocol for the client and the server to understand each other. Prefer to have an interaction diagram between the client and the server. You might need to present the data format for different types of messages.*

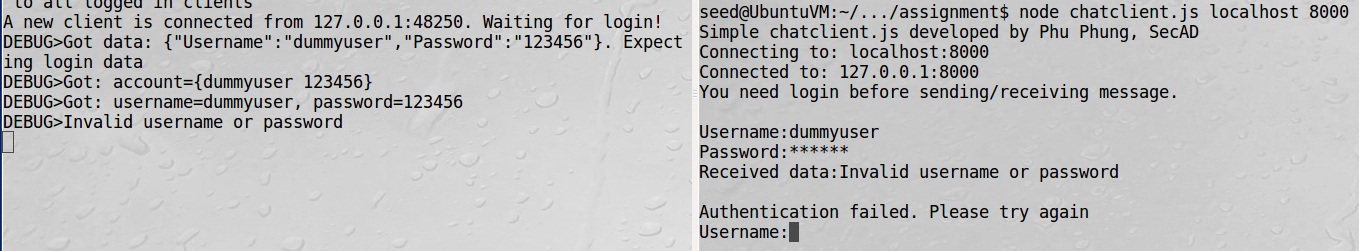
*You also need to discuss how to avoid the data race and other security issues in the programs*

1. Test cases and Demo

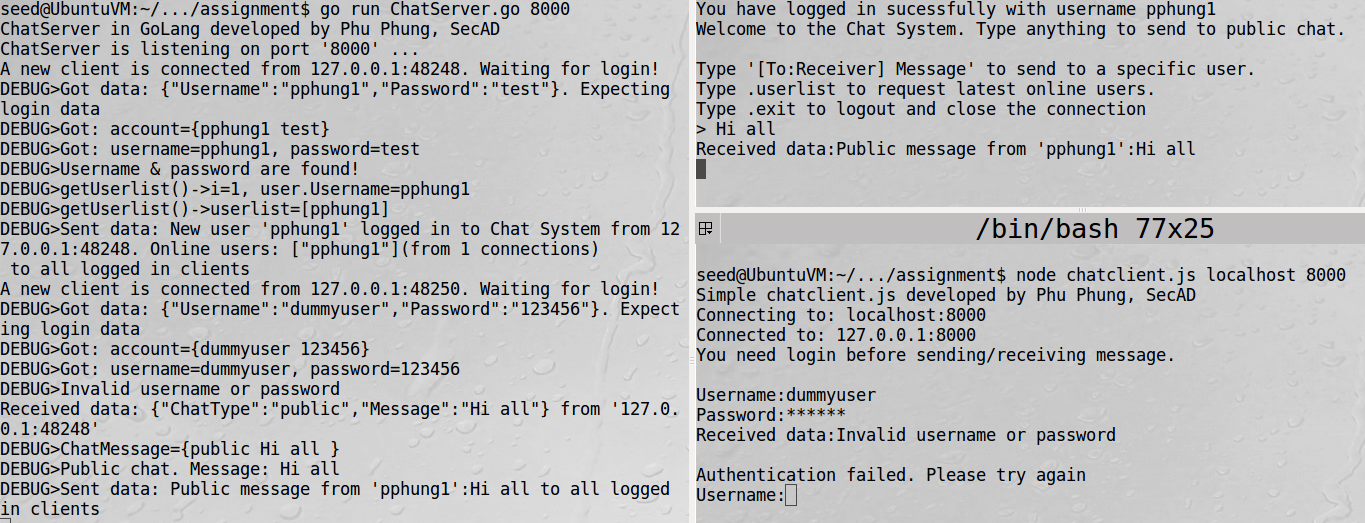
*Test cases should be demonstrated with screenshots and a brief description. You need to organize your test cases according to use-case scenarios in the requirements.   
Some examples (not comprehensive) are presented below.*

**

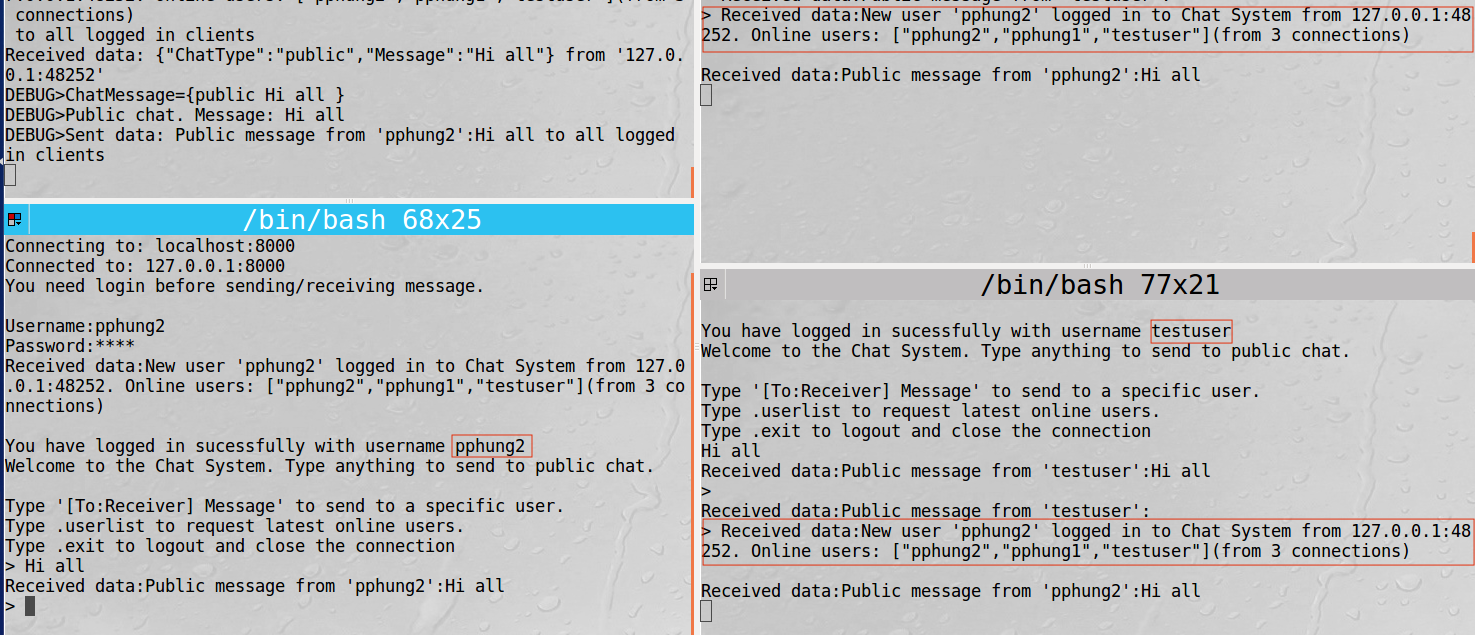
*Test case 1 - A user has been login successfully at the server-side (left), and at the client-side (right). The client program is able to get data from the user and send to the server to be authenticated. Note that the typed password is masked with \*.*

**

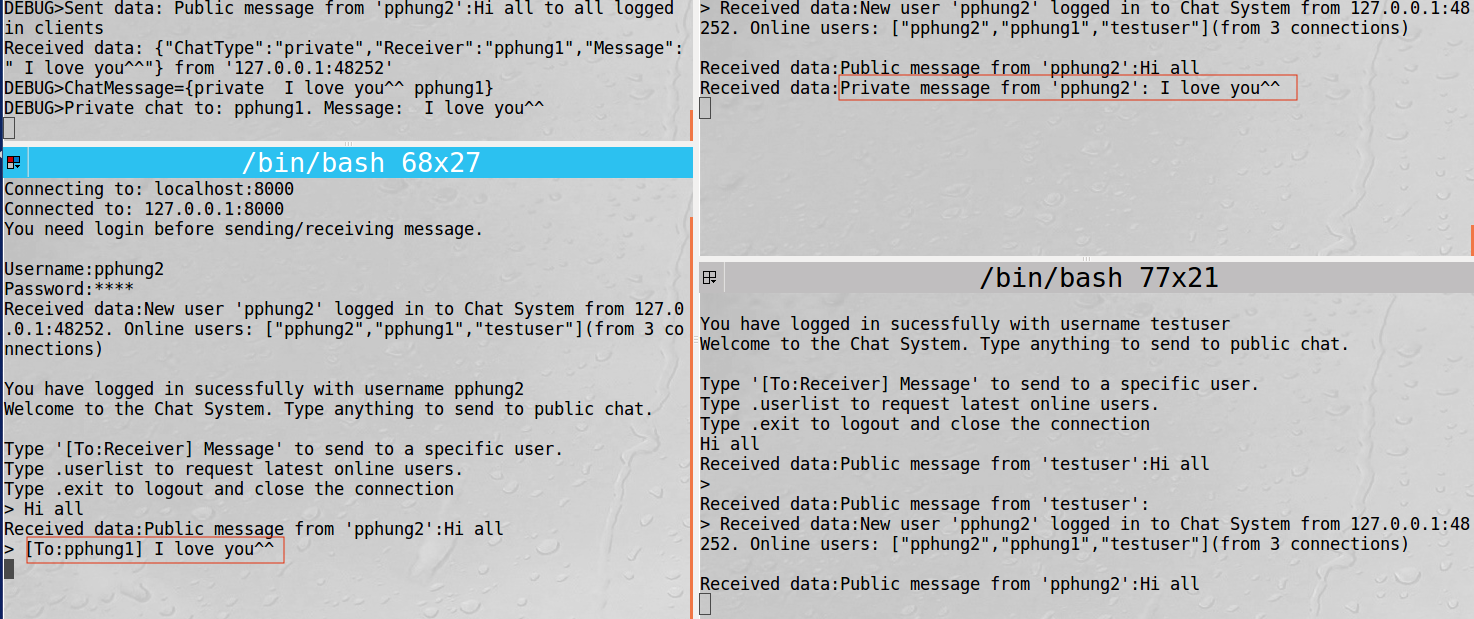
*Test case 2 - A user cannot log in due to an invalid username or password. Left: at the server-side, right: and at the client-side.*

**

*Test case 3 - Logged-in users can receive public chat messages while unlogged-in users cannot.*

**

*Test case 4 & 5. All logged-in users receive a notification with the updated user list when a new user is logged in successfully. Public chat messages are sent to all logged-in users.*

**

*Test case 6. Private messages can only be received by the individual receiver.*

### Appendix

*Include the entire source code in TEXT, not screenshot, as an appendix in your report. Please format and indent your code properly. You should use the “Code Blocks” add-on in Google Docs for this purpose.*

### Submission

You need to submit 3 following files separately on Isidore by the deadline:

1. secad-assignment-YourUDID.pdf: Your report in PDF
2. ChatServer.go: the source code of your server program in Go
3. chatclient.js: the source code of your client program in Node.js