Java 3

Project AT3

Question 1

Project Proposal

Student Name: Alan Pedersen

Student ID: P225139

Date: 21/05/2020

Table of Contents

[Table of Figures i](#_Toc40966668)

[Question 1 1](#_Toc40966669)

[Introduction 1](#_Toc40966670)

[Data structures 1](#_Toc40966671)

[Use of hashing 1](#_Toc40966672)

[Proposed Sorting Algorithm 2](#_Toc40966673)

[Proposed Search Algorithm 2](#_Toc40966674)

[3rd party libraries 2](#_Toc40966675)

[Proposed GUI 2](#_Toc40966676)

[Proposed Source Control 4](#_Toc40966677)

[Coding Standards 4](#_Toc40966678)

[Testing 4](#_Toc40966679)

# Table of Figures

[Figure 1: Client class diagram 1](#_Toc40966680)

[Figure 2: server application design 3](#_Toc40966681)

[Figure 3: client application design 4](#_Toc40966682)

# Question 1

You are expected to produce a document including UML.

## Introduction

The proposed project is a message exchange system using a central server and client applications. Access to the system will be controlled using user accounts and passwords. Users must be logged into the system to send messages. A database will be used for long term storage of user details and activity logging.

The following features will be included to meet the requirements of the project.

## Data structures

The server application will manage the user database.

The details for each user will be stored in an object. The following class diagram shows the proposed structure for the Client class:

|  |
| --- |
| Client Class |
| Attributes:  - name : String  - passwordSalt : byte[]  - passwordHash : String  - activeFlag : boolean  - ServerSocket : Socket  - readStream : DataInputStream  - writeStream : DataOutputStream  - left : Client  - right : Client |
| Methods:  + Getter & Setter methods  + string ToString() |

Figure 1: Client class diagram

The objects will be stored in a self-balancing binary tree structure.

The server application will use MySQL for off-line storage of the user details. A single table storing client name, Salt value and hashed password will be used.

A database table will be used to record system activity such as login requests, log out events and disconnections.

## Use of hashing

Hashing will be used to encrypt passwords for storage. Each user will be allocated a password when their user account is created. The system will generate a random number to be used as a salting value. The salt value will be combined with the password and the combined value hashed. Only the hashed salted password will be stored in the system.

Submitted passwords will be validated by applying the password creation algorithm to the password being submitted and comparing the result with the stored version.

## Proposed Sorting Algorithm

The use of a self-balancing binary tree structure will automatically sort the client list.

The collection of Client objects will be kept in a self-balancing binary tree using the client name as the key value. The structure will be comprised of a series of nodes, each node holding the data for one client as well as two links to child nodes. The nodes will be ordered alphabetically by name, names before the current name will be stored in the left node, names after the current name will be stored in the right node.

A sorted list of clients can be generated by traversing the tree structure.

## Proposed Search Algorithm

Client records will be searched for by traversing the tree structure.

The search will start at the root node of the tree. if the value being searched for matches the node that node will be returned. If the value being searched for is before the node value the left tree will be searched, if the search value is after the node vale the right tree will be searched. The process is recursive searching the tree structure until either a match is found or an unoccupied branch is encountered in which case the search will return no match.

## 3rd party libraries

The application will access the MySQL database.

## Proposed GUI

The system will use two applications, a server application and a client application.

The proposed design for the server application is presented below:

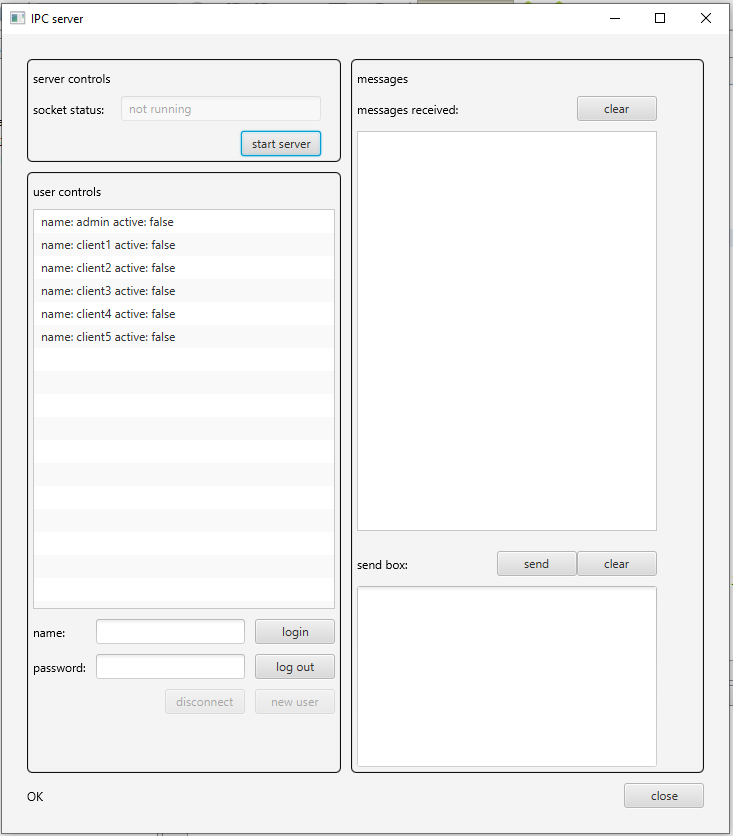


Figure 2: server application design

The proposed design for the client application is presented below:

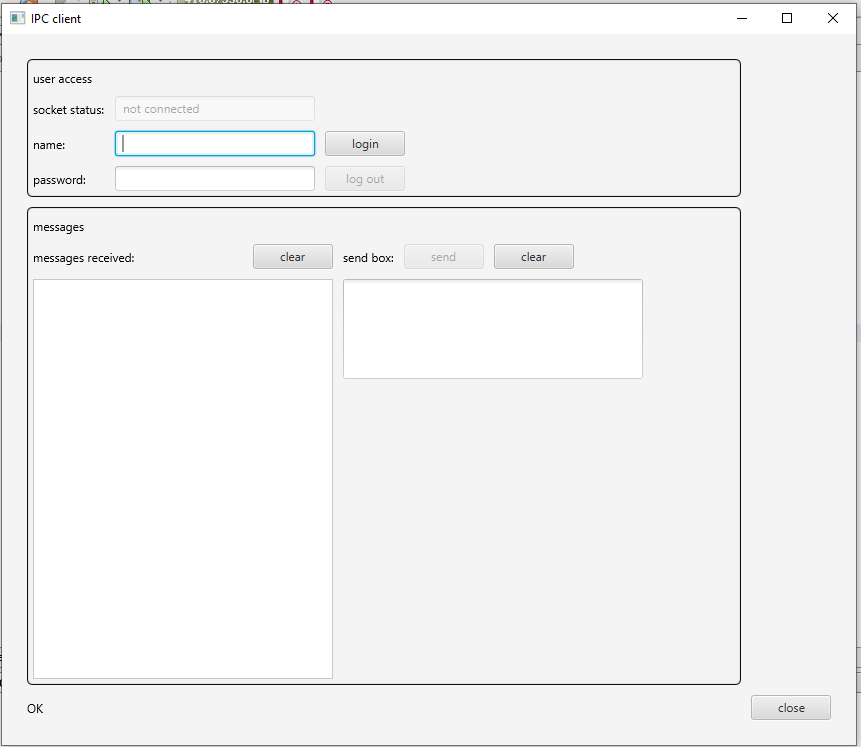


Figure 3: client application design

## Proposed Source Control

GitHub was selected to manage the source code. Git is a distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows. (1)

## Coding Standards

Coding standards will follow Java coding conventions as recommended by Sun Microsystems.

## Testing

All aspects of the system will be tested.

* All user input will be validated before use. The validation routines will be tested by submitting invalid values.
* The user creation routines will be tested by submitting incomplete data and invalid values.
* The database storage will be tested by submitting and retrieving values
* The binary tree will be tested by adding and removing nodes and checking the structure after the process
* The search routine will be tested by searching using key values that are known to exist and key values that are known not to exist.
* The user login process will be tested with correct and incorrect values
* The monitoring functions will be observed for correct functioning during client login testing
* Messaging functions will be tested by sending messages and observing the results