Computer Networks - Homework 2

Sîrcu Cătălin-Alexandru

Facultatea de Informatica

1 Introduction

The idea behind the ”Password Manager” is to develop a client/server application that offers the functionality of a password manager. The application will support multiple user accounts, each user being able to login using a master password. Each user will be able to organise his saved passwords in different categories, and for each password he will have the possibility to insert data regarding the passwords, such as: title, username, URL, notes, category etc..

1.1 Motivation

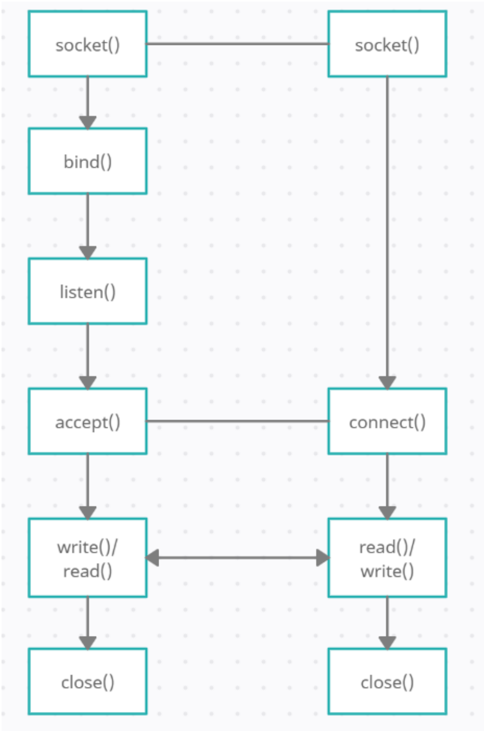
Judging by the fact that we live in an era that we need as many accounts on certain things like: shops, banking, e-mail addresses, phones, bill payments and so on...
I consider that such an app as a password manager will be useful for the user for creating passwords on certain categories and also in case of forgetting a password, the manager will definitely help you remember it.

2 Technologies used

2.1 What is TCP?

TCP is a protocol that is connection-oriented, meaning that once a connection has been established, the data can be transmitted in both directions and also it has built-in systems to check for errors and to guarantee data will be delivered in the order it was sent, making it the perfect protocol for transferring information.

2.2 How does TCP work?



The stages of a server:
-socket(): creates an end-point connection
-bind(): attaches a local address to a socket.
-listen(): gives the permission to a socket to accept connections.
-accept(): blocks the caller until a connection request arrives.
-close(): ends the connection.
The stages of a client:
-socket(): creates an end-point connection(same process as the server).
-connect(): it attempt to establish the connection.
-close(): ends the connection.

2.3 Why TCP is relevant to this project?

In the "Password Manager" the TCP protocol will be needed because the communication between server and client is done by using text, such as login, quit, create, insert password, etc and is reliable as it guarantees the delivery of data to the destination router.
Even though UDP protocol is faster comparing to the TCP, because the UDP

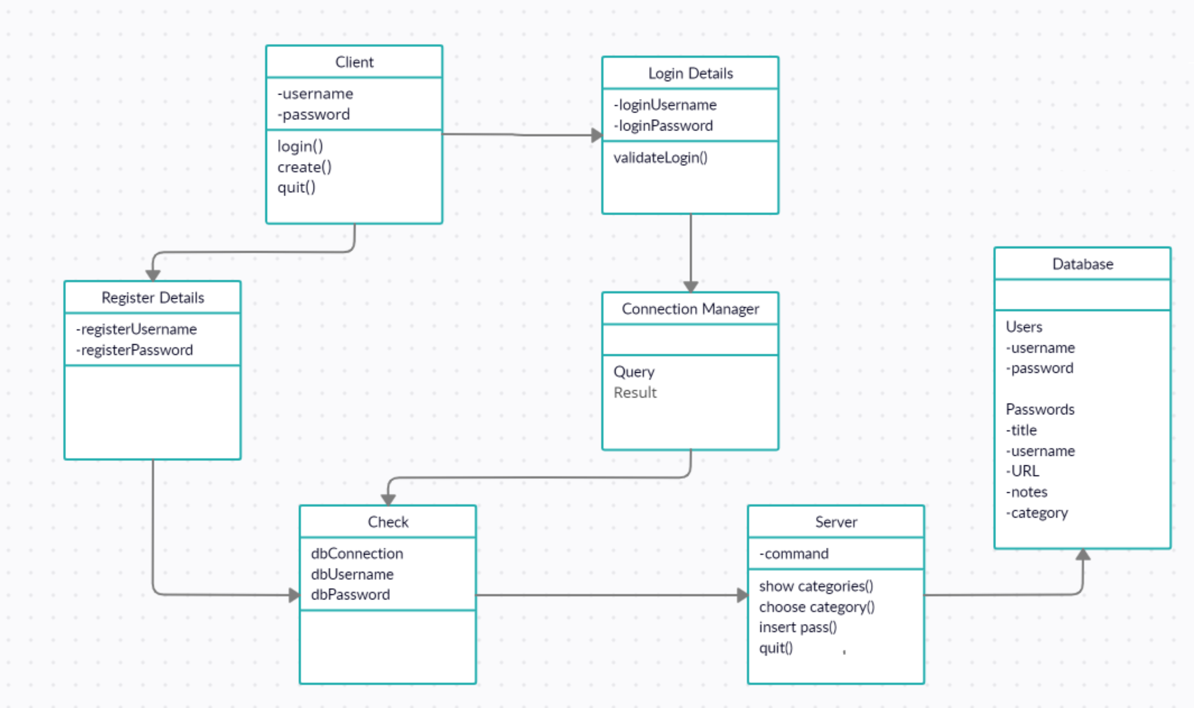
doesn't use the acknowledgement control and has a basic error checking mechanism, thus when it comes to passwords and accounts we can't trade safety for speed.

3 Application architecture

3.1 Concepts involved

For this project I've chosen to work with a database being much simpler comparing the work with I/O files. The technology that will help working with a database is SQLITE which has some interesting properties:
-better performance, reading and writing operations are faster when working with it than with I/O files.
-no installation is required, you just download the libraries and you are free to go.
-reliable, being less bug prone rather than custom written file I/O.

3.2 Architectural Diagram



4 Implementation details

The TCP concurrent server is going to wait for connections from the clients. The client will request the server to connect to it, and client's request can be either accepted or refused. In case his request is going to be refused, the connection between the server and the client will be closed. If the client is accepted, the server will handle it's clients using a select() that receives a descriptors set and it removes those that cannot be read, meaning that it selects only the active ones.
After the connection has been successfully, the unlogged user will have the next commands to use:
-login: where the user will have to type his username and password of his account and the database will check if that user exists or not such that he will receive a message and based on that he will be able to do the things that a logged user can do.
-create: if the user connected doesn't have an account he will be able to create one, and after that to login into the system.
-quit: the user will disconnect from the server. When the client has logged in he will be able to access this commands:
-show categories: when typing this command the user will be able to see the categories for passwords he has in the database.
-choose category: this command will allow the user to type the category of passwords in order to see his passwords from the given category.
-insert pass: this command will allow to enter in his password category different attributes such as: title, URL, username, notes and the category.
-quit: the logged user will be able to disconnect from the server.

5 Conclusions

5.1 How the solution could be improved

-adding an admin role.
-adding other commands that could benefit the user such as "modify" which can help him to modify the passwords and the categories that he had created and a "delete" command which could help him delete a password or a password category.
-using a password that is securely encrypted when logging into the system to not let the admin see the password of the user.
-graphic interface for a better interaction with the user.
-adding various commands for the admin role such as "erase" that could let him delete a certain user from the system.

5.2 How is it going so far

So far in this project I got comfortable with working with the database.

6 Bibliography

https://www.lifesize.com/en/blog/tcp-vs-udp/
https://profs.info.uaic.ro/~computernetworks/cursullaboratorul.php
https://www.tutorialspoint.com/sqlite/sqlite_c_cpp.htm
https://www.javatpoint.com/sqlite-advantages-and-disadvantages
https://www.sqlite.org/cintro.html
https://www.geeksforgeeks.org/differences-between-tcp-and-udp/
https://www.geeksforgeeks.org/socket-programming-cc/