htl-tk-chat 0.0.1

Generated by Doxygen 1.9.1

1 htl-tk-chat	1
1.1 Build documentation	 1
1.2 Project description	 1
1.2.1 Planned features:	 1
1.3 What we use, what we do	 2
1.4 Rules	 2
1.5 Milestone 1	 2
2 Namespace Index	3
2.1 Namespace List	 3
3 Hierarchical Index	5
3.1 Class Hierarchy	 5
4 Class Index	7
4.1 Class List	 7
5 Namespace Documentation	9
5.1 chat_client Namespace Reference	9
5.1.1 Detailed Description	9
5.1.2 Function Documentation	9
5.1.2.1 main_read_loop()	 10
5.1.2.2 main_write_loop()	 10
5.2 chat_server Namespace Reference	 10
5.2.1 Detailed Description	 10
5.2.2 Function Documentation	 11
5.2.2.1 text_message()	 11
5.3 client Namespace Reference	 11
5.3.1 Detailed Description	 11
5.3.2 Function Documentation	 11
5.3.2.1 connect_server()	 12
5.3.2.2 disconnect_server()	 12
5.3.2.3 load_ui_file()	 12
5.3.2.4 send_msg()	 12
5.4 client_functions Namespace Reference	 13
5.4.1 Detailed Description	 13
5.4.2 Function Documentation	
5.4.2.1 authenticate()	 13
5.4.2.2 close_connection()	13
5.4.2.3 get_message()	14
5.4.2.4 text_message()	14
5.5 database_functions Namespace Reference	14
5.5.1 Detailed Description	
5.6 settingsstub Namespace Reference	
Ø	

5.6.1 Detailed Description	15
5.6.2 Function Documentation	15
5.6.2.1 connect_server()	15
5.6.2.2 disconnect_server()	15
5.6.2.3 init_settings_window()	15
5.7 userstub Namespace Reference	16
5.7.1 Detailed Description	16
5.7.2 Function Documentation	16
5.7.2.1 test_combo_box()	16
5.7.2.2 test_connection()	16
5.7.2.3 test_user_table()	17
5.7.3 Variable Documentation	17
5.7.3.1 test_users	17
6 Class Documentation	19
6.1 chat_client.Client Class Reference	19
6.1.1 Detailed Description	19
6.2 database_functions.Database Class Reference	19
6.2.1 Detailed Description	20
6.2.2 Member Function Documentation	20
6.2.2.1 get_message()	20
6.2.2.2 init_db()	20
6.2.2.3 write_message()	20
6.3 chat_server.ThreadedTCPRequestHandler Class Reference	21
6.3.1 Detailed Description	21
6.3.2 Member Function Documentation	21
6.3.2.1 handle()	21
6.4 chat_server.ThreadedTCPServer Class Reference	21
6.4.1 Detailed Description	22
Index 2	23

htl-tk-chat

Watch the development progress at https://gitea.escpe.net/cc69222/htl-tk-chat

1.1 Build documentation

To build the documentation run: make doc

After this you can open the documentation.pdf.

For lazy people or Windows folks, a precompiled version of the documentation can be found here.

1.2 Project description

A Server and client for a self-written chat application. The client interface uses Qt 6, the server uses sockets and msgpack.

1.2.1 Planned features:

- Selecting users to send messages to ("Public", "Private")
- Notify users when new messages are available (for them)
- · database to store message history
- · attachments (files)
- · embedded pictures in chat
- authentification / identification of users via username
- encryption ? (GPG) or SSL -> Let's Encrypt for server

2 htl-tk-chat

1.3 What we use, what we do

- · sockets, tcp streams
- · JSON for message content and metadata
- · msgpack for message content and metadata, because it is faster and encodes in binary instead of strings.
- Server, self written, hosted at home (port forwarding)
- · Database server-side for message-history, users, authentification
- For the Database SQLite, cause it is a lightweight disk-based database and doesn't require a separate server process. Also the sqlite3 python module is in The Python Standard Library.
- We don't care about networking failures, we let tcp handle that.
- Interface in tkinter or Qt 6 (via PyQt6)
- · Doxygen for documentation

1.4 Rules

- · Commit messages should make sense
- If you're planning to implement a new feature, open an issue with the correct label
- Working on the development branch and after the one week sprint make a pull request, to merge it back to the master.
- · Write sensible reviews and document your Code

1.5 Milestone 1

- · Server is running and accessible from the internet
- Clients send messages to Server, the Server broadcasts the message to all Clients.

Namespace Index

2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

chat_client	9
chat_server	10
client	11
client_functions	13
database_functions	14
settingsstub	14
userstub	16

Namespace Index

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

socketserver.BaseRequestHandler	
chat_server.ThreadedTCPRequestHandler	21
chat_client.Client	19
database_functions.Database	19
socketserver.TCPServer	
chat_server.ThreadedTCPServer	21
socketserver.ThreadingMixIn	
chat_server.ThreadedTCPServer	21

6 Hierarchical Index

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

chat_client.Client	19
database_functions.Database	19
chat_server.ThreadedTCPRequestHandler	21
chat server ThreadedTCPServer	21

8 Class Index

Namespace Documentation

5.1 chat_client Namespace Reference

Classes

class Client

Functions

- def main_read_loop (sock)
- def main_write_loop (sock, user)

Variables

- **host** = input("Server address: ")
- **port** = int(input("Server port: "))
- bool **shutdown** = False
- my_client = Client(host, port)
- **user** = input("username: ")
- read_thread = threading.Thread(target=main_read_loop, args=(my_client.sock,))
- write_thread = threading.Thread(target=main_write_loop, args=(my_client.sock, user,))

5.1.1 Detailed Description

CLI bassed Chat Client

5.1.2 Function Documentation

5.1.2.1 main_read_loop()

Reads the socket in a loop

Definition at line 18 of file chat_client.py.

5.1.2.2 main_write_loop()

Sends messages in a loop

Definition at line 32 of file chat_client.py.

5.2 chat_server Namespace Reference

Classes

- class ThreadedTCPRequestHandler
- class ThreadedTCPServer

Functions

- def unpack_message (message)
- def text_message (sock, str text, str author, str recipient="all")

Variables

- HOST
- PORT
- **server** = ThreadedTCPServer((HOST, PORT), ThreadedTCPRequestHandler)
- **server_thread** = threading.Thread(target=server.serve_forever)

5.2.1 Detailed Description

The Chat Server

This is the Chat server, it is the backbone of the communication between the clients.

5.2.2 Function Documentation

5.2.2.1 text_message()

Definition at line 56 of file chat_server.py.

5.3 client Namespace Reference

Functions

```
def connect_server ()def disconnect_server ()def send_msg ()def load_ui_file (filename)
```

Variables

```
    app = QApplication(sys.argv)
    def window = load_ui_file("mainwindow.ui")
    def settings = load_ui_file("settingswindow.ui")
```

5.3.1 Detailed Description

```
The Chat Client
This is the chat client with Qt6 frontend.
```

5.3.2 Function Documentation

5.3.2.1 connect_server()

```
def client.connect_server ( )

Connect to a chat server

Connect to a chat server via the settings window and the information provided through the same window. This function is called when the user presses the 'Connect' button in the settings window. The function is a stub.
```

Definition at line 25 of file client.py.

5.3.2.2 disconnect_server()

```
def client.disconnect_server ( )
```

Disconnect to a chat server

Disconnect from a chat server via the settings window. This function is called when the user presses the 'Cancel' button for server connection in the settings window. The function is a stub.

Definition at line 56 of file client.py.

5.3.2.3 load_ui_file()

UI File Loader

Path independent loader function for QTCreators .ui files

Definition at line 88 of file client.py.

5.3.2.4 send_msg()

```
def client.send_msg ( )
```

Send message function

This function is called when the user presses Enter, to send the message. The function isn't fully implemented at the moment.

Definition at line 71 of file client.py.

5.4 client functions Namespace Reference

Functions

- def authenticate (sock, str username)
- def text_message (sock, str text, str author, str recipient="all")
- def get_message (sock)
- def close connection (sock)

5.4.1 Detailed Description

```
client_functions
Adds functions for the client to use
```

This module serves as some kind of API for the chat client. The client should never send messages to the server directly, but rather through this api. This makes it easier to programm the GUI later.

5.4.2 Function Documentation

5.4.2.1 authenticate()

Authentication function

Takes a socket and a username as arguments and sends it to the server as an auth type message. The auth message has the values user for the username identification and time, to know when the user logs in. A value for the ip address is not necessary, since the socket already knows this.

Definition at line 13 of file client_functions.py.

5.4.2.2 close_connection()

Close function

Closes the server connection and notifies the server about it.

Definition at line 59 of file client_functions.py.

5.4.2.3 get_message()

```
\begin{array}{c} \texttt{def client\_functions.get\_message (} \\ & \textit{sock )} \end{array}
```

Get Message function

Takes a socket argument and returns the message in the pipe.

Definition at line 47 of file client_functions.py.

5.4.2.4 text_message()

Text Message function

Takes a socket, text, author and recipient as arguments and sends it to the server.

Definition at line 31 of file client_functions.py.

5.5 database_functions Namespace Reference

Classes

• class Database

5.5.1 Detailed Description

```
database functions

Adds database functions for the server to use

This library adds easy to use functions to store message data in a database for (somewhat) persistent data storage.
```

5.6 settingsstub Namespace Reference

Functions

- def connect_server (settings)
- def disconnect_server ()
- def init_settings_window (settings)

5.6.1 Detailed Description

Settings stub Stub for settings window since server is not yet implemented.

5.6.2 Function Documentation

5.6.2.1 connect_server()

Connect to a chat server

Connect to a chat server via the settings window and the information provided through the same window. This function is called when the user presses the 'Connect' button in the settings window. The function is a stub.

Definition at line 14 of file settingsstub.py.

5.6.2.2 disconnect_server()

```
def settingsstub.disconnect_server ( )
```

Disconnect from a chat server

Disconnect from a chat server via the settings window. This function is called when the user presses the 'Cancel' button for server connection in the settings window. The function is a stub.

Definition at line 48 of file settingsstub.py.

5.6.2.3 init_settings_window()

Initialize settings-window
Initialize functionality of settings-window elements

Definition at line 62 of file settingsstub.py.

5.7 userstub Namespace Reference

Functions

- def test_connection (ip)
- def test_user_table (window)
- def test_combo_box (window)

Variables

· list test users

5.7.1 Detailed Description

This is just a stub for users since other functionalities are not implemented at the moment.

5.7.2 Function Documentation

5.7.2.1 test_combo_box()

Definition at line 51 of file userstub.py.

5.7.2.2 test_connection()

```
\label{eq:connection} \begin{array}{c} \text{def userstub.test\_connection (} \\ ip \end{array}) Test the Connection \text{pings the ip to get the connction status.} Not implemented at the moment.}
```

Definition at line 14 of file userstub.py.

5.7.2.3 test_user_table()

```
def userstub.test_user_table (
             window )
Adds test users to the table
This function adds test users to the QTableWidget
on the left side of the window.
```

Definition at line 23 of file userstub.py.

5.7.3 Variable Documentation

5.7.3.1 test_users

list userstub.test_users

Definition at line 8 of file userstub.py.

Class Documentation

6.1 chat_client.Client Class Reference

Public Member Functions

• def __init__ (self, str ip, int port)

Public Attributes

sock

6.1.1 Detailed Description

Intializes the Socket which is available as self.sock

Definition at line 9 of file chat_client.py.

The documentation for this class was generated from the following file:

client/chat_client.py

6.2 database_functions.Database Class Reference

Public Member Functions

- def init_db (self)
- def write_message (self, message, author, receipient, date)
- def get_message (self)

Private Member Functions

def <u>__dict_factory</u> (self, cursor, row)

20 Class Documentation

6.2.1 Detailed Description

This class serves as a de facto database providing functions for simple access to the database.

Definition at line 14 of file database_functions.py.

6.2.2 Member Function Documentation

Checks if all the needed tables exist or creates them if not.

Definition at line 28 of file database_functions.py.

6.2.2.3 write_message()

Writes the current message to the message history

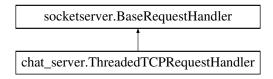
Definition at line 50 of file database functions.py.

The documentation for this class was generated from the following file:

• server/database_functions.py

6.3 chat_server.ThreadedTCPRequestHandler Class Reference

Inheritance diagram for chat_server.ThreadedTCPRequestHandler:



Public Member Functions

• def handle (self)

6.3.1 Detailed Description

Handles the request in a thread

Definition at line 14 of file chat_server.py.

6.3.2 Member Function Documentation

6.3.2.1 handle()

```
\label{lem:def:chat_server.ThreadedTCPRequestHandler.handle ( \\ \textit{self} \ ) function gets a request offer from the socket and prints the message and the Thread it is in. It handles the authentication and the messages. At the moment it just broadcasts the messages
```

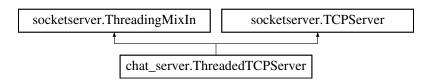
Definition at line 18 of file chat_server.py.

The documentation for this class was generated from the following file:

server/chat_server.py

6.4 chat_server.ThreadedTCPServer Class Reference

Inheritance diagram for chat_server.ThreadedTCPServer:



22 Class Documentation

Public Member Functions

• def __init__ (self, *args)

Public Attributes

connected_clients

6.4.1 Detailed Description

TCPServer with threading support. Spawns a new Thread for every response.

Definition at line 42 of file chat_server.py.

The documentation for this class was generated from the following file:

server/chat_server.py

Index

```
main read loop
authenticate
    client_functions, 13
                                                             chat_client, 9
                                                        main_write_loop
chat_client, 9
                                                             chat_client, 10
     main_read_loop, 9
     main_write_loop, 10
                                                        send_msg
chat_client.Client, 19
                                                             client, 12
chat_server, 10
                                                        settingsstub, 14
    text_message, 11
                                                             connect_server, 15
chat\_server. Threaded TCPR equest Handler, {\color{red}21}
                                                             disconnect_server, 15
     handle, 21
                                                             init_settings_window, 15
chat_server.ThreadedTCPServer, 21
                                                        test combo box
client, 11
     connect_server, 11
                                                             userstub, 16
                                                        test connection
     disconnect_server, 12
                                                             userstub, 16
     load ui file, 12
                                                        test_user_table
     send msg, 12
                                                             userstub, 16
client_functions, 13
     authenticate, 13
                                                        test users
                                                             userstub, 17
     close_connection, 13
                                                        text_message
     get_message, 13
                                                             chat_server, 11
    text_message, 14
                                                             client_functions, 14
close connection
    client functions, 13
                                                        userstub, 16
connect server
                                                             test combo box, 16
    client, 11
                                                             test_connection, 16
     settingsstub, 15
                                                             test_user_table, 16
                                                             test_users, 17
database_functions, 14
database_functions.Database, 19
                                                        write_message
     get_message, 20
                                                              database_functions.Database, 20
     init_db, 20
     write_message, 20
disconnect_server
     client, 12
    settingsstub, 15
get_message
     client_functions, 13
     database_functions.Database, 20
handle
     chat_server.ThreadedTCPRequestHandler, 21
init db
     database_functions.Database, 20
init_settings_window
     settingsstub, 15
load_ui_file
     client, 12
```