

TCPProtocolHelper

typedef QByteArray (TCPProtocolHelper::*funcPtr)()

- _ptr : funcPtr

typedef void (TCPProtocolHelper::*handlePtr)()

-_handlePtr : handlePtr

typedef std::map<ProtocolType, funcPtr>map_funcs

- functions : map_funcs

typedef std::map<protocolClient, handlePtr>handle_funcs

- handle Functions : handle_funcs

- _clientID : int

+ createRequest(ProtocolType type) : QByteArray

+ handleRequest(qint8 type) : void

+ parseLoginSuccess(QByteArray &array) : void

+ getClientId() : int

+ <<signal>>handshakeSuccess() : QByteArray

+ <<signal>>loginSuccess() : QByteArray

+ <<signal>>loginError() : void

+ <<signal>>addContactSuccess(): void

- createHandshake() : QByteArray

- createCallRequest() : QByteArray

- acceptCallRequest() : QByteArray

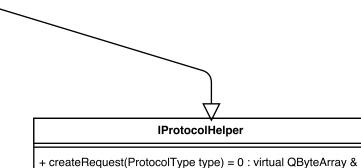
- addContactRequest() : QByteArray

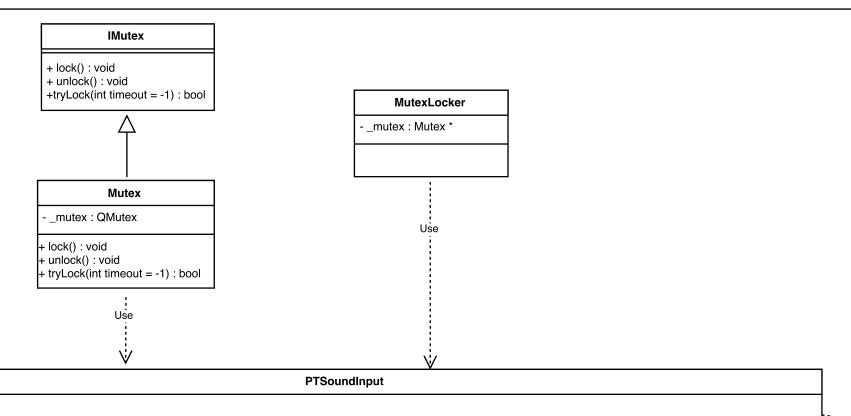
- handleHandshake() : void

- handleLogin() : void

- errorLogin() : void

- handleContactSuccess(): void



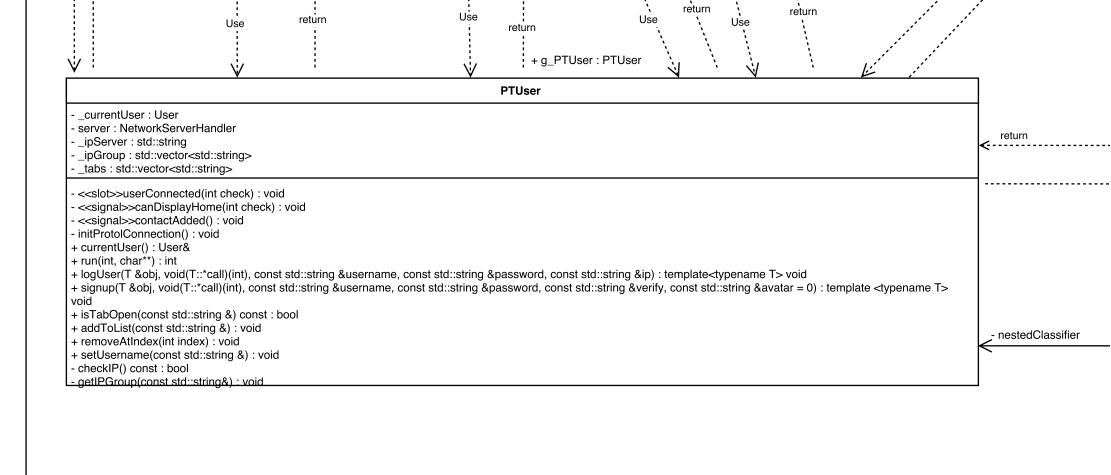


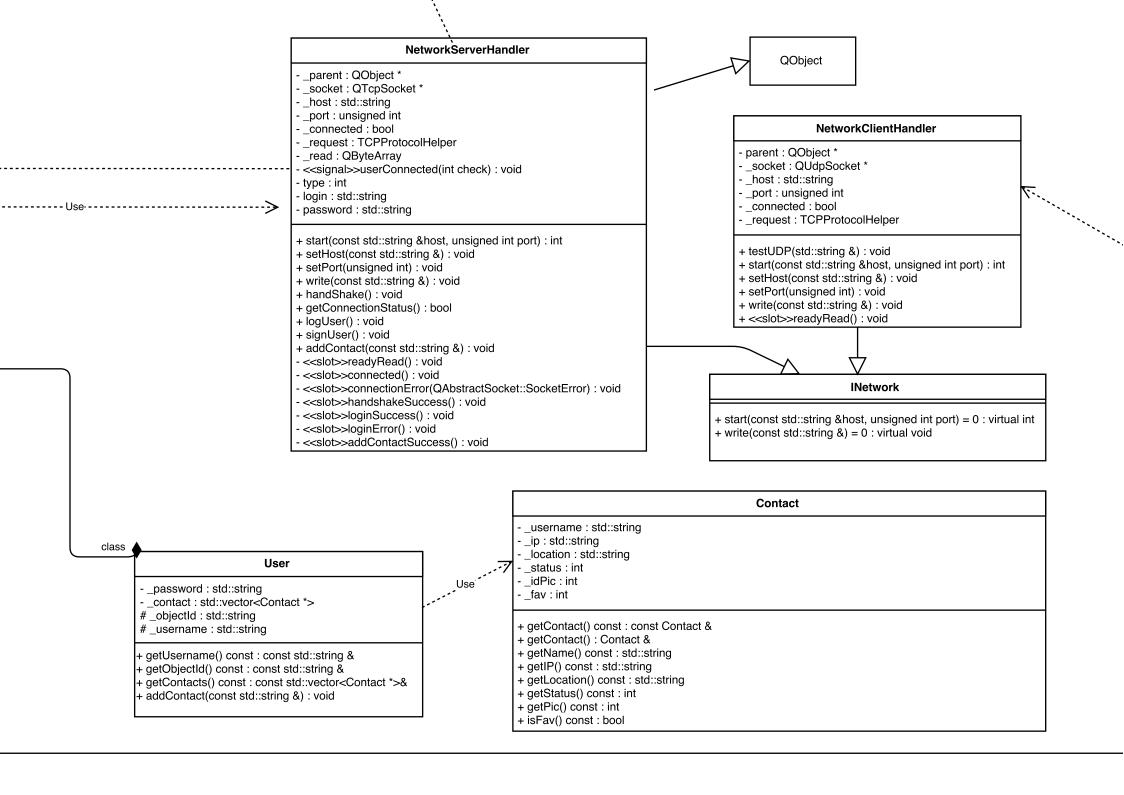
Use

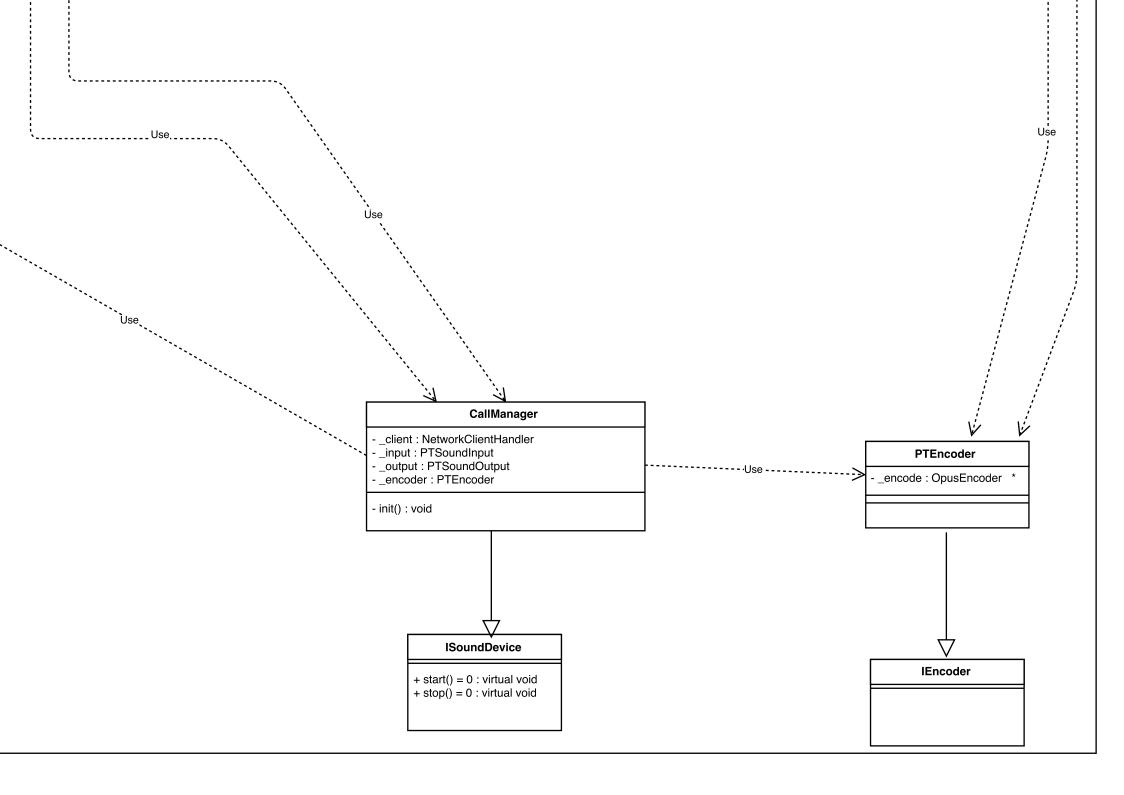
- stream : PaStream *
- _params : PaStreamParameters
- initInput(): void
- recordCallback(const void *input, void *output, unsigned long frameCount, const PaStreamCallbackTimeInfo* timeInfo, PaStreamCallbackFlags statusFlags,void *userData): static int
- + start() : void
- + stop(): void

PTSoundOutput

- _stream : PaStream *
- _params : PaStreamParameters
- initOutput(): void
- playCallback(const void *input, void *output, unsigned long frameCount, const PaStreamCallbackTimeInfo* timeInfo, PaStreamCallbackFlags statusFlags,void *userData): static int
- + start() : void
- + stop() : void







Account

Use

+ state : typedef enum

- _socket : boost::shared_ptr<TCPConnection>

- login : std::string - _passwd : std::string - state : Account::state

- profilePicture : short - location : std::string

nicknames : std::vector<std::pair<Account*:std::string>>

- _contactsList : std::vector<Account*> - favoritesList : std::vector<Account*>

- _id : const std::string

+ setLogin(std::string &login) : void

+ setState(Account::state state) : void

+ setLocation(std::string location) : void

+ setNickname(std::string &id, std::string &nickname) : void

+ setProfilePicture(short) : void

+ getData() const : std::vector<std::string>

+ setSocket(boost::shared_ptr<TCPConnection> socket) : void

+ getLogin(): std::string & + getLocation(): std::string &

+ getPasswd() : std::string &

+ getSocket(): boost::shared ptr<TCPConnection> &

+ getContactList() : std::vector<Account*> &

+ getNicknames(): std::vector<std::pair<std::string,std::string>> &

+ getContactByID(std::string &) : Account *

+ getState(): Account::state

+ isAlreadyAContactOf(Account *) : bool

+ getProfilePictureID(): short

+ getNicknameIfExisting(Account &account) : std::string &

+ addContact(Account *) : bool

+ getID(): std::string

+ removeContact(std::string &ID) : bool

+ addToFavorite(Account *): bool

+ removeFromFavorite(std::string &) : bool

+ getFormatedContactList(): std::vector<std::string> &

+ generateRandomID(size t length) : void

+ operator==(Account &): bool + setID(const std::string &id) : void

Server

- _allAccounts : std::vector<Account*>

net : Network *

- _date : boost::posix_time::ptime

+ doesUsernameExist(std::string &) : bool

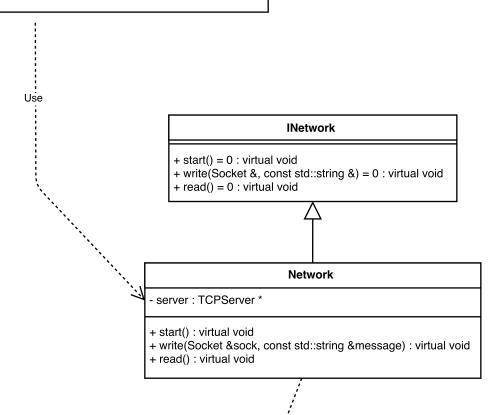
+ isPasswdCorrectForAccount(std::string &, std::string &): bool

+ addAccount(std::string &, std::string &, short) : void

+ getAccountByID(std::string &) : Account *

+ getAccountByUsername(std::string &) : Account *

+ getNetwork() const : Network *



Sender

- + send(Response *response) : static void
- + specialSending(Response *response) : static void

Response

- _toClient : Account *
- _toSocket : boost::shared_ptr<TCPConnection> *
- _sizeData = 0 : short
- _response : std::string
- + setSizeData(std::vector<std::string> data) : void
- + setResponse(CommunicationServer answerType, std::vector<std::string> data) : void
- data) : void
- + getClient() : Account *
- + getSocket() : TCPConnection::pointer &
- + getResponse() : std::string &
- + extractBitsetValue(int startBit, int endBit, std::bitset<16> dataContainer) : int

return

ProtocolClient

typedef void(ProtocolClient::*funcs)(DataFromClient &data)

_ptr : funcs

typedef std::map<CommunicationClient, funcs>PointersOnFuncs

- functions : PointersOnFuncs
- + handshake(DataFromClient &data) : bool
- + signup(DataFromClient &data) : bool
- + signin(DataFromClient &data) : bool
- + callRequest(DataFromClient &data) : bool
- + hangUp(DataFromClient &fromClient): void
- + sendText(DataFromClient &data) : bool
- + mute(DataFromClient &data) : bool
- + addParticipantToCall(DataFromClient &data) : bool
- + sendFile(DataFromClient &data): bool
- + pong(DataFromClient &data) : bool
- + acceptCall(DataFromClient &data) : bool
- + declineCall(DataFromClient &data) : bool
- + addContact(DataFromClient &data) : bool
- + acceptInvitation(DataFromClient &fromClient): void
- + declineInvitation(DataFromClient &fromClient): void
- + modifyStatus(DataFromClient &data) : bool
- + modifyLogin(DataFromClient &data) : bool
- + modifyLocation(DataFromClient &data) : bool
- + addToFavorites(DataFromClient &data) : bool
- + removeFromFavorites(DataFromClient &fromClient): void
- + addNickname(DataFromClient &data) : bool
- + removeContact(DataFromClient &data) : bool
- + modifyProfilePicture(DataFromClient &data) : bool
- + listenToMail(DataFromClient &data): bool
- + success(DataFromClient &fromClient): void
- + error(DataFromClient &fromClient) : void
- + affectTCPConnectionToAccountWithUsername(std::string username) : void
- + initMethod(): void
- + methodChecker(DataFromClient &fromClient): void



- _acceptor : boost::asio::ip::tcp::acceptor
- _newConnection : TCPConnection::pointer
- TCPList : std::list<TCPConnection::pointer>
- + writeOnServer(const std::string &message) : void
- + readOnServer(): void
- + getList() : std::list<TCPConnection::pointer> *
- startAccept() : void
- handleAccept(TCPConnection::pointer newConnection, const boost::system::error_code& error): void

TCPConnection

- _socket : boost::asio::ip::tcp::socket
- _response : boost::asio::streambuf
- readMessage : std::string
- _account = NULL : Account *
- + create(boost::asio::io_service& ioService) : static boost::shared_ptr<TCPConnection>
- + getSocket(): boost::asio::ip::tcp::socket&
- + asyncWrite(const std::string &message) : void
- + asyncRead(): void
- + close(): void
- handleWrite(const boost::system::error_code&, size_t) : void
- handleRead(const boost::system::error_code& error) : void

return -

