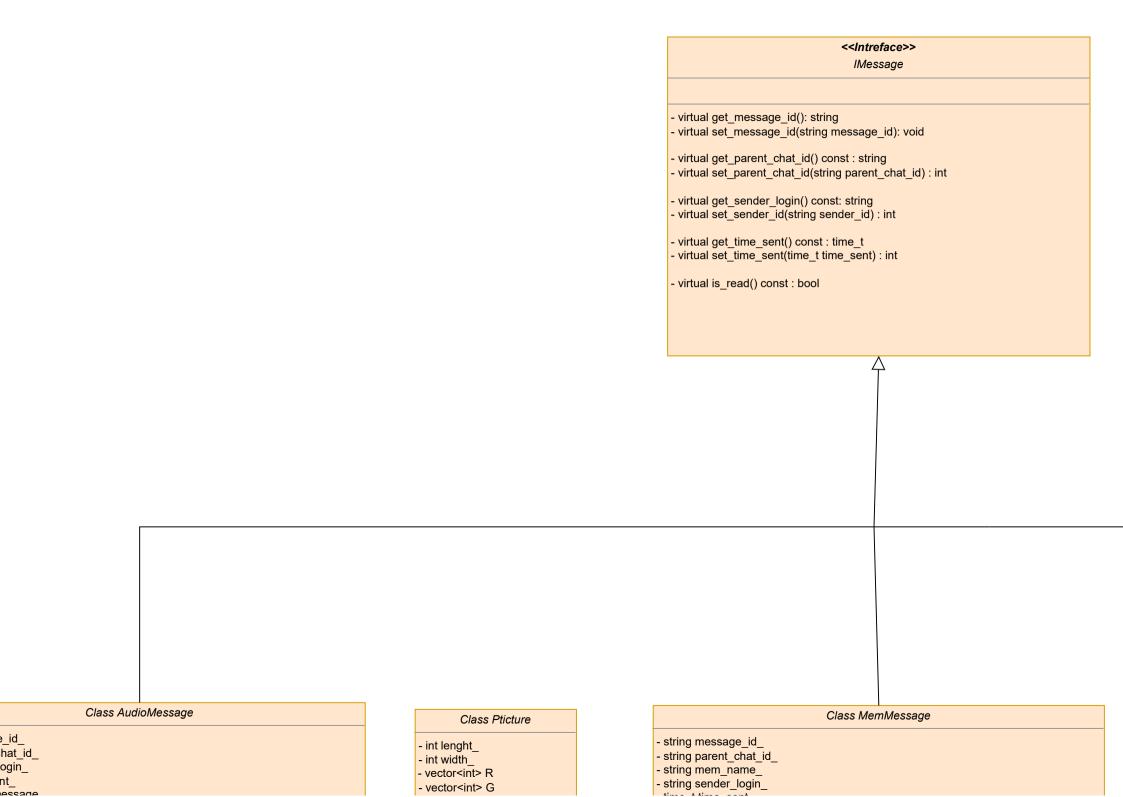
string messagestring parent_cstring sender_l

- time_t time_se



Class TextMessage - string message_id_ - string parent_chat_id_ - string sender_id_ - time_t time_sent_

Class User

- string loginstring passwordstring token
- vector<string> chat_list

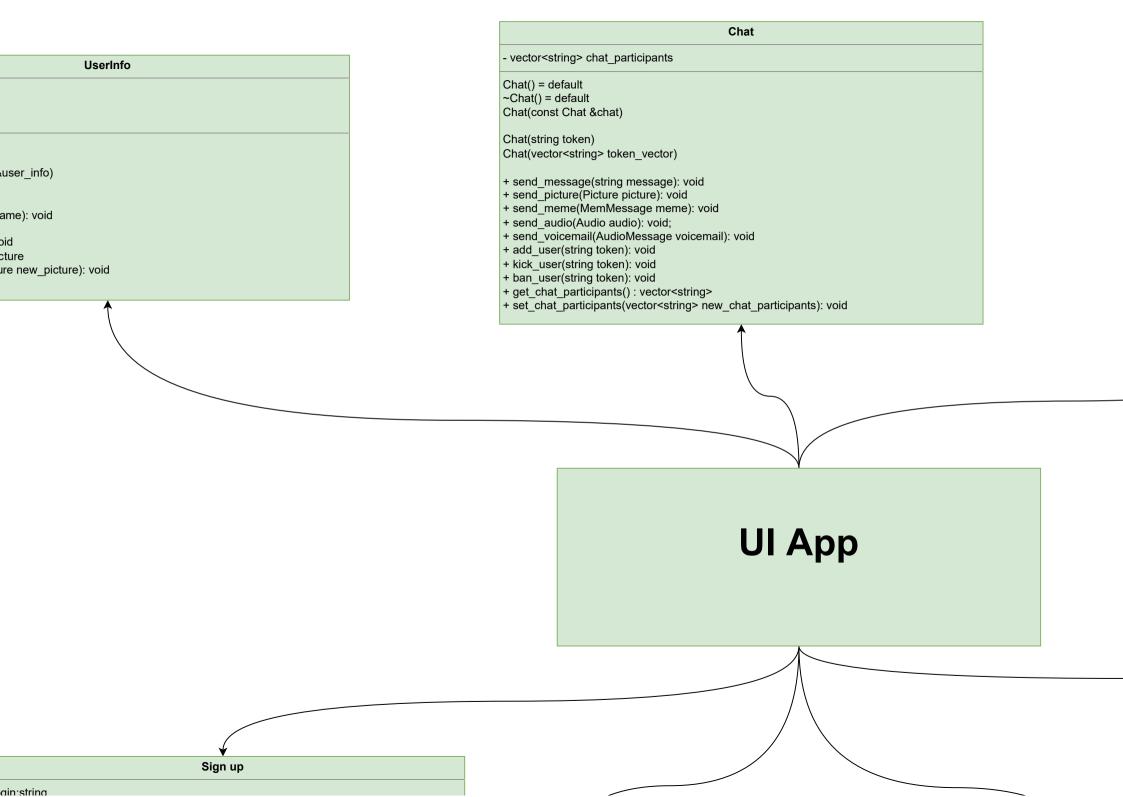
- string id_: - vector<IMessage> messages_: - vector<string> participants_: - time_t date_creaton_:

- string name
- string id
- Picture profile_picture

UserInfo() = default ~UserInfo() = default UserInfo(const UserInfo &

- + get_name(): string

- + set_name(string new_n + get_id(): string + set_id(string new_id): ve + get_profile_picture(): Pi + set_profile_picture(Picture



Search

string query

Chat() = default

~Chat() = default

Chat(const Chat &chat)

- + search_by_query(string query) : vector<string>
- + get_query(): string
- + set_query(string new_query): void

Chat List

- chat_list: chat[]

- + select_chat(chat_id: string): void
- + remove_chat(chat_id: string): void

```
Class Audio

- size_t channels_
- size_t duration_
- size_t sample_rate_
- size_t buffer_size_
- vector<string> buffer

+ play(): void
+ record(): void
+ Audio()
+ ~Audio()
```

- bool is_read_ AudioMessage() ~AudioMessage

AudioMessage(

: parent_chat_ text_messag

AudioMessage(

const std::string

time_t time_sen

+ get_message_

+ get_parent_ch

+ get_sender_lo

+ get_time_sent

+ get_message_

+ play_audio(au + stop_audio(au + transfer_to_te

+ set_message_

+is_read() const

: messag

sender_ time_se

const

```
() = default
const std::string &parent_chat_id, const std::string &sender_login,
std::string &text message, time t time sent)
id (parent chat id), sender login (sender login),
ge (text message), time sent (time sent), is read(false)
const std::string &message id, const std::string &parent chat id,
&sender login, std::string text message,
t, bool is read)
e id (message id), parent chat id (parent chat id),
login_(sender_login), audio_message_(std::move(text_message)),
ent (time sent), is read (is read)
id() const : string
at id() const : string
gin() const : string
() const : time t
content() const : Audio
dio t): void
dio t): void
kt(audiot t) : string
id(string id): void
: bool
```

= default

vector<int> B

+ Picture()

+ ~Picture()

```
- bool is read
MemMessage() = default
~MemMessage() = default
MemMessage(const std::string &parent chat id, const std::string &sender login,
         const std::string &text message, time t time sent)
  : parent chat id (parent chat id), sender login (sender login),
   text message (text message), time sent (time sent), is read(false)
MemMessage(const std::string &message id, const std::string &parent chat id,
const std::string &sender login, Picture& mem picture,
time t time sent, bool is read)
      : message id (message id), parent chat id (parent chat id),
       sender login (sender login), mem picture (std::move(mem picture)),
        time sent (time sent), is read (is read)
+ get message id() const : string
+ get parent chat id() const : string
+ get sender login() const : string
+ get_message_picture() const : Picture&
+ get time sent() const : time t
+ set message id(string id): void
+is read() const: bool
```

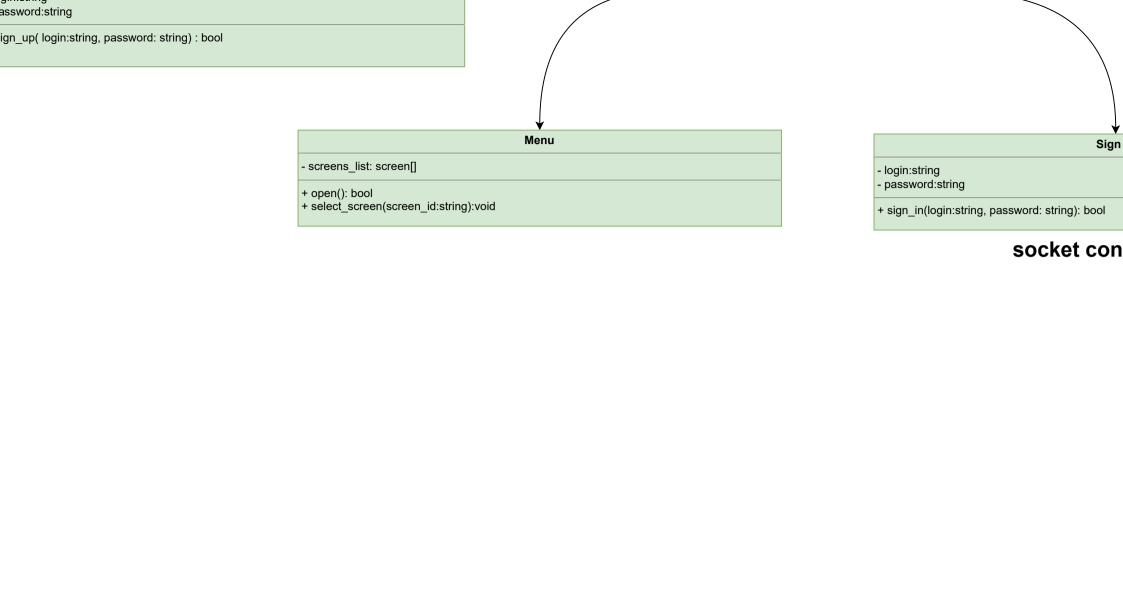
- time_t time_sent_ - Picture mem_picture

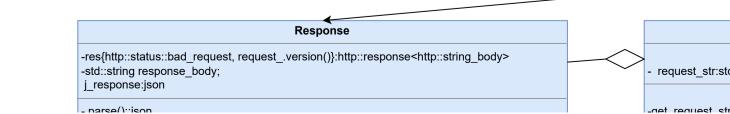
```
- string text message
- bool is read
TextMessage() = default
~TextMessage() = default
TextMessage (const std::string &parent_chat_id, const std::string &sender_login,
          const std::string &text message, time t time sent)
  : parent chat id (parent chat id), sender login (sender login),
   text message (text message), time sent (time sent), is read(false)
TextMessage (const std::string &message id, const std::string &parent chat id,
const std::string &sender login, std::string text message,
time t time sent, bool is read)
       : message id (message id), parent chat id (parent chat id),
       sender login (sender login), text message (std::move(text message)),
        time sent (time sent), is read (is read)
+ get message id() const : string
+ get parent chat id() const : string
+ get sender login() const : string
+ get_message_content() const : string
+ get time sent() const : time t
+ set message id(string id): void
+is read() const: bool
```

```
User() = default
~User() = default
User(const std::string &user login)
: login(user login), status(0)
User(const std::string &user login, const std::string &user passwor
: login(user login), password(user password), status(0)
// constructor-move
User(std::string &user login, std::string &user password,
    std::string user token, int user status)
  : login(std::move(user_login)), password(std::move(user_passw
    token(std::move(user token)), status(user status)
+ get login() const : sring
+ get_password() const : string
+ get_token() const : string
+ get active status() const : int
+ get chat list() const : vector<string>
+ set login(const srting &login): void
+ set password(const string &login): void
+ set token(const string &token): void
+ set status(int active status): void
+ add chat to list()
+ operator==(const User &user) const : bool
+ operator!=(const User &user) const : bool
```

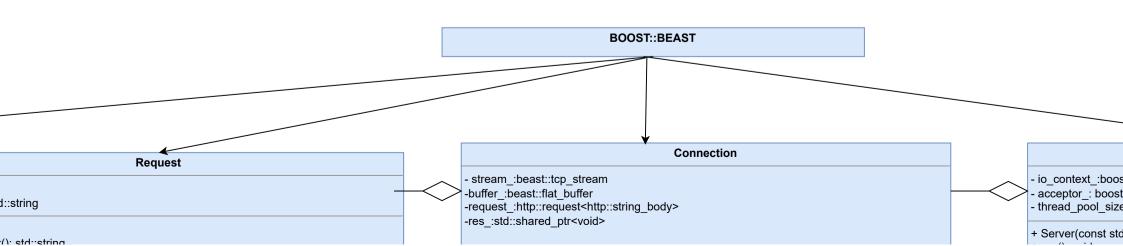
```
Chat() = default
~Chat() = default
Chat(vector<string> participants, time t date creation)
  : participants_(std::move(participants)),
     date creation (date creation)
Chat(string id,
     vector<IMessage&> messages,
     vector<string> participants, time t date creation)
       id (std::move(id)),
       messages_(std::move(messages)),
       participants (std::move(participants)),
       date creation (date creation) {
+ get last message() const : IMessage&
+ get_participants() const : vector<string>
+ get_messages() const : vector<IMessage&>
+ get_chat_date_creation() const : time_t
+ get_id() const : string
+ push_new_message(const IMessage& new_message) : void
+ add_new_participant(string& new_participant_id) : void
+ is_empty() const : bool
+ is dialogue() const : bool
+ is monologue() const : bool
+ is_polilogue() const : bool
+ operator==(const Chat &chat) const : bool
+ operator!=(const Chat &chat) const : bool
```

ord)),





nect???



data flow from server Server st::asio::io_context ::asio::ip::tcp::acceptor e_: std::size_t ::string& address, const std::string& port,std::size_t thread_pool_size)

<<interface>> **IDataBase** - virtual int save(const string & table, vctor <string> values, string where) - virtual int update(const string & table, vctor <string> values, string where) - virtual int insert(const string & table, vctor <string> values, string where) - virtual int delete(const string & table, vctor <string> values, string where) virtual result select(const string &table, strig where, vector <string> columns) Postgre DB - shared ptr <connection> PG connection + Postgre_DB(string host, string port): + ~Postgre DB() + init_tables(): int + add user(const User& user) : int + get user by login(string login) const: User + find user by login(string login) const : bool + delete_user(User& user) : int + change user login(User& user, string new login): int + change user password(User& user, string new password): int + add message(Message& message): int + get last N messages from chat(string chat id, int num of messages) const: vector <Message> + delete message(Message& message) : int + add chat(Chat& chat): int + find chat by participants(vector<string> participants) const : string + get_participants_from_chat(const Chat& chat) const : vector<string> + get all chats by login(string login) const: vector<Chat> + delete chat(Chat& chat) : int



+get_user_id(std::string):int
+change_user_login():int
+change_user_password():int
+get_all_chats_by_login(std::string):vector
+add_chat():int
+del_chat():int

+get_user_id(sto +get_user_by_lo +get_all_chats_l +add_chat():std:: +del_chat():std:: -what_type_requ (). 314..311119 d::string):int ogin(std::string):std::string by_login(std::string):vector :string string uest(std::string):bool

+ Connection(boost::asio::ip::tcp socket&& socket)

- + start(): void do_read():void
- -handle_read(boost::beast::error_code e,std::size_t bytes_transferred):void
 handle_write(bool close, boost::beast::error_code e,std::size_t bytes_transferred):void
- -do_close():void

- | + run(): void - start_accept():vo
- -handle_accept(bo
- handle_stop():vo

id post::beast::error_code e, tcp::socket socket):void id

