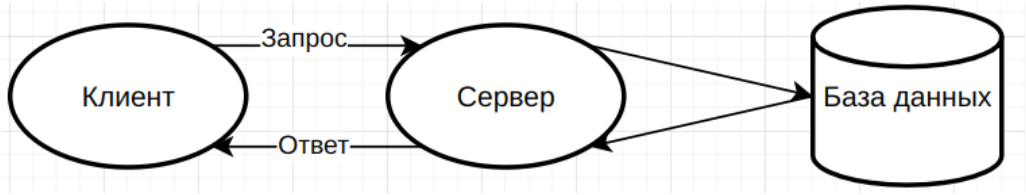
Лабораторная работа 2

ОПИСАНИЕ АРХИТЕКТУРЫ ПРИЛОЖЕНИЯ

Чат-мессенджер имеет клиент-серверную архитектуру.



Клиент – локальный компьютер на стороне пользователя, генерирующий запросы к серверу на предоставление данных или выполнение определенных действий.

Сервер – программная часть, предназначенная для обслуживания таких запросов, предоставления доступа к ресурсам и сохранении данных.

База данных – хранилище данных приложения.

ОПИСАНИЕ СУЩНОСТЕЙ ПРОГРАММЫ

Клиентская часть

Клиент (class Client)

Свойства:

- username – пользовательский никнейм;
- email – электронная почта;
- password – пароль;
- connection – поле типа IConnection

Методы:

- login()

Осуществляет попытку авторизации пользователя, отправляя запрос серверу. Если до этого аккаунт с такими данными не существовал, авторизация заканчивается неудачно. В таком случае пользователь может начать регистрацию и снова попытаться авторизоваться. Если данные регистрации не конфликтуют с тем, что хранится в базе данных, создается новый аккаунт. Пользователь переходит к этапу входа в аккаунт. Ему необходимо ввести данные существующего аккаунта. Если сервер не отклоняет запрос, то открывается доступ к аккаунту.

- createAccount()

В случае успешного доступа к аккаунту необходимо создать класс типа IAccount. В метод передается та информация, которая была получена от сервера после успешного входа в аккаунт.

Аккаунт (abstract class IAccount)

Свойства:

- user_id – уникальный пользовательский идентификатор;
- connection – поле типа IConnection

Методы:

- logout()

Осуществляет выход из аккаунта.

- sendMessage(chat_type, chat_id, msg);
  Отправляет сообщение типа IMessage. Сообщение сохраняется в базе данных в таблице, которая определяется типом чата chat_type. Чат уже должен существовать до отправления.
- deleteMessage(chat_type, chat_id, msg_id);
  Удаляет сообщение с идентификатором msg_id из чата типа chat_type с идентификатором chat_id.
- searchUser(username);
  Находит пользователя с никнеймом username. Возвращает его пользовательский идентификатор.
- getUserInfo(user_id);
  Возвращает информацию о пользователе идентификатором user_id в виде структуры StatusInfo, если доступ к аккаунту пользователя не заблокирован.
- blockUser(user_id);
  Блокирует доступ к текущему аккаунту пользователю с идентификатором user_id.
- unblockUser(user_id);
  Открывает доступ к текущему аккаунту пользователю с идентификатором user_id.
- createIndChat(first_user_id, second_user_id);
  Создает чат для общения между двумя пользователями, если между ними нет блокировок.
- createBroadChat(chat_type, chat_info);
  Создает чат типа chat_type с данными в структуре типа chat_info.
- searchBroadChat(chat_type, chat_name)
  Находит чат типа chat_type с именем chat_name. Возвращает chat_id.
- deleteChat(chat_id, chat_type);
  Удаляет чат типа chat_type с уникальным идентификатором chat_id.
- addChatSubs(chat_id, chat_type, user_id);
  Добавляет пользователя с идентификатором user_id в чат типа chat_type с идентификатором chat_id.
- deleteChatSubs(chat_id, chat_type, user_id);
  Удаляет пользователя с идентификатором user_id из чата типа chat_type с идентификатором chat_id.
- getBroadChatInfo(chat_type, chat_id)
  Получает всю информацию о группе или канале: статусная информация участников, идентификатор владельца, название чата.

Пользовательский аккаунт (class UserAccount, implement IAccount)

Свойства:
- поля, наследуемые от абстрактного класса IAccount;
- status_info – поле типа StatusInfo;

Методы:
- методы, наследуемые от абстрактного класса IAccount;
- changeStatusInfo(status_info)
  Изменяет информацию о пользователе.

Аккаунт суперпользователя (class AdminAccount, implement IAccount)

Свойства:
- поля, наследуемые от абстрактного класса IAccount;
- username – никнейм суперпользователя;

Методы:

- методы, наследуемые от абстрактного класса IAccount;
- banUser(user_id)
  Осуществляет бан пользователя. Забанненый пользователь может отправлять сообщения только суперпользователю и не может изменять информацию о себе.
- unbanUser(user_id)
  Отменяет бан пользователя. Пользователю возвращается возможность взаимодействия с приложением на прежнем уровне.
- banChat(chat_id, chat_type)
  Осуществляет бан чата. В забаненный чат никто не может отправлять сообщения.
- unbanChat(chat_id, chat_type)
  Отменяет бан чата. Чат открывается для пользователей.

Статусная информация (structure StatusInfo)

Свойства:
- username - пользовательский никнейм;
- birthday – день рождения пользователя;
- text_status – статусная информация;
- image – картинка профиля пользователя;

Информация о канале/группе (structure BroadChatInfo)

Свойства:
- owner_id – идентификатор владельца;
- chat_name – имя канала/группы;

Чат (abstract class IChat)

Свойства:
- msgs_list – список сообщений;
- chat_id – идентификатор чата;

Методы:
- getChatId()l
  Возвращает идентификатор чата.
- getMessages();
  Возвращает список всех сообщений.
- deleteMessages();
  Удаляет все сообщения.
- addMessage(msg);
  Добавляет сообщение типа IMessage.
- receiveNewMessages(Date since);
  Возвращает список сообщений, которые были созданы после указанной даты.
- deleteMessage(msg_id);
  Удаляет сообщение с идентификатором msg_id.

Индивидуальный чат (class IndChat, implement IChat)

Свойства:
- *поля, наследуемые от абстрактного класса IChat.*

Методы:
- *методы, наследуемые от абстрактного класса IChat.*

Многопользовательский чат (abstract class IBroadChat)

Свойства:

- поля, наследуемые от класса IChat;
- users_id - список идентификаторов участников чата;
- owner_id – идентификатор пользователя;
- chat_name – название чата;

Методы:

- методы, наследуемые от класса IChat;
- inviteUser(user_id)
  Позволяет пригласить пользователя с идентификатором user_id в чат.
- deleteUser(user_id)
  Удаляет пользователя с идентификатором user_id из чата. На это имеет право только суперпользователь или владелец.
- banUser(user_id)
  Блокирует доступ к чату пользователю с идентификатором user_id. На это имеет право только суперпользователь или владелец.

Чат группы (class GroupChat, implements IBroadChat)

Свойства:

- поля, наследуемые от класса IBroadChat

Методы:

- методы, наследуемые от класса IBroadChat. Реализация состоит в том, что любой желающий может писать сообщения в группу.
- GetUsers()
  Возвращает список никнеймов всех пользователей группы.

Чат канала (class ChannelChat, implements IBroadChat)

Свойства:

- поля, наследуемые от класса IBroadChat

Методы:

- методы, наследуемые от класса IBroadChat. Реализация состоит в том, что только владелец может писать сообщения в канал.

- методы, наследуемые от класса IBroadChat. Реализация состоит в том, что только владелец может писать сообщения в группу.

Сообщение (abtract class IMessage)

Свойства:

- msg_type – тип сообщения (текст, аудио, картинка);
- msg_id – идентификатор сообщения;

Методы:

- getRawData()
  Возвращает содержимое сообщения в виде байтов.
- GetMsgType()
  Возвращает тип сообщения.

Однопользовательское сообщение (class IndMessage, implements IMessage)

Свойства:

- поля, наследуемые от класса IMessage;
- receiver_id – идентификатор получателя;
- sender_id – идентификатор отправителя;
- content – поле типа ImessageContent;

Методы:

- методы, наследуемые от класса IMessage
- getEdgesId()
  Возвращает идентификаторы получателя и отправителя.

Многопользовательское сообщение (BroadMessage, implements IMessage)

Свойства:

- поля, наследуемые от класса IMessage;
- sender_id – идентификатор отправителя;
- content – поле типа ImessageContent;

Методы:

- методы, наследуемые от класса IMessage;
- getSenderId()
  Возвращает идентификатор отправителя;

Содержимое сообщения (abstract class IMessageContent)

Свойства:

- send_date – время отправления сообщения

Методы:

- getRawData()
  Возвращает содержимое в виде байтов.

Текстовое содержимое сообщения (TextMessageContent, implement IMessageContent)

Свойства:

- поля, наследуемые от класса IMessageContent;
- text_data – текстовые данные;

Методы:

- методы, наследуемые от класса ImessageContent;

Содержимое сообщения в виде аудио (AudioMessageContent, implement IMessageContent)

Свойства:

- поля, наследуемые от класса ImessageContent;
- autio_data – данные в виде аудио;
- duration – время аудио;

Методы:

- методы, наследуемые от класса ImessageContent;

Содержимое сообщения в виде картинки (ImageMessageContent, implement IMessageContent)

Свойства:

- поля, наследуемые от класса ImessageContent;
- image_pixels – пиксели картинки

Методы:

- методы, наследуемые от класса ImessageContent;

Класс соединения (abstract class IDonnection)

Методы:

- send()
  Отправляет запрос.
- receive()
  Получает ответ.

Класс соединения на сокетах (class SocketConnection, implements IConnection)

Свойства:

- socket – сокет соединения

Методы:

- методы, наследуемые от класса Iconnection

Сервер (class Server)

Свойства:

- *socket – сокет соединения с сервером;*
- *data_base – клиент базы данных;*

Методы:

- setup()
  Настраивает сервер на работу.
- startAccept()
  Отвечает за принятие соединений.
- HandleNewConnection()
  Обрабатывает новое соединение, создавая новый поток.
- closeConnection()
  Закрывает сокет сервера и все соединения.

База данных (absract class IDataBase)

Методы:

- *setup()*
  Настраивает БД на работу: проверяет и создает таблицы, настраивает соединения.
- *startConnection()*
  Открывает соединение для подключения со стороны сервера.
- *closeConnection()*
  Закрывает соединение с сервером.
- *getUserInfo(username)/getUserInfo(user_id)*
  Возвращает информацию о пользователе в виде структуры StatusInfo.
- *updateUser(user_id)*
  Обновляет информацию о пользователе с идентификатором user_id.
- *deleteUser(user_id)*
  Удаляет пользователя с идентификатором user_id.
- *getBroadChatInfo(chat_id, chat_type)*
  Возвращает информацию о чате типа chat_type с идентификатором chat_id в виде структуры BroadChatInfo.

- *addBroadChat(chat_type, chat_info)*
  Создает чат типа chat_type с информацией в виде структуры BroadChatInfo().
- *updateBroadChat(chat_type, chat_info)*
  Обновляет чат типа chat_type с информацией в виде структуры BroadChatInfo().
- *deleteChat(chat_type, chat_id)*
  Удаляет чат типа chat_type с идентификатором chat_id.
- *getMessages(chat_type, chat_id, date)*
  Возвращает все сообщения из чата типа chat_type с идентификатором chat_id, начиная с даты date.
- *addMessage(msg, chat_type, chat_id)*
  Добавляет сообщение msg типа IMessage в чат типа chat_type с идентификатором chat_id.
- *delMessage(msg_id, chat_type, chat_id)*
  Удаляет сообщение с идентификатором msg_id из чата типа chat_type и идентификатором chat_id.
- *delAllMessages(chat_type, chat_id)*
  Удаляет все сообщения из чата типа chat_type с идентификатором chat_id.

MySQL база данных (MySQLDataBase, implements IDataBase)

Методы:

- методы, наследуемые из класса IDataBase