1

Технически Университет – Варна  
Катедра „СИТ“

**Проект ПМС и АБТМУ**Система обмен съобщения Viber 2.0

Изготвили:

инж. Александрина Ръженкова (фак. № 21651237)  
инж. Владислав Маринов (фак. № 21651228)  
инж. Йоан Енчев (фак. № 21651230)  
Специалност: СИ  
Курс: I

**Съдържание**

[**1.** **Въведение** 3](#_Toc92127320)

[**2.** **Анализ на проблема** 4](#_Toc92127321)

[2.1. Функционални изисквания 4](#_Toc92127322)

[2.2. Концептуален модел 4](#_Toc92127323)

[2.3. ERD 5](#_Toc92127324)

[2.4. Use case диаграма 5](#_Toc92127325)

[2.5. UML диаграма 6](#_Toc92127327)

[**3.** **Структура и архитектура на проекта** 7](#_Toc92127328)

[3.1. Структура 7](#_Toc92127329)

[3.2. Архитектурни особености 7](#_Toc92127330)

[**4.** **Проектиране на системата** 8](#_Toc92127331)

[4.1. Проектиране на сървър 8](#_Toc92127332)

[4.2. Проектиране на клиент 8](#_Toc92127333)

[4.3. Описание на комуникационният протокол 8](#_Toc92127334)

[**5.** **Реализация на бизнес логиката, графичния интерфейс** **и тестове** 11](#_Toc92127335)

# **Въведение**

Да се изгради цялостна система за обмен на съобщения. Сървърната част се реализира чрез Java технологии. Клиентската част е Андроид клиент.

Системата трябва да поддържа следните неща:

* Автентикация в системата с потребителско име и парола
* Обмен на текстови съобщения
* Редактиране на изпратеното съобщение
* История на съобщенията
* Заявка за приятелство (потребителите виждат само хората от приятелския лист, но могат да търсят по потребителско име и да изпращат покана за приятелство)
  + Приемане
  + Изчакване
  + Отказано
* Комуникацията да става по сокет

# **Анализ на проблема**

## Функционални изисквания

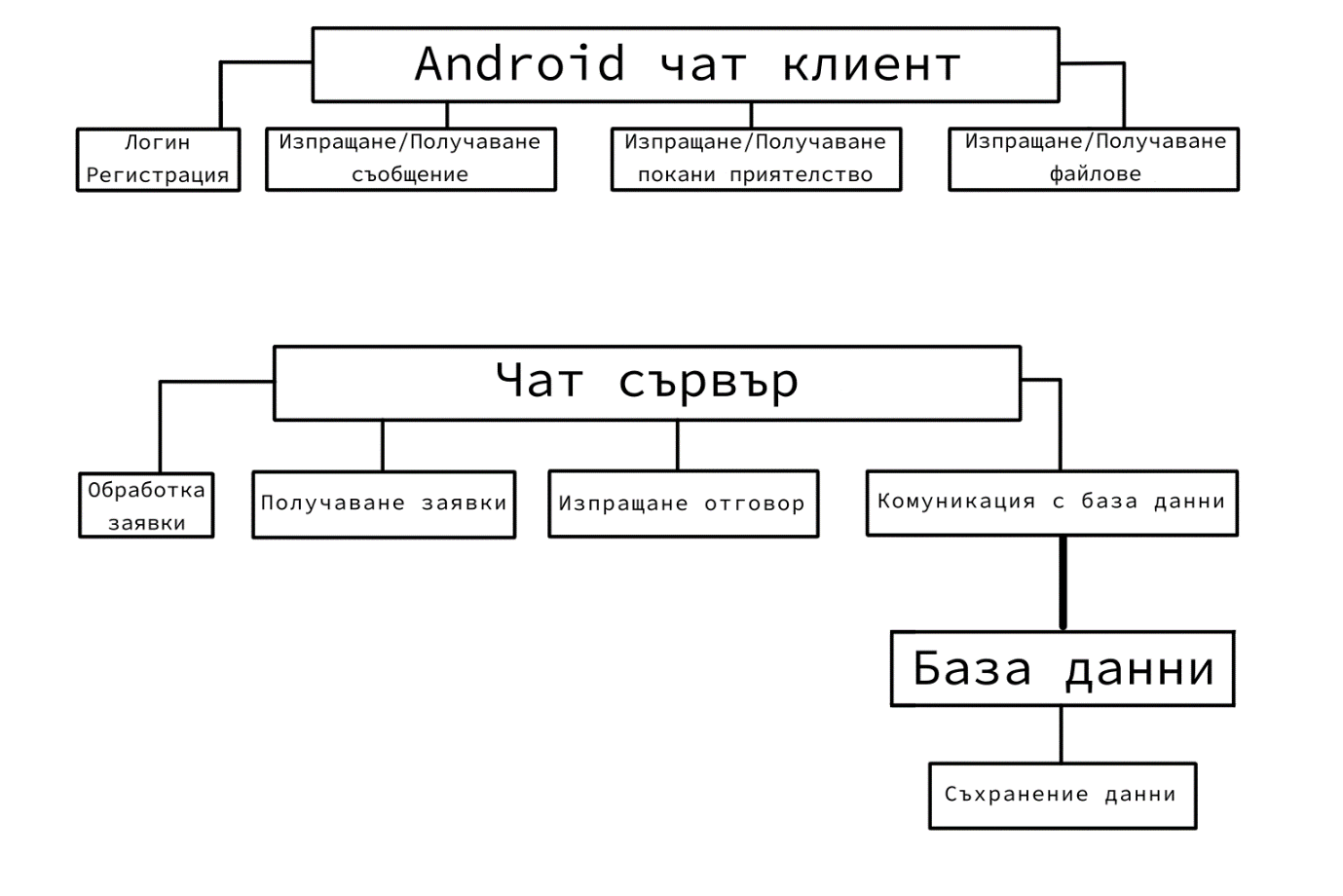
Чат системата трябва да може да изпраща съобщения между потребителите. Всеки потребител има възможността да се присъедини или да бъде поканен в чат група. Под чат група се разбира чат с повече от двама участници. За да може да бъде поканен потребител в чат, той трябва да е приятел с поне един от участниците в групата. За тази цел е възможно потребителите да изпращат покани на другите потребители.

Чат клиента трябва да предоставя горните възможности на клиента под формата на Андроид базирано приложение. То трябва да може да комуникира със сървър, обслужващ база данни, съхраняваща данните на потребителите като изпратени съобщения, приятелства и други.

Сървърът следва да приема заявки от Андроид клиента, да ги обработи, да събере данните от базата данни, ако има такива, да сформира отговор и да изпрати отговора. За целта сървърът слуша на определен порт (8080) за връзка, при получена заявка да я парсира, събере желаните от клиента данни, или да впише изпратени данни. Отговор се връща на всички получени заявки. В случай на вътрешна грешка или невалидна заявка се връща подобаващ отговор.

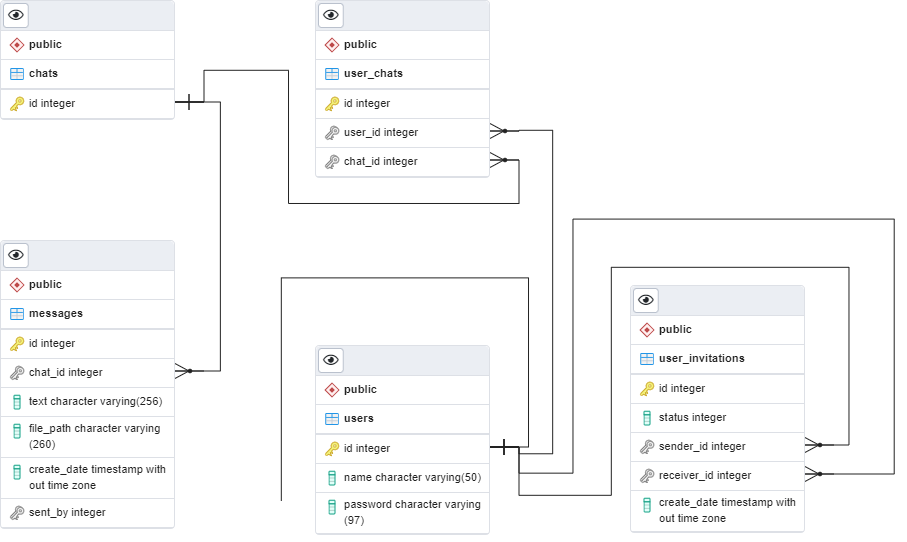
Базата данни трябва да е с подходящи релации за безпроблемно следене на потребители и техните съобщения, групи и приятелства.

## Концептуален модел



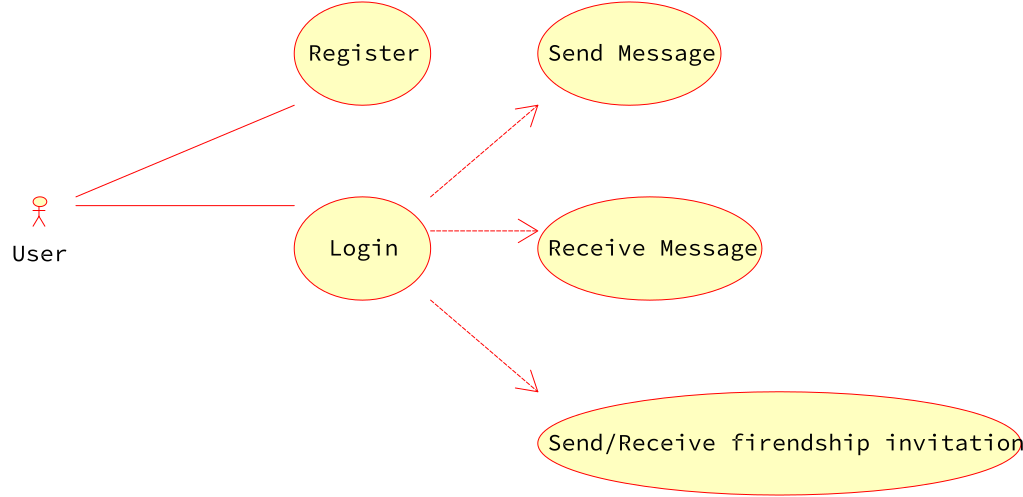
*Фиг 2.1 Концептуален модел на системата*

## ERD



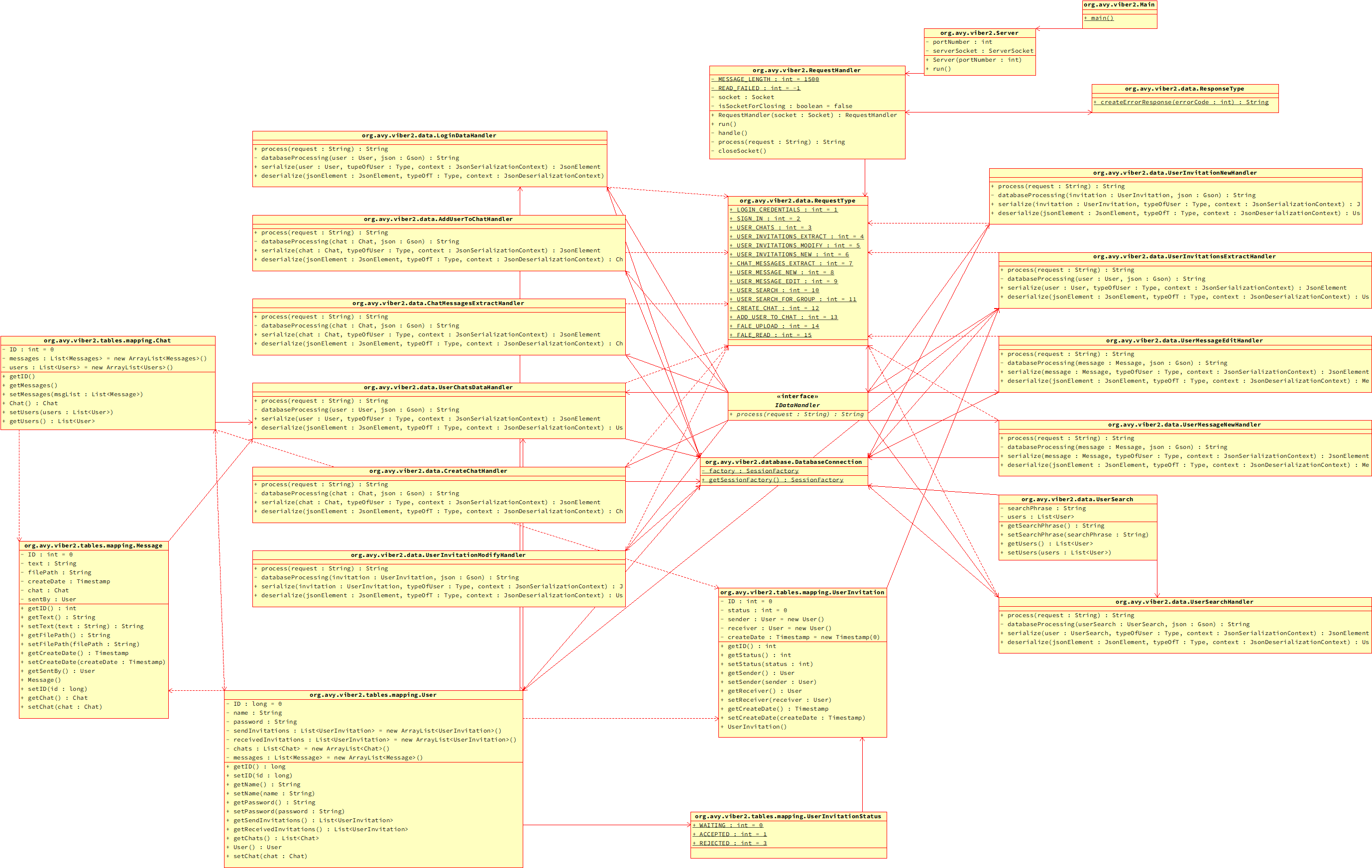
*Фиг 2.2 Entity Relationship Diagram*

## Use case диаграма



*Фиг 2.3 UML диаграма*

## UML диаграма



*Фиг 2.4 Use case диаграма*

# **Структура и архитектура на проекта**

## Структура

Пакет org.avy.viber2

* Main
* Server
* RequestHandler

Пакет org.avy.viber2.data

* AdditionalMessageData
* AddUserToChatHandler
* ChatMessagesExtractHandler
* CreateChatHandler
* FileReadHandler
* IDataHandler
* LoginDataHandler
* RequestType
* ResponseType
* UserChatsDataHandler
* UserInvitationModifyHandler
* UserInvitationNewHandler
* UserInvitationsExtractHandler
* UserMessageEditHandler
* UserMessageNewHandler
* UserSearch
* UserSearchHandler

Пакет org.avy.viber2.database

* DatabaseConnection

Пакет org.avy.viber2. tables. Mapping

* AdditionalData
* Chat
* Message
* User

## Архитектурни особености

За живота на сървъра, връзката с базата данни се осъществява еднократно и при нужда само се достъпва, без да се инициализира отново. Това е постигнато чрез шаблона „Singleton“.

Всяка заявка от сървъра наследява интерфейс, който задължава пренаписването на метод за сериализация, десериализация и обработката на заявката. Всеки клиент, отворя връзка със сървъра през сокета на нова нишка. Така могат да се обработват няколко заявки от различни клиента едновременно и асинхронно.

# **Проектиране на системата**

## Проектиране на сървър

За програмно обезпечение е използван езикът Java. Базата данни (БД) е осъществена чрез PostgreSQL 14.1 RDBMS и за осигуряване на връзка към БД е използван Hibernate ORM 5.6.0.Final. Комуникацията е посредством JSON чрез библиотеката Gson. Програмната среда е Eclipse и Java 17.

## Проектиране на клиент

За клиентската част, отново е използван езикът Java. Програмната среда е Android Studio и Java 1.8.

## Описание на комуникационният протокол

Комуникационният протокол се осъществява чрез JSON съобщения през TCP/IP. Съществуват девет съобщения

|  |  |
| --- | --- |
| **LOGIN\_CREDENTIALS** | Клиент иска да влезе |
| **SIGN\_IN** | Клиент иска да се впише |
| **USER\_CHATS** | Клиент иска чатовете си |
| **USER\_INVITATIONS\_EXTRACT** | Клиент иска поканите си |
| **USER\_INVITATIONS\_MODIFY** | Клиент отговаря на покана |
| **USER\_INVITATIONS\_NEW** | Клиент заявява нова покана |
| **CHAT\_MESSAGES\_EXTRACT** | Клиент иска съобщенията си |
| **USER\_MESSAGE\_NEW** | Клиент пише ново съобщение |
| **USER\_MESSAGE\_EDIT** | Клиент редактира съобщение |
| **USER\_SEARCH** | Клиент търси друг клиент |
| **USER\_SEARCH\_FOR\_GROUP** | Клиент търси група |
| **CREATE\_CHAT** | Клиент създава чат |
| **ADD\_USER\_TO\_CHAT** | Клиент добавя друг клиент в чат |

Клиента сформира заявка, според типа на заявката: {"requestType":"<тип-заявка>",<тяло на заявката>}

Типът на заявката и номера на заявката е зададен в класа „org.avy.viber2.data.RequestType“

При заявка „**LOGIN\_CREDENTIALS**“, сървърът връща отговор **ID** на потребителят желаещ да влезе в акаунта си. В случай на грешно име или парола, или несъществуващ потребител се връща **ID 0**, което се интерпретира от клиента като „**грешни данни**“.

Примерна заявка: {"requestType:"1","name":"avy","password","<парола>"}

Примерен отговор: {"requestType:"1","userID":"658689"}

При заявка “**SIGN\_IN**”, сървърът записва в БД новият потребител и автоматично присвоява идентификатор на новия потребителя.

Примерна заявка: {"requestType:"2","name":"avy","password","<парола>"}

Примерен отговор: {"requestType:"2","userID":"658689"}

При заявка “**USER\_CHATS**”, сървърът връща всички чатове на клиента.

Примерна заявка: {"requestType:"3","name":"avy","password","<парола>"}

Примерен отговор: {"requestType:"3","userID":"658689"}

При заявка “**USER\_INVITATIONS\_EXTRACT”**, сървърът връща всички покани за клиента по неговото ID.

Примерна заявка: {"requestType:"4","id":"658689"}

Примерен отговор: {"requestType:"4","invitations":[{"invitationID":"807883","status":"2","sender":{"senderID":"658689","name":"avy"},"receiver":{"receiverID":"7869787986","name":"hristo"}}]}

При заявка **“USER\_INVITATIONS\_MODIFY**“, сървърът променя състоянието на поканата при приемане или отказване на покана.

Примерна заявка: {"requestType":"5","invitationID":"807883","status":"2"}

Примерен отговор: {"requestType":5,"invitationID ":1,"status":1}

При заявка „**USER\_INVITATIONS\_NEW“**, сървърът запомня нова покана със статус по подразбиране **0**.

Примерна заявка: {"requestType":"6","sender":"658689","receiver":"7869787986"}

Примерен отговор:{"requestType":6}

При заявка „**CHAT\_MESSAGES\_EXTRACT“**, сървърът взема съобщенията на клиента и му ги изпраща.

Примерна заявка: {"requestType":"7","chatID":"2"}

Примерен отговор: {"requestType":7,"messages":[]}

При заявка „**USER\_MESSAGE\_NEW“**, сървърът приема ново съобщение.

Примерна заявка: {"requestType":"8","chatID":"2","text":"Hey!","user":"658689"}

Примерен отговор: {"requestType":8,"chatID":2,"message":{"id":6,"message":"hello","date":"2022-01-03 13:16:22.812","sender\_username":"","sender\_id":2}}

При заявка „**USER\_MESSAGE\_EDIT**“, сървърът пренаписва редактирано съобщение.

Примерна заявка: {"requestType":"9","chatID":"2","text":"hello2","user":"2","messageID":"8"}

Примерен отговор: {"requestType":9,"chatID":0,"message":{"id":0,"message":""}}

При заявка „**USER\_SEARCH**“, сървърът търси потребител по зададено име

Примерна заявка: {"requestType":"10","phrase":"kir"}

Примерен отговор: {"requestType":10,"users":[{"name":"kiril","id":2}]}

При заявка „**USER\_SEARCH\_FOR\_GROUP**“, сървърът търси група по зададено име

Примерна заявка: {"requestType":"11","phrase":"kir"}

Примерен отговор: {"requestType":11,"users":[{"name":"kiril","id":2}]}

При заявка „**CREATE\_CHAT**“, сървърът създава нов чат

Примерна заявка: {"requestType":"12","userID":"2"}

Примерен отговор: {"requestType":12,"chatID":22}

При заявка „**ADD\_USER\_TO\_CHAT**“, сървърът добавя нов потребител до

Примерна заявка: {"requestType":"13","chatID":"2","userID":"1"}

Примерен отговор: {"requestType":13,"userID":1}

# **Реализация на бизнес логиката, графичния интерфейс** **и тестове**

При стартиране на приложението, на потребителя са предоставени опции за регистрация и вписване. В случай че потребителя няма акаунт, то следва да се създаде такъв. Ако има, може да влезе в системата като въведе името и паролата си, които е вписал при регистрацията.

|  |  |
| --- | --- |
| Login | Register |
| *Фиг. 5.1 Страница за вписване* | *Фиг. 5.2 Страница за регистрация* |

След вписване в системата на клиента е предоставен списък с чатовете му. Тук той може да види с кого е писал последно. Може да избере някой от чатовете и да продължи кореспонденцията, ако желае. Предоставени са опции за добавяне на нови чатове и преглед на изпратени или получени покани

При избор на опцията за преглед на покани, потребителят може да види получени нови покани за приятелство и покани, който той/тя е изпратил.

|  |  |
| --- | --- |
|  |  |
| *Фиг 5.3 Чатове на потребителя* | *Фиг 5.4 Чат между потребители* |

|  |  |
| --- | --- |
|  |  |
| *Фиг 5.5 Търсене на потребители* | *Фиг 5.6 Покани за приятелство* |

При избор за търсене на други потребители, потребителят може да намери друг такъв по зададено име. Предоставена е и възможност за създаване на нова група.