

Разработка группового проекта

Версия 0.1

18 марта 2016 г.

1 Задание

Написание многопоточного чата на ≈ 1000 человек. Возможности:

1. написание сообщений самому себе, другому пользователю или в комнату,
2. возможность отправления статических картинок и ссылок,
3. поддержка шифрования,
4. хранение истории; получение истории и статистики,
5. доставка непрочитанных сообщений при запуске,
6. возможность пересылать сообщения,
7. возможность узнать, прочитано сообщение или нет,
8. передача файлов,
9. получение статуса пользователя,
10. mb общение p2p с шифрованием,
11. mb возможность отмечать сообщения избранными.

2 Термины

1. *user* — пользователь:

- логин
- `user_id`
- пароль

2. *room* — комната:

- `room_id`
- список пользователей — `[user_id]`
- название

3. *message* — сообщение:

- `user_id` отправителя
- `room_id` получателей
- тема
- время
- текст

4. *picture* — картинка:

- `picture_id`
- данные изображения

3 Реализация

Выбрана клиент-серверная архитектура.

Функции сервера:

1. хранение истории сообщений,
2. загрузка и хранение картинок,
3. пересылка сообщений от одного пользователя в комнату,
4. доставка сообщений для появившегося в сети пользователя,
5. ответы на запросы пользователя,
6. периодическая проверка связи с клиентом.

Функции клиента:

1. кэширование последних сообщений и картинок (для desktop- и android-приложений),
2. отправка сообщений на сервер,
3. отправка запросов пользователя.

Методы сервера:

1. `is_alive`
2. `new_msg`
3. `send_data`
4. `new_room`

Методы клиента:

1. `connect`
2. `still_alive`
3. `send_msg`
4. `send_data`
5. `new_room`
6. `get_history`
7. `get_statistics`
8. `get_user_status`
9. `get_user_id`
10. `get_rooms_id`
11. `get_room_users_id`
12. `create_room`
13. `synchronize`
14. `disconnect`

4 Некоторые детали реализации

Сервер работает постоянно, поддерживая соединения со всеми клиентами в сети. При подключении клиент отправляет свои данные, и после их проверки сервер подтверждает подключение и начинает синхронизацию.

Когда сообщение от отправителя приходит на сервер, он отправляет его получателю с теми же параметрами.

Тяжелые объекты вроде картинок, истории или статистики, можно отправлять и получать в отдельном потоке.

Картинки и ссылки в сообщении можно передавать, используя теги `##[pict_id = ...]##` и `##[href = ...]##`.

Также нужно предусмотреть систему безопасности: загрузить картинку, историю, сообщение

или статистику можно только являясь связанным с ними.