

Общее техническое задание

Разработать многофункциональный чат для компании SimbirSoft.

Условные обозначения

- **Пользователь** - авторизованный пользователь, который может общаться в чате;
- **Модератор** - авторизованный пользователь, с расширенными правами;
- **Администратор** - авторизованный пользователь, с расширенными правами;
- **Комната** - общий диалог > 2 пользователей;
- **Сообщение** - текст, написанный пользователем в чате;
- **Управляющая команда** - текст начинающийся на '/' и способный выполнять определенные действия;
- **Приватное сообщение** - комната на 2 человек;
- **Чат-бот** - комната со специальным 'пользователем', способным, принимая определенные команды, делать определенные действия;

Матрицы действий

Пользователи:

Доступные действия/Пользователь	Пользователь	Модератор	Администратор
Блокировка пользователей	-	+	+
Разблокировка пользователей	-	+	+
Назначение модераторов	-	-	+
Удаление модераторов	-	-	+

Сообщения:

Доступные действия/ Пользователь	Пользователь	Модератор	Администратор	Заблокированный пользователь
Отправка сообщений	+	+	+	-
Принятие сообщений	+	+	+	+
Удаление сообщений	-	+	+	-

Комнаты:

Доступные действия/ Пользователь	Владелец комнаты	Пользователь	Модератор	Администратор	Заблокированный пользователь
Создание общих комнат	+	+	+	+	-
Создание частных комнат	+	+	+	+	-
Добавление пользователей	+	+	+	+	-
Удаление пользователей	+	-	-	+	-
Переименование комнаты	+	-	-	+	-

Команды

Комнаты:

1. //room create {Название комнаты} - создает комнаты;
-с закрытая комната. Только (владелец, модератор и админ) может добавлять/удалять пользователей из комнаты.
2. //room remove {Название комнаты} - удаляет комнату (владелец и админ);
3. //room rename {Название комнаты} - переименование комнаты (владелец и админ);
4. //room connect {Название комнаты} - войти в комнату;
-l {login пользователя} - добавить пользователя в комнату
5. //room disconnect - выйти из текущей комнаты;

6. `//room disconnect {Название комнаты}` - выйти из заданной комнаты;
-l {login пользователя} - выгоняет пользователя из комнаты (для владельца, модератора и админа).
-m {Количество минут} - время на которое пользователь не сможет войти (для владельца, модератора и админа).

Пользователи:

1. `//user rename {login пользователя}` (владелец и админ);
2. `//user ban`;
-l {login пользователя} - выгоняет пользователя из всех комнат
-m {Количество минут} - время на которое пользователь не сможет войти.
3. `//user moderator {login пользователя}` - действия над модераторами.
-n - назначить пользователя модератором.
-d - "разжаловать" пользователя.

Боты:

1. `//yBot find -k -l {название канала}||{название видео}` - в ответ бот присылает ссылку на ролик;
-v - выводит количество текущих просмотров.
-l - выводит количество лайков под видео.
2. `//yBot help` - список доступных команд для взаимодействия.

Другие:

1. `//help` - выводит список доступных команд.

Прототипы приложения

Если вы разрабатываете с UI, то рабочий прототип приложения + страница регистрации. <https://ninjamock.com/s/FRMCTTx>

REST api

1. Получение списка (login) на данный момент всех пользователей;

Технологии

Java разработка

Java 8, Spring Boot, Spring Web, Spring JPA, Spring Security, Spring DevTools, Spring Websocket, Thymeleaf, Git, Maven, Bootstrap (Materializecss).

.NET разработка

Visual Studio Community, Git

Первая версия приложения

Разработка каркаса чата:

- Только общая комната для всех участников;
- Хранение сообщений в общей очереди (очередь пропадает после рестарта);
- Для входа в чат необходимо указать только login (логин) (без хранения данных).
- Простенький UI:
 - Форма для ввода логина;
 - Общая комната со списком сообщений от всех пользователей и с возможностью отправки сообщений в общий чат.

Отчет

Продемонстрировать работу приложения наставнику.

Java

- Использовать стек Spring технологий (Spring Boot, Spring Web, Spring JPA, Spring Security, Spring DevTools, Spring Websocket);
- Для UI использовать materialize + Thymeleaf;
- Реализовывать в git ветке feature/chat-v1.

.Net

- Использовать ASP.NET MVC/ASP.NET Core для реализации серверной части;
- Для UI использовать на выбор любой из популярных JS фреймворков;
- Для общения с сервером использовать SignalR;