Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

КУРСОВАЯ РАБОТА

“Web сервис для обмена сообщениями и домашними заданиями между студентами и преподавателями”

по дисциплине

Программное обеспечение распределенных вычислительных систем

Выполнил студент Группы № [3540901/81501](https://ruz.spbstu.ru/faculty/95/groups/27682) Сальма Махмуд Тахер

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Работу принял преподаватель доцент Стручков И.В

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2019

**Содержание**

Введение 4

1.1 Интервьюирование заказчика (целевой аудитории) 5

1.2 Анкетирование пользователей 6

1.3 Словарь предметной области 8

1.4 Описание текущей ситуации (как было) 9

1.5 Описание будущей ситуации (как будет) 10

1.6 Описание заинтересованных лиц и конечных пользователей 10

1.7 Бизнес-требования 11

2 Анализ существующих решений 12

2.1 Обзор существующих решений 12

2.1.1 Решение 1. «Telegram» 12

2.1.2 Решение 2. «WhatsApp» 12

2.1.3 Решение 3. «Viber» 13

2.1.4 Решение 4. «Skype» 13

2.1.5 Решение 5. «FaceBook Messenger» 13

2.1.6 Решение 6. «HangOuts» 14

2.1.7 Решение 7. «iMessage» 14

2.1.8 Решение 8. «SMS» 14

2.2 Сравнительный анализ решений 15

2.3 Выводы по анализу 16

3 Концепция решения 16

3.1 Техническое задание 17

3.1.1 Общие сведения 17

3.1.2 Назначение и цели создания системы 17

3.1.3 Требования к системе 17

3.2 Анализ факторов риска проекта 20

4 Проектирование решения 22

4.1 Описание архитектуры/структуры системы 22

4.2 Описание возможных технических решений 23

4.3 Проектирование интерфейса пользователя 23

4.4 Проектирование базы данных 23

5 Программная реализация системы 24

6 Функциональное тестирование решения 29

Тест «Регистрация» 29

Тест «Авторизация» 30

Тест «Добавление сообщений» 31

Тест «Редактирование сообщений» 33

Тест «Подписки и подписчики» 34

Тест «Администратор» 35

Список литературы 37

# Введение

В настоящее время в разрабртке все чаще использется микросервисная ахитектура. Также весьма популярными являются сервисы для обмена сообщениями и публикации заметок и статей.

Целью данной курсовой работы является реализация Web сервиса для публикации статей и заметок, который, в частности, может быть использован для обмена сообщениями и домашними заданиями между студентами и преподавателями.

Для достижения поставленной цели, необходимо решить следующие задачи:

1. Выполнить бизнес-анализ предметной области, включая:
   1. Интервьюирование заказчика
   2. Анкетирование конечных пользователей
   3. Описание заинтересованных лиц и конечных пользователей
   4. Словарь предметной области
   5. Анализ бизнес-требований (в виде схемы бизнес-целей, бизнес-требований и функций системы)
2. Провести сравнительный анализ существующих аналогичных решений на рынке
3. Разработать концепцию решения, включая:
   1. Описание общей функциональности решения
   2. Детализация требований к решению (техническое задание)
   3. Описание границ решения
   4. Описание возможных архитектурных и технических решений
4. Спроектировать разрабатываемую систему, включая:
   1. Проектирование архитектуры/структуры системы
   2. Проектирование базы данных
   3. Прототипирование интерфейса пользователя
5. Выполнить программную реализацию первой версии системы (продукта)
6. Выполнить функциональное тестирование системы по тестовым сценариям

# Бизнес-анализ

В соответсвии с учебным планом, в рамках курсовой работы, была поставлена задача разработать веб-сервис для студентов, который бы позволял обмениваться сообщениями, домашними заданиями и помогал бы им в учебе, а также позволял преподавателям контролировать выполнение домашних заданий и отвечать на вопросы студентов.

В настоящее время студенты и преподаватели для указанных целей пользуются сторонними ресурсами, такими как WatsApp, Telegramm и пр., а также традиционные sms. Однако, гораздо удобнее иметь в распоряжении свой собственный сервис для обмена сообщениями.

Сегодня на рынке присутствует огромное количество мессенджеров и средств для связи. На первый взгляд, они решают одинаковые задачи. Анализ конкурентов разрабатываемого продукта будет проведен в следующих разделах данной курсовой работы.

Немаловажным фактором является наличие нишевой аудитории – это все студенты и преподаватели нашего учебного заведения, и в перспективе распространение продукта среди всех университетов страны.

## Интервьюирование целевой аудитории

Приступая к анализу наиболее популярных мессенджеров, прежде всего определимся, какая нишевая аудитория соответствует задачам, которые эти мессенджеры могут решить, а также какие цели у создателей мессенджера и пользователя.

Как правило, основной целевой аудиторией является молодежь и люди среднего возраста, примерно 15 – 45 лет разных профессий, в том числе студенты и преподаватели.

Как видно, круг пользователей довольно широк, поэтому в процессе разработки необходимо предусмотреть возможность поддержки большого количества пользователей.

Для опроса целевой аудитории были подготовлены следующие вопросы:

1. Ваше имя.
2. Ваш возраст.
3. Вы студент/преподаватель?
4. Ваши профессиональные интересы.
5. Ваши увлечения.
6. Какие у вас основные цели?
7. С какими проблемами вы сталкиваетесь в достижении основных целей?
8. Что вы цените больше всего?

Перечисленные выше вопросы были использованы в составлении анкет в Google-forms и проведения анкетирования пользователей.

Результаты анкетирования представлены в следующем разделе.

## Анкетирование пользователей

Ниже представлены результаты анкетирования пользователей, полученные при помощи Google-forms.

Александр, 19 лет, студент.

**Профессия**: учится на дизайнера.

**Цели:**

- хочет расширить свой профессиональный кругозор, освоить профессию дизайнера;

- научные интересы, учеба, посещение научно-популярных лекций;

- пользоваться для учебы специальным мессенджером.

**Проблемы**:

- переживает, что не сможет вовремя получать качественную обратную связь от преподавателя;

- медленная работа имеющихся мессенджеров.

**Интересы**:

- посещение выставок, лекций;

- просмотр вебинаров;

- активное использование социальных сетей.

**Ценит**:

- свое время;

- общение с одногруппниками и сверстниками.

Николай, 23года, выпускник.

**Профессия**: программист.

**Цели:**

- во время учебы занимался версткой университетского сайта, поэтому решил попробовать себя в веб-дизайне;

- получать крепкий фундамент знаний по новой профессии;

- работать над проектом и быть всегда на связи с коллегами.

**Проблемы**:

- не всегда получаетсяиметь качественную обратную связь от преподавателя и сокурсников;

- перегруженный функционал.

**Интересы**:

- рисование;

- футбол.

**Ценит**:

- свое время;

Мария, 33 года, домохозяйка.

**Профессия**: окончила университет по специальности «Маркетинг», замужем, имеет двоих детей.

**Цели:**

- хочет освоить SMM и подрабатывать удаленно на дому;

- есть свободное время, поэтому планирует поступать на заочное отделение.

**Проблемы**:

- свободного времени не хватает, т.к. имеются маленькие дети;

- надеется, что современные средства связи позволят учиться дома.

**Интересы**:

- реклама, интернет-маркетинг;

- продвижение товаров в социальных сетях.

**Ценит**:

- своих детей;

- заработок через интернет.

Анкетирование целевой аудитории показало потребность пользователей в оперативной и качественной связи.

## Словарь предметной области

Для того, чтобы иметь возможность общаться с заказчиком на одном языке, составим словарь предметной области. Данный словарь послужит основой для userstory и объектов разрабатываемой системы.

Таблица 1.1 – Словарь предметной области

|  |  |  |  |
| --- | --- | --- | --- |
| № | Термин или понятие | Описание понятия  или расшифровка термина | Аббревиатура |
| 1 | USER | Обычный пользователь без права администратора |  |
| 2 | ADMIN | Пользователь с правами администратора |  |
| 3 | Регистрация | Создание нового пользователя в системе, предусматривает наличие уникального имени и пароля |  |
| 4 | Авторизация | Вход зарегистрированного пользователя в систему, предусматривает поиск введенных имени и пароля в базе данных |  |
| 5 | Клиент | Устройство для входа в систему, имеющее графический интерфейс |  |
| 6 | Подписка | Пользователь, все посты которого доступны в первую очередь тем пользователям, которые на него подписаны. |  |

## Описание текущей ситуации (как было)

В настоящее время студенты и преподаватели для общения друг с другом и обмена домашними заданиями пользуются сторонними ресурсами, такими как WhatsApp , Telegram и пр., а также традиционными sms-технологиями. Однако, гораздо удобнее иметь в распоряжении свой собственный сервис для обмена сообщениями.

Анкетирование пользователей выявило, что студенты часто не могут оперативно связаться с преподавателем чтобы задать ему вопросы по домашнему заданию или сообщить, например, что не успевают выполнить д/з вовремя.

Преподавателей гораздо меньше чем студентов, поэтому чатов у преподавателей больше. Часто используемые средства связи (WatsApp, Telegramm, sms) имеют неудобную навигацию между чатами, из-за чего некоторые сообщения могут затеряться, а вопросы студентов – остаться без ответа.

Представим описание текущей ситуации в виде таблицы 1.2.

Таблица 1.2 – Описание текущей ситуации

|  |  |
| --- | --- |
| Проблема | Студенты часто не имеют возможности связаться с преподавателем оперативно. Из-за большого количества студентов, преподаватель может не заметить или не успеть ответить на вопросы. |
| Воздействует на | Студенты и преподаватели |
| Результатом чего является | Вопросы от студентов по д/з долго остаются без ответа, что отрицательно сказывается на успеваемости. |
| Сейчас решается | Преподаватели и студенты для обмена информацией пользуются сторонними ресурсами, такими как WatsApp, Telegramm и пр., а также традиционными sms-технологиями. |
| Имеющиеся решения не | Часто используемые средства связи (WatsApp, Telegramm, sms) имеют неудобную навигацию между чатами, из-за чего некоторые сообщения могут затеряться, а вопросы студентов – остаться без ответа. |

## Описание будущей ситуации (как будет)

Разрабатываемый веб-сервис для публикаци заметок и статей и обмена сообщениями для студентов и преподавателей позволит оперативно обмениваться домашними заданиями, решениями, сообщениями. Позволит студентам вовремя узнавать об изменениях в д/з или расписании занятий. Преподаватели смогут быстро отвечать на вопросы студентов по теме занятия или по домашнему заданию.

Возможность отслеживать процесс учебы и работы студентов над проектами. Поддержка и отображение прикрепляемого контента в чате. Постановка задач для решения в рамках домашних заданий.

## Бизнес-требования

Основные цели создания и внедрения разрабатываемого веб-сервиса:

- оптимально быстрый и комфортный сервис общения;

- привлечение в перспективе большого количества пользователей за счет сетевого эффекта;

- ведение пабликов с вопросами и статьями для улучшения качества общения и повышения социализации.

Основные цели, которые преследует пользователь при выборе средства связи:

- удобный обмен информацией с другими пользователями (смс чат, видео чат, обмен ссылками, фотографиями, видео и файлами различных форматов);

- быстрый способ доступа к одногруппникам и преподавателям;

- удобный инструмент для деловой переписки (ведения рабочих, учебных проектов).

Существующие решения могут удовлетворить указанные потребности, но только частично. Основные достоинства и недостатки существующих мессенджеров и средств связи перечислены в следующем разделе.

Как видно из перечисленных выше целей создателей и пользователей, мессенджер – это незаменимая для современных людей программа, которая позволяет общаться друг с другом при помощи сообщений и не только.

# Анализ существующих решений

Для того, чтобы создать хороший и удобный сервис для связи студентов и преподавателей, который будет нравиться пользователям, необходимо проанализировать аналогичные продукты конкурентов.

## Обзор существующих решений

### Решение 1. «Telegram»

Один из самых популярных сегодня мессенджеров – Telegram. Перечислим его достоинства [2]:

* - удобный интерфейс;
* - возможность создания собственной темы;
* - добавление умных ботов в чаты;
* - секретный чат;
* - групповые чаты;
* - открытый исходный код;
* - интегрированный редактор фотографий;
* - протокол шифрования MTProto;

В качестве недостатка Telegram следует отметить отсутствие возможности видеозвонка.

### Решение 2. «WhatsApp»

Еще один популярный мессенджер – WhatsApp. Перечислимегопреимущества:

* протокол HTTPS;
* оффлайн кэширование сообщения;
* анализ данных пользователя;
* возможность создавать групповые чаты;
* голосовые вызовы;
* не требует регистрации.

Заметное искажение качества передаваемых через WhatsApp медиафайлов может омрачить впечатление от сервиса или даже побудить от него отказаться. Кроме того, WhatsApp не сообщает пользователю об отсутствии сети, из-за чего пользователь может отправить сообщения в кэш, не зная об этом. Но самым существенным недостатком WhatsApp является то, что он не работает без сматрфона. Десктоп приложение WhatsApp не является полноценным приложением, а используется как дополнение к мобильному приложению.

### Решение 3. «Viber»

В последнее время набирает популярность новый сервис сообщений – Viber. Егоположительныестороны:

* огромный выбор эмоджи и стикеров;
* уникальная игровая платформа;
* аудио и видео звонки;
* возможность создавать групповые чаты;
* работа уведомлений в оффлайн режиме;

Главным недостатком Viber является большое количество спама и рекламы.

### Решение 4. «Skype»

Уже много лет остается достаточно популярным Skype. Основнымиегодостоинствамиявляются:

* возможность создания групповых чатов;
* возможность проведения видеоконференций;
* возможность добавления ботов;
* простой и понятный интерфейс.

К недостаткам можно отнести следующее:

* - отсутствие шифрования;
* - плохая синхронизация с различными устройствами;
* требует регистрации пользователя.

### Решение 5. «FaceBook Messenger»

Отметим также достаточно популярный на Западе FaceBookMessenger. Егодостоинстваследующие:

* простой интерфейс;
* добавление умных ботов в чаты;
* отчеты о прочтении сообщений;
* возможность осуществлять электронные платежи;
* не требует регистрации.

Существенным недостатком FaceBookMessenger является ограничение удаления сообщений: сообщения можно удалять только на своем устройстве.

### Решение 6. «HangOuts»

Также следует упомянуть HangOuts, который является не только сервисом для отправки сообщений, но также может быть использован как платформа для проведения вебинаров. Егоплюсы:

* групповые чаты;
* синхронизация с Google сервисами;
* хранение на сервере истории.

Егонедостатки:

* - не очень удобный интерфейс;
* - не большое количество пользователей;
* снижает качество передаваемых снимков.

### Решение 7. «iMessage»

Упомянем также iMessage , который интегрирован в каждое устройство Apple. Его основным достоинством является безопасный режим для отправки сообщений. Однако он имеет ряд существенных недостатков:

* - привязывает номер телефона к учетной записи ID;
* - доступен исключительно на устройствах компании Apple;
* - если iMessage не смог добавить сообщение, то оно отправится как СМС без шифрования.

### Решение 8. «SMS»

SMS технология, в сравнении со специальными сервисами для отправки сообщений, является несколько устаревшей, хотя и остается достаточно популярной среди пользователей мобильных устройств. К достоинствам можно отнести возможность отправки сообщения с простого мобильного телефона (не смартфона) и в отсутствие подключения к интернету. Перечислим некоторые недостатки данной технологии:

* - высокая стоимость отправки одного сообщения;
* - отправка медиа файлов занимает много времени и дорого;
* - отсутствует возможность создания групповых чатов;
* - доступна только на мобильных телефонах.

## Сравнительный анализ решений

В таблице 2.1 наглядно представлено сравнение функциональных возможностей наиболее популярных в настоящее время в России мессенджеров и программ для связи.

Таблица 2.1- Сравнение функционала популярных мессенджеров

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Функция** | **WatsApp** | **Telegram** | **Skype** | **MailAgent** | **icq** | **imo** | **Viber** |
| Отправка/получение сообщений через интернет | + | + | + | + | + | + | + |
| Связь с группойлюдей | + | + | + | + | + | + | + |
| Обменфото/видео/файлами | + | + | + | - |  | + | + |
| Кросплатформенность | + | + | + | + | + | + | - |
| Синхронизация сообщений со всеми устройствами | + | + | + | + | + | + | - |
| Аудиозвонки | + | + | + | - |  | + | + |
| Видеозвонки | + |  | + | - | + | + | + |
| Передачаголосовогосообщения | + | + | - | - |  | + |  |
| Шифрование | сквозное | MTProto | - | - | + | - | End-to-end |
| Обменместоположениями | - | + | - | - | - | - | + |
| Демонстрацияэкрана | - | - | + | - | - | - | - |

## Выводы по анализу

Kак видно, большинство приложений для отправки сообщений имеет похожий функционал. Некоторые из них имеют уникальные функции, которые являются специфическими, как например демонстрация экрана в Skype. Как показывает анализ, наиболее важными функциями являются отправка/прием сообщений, обмен данными, аудио/видео звонки, чаты. Поскольку, современные пользователи одновременно пользуются различными устройствами, то кроссплатформенность и возможность синхронизации сообщений на всех устройствах также является необходимым преимуществом.

# Концепция решения

Концепция решения описана техническим заданием ниже. Техническое задание показывает понимание потребностей пользователей и заинтересованных лиц.

Техническое задание описывает будущие возможности и функционал разрабатываемой системы.

## Техническое задание

Данное техническое задание является основным документом разрабатываемого веб сервиса. Разработка ведется строго в соответствии с техническим заданием.

### Назначение и цели создания системы

Разрабатываемая система предназначена для обеспечения оперативной связи между студентами и преподавателями для обмена домашними заданиями, решениями, вопросами и ответами.

Конечной целью внедрения разрабатываемой системы является повышение эффективности учебного процесса.

### Требования к системе:

#### Требования к структуре и функционированию системы

1. Сервер должен быть реализован на Java.
2. Базой данных выступает MySql.
3. 8
4. Инстанс сервера должен иметь возможность взаимодействовать с другими своими инстансами. Например, если *пользователь А* хочет отправить сообщение *пользователю Б*, клиент (или клиенты) которого сейчас подключен к другому инстансу (к другим инстансу), все клиенты *пользователя Б* должны получить это сообщение мгновенно.

#### Требования к численности и квалификации пользователей

1. Численность пользователей может быть не ограничена.
2. Каждый пользователь может подключиться с нескольких клиентов одновременно.
3. Клиентом в терминологии ТЗ называется подключенное к WS соединению устройство.
4. Все клиенты пользователя должны моментально получать новые сообщения. Так же, если один из клиентов написал сообщение, остальные клиенты должны получить информацию об этом.
5. Создатель группы может добавлять и удалять участников из группы. Так же он может создавать уникальную ссылку, перейдя по которой, другой пользователь подключиться к группе. Такая ссылка действует неограниченное количество раз, но создатель группы может ее отключить.
6. Другие участники могут только добавлять тех, кто есть у них в списке чатов.

#### Требования к дизайну и реализации

Перечислим факторы, которые ограничат возможности, доступные разработчикам:

* Следует использовать язык программирования Java, для создания клиент-серверной архитектуры следует использоватьSpring;
* Для обеспечения безопасного соединения использовать SpringSecurity;
* В качестве базы данных использовать MySql
* Для создания удобного пользовательского интерфейса использовать Bootstrap.
* Клиентом может быть любойWeb-браузер;
* формат обмена данными: текстовый и файловый.

#### Требования к защите информации, безопасности и сохранности данных

Для обеспечения безопасности хранения паролей пользователей в базе данных использовать шифрование MD5 и движок flyway-core.

Для безопасного входа в систему использовать валидацию полей авторизации и регистрации при помощи spring-boot-starter-security.

Хранение передаваемых сообщений осуществляется в базе данных MySql, для обеспечения сохранности данных доступ к базе данных должен быть ограничен правами администратора.

#### Требования к внешнему интерфейсу

Внешний интерфейс должен обязательно иметь окно регистрации и авторизации пользователя.

* На главной странице располагается список всех сообщений.
* На отдельной странице располагается список сообщений пользователя.
* На каждой странице имеется навигация.
* Окно с профилем пользователя включает возможность смены пароля пользователем. При этом новый пароль перезаписывается в базу данных с шифрованием MD5.

Доступ к списку пользователей и возможность редактирования роли пользователя может быть только у администратора.

## Анализ факторов риска проекта

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Тип (категория риска) | Риск | Последствия | Меры по предотвращению | Меры по минимизации последствий | Вероятность свершения риска [P] | Степень ущерба [L] | Влияние (фактор риска)  [R] = [P] x [L] |
| Проектный (связанный с требованиями)  Технический (технологический) | Перехват паролей  Блокировка приложения Роскомнадзором | Утечка данных сервера  Не доступность приложение всем пользователям | Использование протокола шифрования и https соединения, аутентификация  Использование системы хранения сообщений | Шифрование сообщений  Хранение сообщений на стороне клиента | 5  8 | 6  7 | 30  56 |

# 

# Проектирование решения

## Описание архитектуры/структуры системы

Разрабатываемая система относится к распределенным многоуровневым сетевым системам. Рассмотрим архитектуру и принцип работы с точки зрения программирования сетевого приложения.

При создании мессенджера так или иначе придется иметь дело с сетью. Знание сетевых технологий, понимание работы стека протоколов TCP/IP, клиент серверной архитектуры и протоколов прикладного уровня – это основа для разработки сетевых приложений.

Разрабатывая приложение, архитектор или программист всегда должен ответить на ряд вопросов, от которых зависит то, как будут использованы сетевые сервисы:

* масштабирование приложения;
* производительность приложения;
* безопасность приложения.

Все перечисленные вопросы в том числе зависят и от архитектуры сети и используемых сетевых протоколов.

На данный момент используются следующие сетевые стандарты:пользовательские протоколоы (HTTP, SMTP, XMPP), а также большая часть протоколов, необходимых для нормальной работы системы в сети (DHCP, DNS, NTP). Прикладные протоколы реализуются за счёт программ, работающих в пользовательском пространстве. Для доступа к стеку TCP/IP они запрашивают сокет через интерфейсы ядра операционной системы.

Если используется надёжное соединение, то применяется SSL (SocketSecureLayer), фактически TLS (TransportLayerSecurity). В таком случае прикладной протокол инкапсулируется в TLS сессию, которая уже использует открытый сокет. TLS, который можно отнести к сеансовому уровню модели OSI/ISO, используется в пользовательском пространстве.

К транспортному уровню относятся STREAM-протоколы, использующие STREAM-сокеты TCP и SCTP. Принадлежа к транспортному уровню, они обладают определенными чертами сеансового уровня. Данные STREAM-протоколов называются сегментами: например, TCP-сегмент. Один сегмент несёт фрагмент установленной сессии.

Кроме того, к транспортному уровню относятся DATAGRAM-протоколы, использующие DATAGRAM-сокеты: UDP и DCCP. Данные в этих протоколах называются дейтаграммами. Такие протоколы не осуществляют контроля доставки и порядка получения сообщений.

Транспортные протоколы позволяют идентифицировать приложения на хосте. Для этого используется двухбайтное пространство STREAM-портов (обычно называемых TCP-портами, даже если используется SCTP) и не связанное с ним пространство DATAGRAM-портов (обычно называемых UDP-портами, даже если используется DCCP). Именно поэтому в стеке протоколов TCP/IP транспортный уровень осуществляет механизм межпроцессного взаимодействия. Программы пользовательского окружения лишь запрашивают доступ к сокету (аналогично доступу к файлам), в то время как механизм сборки пакетов и контроля сессий реализуется на уровне ядра.

## Описание возможных технических решений

Back-end разрабатываемойсистемыреализованна Java, сиспользованиемфреймворков Spring MVC, Spring Boot, Spring Security.

Springframework в настоящее время является самым популярным фреймворком для разработки сетевых приложений.

## Проектирование интерфейса пользователя

Интерфейс пользователя в разрабатываемой системе реализован при помощи Bootstrap, который предоставляет готовые решения для блоков html разметки и css стилей

## Проектирование базы данных

ERдиаграмма базы данных показана на рис.4.1.

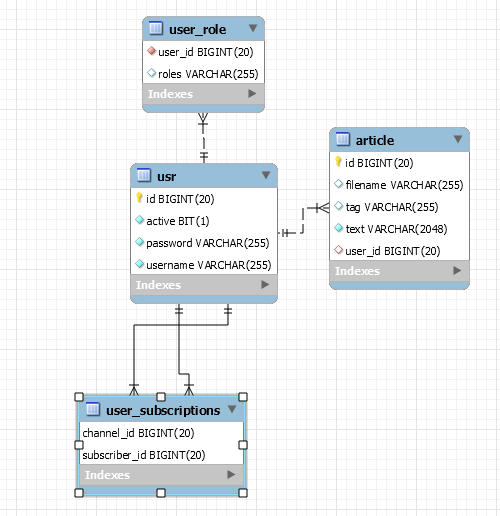


Рис.4.1 – ER-диаграмма веб сервиса для отправки сообщений

# Программная реализация системы

Начнем создание приложения, используя maven и SpringMVC. Необходимые зависимости можно скопировать с официального сайта spring.io, там же есть гайды по работе со Spring.

Подключаем parent зависимость от Springboot – основная обертка фреймворка. Основные зависимости проекта:starter-web, который позволяет веб использовать, devtools– инструмент, который позволяет быстро перезагружать приложение.

Версию Java используем 1.8.

В соответствии с mvc шаблоном создаем папку controller и в ней класс MainController, который представляет собой программный модуль, который по указанному пути слушает запросы от пользователя и возвращает данные в соответствии с бизнес логикой.

Для того, чтобы запускать приложение, необходимо создать точку входа в программу. Для этого в корневом пакете проекта создадим класс Main. Код файла запуска выглядит следующим образом:

**import** org.springframework.boot.SpringApplication;  
**import** org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
**public class** Application {  
  
**public static void** main(String[] args) {  
SpringApplication.*run*(Application.**class**, args);  
 }  
  
}

Для того, чтобы иметь возможность работать с базой данных, добавим зависимости jpa и mysql-connector. Также необходимо добавить некоторые конфиги для подключения к СУБД. Для этого создадим в папке resources файл application.properties и в нем пропишем следующие настройки:

spring.jpa.hibernate.ddl-auto=create

spring.datasource.url=jdbc:mysql://localhost:3306/db\_example

spring.datasource.username=springuser

spring.datasource.password=ThePassword

Для того, чтобы создать таблицу в базе данных, создадим новую сущность. Для этого в папке domain создаем новый класс, назовем его Article. Класс помечается аннотацией @Entity специально для того, чтобы указать Spring, что это не просто класс-сущность, а таблица базы данных. Следующий фрагмент кода описывает идентификатор, который служит для того, чтобы различать записи в таблице:

@Id  
@GeneratedValue(strategy= GenerationType.***AUTO***)  
**private** Long **id**;

Также в данном классе-сущности будем хранить приватные строковые поля с текстом сообщения и тегом:

@NotBlank(message = **"Field cannot empty"**)  
@Length(max = 2048, message = **"Too long"**)  
**private** String **text**;  
@Length(max = 255, message = **"Too long"**)  
**private** String **tag**;

Т.к. поля приватные, необходимо сгенерировать аксессоры – геттеры и сеттеры.

Далее в папке repos создадим CRUD репозиторий, который позволяет получать список полей, либо найти сообщение по идентификатору, а также сохранять в него данные. Кроме того, создадим в нашем репозитории метод, позволяющий находить сообщения п тегу.

**import** com.student.crecer.domain.Article;  
**import** org.springframework.data.repository.CrudRepository;  
  
**import** java.util.List;  
  
**public interface** ArticleRepo**extends** CrudRepository<Article, Long> {  
  
 List<Article>findByTag(String tag);  
}

Теперь нужно создать view для только что созданной сущности. Предварительно подключим шаблонизаторfreemarker в maven зависимости.Далее попути resources/templatesсоз даем файл шаблона. Также в классе-контроллере пропишем методы добавления сообщений и поиска сообщений по тегам:

@GetMapping(**"/main"**)  
**public** String main(@RequestParam(required = **false**, defaultValue = **""**) String filter, Model model) {  
Iterable<Article> articles;  
  
**if** (filter != **null** && !filter.isEmpty()) {  
 articles = **articleRepo**.findByTag(filter);  
 } **else** {  
 articles = **articleRepo**.findAll();  
 }  
  
model.addAttribute(**"articles"**, articles);  
model.addAttribute(**"filter"**, filter);  
**return "main"**;  
}  
  
@PostMapping(**"/main"**)  
**public** String add(  
@AuthenticationPrincipalUser user,  
@Valid Article article,  
BindingResultbindingResult,  
 Model model,  
@RequestParam(**"file"**) MultipartFile file  
) **throws** IOException {  
article.setAvtor(user);  
  
**if** (bindingResult.hasErrors()) {  
 Map<String,String> errors = ControllerUtils.*getErrors*(bindingResult);  
model.mergeAttributes(errors);  
model.addAttribute(**"article"**, article);  
 } **else** {  
saveFile(article, file);  
  
model.addAttribute(**"article"**, **null**);  
**articleRepo**.save(article);  
 }  
Iterable<Article> articles = **articleRepo**.findAll();  
model.addAttribute(**"articles"**, articles);  
  
**return "main"**;  
}

Полный код проекта представлен в приложении к данной курсовой работе.

На рис.5.1 представлена UML-диаграмма на уровне пакетов. На диаграмме рис.5.1 показаны зависимости между классами из различных пакетов проекта.

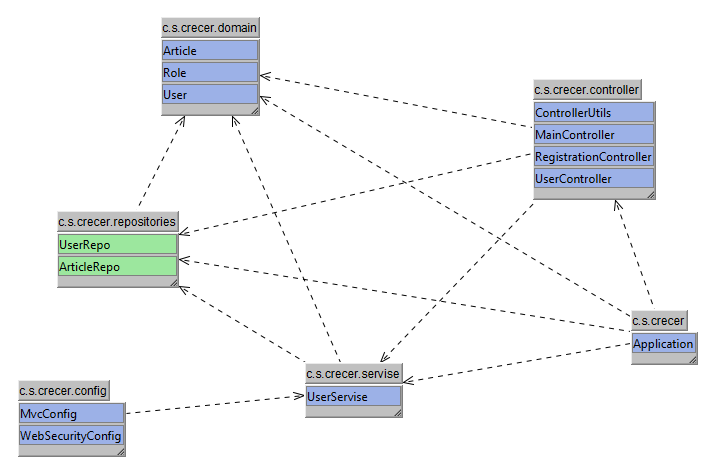


Рис.5.1 - UML-диаграмма на уровне пакетов

На рис.5.2 представлена полная UML-диаграмма на уровне классов, показывающая все поля и методы всех классов приложения и связи между классами.

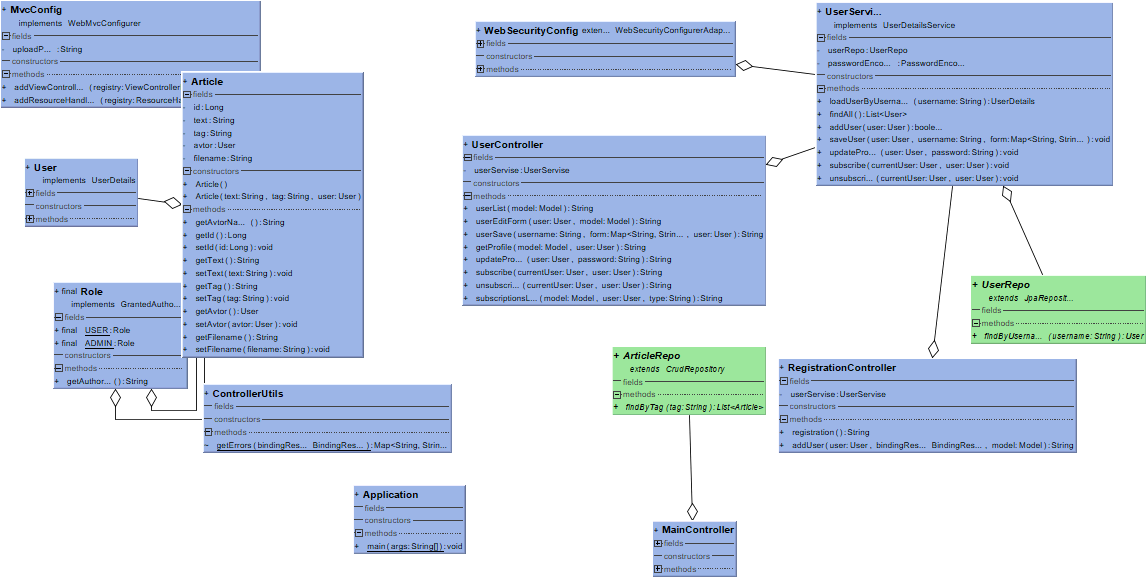


Рис.5.2 - UML-диаграмма на уровне классов

# Функциональное тестирование решения

Произведем функциональное тестирование разрабатываемой системы. В рамках тестирование динамически проверим поведение программы на соответствие ожидаемому.

## Тест «Регистрация»

|  |  |
| --- | --- |
| Идентификатор тест-варианта | 1 |
| Набор входных данных | Имя пользователя, пароль, подтверждение пароля |
| Ожидаемые результаты | Вход в систему, добавление пользователя в базу данных, шифрование пароля. |
| Выполняемые действия | В соответствующие поля вводятся имя пользователя, пароль и подтверждение пароля. Если пользователь с таким именем уже существует, система выводит об этом сообщение. Имя пользователя должно быть уникальным. Если хотя бы одно поле не заполнено или подтверждение пароля и пароль не совпадают, выводится сообщение валидации. |



Рис.6.1 – Окно входа

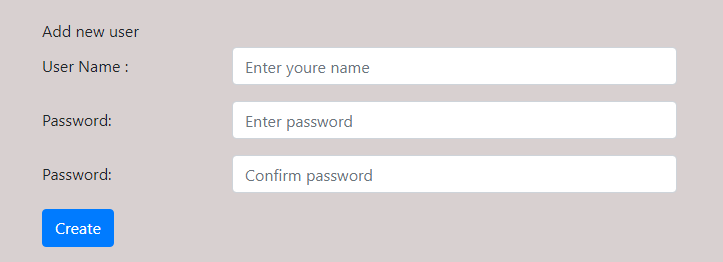


Рис.6.2 – Окно регистрации

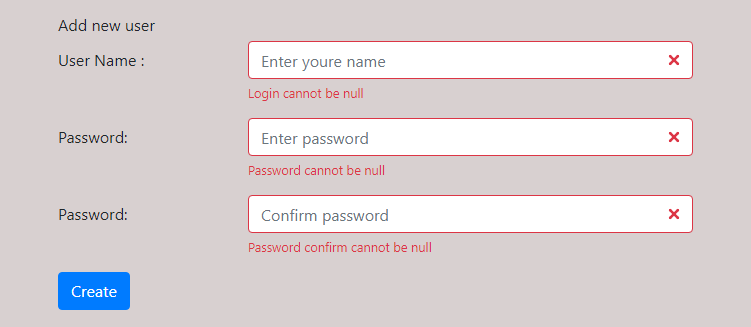


Рис.6.3 – Попытка зарегистрироваться когда все поля пустые

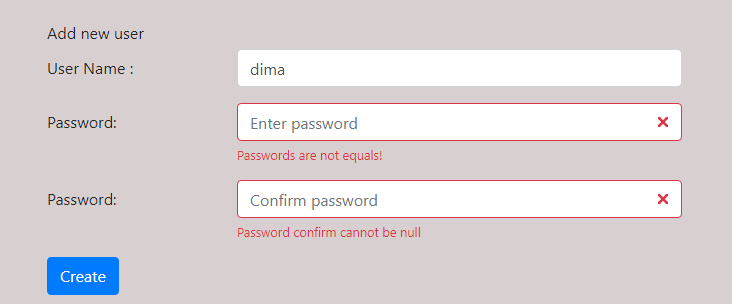


Рис.6.4 – Попытка зарегистрироваться когда одно поле заполнено

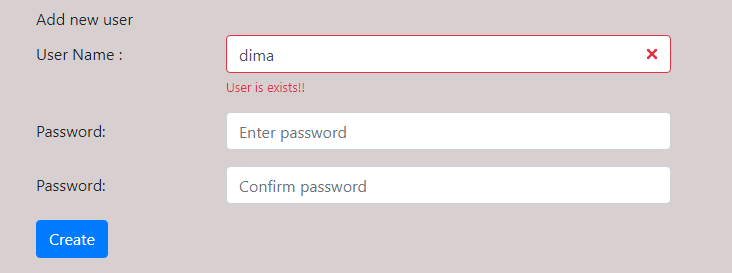


Рис.6.5 – Попытка зарегистрироваться с уже существующим в базе данных пользователем

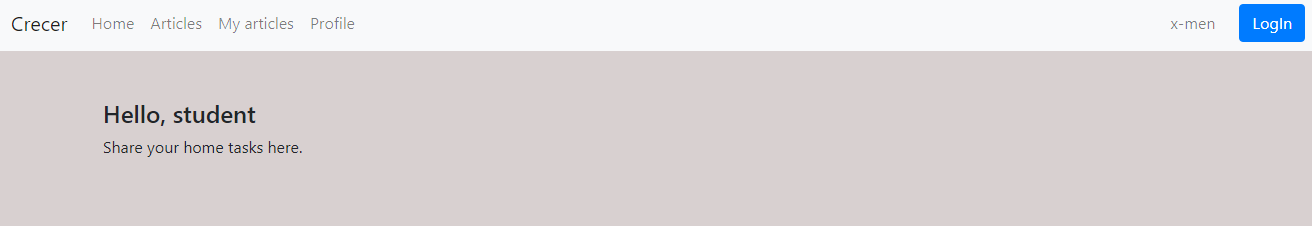


Рис.6.6 – Удачная попытка регистрации нового пользователя

## Тест «Авторизация»

|  |  |
| --- | --- |
| Идентификатор тест-варианта | 2 |
| Набор входных данных | Имя пользователя, пароль |
| Ожидаемые результаты | Вход в систему, доступ к остальным функциям системы |
| Выполняемые действия | В соответствующие поля вводятся имя пользователя и пароль. Если хотя бы одно поле не заполнено или имя пользователя не существует в базе данных или пароль не верный, выводится сообщение валидации. |

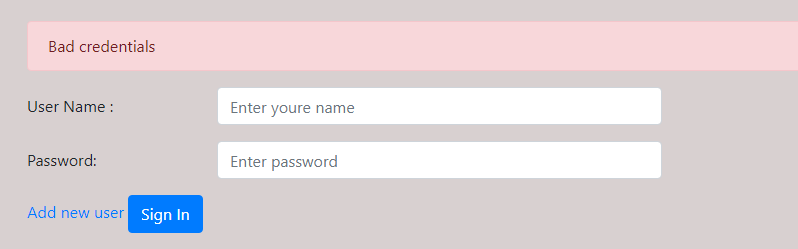


Рис.6.7 – Попытка авторизации с пустыми полями



Рис.6.8 – Успешная авторизация пользователя

## Тест «Добавление сообщений»

|  |  |
| --- | --- |
| Идентификатор тест-варианта | 3 |
| Набор входных данных | Текст сообщения, тэг, файл |
| Ожидаемые результаты | Добавление нового сообщение в базу данных |
| Выполняемые действия | В соответствующие поля вводятся текст сообщения и тег. Для выбора файла появляется окно диалога. После выбора файла на странице появляется картинка. |

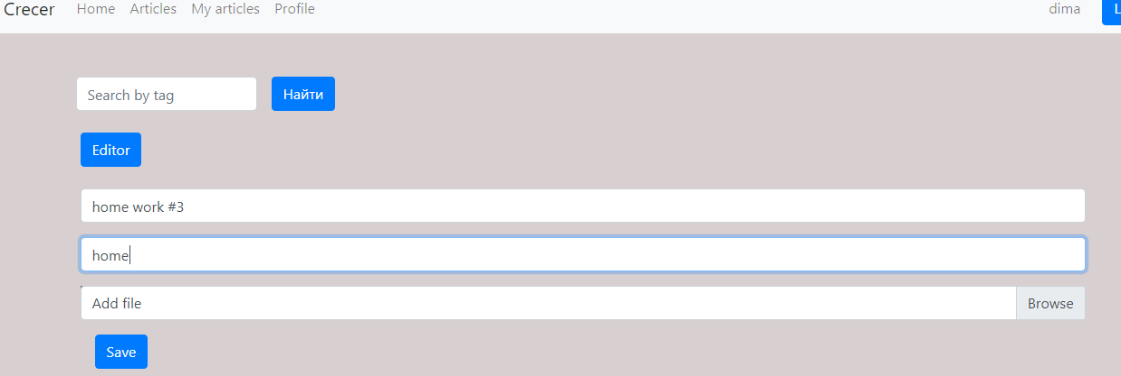


Рис.6.9 – Ввод текста сообщения

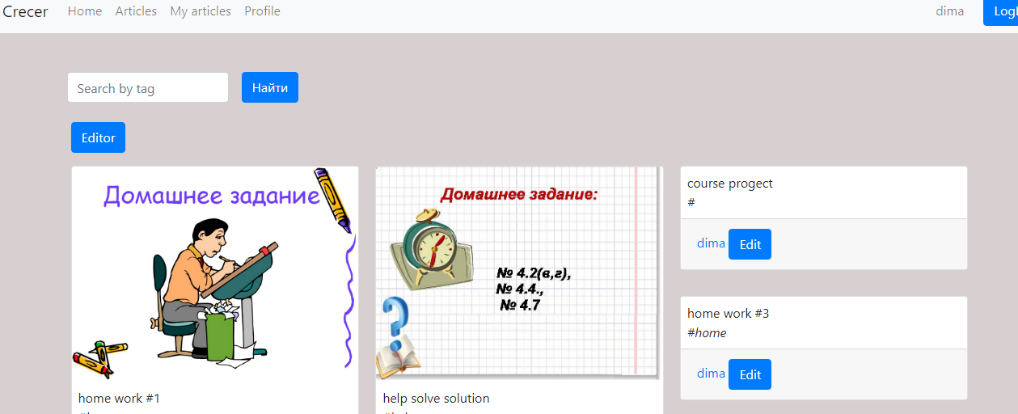


Рис.6.10 – Только что добавленное сообщение появилось в списке сообщений на главной странице

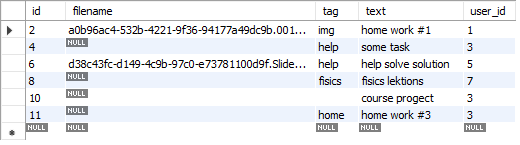


Рис.6.11 – Только что добавленное сообщение появилось в базе данных

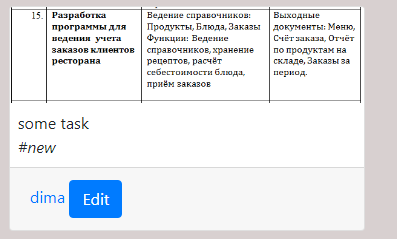


Рис.6.12 – Публикация файла

## Тест «Редактирование сообщений»

|  |  |
| --- | --- |
| Идентификатор тест-варианта | 4 |
| Набор входных данных | Текст сообщения, тэг, файл |
| Ожидаемые результаты | Обновление сообщения в базе данных |
| Выполняемые действия | Кнопка Edit доступна только автору сообщения. В соответствующие поля вводятся текст сообщения и тег. Для выбора файла появляется окно диалога. |

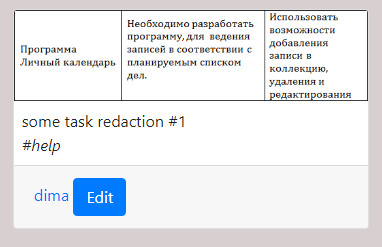


Рис.6.13 – Отредактированное сообщение

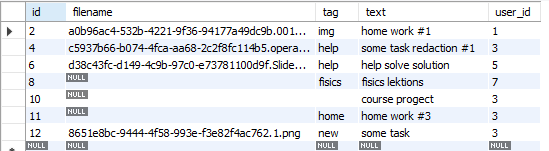


Рис.6.14 – Отредактированное сообщение обновлено в базе данных

## Тест «Подписки и подписчики»

|  |  |
| --- | --- |
| Идентификатор тест-варианта | 5 |
| Набор входных данных | Имя пользователя и пароль |
| Ожидаемые результаты | Пользователь может зайти на страницу другого пользователя, увидеть все его посты и подписаться/отписаться, нажав соответствующую кнопку. |
| Выполняемые действия | На странице Myarticlesотображаютсявсе статьи авторизованного пользователя и количество его подписок и подписчиков. |

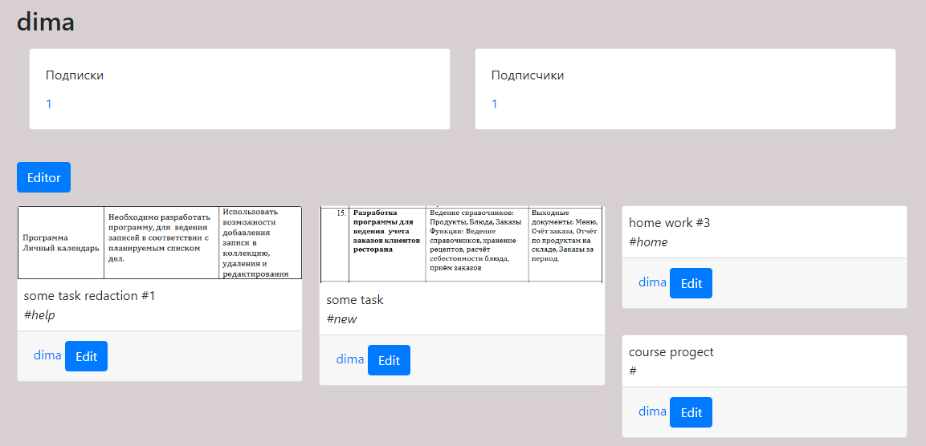


Рис.6.15 – Отображение всех постов авторизованного пользователя и его подписки и подписчики

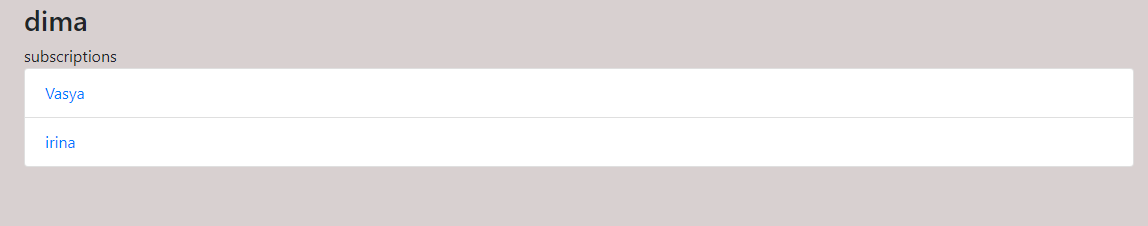


Рис.6.16 – Список подписок

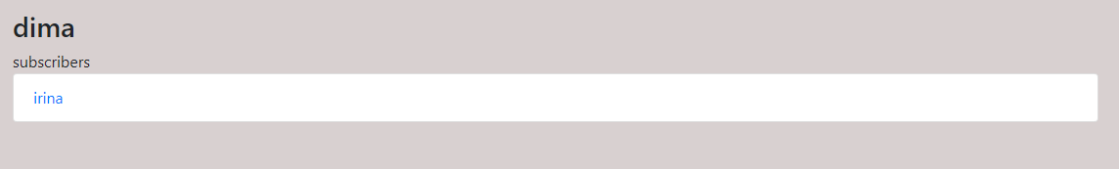
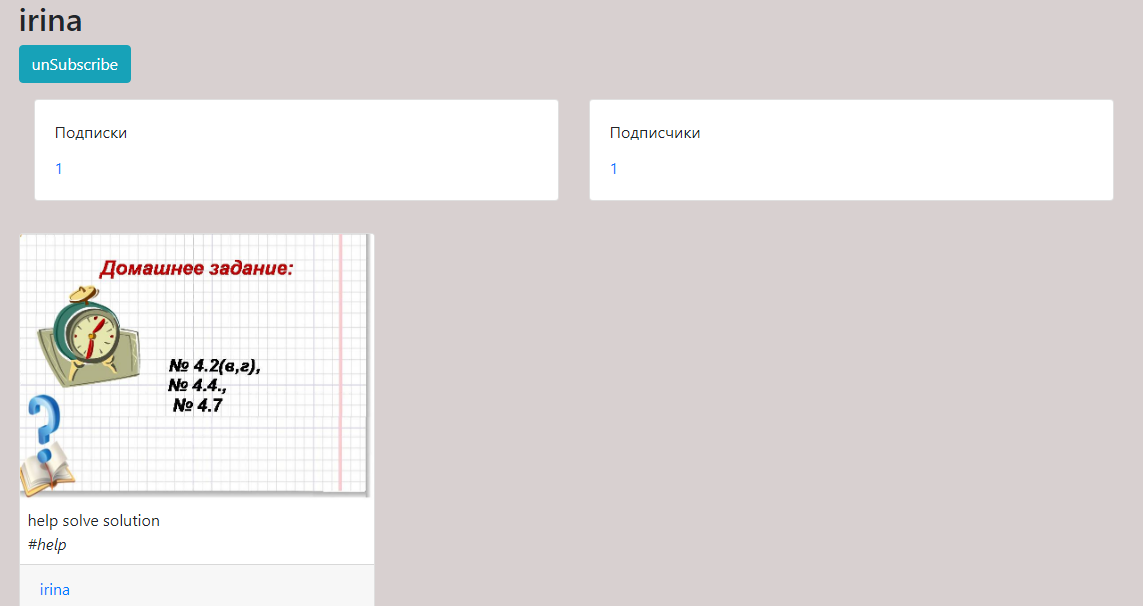


Рис.6.17 – Список подписчиков



6.18 – Канал, на который подписан авторизованный пользователь

## Тест «Администратор»

|  |  |
| --- | --- |
| Идентификатор тест-варианта | 6 |
| Набор входных данных | Имя администратора и пароль |
| Ожидаемые результаты | Администратор имеет возможность просматривать список всех зарегистрированных пользователей и добавлять выбранным пользователям роль администратора |
| Выполняемые действия | Войти под именем админа, просмотреть список пользователей и их роли |



Рис.6.19 – Вход под именем администратора

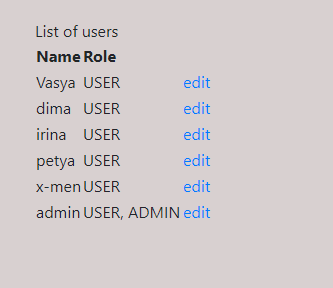


Рис.6.20 – Список всех зарегистрированных пользователей и их роли

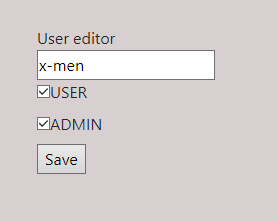


Рис.6.21 – Редактирование пользователя

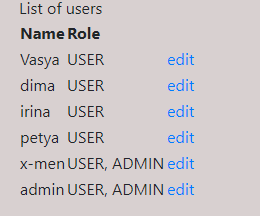


Рис.6.22 – Результат редактирования в списке пользователей

# Список литературы

1. **Карл Вигерс**Разработка требований к программному обеспечению.
2. **Дин Лэффингуэлл, Дон Уидриг** Принципы работы с требованиями к программному обеспечению. Унифицированный подход
3. **ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы**  
   <http://www.rugost.com/index.php?option=com_content&view=article&id=96:gost-34602-89&catid=22:34&Itemid=53>
4. **Техническое задание на сайт** <https://habrahabr.ru/post/138749/>
5. <http://artamonov.ru/2007/01/29/spring-v-ne-web-prilozheniyah/>
6. <https://tlgrm.ru>
7. <https://www.whatsapp.com>
8. <http://fb.ru/article/338879/rest-api---chto-eto-rest-perevod-representational-state-transfer>
9. <https://www.websocket.org/echo.html>