**Санкт-Петербургский политехнический университет Петра Великого**

**Институт компьютерных наук и технологий**

**Кафедра компьютерных систем и программных технологий**

**Курсовая работа**

**Реализация приложения «Sweater» с помощью «Spring Framework»**

по дисциплине «ПО распределённых вычислительных систем»

Работу выполнил Гагаркин И.Ю.  
студент группы 23541/3

Работу принял Стручков И.В.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018г.

Санкт-Петербург

2018

**Оглавление**

[**Анализ задания** 3](#_Toc8976454)

[**1.** **Формулировка индивидуального задания** 3](#_Toc8976455)

[**2.** **Требования, которым должно удовлетворять приложение** 3](#_Toc8976456)

[**3.** **Основные варианты использования** 4](#_Toc8976457)

[**Пользователь** 4](#_Toc8976458)

[**Модератор** 5](#_Toc8976459)

[**4.** **Описание модели предметной области** 7](#_Toc8976460)

[**Вовлеченные Сущности** 7](#_Toc8976461)

[**Реализация задания с помощью «Spring Framework»** 8](#_Toc8976462)

[**1.** **Объектно-ориентированное проектирование с учётом особенностей технологии** 8](#_Toc8976463)

[**2.** **Полный текст программы** 11](#_Toc8976464)

[**3.** **Методика и результаты тестирования** 12](#_Toc8976465)

[**4.** **Инструкция системному администратору по развёртыванию приложения** 13](#_Toc8976466)

[**Вывод** 13](#_Toc8976467)

[**Литература** 14](#_Toc8976468)

**Анализ задания**

1. **Формулировка индивидуального задания**

Реализовать приложение «Sweater» с помощью Spring Framework

1. **Требования, которым должно удовлетворять приложение**

Приложение должно строиться, используя следующие принципы и инструменты:

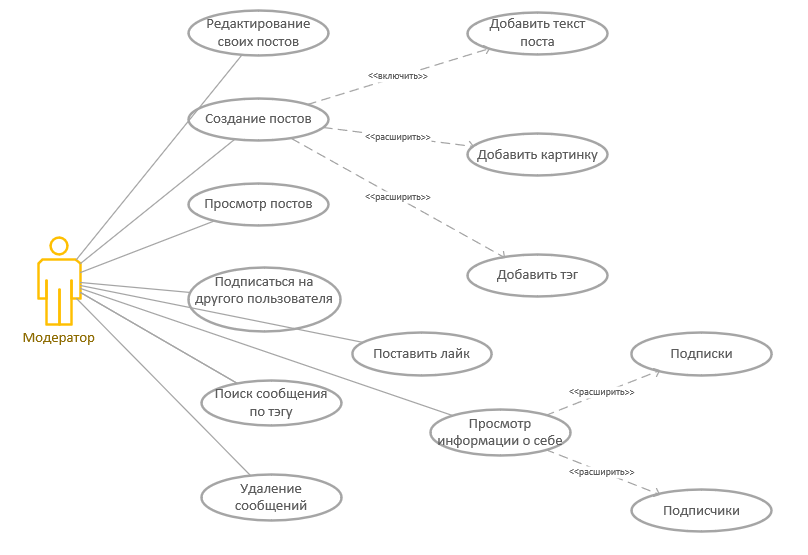
* Spring Fraemwork
* Rest
* Паттерны

1. **Основные варианты использования**

**Пользователь**

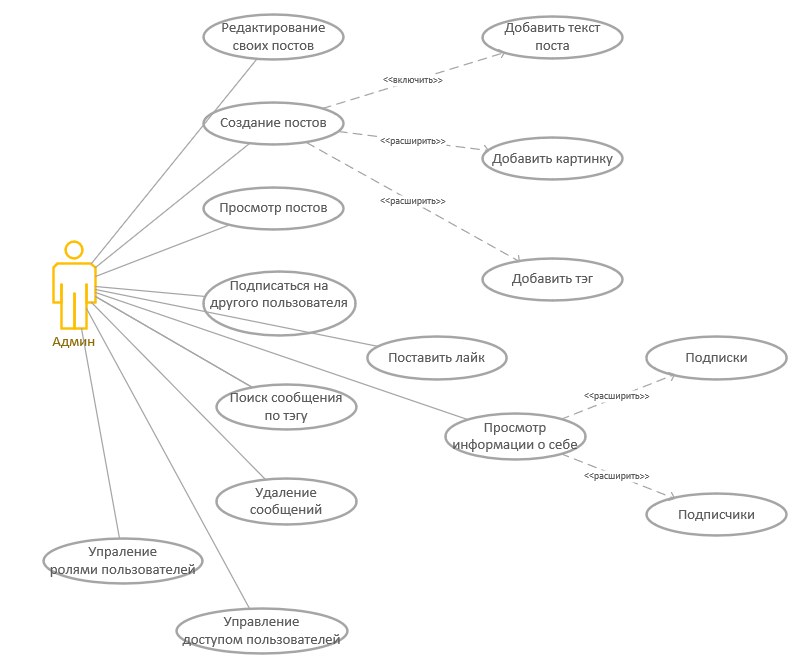


**Модератор**



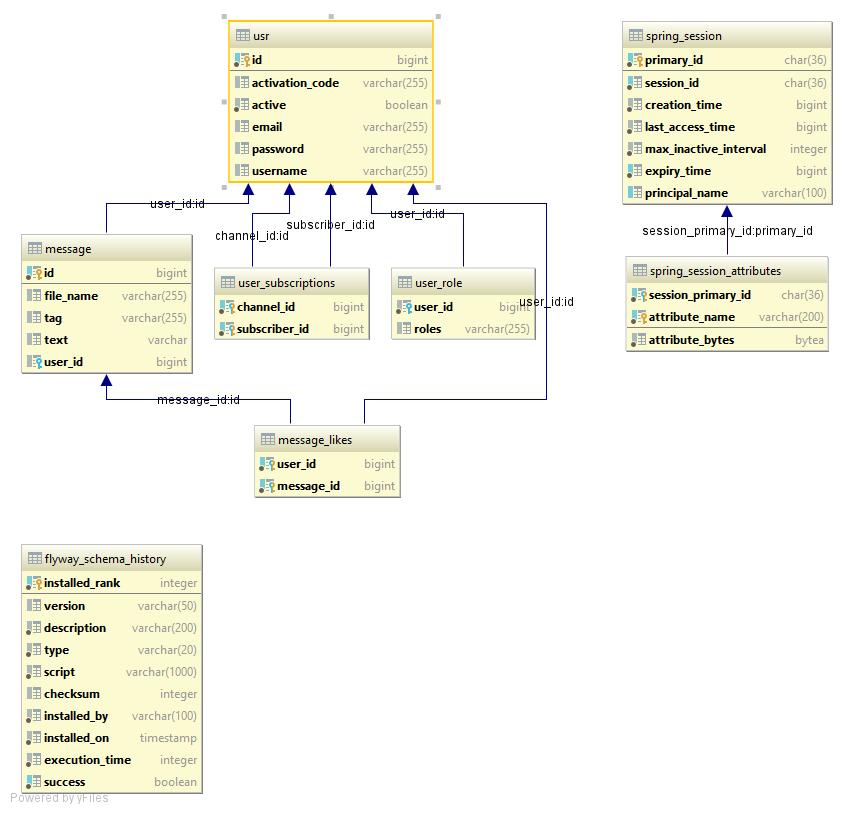
Модератору доступны все те же действия, что и пользователю, но также он имеет возможность удалять свои или чужие сообщения.

**Администратор**



Админу доступны все те же действия, что и модератору, но также он имеет возможность управлять ролями пользователей и их доступом в систему.

1. **Описание модели предметной области**



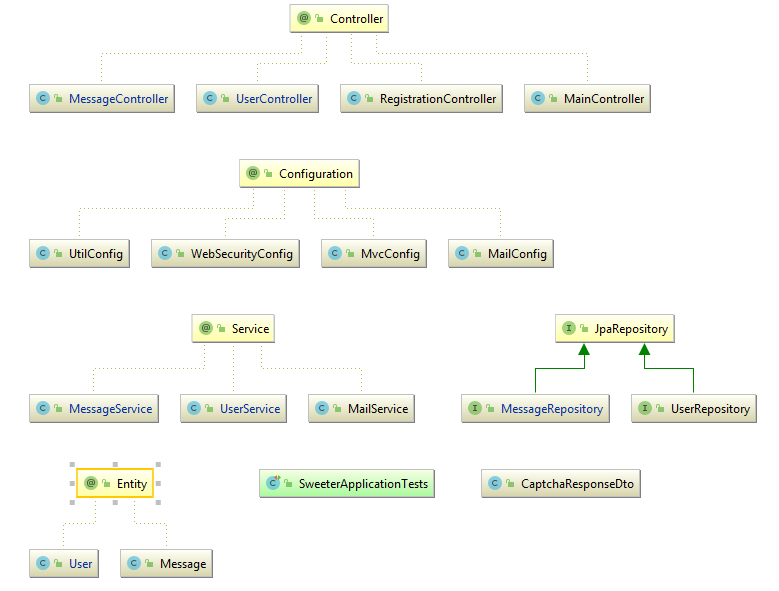
**Вовлеченные Сущности**

В результате разработки вариантов использования были выделены следующие вовлеченные сущности:

* Пользователь: клиент; логин и пароль; код активации
* Сообщение: название, прикрепленной картинки, тэг, текст сообщения
* Подписки пользователей
* Лайки пользователей

**Реализация задания с помощью «Spring Framework»**

1. **Объектно-ориентированное проектирование с учётом особенностей технологии**



**@SpringBootApplication**

Spring Boot поддерживает новую аннотацию @SpringBootApplication, которая эквивалентна использованию @Configuration, @EnableAutoConfiguration и @ComponentScan с их атрибутами по умолчанию.

Таким образом, достаточно создать класс, аннотированный с помощью @SpringBootApplication, а Spring Boot включит автоматическую настройку и отсканирует ресурсы в текущем пакете:

**@Configuration**

Это маркер, что данный производит регистрацию дополнительных компонентов или импорт дополнительных классов конфигурации.

**MailConfig**

Данный класс отвечает за настройку почтовой рассылки сервером.

**MvcConfig**

Данный класс производит настройку веб-приложения, а именно – конфигурирование ресурсов представления.

**WebSecurityConfig**

Данный класс производит настройку безопасности и отвечает за аутентификацию и авторизацию пользователей.

**@Controller**

Реализует компонент «Контроллер» из шаблона проектирования MVC (model-view-controller). Определяет, какое представление должно быть отображено в данный момент, а также интерпретирует действия пользователя и повещает модель о необходимости изменений.

**MainController**

Контроллер имеет единственный метод, который возвращает главную страницу приложения – index;

**RegistrationController**

Контроллер отвечает за регистрацию пользователя, а также за активацию аккаунта пользователя.

**UserController**

Контроллер отвечает за управление информацией о пользователях.

**MessageController**

Контроллер отвечает за управление информацией о сообщениях.

**@Entity**

Это маркер, что данный POJO является отображением таблицы базы данных.

**User** – класс пользователя.

**Message –** класс сообщения.

**JpaRepository** – данный интерфейс предоставляет удобные методы для работы с базой данных.

**MessageRepository –** данный интерфейс предоставляет методы для работыс даннымио сообщениях.

**UserRepository –** данныйинтерфейс предоставляет методы для работы с данными о пользователях.

**@Service**

Это маркер, который указывает на то, что класс реализует бизнес-логику приложения. Принято соблюдать следующий принцип взаимодействия между компонентами проекта: controller-service-domain

**MailService**

Данный класс отвечает за отправление почтовых сообщения пользователям.

**MessageService**

Данный класс описывает бизнес-логику приложения при работе с сущностями сообщений.

**UserService**

Данный класс описывает бизнес-логику приложения при работе с сущностями пользователей.

1. **Полный текст программы**

См. Приложение.

1. **Методика и результаты тестирования**

При разработке приложения были написаны модульные тесты, функциональное тестирование проводилось вручную.

|  |  |  |
| --- | --- | --- |
| **Вариант тестирования** | **Ожидаемый результат** | **Фактический результат** |
| Регистрация нового пользователя | Добавлен новый пользователь | Совпадает с ожидаемым |
| Нельзя использовать уже занятый логин | Совпадает с ожидаемым |
| Все поля формы регистрации должны пройти валидацию | Совпадает с ожидаемым |
| Авторизация | Уведомить, если логин или пароль не прошли валидацию | Совпадает с ожидаемым |
| Не авторизированным пользователям видна только страница приветствия | Совпадает с ожидаемым |
| Авторизованный пользователь может менять почту и пароль | Совпадает с ожидаемым |
| Роль пользователя | Может создать пост | Совпадает с ожидаемым |
| Может подписываться на других пользователей | Совпадает с ожидаемым |
| Может отписываться от других пользователей | Совпадает с ожидаемым |
| Может «лайкать» пост | Совпадает с ожидаемым |
| Может просматривать информацию о подписках и подписчиках | Совпадает с ожидаемым |
| Может искать посты по тэгам | Совпадает с ожидаемым |
| Может просматривать ленту постов | Совпадает с ожидаемым |
| Роль модератора | Может удалять сообщения | Совпадает с ожидаемым |
| Роль администратора | Может изменять роль пользователя | Совпадает с ожидаемым |
| Может блокировать доступ в систему | Совпадает с ожидаемым |

1. **Инструкция системному администратору по развёртыванию приложения**
2. В файле application.properties указать настройки для приложения:

* Connection string для подключения к базе данных
* Логин и пароль для подключения к базе данных
* Директорию, для хранения пользовательских картинок из сообщений
* Настройки для электронной почты: адрес почтового сервера, email-адрес, пароль, порт, протокол.
* Secret ключ для google-сервиса recaptcha

1. Собрать архив с помощью gradle из корневой папки проекта

*Gradle jar*

1. Скопировать на сервер
2. Запустить приложение

*java -jar sweater.jar*

1. Войти в систему (логин: admin, пароль: 1)

**Вывод**

В данной работе было спроектировано и разработано приложение «Sweater». В процессе проектирования были закреплены на практике знания о паттернах и внутреннем устройстве Spring Framework.

Так же были получены навыки интеграции с внешними сервисами – Google ReCaptcha, mail.yandex.ru. Для управления миграциями базы данных использовалась библиотека Flyway, что научило решать проблемы с версионностью схемы БД.

При разработке я отказался от реализации REST-сервиса из-за возрастающей сложности на стороне отображения (frontend), но у меня получилось добиться увеличения скорости работы приложения за счет использования библиотеки turbolinks.js, которая позволяет браузеру не перестраивать всю страницу при переходе по ссылкам, а только измененные части.

**Литература**

1. Буч Г., Рамбо Д., Якобсон А. Язык UML. Руководство пользователя. ДМК, 2006.

2. Фаулер М. UML. Основы, 3-е издание. Символ-Плюс, 2006.

3. Гранд М. Шаблоны проектирования в Java. BHV-СПб, 2004.

4. Крэг Ларман. Применение UML 2.0 и шаблонов проектирования, 3-е издание. Вильямс, 2007.

5. Гайд по разработке приложений на Spring - https://spring.io/guides