

**МИНИСТЕРСТВА НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**  
Федеральное государственное автономное образовательное учреждение  
высшего профессионального образования  
**Санкт-Петербургский политехнический университет Петра Великого**

---

Институт компьютерных наук и технологий  
Кафедра компьютерных систем и программных технологий

# Курсовой проект

Наименование: CopyritersShop с использованием «Spring  
Framework»

по курсу: "«Программное обеспечение распределённых  
вычислительных систем»"

Выполнил студент гр. 3540901/81501

Селиверстов Я.А.

Руководитель

Стручков И.В.

«\_\_»\_\_\_\_\_ 2019г.

Санкт-Петербург  
2019

# ОГЛАВЛЕНИЕ

<b>Глава 1. Анализ задания .....</b>	<b>3</b>
1. <b>Формулировка индивидуального задания .....</b>	<b>3</b>
2. <b>Требования, которым должно удовлетворять приложение.....</b>	<b>3</b>
3. <b>Основные варианты использования .....</b>	<b>3</b>
4. <b>Описание модели предметной области .....</b>	<b>4</b>
5. <b>Вовлеченные Сущности .....</b>	<b>4</b>
<b>Глава 2. Реализация задания с помощью «Spring Framework» .....</b>	<b>5</b>
1. <b>Объектно-ориентированное проектирование с учётом особенностей технологии.....</b>	<b>5</b>
2. <b>Диаграмма последовательности.....</b>	<b>6</b>
3. <b>Тестирование.....</b>	<b>7</b>
4. <b>Интерфейс приложения .....</b>	<b>7</b>
5. <b>Инструкция системному администратору по развёртыванию приложения .....</b>	<b>8</b>
6. <b>Ведение базы данных заказов .....</b>	<b>8</b>
7. <b>Инструкция пользователю по запуску приложения .....</b>	<b>9</b>
<b>Вывод.....</b>	<b>10</b>
<b>Литература .....</b>	<b>11</b>

# **Глава 1. Анализ задания**

## **1. Формулировка индивидуального задания**

Реализовать приложение «CopyritersShop» на языке Kotlin с помощью Spring Framework. Система CopyritersShop предназначена для автоматизации процесса заказа услуг по наполнению web-сайта текстами и написанию статей на заказ. Данная система предоставляет владельцу сайта найти и выбрать копирайтера, являющегося специалистом в определенной области, способного написать контент, либо промо-материал для сайта.

## **2. Требования, которым должно удовлетворять приложение**

Приложение должно строиться, используя следующие принципы и инструменты:

- Spring Framework
- Rest
- Паттерны

## **3. Основные варианты использования**

### **1. Создание заказа**

1. Клиент открывает главную страницу, на которой генерируется список заказов
2. Клиент генерирует заказ
3. Клиент оформляет заказ
4. Клиент подтверждает оформление заказа.

**Примечание.** Для простоты демонстрации реализован генератор заказов

### **2. Просмотр заказа оператором**

1. Оператор выполняет вход в систему
2. Оператор просматривает список заказов, которые ожидают его сопровождения

3. Оператор выбирает заказ для сопровождения
4. Оператор звонит клиенту для точного подтверждения
5. Оператор выставляет статус заказа.

### 3. Выполнение заказа

1. Copyriter выполняет вход в систему
2. Copyriter просматривает список заказов, которые готовы и ожидают выполнения
3. Copyriter выбирает заказы, которые хочет выполнить
4. Copyriter смотрит информацию
5. Copyriter принимает оплату от клиента.

### 4. Описание модели предметной области

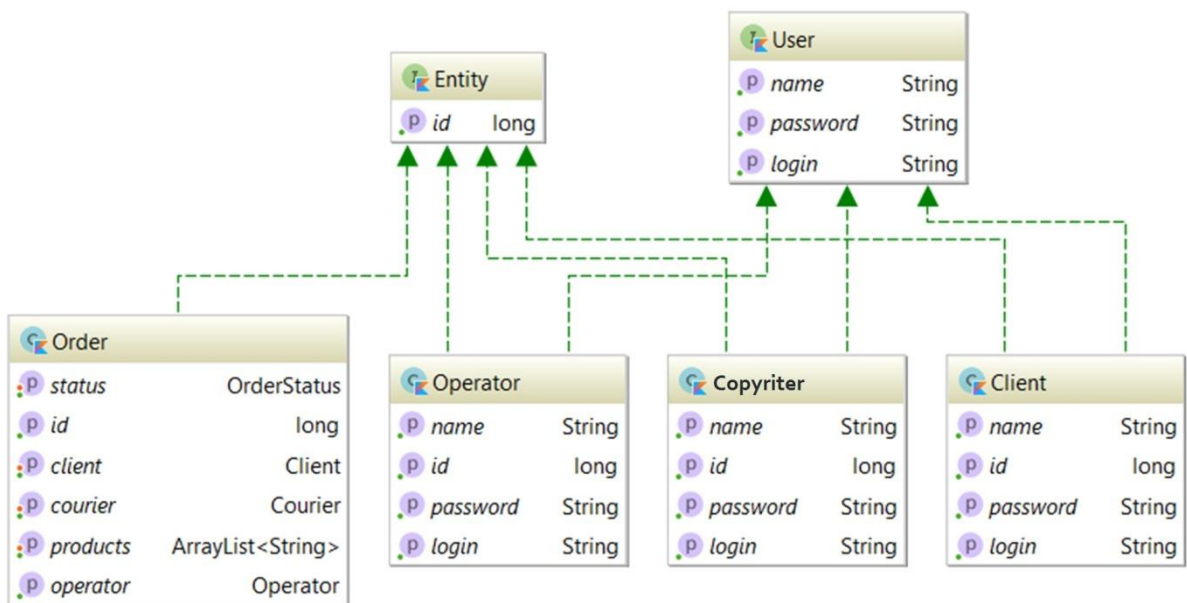


Рис.1

### 5. Вовлеченные Сущности

В результате разработки вариантов использования были выделены следующие вовлеченные сущности: Клиент, Оператор, Copyriter, Заказ.

## Глава 2. Реализация задания с помощью «Spring Framework»

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

Диаграммы классов представлены на рис.2

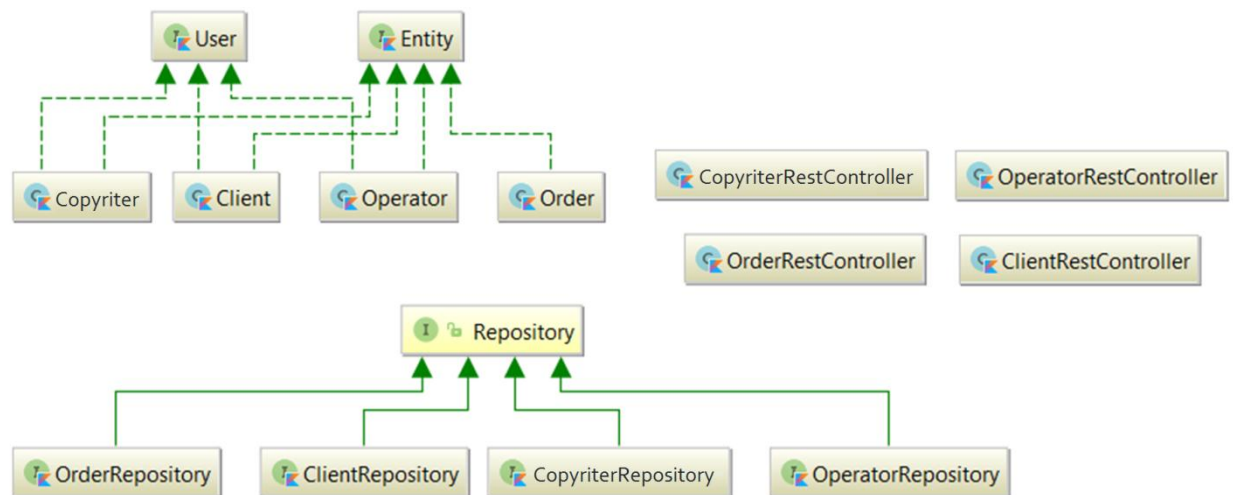


Рис.2

@SpringBootApplication – сканирует все контроллеры, сервисы и прочие составляющие спринга, чтобы все это вместе запустить

Application – класс запуска программы

@RestController – принимает запросы пользователя и возвращает данные в JSON формате

OrderRestController – контроллер для работы с сущностью заказа

@Entity – это аннотация, который берет соответствующий класс и на основе его полей делает таблицу в БД.

Order – класс заказа

User – класс пользователя

CrudRepository – обеспечивает сложные функциональные возможности CRUD для класса сущностей

## 2. Диаграмма последовательности

Диаграммы последовательности представлены на рис.3-5.



Рис.3



Рис.4



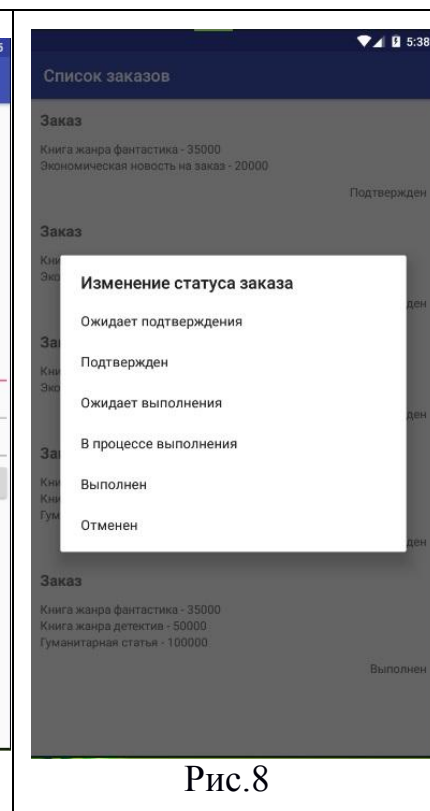
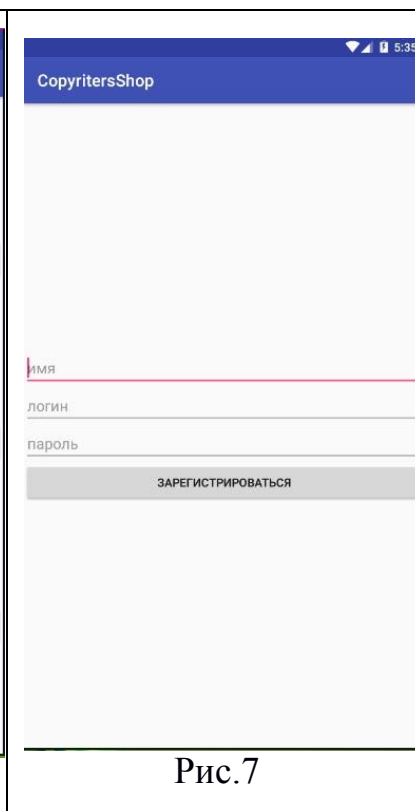
Рис.5

### 3. Тестирование

Вариант тестирования	Ожидаемый результат	Фактический результат
Регистрация нового пользователя	Добавлен новый пользователь	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
Роль Оператора	Оператор может изменять статус заказа	Совпадает с ожидаемым
	Оператор может подтвердить заказ	Совпадает с ожидаемым
Роль Copyriter	Copyriter может взять заказы, готовые к выполнению	Совпадает с ожидаемым
Роль Клиента	Клиент может создать заказ	Совпадает с ожидаемым

### 4. Интерфейс приложения

Интерфейс приложения представлен на рис. 6-8.



## 5. Инструкция системному администратору по развёртыванию приложения

1. Установить NoxPlayer  
<https://ru.bignox.com/>
2. Установить IntelliJ IDEA Ultimate  
<https://jetbrains.ru/products/idea/>
3. Скачать архив copyriter.rar
4. Скачать архив copyritersShop.rar
5. Разархивировать copyriter.rar
6. Разархивировать copyritersShop.rar
7. В IntelliJ IDEA Ultimate запустить проект copyriter
8. В Multi-Drive Nox создать эмулятор Android 7, перенести в него copyritersShop и прописать IP и порт: 192.168.1.70:8080.
9. Прописать в адресной строке хост: localhost:8080/h2-console (рис.6)

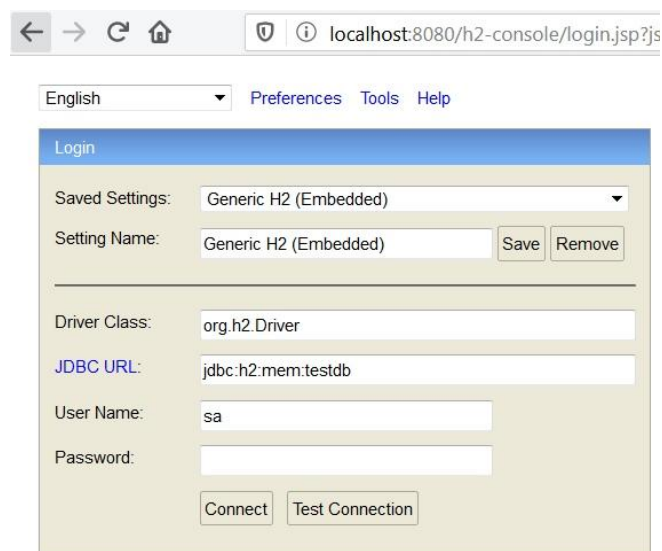


Рис.9.

## 6. Ведение базы данных заказов

Структура БД представлена на рис. 10-12.

SELECT * FROM ORDERREQUEST					
ID	PRODUCTS	STATUS	CLIENT_ID	COPYRITER_ID	OPERATOR_ID
1	aced0005737200136a617661	4	1	null	null
2	aced0005737200136a617661	0	1	null	null

(2 rows, 10 ms)

Рис.10



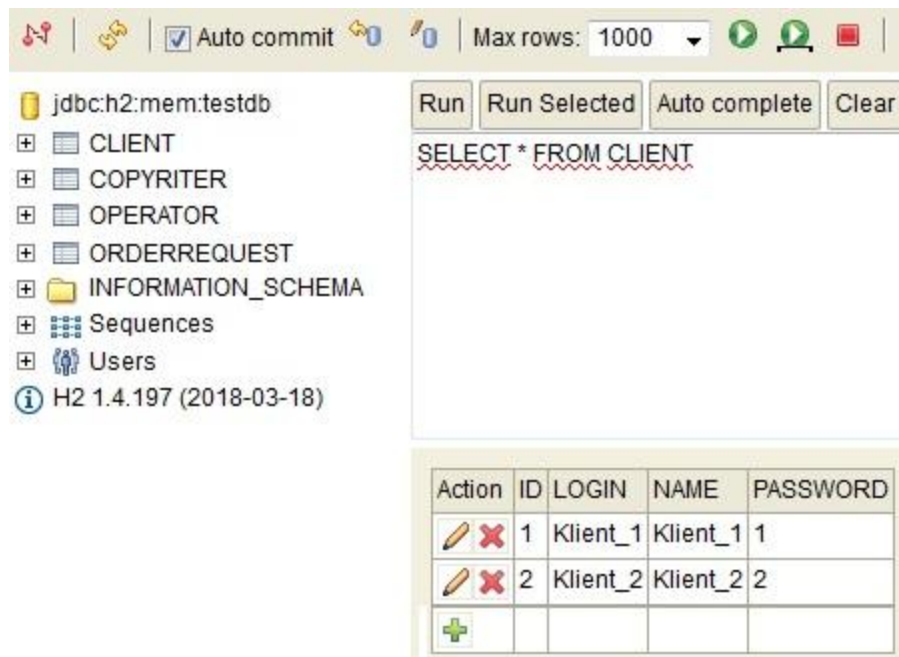


Рис.11

## 7. Инструкция пользователю по запуску приложения

1. Зарегистрироваться в системе
2. Выбрать понравившийся заказ (для простоты демонстрации реализован генератор заказов)
3. Оформить заказ
4. Дождаться выполнения заказа

## Вывод

В результате работы на курсовым проектом было спроектировано и разработано приложение «CopyritersShop». В процессе проектирования были закреплены на практике знания о Spring Framework.

Для реализации проекта использовался Spring Boot.

Степень выполнения поставленного задания составляет 60%. Текущая реализация требует улучшения и некоторых доработок по следующим направлениям:

- Надежность данных при авторизации
- Расширения функциональности для пользователей

В текущей реализации продемонстрированы такие фундаментальные навыки как построение клиент-серверной системы, организация взаимодействия компонентов посредством REST API.

Код программы выложен в репозитории на GitHub.

## Литература

1. Фаулер М. UML. Основы, 3-е издание. Символ-Плюс, 2006.
2. Гранд М. Шаблоны проектирования в Java. БНВ-СПб, 2004.
3. Крэг Ларман. Применение UML 2.0 и шаблонов проектирования, 3-е издание. Вильямс, 2007.
4. Гайд по разработке приложений на Spring - <https://spring.io/guides>