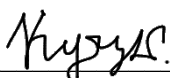


**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Образовательная программа бакалавриата «Программная инженерия»

**СОГЛАСОВАНО**

Руководитель проекта,  
Приглашенный преподаватель  
Базовой кафедры «Системное  
программирование» Института системного  
программирования им. В.П. Иванникова  
РАН (ИСП РАН)

 А.А. Кучук  
«16» мая 2021 г.

**УТВЕРЖДАЮ**

Академический руководитель  
образовательной программы  
«Программная инженерия»  
профессор департамента программной  
инженерии, канд. техн. наук

В. В. Шилов  
«\_\_» \_\_\_\_\_ 2021 г.


**КЛИЕНТ-СЕРВЕРНОЕ ПРИЛОЖЕНИЕ ГЕОТАРГЕТИРОВАННОЙ РЕКЛАМЫ  
«ГЕОФУД». СЕРВЕР**

**Пояснительная записка**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.05.06-01 81 01-1-ЛУ**

Исполнитель:

 студент группы БПИ199  
/ Т. Р. Асатиани /  
«16» мая 2021 г.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	RU.17701729.04.01-01 81 01-1-ЛУ

**Москва 2021**

УТВЕРЖДЕН  
RU.17701729.04.01-01 81 01-1-ЛУ

КЛИЕНТ-СЕРВЕРНОЕ ПРИЛОЖЕНИЕ ГЕОТАРГЕТИРОВАННОЙ РЕКЛАМЫ  
«ГЕОФУД». СЕРВЕР

Пояснительная записка

RU.17701729.04.01-01 81 01-1

Листов 30

<i>Инв. № подл</i>	<i>Подп. и дата</i>
RU.17701729.04.01-01 81 01-1-ЛУ	
<i>Взам. инв. №</i>	<i>Инв. № дубл.</i>
<i>Подп. и дата</i>	<i>Подп. и дата</i>

Москва 2021

## Содержание

<b>АННОТАЦИЯ.....</b>	<b>3</b>
<b>1. ВВЕДЕНИЕ.....</b>	<b>4</b>
1.1. Наименование программы .....	4
1.2. Документы, на основании которых ведется разработка программ.....	4
<b>2. Назначение и область применения.....</b>	<b>5</b>
2.1. Назначение .....	5
2.2. Область применения.....	5
<b>3. Технические характеристики .....</b>	<b>6</b>
3.1. Постановка задачи на разработку программы .....	6
3.2. Описание алгоритма и функционирования программы.....	6
3.3. Обоснование выбора архитектуры приложения .....	9
3.4. Описание и обоснование выбора технических и программных средств .....	9
<b>4. Ожидаемые технко-экономические показатели .....</b>	<b>11</b>
4.1. Предполагаемая потребность .....	11
4.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами.....	11
<b>5. Список литературы.....</b>	<b>12</b>
<b>ПРИЛОЖЕНИЕ 1 .....</b>	<b>2</b>
<b>ТЕРМИНОЛОГИЯ .....</b>	<b>2</b>
<b>ПРИЛОЖЕНИЕ 2 .....</b>	<b>3</b>
<b>ПРЕДСТАВЛЕНИЕ ТАБЛИЦ БАЗЫ ДАННЫХ.....</b>	<b>3</b>
<b>ПРИЛОЖЕНИЕ 3 .....</b>	<b>5</b>
<b>ДИАГРАММА КЛАССОВ.....</b>	<b>5</b>
<b>ПРИЛОЖЕНИЕ 4 .....</b>	<b>6</b>
<b>ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ.....</b>	<b>6</b>
<b>ПРИЛОЖЕНИЕ 5 .....</b>	<b>8</b>
<b>ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ МЕТОДОВ И СВОЙСТВ СЕРВЕРА .....</b>	<b>8</b>
<b>6. Лист регистрации изменений .....</b>	<b>30</b>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛУ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## RU.17701729.04.01-01 81 01-1 АННОТАЦИЯ

В данном программном документе приведена пояснительная записка к программе «GeoFood» («Клиент-серверное приложение геотаргетированной рекламы «Геофуд». Сервер»), предназначенной для применения пользователями в повседневной жизни для облегчения выбора товаров в магазинах.

В разделе «Введение» указано наименование и краткая характеристика области применения программы.

В разделе «Назначение и область применения» указано функциональное назначение программы, эксплуатационное назначение программы и краткая характеристика области применения программы.

В разделе «Технические характеристики» содержатся следующие подразделы:

- постановка задачи на разработку программ;
- описание алгоритма и функционирования программы с обоснованием выбора схемы алгоритма решения задачи;
- описание и обоснование выбора метода организации входных и выходных данных;
- описание и обоснование выбора состава технических и программных средств.

В разделе «Ожидаемые технико-экономические показатели» указана предполагаемая потребность и экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

Настоящий документ разработан в соответствии с требованиями:

- 1) ГОСТ 19.103-77 Обозначения программ и программных документов [1];
- 2) ГОСТ 19.104-78 Основные надписи [2];
- 3) ГОСТ 19.105-78 Общие требования к программным документам [3];
- 4) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [4];
- 5) ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению [5].

Изменения к техническому заданию оформляются согласно ГОСТ 19.603-78 [6], ГОСТ 19.604-78 [7].

Перед прочтением данного документа рекомендуется ознакомиться с терминологией, приведённой в Приложении 1 настоящей пояснительной записки.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛЮ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**1. ВВЕДЕНИЕ****1.1. Наименование программы**

Наименование темы разработки: «Клиент-серверное приложение геотаргетированной рекламы «ГеоФуд».

Наименование темы разработки на английском языке: «Client-server application of geo-targeted advertising «GeoFood».

**1.2. Документы, на основании которых ведется разработка программ**

Основанием для разработки является учебный план подготовки бакалавров по направлению 09.03.04 "Программная инженерия" и утвержденная академическим руководителем тема курсового проекта.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛУ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## 2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

### 2.1. Назначение

#### 2.1.1. Функциональное назначение

Программа должна выполнять следующие функции:

- 1) Отслеживать перемещения пользователей;
- 2) Анализировать полученные данные и строить по ним вероятные маршруты пользователей;
- 3) Предоставлять возможность компаниям предоставлять уникальные акционные предложения для будущих потенциальных клиентов;
- 4) Предлагать пользователям персонализированные акции.

#### 2.1.2. Эксплуатационное назначение

Приложение должно помочь свести потенциального потребителя и компанию, предоставляющую услуги питания.

Сервис предоставляет акции для пользователей, которые постоянно проходят мимо заведения, но не пользуются его услугами, однако могут стать его постоянными клиентами. Персональные акции могут стать мотиватором для похода в новое кафе, ресторан и т.п. Клиент впервые для себя открывает новое место и возможно останется его постоянным посетителем.

Приложение учитывает индивидуальные предпочтения пользователей и подстраивает рекламу под его потребности, например утром - кофе вечером - рестораны.

### 2.2. Область применения

Приложение будет применяться заведениями в области общественного питания в целях привлечения клиентов.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛЮ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

### 3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

#### 3.1. Постановка задачи на разработку программы

##### 3.1.1. Разрабатываемое приложение должно иметь следующие функции:

- 1) Регистрация двух типов пользователей – User и Admin.
- 2) Аутентификация пользователей;
- 3) Регистрация и хранение магазинов.
- 4) Регистрация и хранение акций
- 5) Получение общих акций ресторана;
- 6) Получение персонализированных акций;
- 7) Получение магазинов в определенном радиусе от точки;
- 8) Получение заведений определенного типа;
- 9) Анализ перемещений пользователя.

#### 3.2. Описание алгоритма и функционирования программы

##### 3.2.1. Архитектура приложения

Приложение написано на Spring Boot с применением REST API архитектуры.

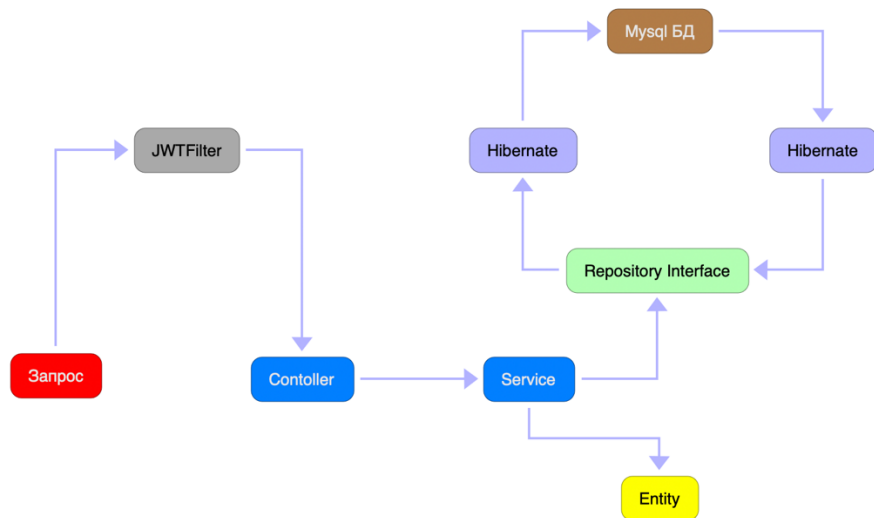


рис. 1

- 1) JWT – библиотека, отвечающая за аутентификацию пользователей и фильтрацию запросов по токену.
- 2) Spring RestControllers – обрабатывают заданные URL, одержат основную бизнес-логику.
- 3) MySQL – хранение данных.
- 4) Hibernate – соединение с MySQL.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-Л1У				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



рис. 2

Модель “Объект – Сервис – Интерфейс – Фреймворк”, изображенная на рис. 2 позволяет разделить бизнес-логику и реализацию. Благодаря такой модели любое изменение в реализации не будет затрагивать бизнес-логику.

### 3.2.2. Организация хранения данных в MySQL БД.

Данные хранятся в базе данных “geofood” и разделены на 5 таблиц (см. рис. 3):

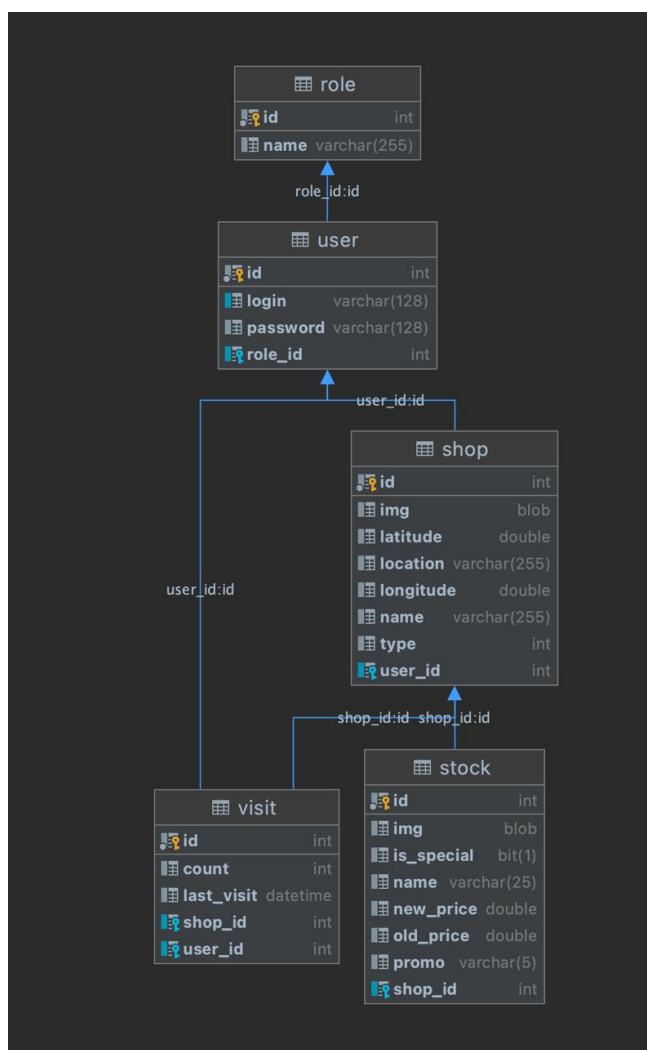


рис. 3

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-1/1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



**RU.17701729.04.01-01 81 01-1**

- 1) Role – статическая таблица для хранения идентификатора роли пользователя. Имеет одно поле - “name”, и содержит только две записи: USER\_ROLE и ADMIN\_ROLE.
- 2) User – таблица для хранения информации о пользователе. Имеет поля:
  - a. login – соответственно логин пользователя;
  - b. password – пароль, хранящийся в зашифрованном виде;
  - c. role\_id – id записи в таблице “role” для идентификации роли пользователя (admin/user);
- 3) Shop – таблица для хранения данных о магазине. Поля таблицы:
  - a. img – изображение-иконка магазина, хранящееся в формате BLOB;
  - b. latitude – широта положения точки;
  - c. longitude – широта положения точки;
  - d. location – адрес магазина;
  - e. name – название магазина;
  - f. type – типа магазина, хранящийся в формате int, идентифицируется на стороне клиента с помощью перечисления.
  - g. user\_id – id сущности с ролью “ADMIN\_ROLE” в таблице “user”, которая является владельцем магазина.
- 4) Stock – таблица для хранения данных об акции. Поля таблицы:
  - a. img – изображение-иконка акции, хранящееся в формате BLOB;
  - b. is\_special – boolean параметр, который разделяет акции на специальные и обычные;
  - c. name – название акции;
  - d. new\_price – новая цена продукта;
  - e. old\_price – старая цена продукта;
  - f. promo – 5 латинских символов представляющих собой промо-код акции;
  - g. shop\_id – id сущности в таблице “shop” – магазина, предоставляющего данную акцию.
- 5) Visit – таблица для хранения записи о посещениях пользователем отдельных магазинов. Таблица позволяет хранить записи о посещениях каждым пользователем каждого магазина и реализовывать анализ его предпочтений. Каждая запись в таблице отвечает за отдельное сочетание User-Shop. Поля таблицы:
  - a. count – кол-во посещений конкретного магазина.
  - b. last\_visit – дата последнего посещения.
  - c. shop\_id – id сущности в таблице “shop” – магазина, который посетил пользователь.
  - d. user\_id – id пользователя с ролью “USER\_ROLE” в таблице “user”, который посещал магазин.

Все типы полей в таблицах БД можно увидеть на рис. 3.

**3.2.3. Синхронизация положения пользователя с клиентом.**

Сервер может принимать запросы о местонахождении пользователя и анализировать магазины в некотором радиусе от пользователя создавая записи о посещениях.

**3.2.4. Обработка запросов на сервере.**

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛУ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## RU.17701729.04.01-01 81 01-1

Запросы приходят на прописанные URL и имеют строгую маршрутизацию. Пред обработкой, запрос проходит фильтрацию и анализ токена, а после попадает в Spring RestController, который содержит бизнес-логику.

### 3.2.4.1. Безопасность с помощью токена и фильтрация запросов .

Сервер создает токен с помощью библиотеки JWT, в который заворачивает логин пользователя, чтобы потом его оттуда забрать в фильтре, когда пользователь будет делать запрос.

Фильтрация основана на ролях Admin и User, реализована с помощью springframework.web.filter.GenericFilterBean в которой достаётся логин пользователя из токена, находится пользователь в базе данных и в зависимости от его роли происходит доступ или запрет на доступ к прописанным url.

## 3.3. Обоснование выбора архитектуры приложения

Архитектура REST была выбрана из-за ряда достоинств:

- 1) Масштабируемость – такую архитектуру легко поддерживать и развивать по причине разбиения программы на модули, которые отвечают за свою логику;
- 2) Тестируемость – каждый модуль можно тестировать независимо;
- 3) Отсутствие состояния – в период между запросами серверу не нужно хранить информацию о состоянии клиента и наоборот.
- 4) Единообразие интерфейса – единый формат клиент-серверного взаимодействия, который описывает, что, куда, в каком виде и как отсылать;

## 3.4. Описание и обоснование выбора технических и программных средств

### 3.4.1. Состав технических и программных средств

Для работы программы необходим сервер:

- 1) Ubuntu 20.04;
- 2) 2 ГБ RAM;
- 3) 2 core CPU;
- 4) 20 ГБ SSD;

### 3.4.2. Обоснование выбора технических и программных средств

- 1) Для разработки использовался язык Java + Spring Boot из-за:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛД				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**RU.17701729.04.01-01 81 01-1**

- a. Java изучалась на 2-м курсе ОП, на ней написано огромное кол-во веб. приложений.
- b. Spring Framework - стандарт для разработки серверной части REST API приложений.
- c. Spring Boot – упрощает работу со Spring Framework:
  - i. Авто конфигурация приложения;
  - ii. Помогает напрямую встроить в приложение Tomcat;
  - iii. Создает автономные приложения.
- 2) Hibernate – работа с базой данных.
  - a. Обеспечивает автоматическую генерацию и обновление таблиц в БД;
  - b. Устраняет множество повторяющегося кода, который пишет разработчик при работе с JDBC;
  - c. Предоставляет собственный мощный язык запросов HQL;
  - d. Ускоряет работу приложения.
- 3) Lombok – плагин компилятора, для авто-генерации такого кода как “геттеры”, “сеттеры”, “конструкторы” и т.д.
- 4) Maven – фреймворк для автоматизации сборки проектов на основе описания их структуры в файлах на языке POM.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛЮ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

#### 4. ОЖИДАЕМЫЕ ТЕХНКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

##### 4.1. Предполагаемая потребность

Компании-рекламодатели смогут привлечь новых покупателей благодаря созданию персонализированных рекламных предложений, которые дают плюсы обоим сторонам: пользователи-рекламодатели находят новых клиентов, а клиенты получают индивидуальные акционные предложения.

##### 4.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами

На зарубежном рынке существуют приложения "Instagram", "Facebook" обладающие подобным функционалом: <https://www.instagram.com/>, <https://www.facebook.com/>.

На отечественном рынке существуют интернет ресурсы "Вконтакте", "Яндекс.Карты", "Яндекс.Еда", "2ГИС", "Деливери-клуб" с подобным функционалом: <https://vk.com/>, <https://yandex.ru/maps/>, <https://eda.yandex.ru/>, <https://2gis.ru/>, <https://www.delivery-club.ru/>.

Приложение обладает преимуществами:

- 1) Позволяет пользователям получать как общие, так и индивидуальные акционные предложения;
- 2) Позволяет компаниям-рекламодателям создавать как общие, так и индивидуальные акционные предложения;
- 3) Специальные акции предлагаются пользователям автоматически, на основе его перемещений.
- 4) Позволяет компаниям-рекламодателям просматривать статистику по общим и индивидуальным рекламным предложениям;
- 5) Подбор индивидуальных предложений происходит на основе каждодневных маршрутов пользователя.
- 6) Приложение заточено на индивидуальную ротацию акций для каждого пользователя.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛЮ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## 5. СПИСОК ЛИТЕРАТУРЫ

1. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 15150-69 Машины, приборы и другие технические изделия. Исполнения для различных климатических районов. Категории, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды. – М.: Изд-во стандартов, 1997.
3. ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.401-78. ЕСПД. Текст программы. Требования к содержанию и оформлению. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.404-79. ЕСПД. Пояснительная записка. Требования к содержанию и оформлению. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.505-79. ЕСПД. Руководство оператора. Требования к содержанию и оформлению. . – М.: ИПК Издательство стандартов, 2001.
7. Документация по Spring Boot [Электронный ресурс]// URL: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/> (Дата обращения: 5.05.2021, режим доступа: свободный).
8. Документация по Spring Boot [Электронный ресурс]// URL: <https://docs.spring.io/spring-framework/docs/current/reference/html/> (Дата обращения: 5.05.2021, режим доступа: свободный).
9. Документация по Hibernate [Электронный ресурс]// URL: <https://hibernate.org/orm/documentation/5.4/> (Дата обращения: 5.05.2021, режим доступа: свободный).
10. Г. Шилдт Java Полное Руководство // Издательство “Диалектика-Вильямс”, 2018 .

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛУ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



## ПРИЛОЖЕНИЕ 1

### ТЕРМИНОЛОГИЯ

**Http** – протокол передачи данных;

**Авторизация** – предоставление определённому лицу или группе лиц определённых прав;

**База данных** – организованная структура, предназначенная для хранения, изменения и обработки взаимосвязанной информации;

**Пользователь** – человек, который использует данную программу;

**Сервер** – компонент вычислительной системы, выполняющий функции по запросу клиента, предоставляя ему доступ к определенным ресурсам;

**Клиент** – программа, работающая на устройстве пользователя и позволяющая получать и отправлять информацию на сервер и предоставлять ее пользователю в интерактивном виде;

**Токен** (“Пограмный токен”) – зашифрованная последовательность символов, которая выдается пользователю после успешной авторизации и являются ключом для доступа к службам;

**URL** – система унифицированных адресов электронных ресурсов, или единообразный определитель местонахождения ресурса (файла);

**REST** – архитектурный стиль взаимодействия компонентов распределённого приложения в сети;

**БД** – база данных;

**Геотаргетинг** – в веб-разработке и интернет-маркетинге, метод выдачи посетителю содержимого, соответствующего его географическому положению.

**Акция** – событие, дающее право на приобретение товара или услуги на специальных условиях.

**Магазин** – заведение, ресторан/кафе.

**Промокод** — состоящая из букв и/или цифр совокупность символов, дающая право на приобретение товара или услуги на специальных условиях.

**Репозиторий** — место, где хранятся и поддерживаются какие-либо данные.

RU.17701729.04.01-01 81 01-1

## ПРИЛОЖЕНИЕ 2

## ПРЕДСТАВЛЕНИЕ ТАБЛИЦ БАЗЫ ДАННЫХ

Название таблицы	Представление таблицы
role	
user	
shop	
stock	

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛД				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



RU.17701729.04.01-01 81 01-1

visit	<div><div>visit</div><div><div>id</div><div>int</div></div><div><div>count</div><div>int</div></div><div><div>last_visit</div><div>datetime</div></div><div><div>shop_id</div><div>int</div></div><div><div>user_id</div><div>int</div></div></div>	
-------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-Л1У				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1  
ПРИЛОЖЕНИЕ 3

## ДИАГРАММА КЛАССОВ



Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ИЙ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

## ПРИЛОЖЕНИЕ 4

## ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ

Класс	Назначение
AddShopRequest	Формат запроса "admin/stock/add"
AddStockRequest	Формат запроса /admin/stock/add"
AuthController	Spring @RestController, отвечающий за регистрацию и авторизацию пользователей.
AuthResponse	Формат ответа на запрос " /auth"
AuthRequest	Формат запроса " /auth"
BusinessController	Spring @RestController, отвечающий за добавление/удаление магазинов/акций, а также за обработку перемещений пользователя.
CustomUserDetails	Отвечает за авторизацию пользователя по токenu
CustomUserDetailsService	Отвечает за работу с CustomUserDetails
ControllerUtils	Содержит вспомогательные методы для вызова в классах @RestController
GeofoodApplication	Главный класс, который запускает SpringBoot приложение.
GetStocksRequest	Формат запроса " /user/stocks"
GetStockImgRequest	Формат запроса " /stock/img"
GetShopImgRequest	Формат запроса " /shop/img"
GetNearShopsRequest	Формат запроса " /user/shops"
JwtProvider	Класс, отвечающий за генерацию и валидацию токенов
JwtFilter	Отвечает за фильтрацию пользователей
MovementRequest	Формат запроса " /movement"

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-Л1У				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## RU.17701729.04.01-01 81 01-1

RegistrationRequest	Формат запроса “/reg/user”
RoleEntity	Описывает сущность “роль”
RoleRepository	Интерфейс репозитория для доступа к базе данных
SecurityConfig	Разделяет доступ к URL адресам по разным ролям
ShopEntity	Описывает сущность “магазин”
ShopRepository	Интерфейс репозитория для доступа к базе данных
ShopService	Класс, содержащий методы для работы с таблицей “shop” в базе данных
StockEntity	Описывает сущность “акция”
StockRepository	Интерфейс репозитория для доступа к базе данных
StockService	Класс, содержащий методы для работы с таблицей “stock” в базе данных.
TestSecurityController	Spring @RestController, предназначенный для ручного тестирования работы фильтрации по ролям
UserEntity	Описывает сущность “пользователь”
UserRepository	Интерфейс репозитория для доступа к базе данных
UserService	Класс, содержащий методы для работы с таблицей “user” в базе данных.
Views	Класс, содержащий интерфейсы для правильного формирования JSON
VisitActionEntity	Описывает событие “пользователь посетил магазин”
VisitActionRepository	Интерфейс репозитория для доступа к базе данных
VisitActionService	Класс, содержащий методы для работы с таблицей “visit” в базе данных.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-Л1У				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## RU.17701729.04.01-01 81 01-1

## ПРИЛОЖЕНИЕ 5

## ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ МЕТОДОВ И СВОЙСТВ СЕРВЕРА

## Описание полей, методов и свойств класса AddShopRequest

Класс имеет аннотацию lombok.Data => все “геттеры”, “сеттеры” и т.д. генерируются автоматически.

Поля				
Имя	Модификатор доступа	Тип	Аннотация	Назначение
name	private	String	@NotEmpty	Название магазина
longitude	private	double	@NotEmpty	Долгота
latitude	private	double	@NotEmpty	Широта

## Описание полей, методов и свойств класса AddStockRequest

Класс имеет аннотацию lombok.Data => все “геттеры”, “сеттеры” и т.д. генерируются автоматически.

Поля				
Имя	Модификатор доступа	Тип	Аннотация	Назначение
name	private	String	@NotEmpty	Название акции
newPrice	private	double	@NotEmpty	Новая цена
oldPrice	private	double	@NotEmpty	Старая цена
promo	private	String	@NotEmpty	Промокод
shopName	private	String	@NotEmpty	Название магазина, которому принадлежит акция

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛД				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## RU.17701729.04.01-01 81 01-1

Описание полей, методов и свойств класса AuthController

Поля				
Имя	Модификатор доступа	Тип	Назначение	
userService	private	UserService	Экземпляр класса для работы с БД	
jwtProvider	private	JwtProvider	Класс, отвечающий за генерацию и валидацию токенов	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
registerUser	public	void	@RequestBody @Valid RegistrationRequest registrationRequest	Обработывает запросы, приходящие на "/reg/user"
auth	public	AuthResponse	@RequestBody AuthRequest request	Обработывает запросы, приходящие на "/auth"

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛЮ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**RU.17701729.04.01-01 81 01-1**

## Описание полей, методов и свойств класса AuthResponse

Класс имеет аннотацию lombok.Data => все “геттеры”, “сеттеры” и т.д. генерируются автоматически.

Поля				
Имя	Модификатор доступа	Тип	Аннотация	Назначение
token	private	String		Сгенерированный токен при аутентификации пользователя

## Описание полей, методов и свойств класса AuthRequest

Класс имеет аннотацию lombok.Data => все “геттеры”, “сеттеры” и т.д. генерируются автоматически.

Поля				
Имя	Модификатор доступа	Тип	Аннотация	Назначение
login	private	String	@NotEmpty	Логин пользователя
passwordd	private	String	@NotEmpty	Пароль

## Описание полей, методов и свойств класса BusinessController

Поля			
Имя	Модификатор доступа	Тип	Назначение
userService	private	UserService	Экземпляр класса для работы с БД
shopService	private	ShopService	Экземпляр класса для работы с БД
stockService	private	StockService	Экземпляр класса для работы с БД

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛУ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## RU.17701729.04.01-01 81 01-1

visitActionService	private	VisitActionService	Экземпляр класса для работы с БД	
jwtProvider	private	JwtProvider	Класс, отвечающий за генерацию и валидацию токенов	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
addShops	public	HttpStatus	@RequestHeader("Authorization") String bearer,  @RequestBody AddShopRequest request	Обрабатывает запросы, приходящие на "/admin/shop/add"
getShops	public	ResponseEntity  <Iterable<ShopEntity>>	@RequestHeader("Authorization") String bearer	Обрабатывает запросы, приходящие на "/admin/shop/get"
addStock	public	HttpStatus	@RequestHeader("Authorization") String bearer,  @RequestBody AddStockRequest request	Обрабатывает запросы, приходящие на "/admin/stock/add"
getStocks	public	ResponseEntity  <Iterable<StockEntity>>>	@RequestHeader("Authorization") String bearer	Обрабатывает запросы, приходящие на "/admin/stock/get"

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-Л1У				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



## RU.17701729.04.01-01 81 01-1

getNearShops	public	List<ShopEntity>	@RequestHeader("Authorization") String bearer,  @RequestBody GetNearShopsRequest request	Обрабатывает запросы, приходящие на "/user/shops"
getStocks	public	List<StockEntity>	@RequestHeader("Authorization") String bearer,  @RequestBody GetStocksRequest request	Обрабатывает запросы, приходящие на "/user/stocks"
movement	public	HttpStatus	@RequestHeader("Authorization") String bearer,  @RequestBody MovementRequest request	Обрабатывает запросы, приходящие на "/movement"
getImg	public	byte[]	@RequestBody GetShopImgRequest request	Обрабатывает запросы, приходящие на "/shop/img"
getImg	public	byte[]	@RequestBody GetStockImgRequest request	Обрабатывает запросы, приходящие на "/stock/img"

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛЮ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**RU.17701729.04.01-01 81 01-1**

Описание полей, методов и свойств класса ControllerUtils

Все методы класса имеют модификатор “static”

Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
getTokenFromHeader	public	String	String bearer	Получает токен из “заголовка” http запроса
createShop	public	ShopEntity	AddShopRequest request	Создает новую сущность Shop
createStock	public	StockEntity	AddStockRequest request	Создает новую сущность Stock
shopInAarea	public	boolean	double x, double y, double radius, ShopEntity shop	Проверяет, находится ли магазин в заданном радиусе от передаваемой точки

Описание полей, методов и свойств класса AccountViewController

Поля			
Имя	Модификатор доступа	Тип	Назначение
userService	private	UserService	Экземпляр класса для работы с БД
shopService	private	ShopService	Экземпляр класса для работы с БД
stockService	private	StockService	Экземпляр класса для работы с БД
visitActionService	private	VisitActionService	Экземпляр класса для работы с БД

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-Л1У				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## RU.17701729.04.01-01 81 01-1

path	private	String	Путь, к папке на сервере в которой лежат медиа-файлы для генерации тестовых примеров	
shops	private	List<Shop>	Лист 11-ти магазинов для генерации тестовых примеров	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
main	public	void		Метод запускающий приложение
generatePromo	private	void		Создает строку из 5-и латинских символов и цифр
createUsersWithContent	private	void		Создает админа и пользователя, наполняет контентом базу данных
generateShops	private	void	UserEntity holder, UserEntity user	Создает магазины из shops, наполняет их контентом и сохраняет в базу данных
generateStocks	private	void	ShopEntity holder, boolean isMac	Создает акции, наполняет их контентом и сохраняет в базу данных
createSpecialStock	private	void	ShopEntity holder	Создает “специальные” акции, наполняет их контентом и сохраняет в базу данных

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-Л1У				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**RU.17701729.04.01-01 81 01-1**

createVisit	private	void	UserEntity user, ShopEntity shop, int cnt	Создает события “пользователь посетил магазин” и сохраняет их в базу данных
-------------	---------	------	-------------------------------------------------	--------------------------------------------------------------------------------------

## Описание полей, методов и свойств класса GetStockRequest

Класс имеет аннотацию lombok.Data => все “геттеры”, “сеттеры” и т.д. генерируются автоматически.

Поля				
Имя	Модификатор доступа	Тип	Аннотация	Назначение
id	private	int	@NotEmpty	Id магазина, которому принадлежит акция

## Описание полей, методов и свойств класса GetStockImgRequest

Класс имеет аннотацию lombok.Data => все “геттеры”, “сеттеры” и т.д. генерируются автоматически.

Поля				
Имя	Модификатор доступа	Тип	Аннотация	Назначение
stockId	private	int	@NotEmpty	Id акции, изображение которой будет получено

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-Л1У				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## RU.17701729.04.01-01 81 01-1

## Описание полей, методов и свойств класса GetShopImgRequest

Класс имеет аннотацию lombok.Data => все “геттеры”, “сеттеры” и т.д. генерируются автоматически.

Поля				
Имя	Модификатор доступа	Тип	Аннотация	Назначение
shopId	private	int	@NotEmpty	Id магазина, изображение которого будет получено

## Описание полей, методов и свойств класса GetNearShopsRequest

Класс имеет аннотацию lombok.Data => все “геттеры”, “сеттеры” и т.д. генерируются автоматически.

Поля				
Имя	Модификатор доступа	Тип	Аннотация	Назначение
longitude	private	double	@NotEmpty	Долгота
latitude	private	double	@NotEmpty	Широта
radius	private	double	@NotEmpty	Радиус

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛД				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## RU.17701729.04.01-01 81 01-1

## Описание полей, методов и свойств класса JwtProvider

Поля				
Имя	Модифика тор доступа	Тип	Назначение	
jwtSecret	private	jwtSecret	Строка-ключ для генерации и дешифрации токенов	
Методы				
Имя	Модифик атор доступа	Тип	Аргументы	Назначение
generateToken	public	String	String login	Создает новый токен
validateToken	public	boolean	String token	Проверка соответствия “полученный токен – пользователь”
getLoginFromToken	public	String	String token	Получает логин пользователя из переданного токена

## Описание полей, методов и свойств класса JwtFilter

Поля			
Имя	Модификат ор доступа	Тип	Назначение
AUTHORIZATION	private	String	Название заголовка http запроса, в котором содержится токен
jwtProvider	private	JwtProvider	Класс отвечающий за генерацию и валидацию токенов

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-Л1У				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**RU.17701729.04.01-01 81 01-1**

customUserDetailsService	private	CustomUserDetailsService		
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
doFilter	public	void	ServletRequest servletRequest, ServletResponse servletResponse, FilterChain filterChain	Определяет, имеет ли пользователь, с переданным в запросе токеном, доступ к запрашиваемому пути
getTokenFromRequest	private	String	HttpServletRequest request	Получает токен из заголовка http запроса

## Описание полей, методов и свойств класса MovementRequest

Класс имеет аннотацию lombok.Data => все “геттеры”, “сеттеры” и т.д. генерируются автоматически.

<b>Поля</b>				
Имя	Модификатор доступа	Тип	Аннотация	Назначение
longitude	private	double	@NotEmpty	Долгота
latitude	private	double	@NotEmpty	Широта
date	private	String	@NotEmpty	Дата и время, в которое было совершено перемещение

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-Л1У				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## RU.17701729.04.01-01 81 01-1

## Описание полей, методов и свойств класса RegistrationRequest

Класс имеет аннотацию lombok.Data => все “геттеры”, “сеттеры” и т.д. генерируются автоматически.

Поля				
Имя	Модификатор доступа	Тип	Аннотация	Назначение
login	private	String	@NotEmpty	Логин
password	private	String	@NotEmpty	Пароль
role	private	String		Роль пользователя, если не указана, то по умолчанию ставится роль “USER_ROLE”

## Описание полей, методов и свойств класса RoleEntity

Класс имеет аннотацию lombok.Data => все “геттеры”, “сеттеры” и т.д. генерируются автоматически.

Поля				
Имя	Модификатор доступа	Тип	Аннотация	Назначение
id	private	Integer	@Id, @GeneratedValue, @Column	Id сущности в таблице БД
name	private	String	@NotEmpty	Название роли

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛД				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



## RU.17701729.04.01-01 81 01-1

Описание полей, методов и свойств интерфейса RoleRepository

Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
findByName		RoleEntity	String name	Поиск сущности RoleEntity по полю "name" в базе данных

Описание полей, методов и свойств класса JwtFilter

Поля				
Имя	Модификатор доступа	Тип	Назначение	
jwtFilter	private	JwtFilter	Класс отвечающий за фильтрацию пользователей по ролям	
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
configure	protected	void	HttpSecurity http	Устанавливает настройки доступа к URL для разных ролей
passwordEncoder	public	PasswordEncoder		Возвращает новый экземпляр класса BCryptPasswordEncoder

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-Л1У				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## RU.17701729.04.01-01 81 01-1

## Описание полей, методов и свойств класса ShopEntity

Класс имеет аннотацию lombok.Data => все “геттеры”, “сеттеры” и т.д. генерируются автоматически.

Поля				
Имя	Модификатор доступа	Тип	Аннотация	Назначение
id	private	Integer	@Id, @GeneratedValue, @Column, @JsonView	Id сущности в таблице БД
name	private	String	@Column, @JsonView	Название магазина
longitude	private	double	@Column, @JsonView	Широта
latitude	private	double	@Column, @JsonView	Долгота
location	private	String	@Column, @JsonView	Адрес
type	private	int	@Column, @JsonView	Тип магазина
holder	private	UserEntity	@JsonBackReference, @ManyToOne, @Column, @JsonView	Пользователь “admin”, которому принадлежит магазин
img	private	byte[]	@Lob, @Column, @JsonView	Изображение-иконка магазина
stocks	private	List<StockEntity>	@JsonIgnore @JsonManagedReference, @JsonView, @OneToMany	Лист акций, которые принадлежат магазину
Методы				

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-Л1У				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## RU.17701729.04.01-01 81 01-1

Имя	Модификатор доступа	Тип	Аргументы	Назначение
addStock	public	void	StockEntity newStock	Добавляет переданную акцию в List stocks

## Описание полей, методов и свойств интерфейса ShopRepository

Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
findByName		ShopEntity	String name	Поиск сущности ShopEntity по полю "name" в базе данных

## Описание полей, методов и свойств класса ShopService

Поля				
Имя	Модификатор доступа	Тип	Аннотация	Назначение
shopRepository	private	ShopRepository	@Autowired	Экземпляр интерфейса для работы с БД
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
saveShop	public	ShopEntity	ShopEntity shop	Сохраняет новый экземпляр ShopEntity в базу данных

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛД				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## RU.17701729.04.01-01 81 01-1

getAllShops	public	List<ShopEntity>		Получает все магазины из базы данных
findById	public	ShopEntity	int id	Находит сущность ShopEntity в БД с переданным идентификатором

## Описание полей, методов и свойств класса StockEntity

Класс имеет аннотацию lombok.Data => все “геттеры”, “сеттеры” и т.д. генерируются автоматически.

Поля				
Имя	Модификатор доступа	Тип	Аннотация	Назначение
id	private	Integer	@Id, @GeneratedValue, @Column, @JsonView	Id сущности в таблице БД
name	private	String	@Column, @JsonView	Название магазина
promo	private	String	@Column, @JsonView	Промокод
oldPrice	private	double	@Column, @JsonView	Старая цена
newPrice	private	double	@Column, @JsonView	Новая цена
special	private	boolean	@Column, @JsonView	Указывает, является ли акция “специальной”
img	private	byte[]	@Lob, @Column, @JsonView	Изображение-иконка магазина
shop	private	List<StockEntity>	@JsonBackReference @ManyToOne, @JsonView, @Column	Магазин, которому принадлежит акция

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-Л1У				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## RU.17701729.04.01-01 81 01-1

Описание полей, методов и свойств интерфейса StockRepository

Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
findByName		StockEntity	String name	Поиск сущности StockEntity по полю "name" в базе данных

Описание полей, методов и свойств класса StockService

Поля				
Имя	Модификатор доступа	Тип	Аннотация	Назначение
stockRepository	private	StockRepository	@Autowired	Экземпляр интерфейса для работы с БД
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
saveStock	public	StockEntity	StockEntity stock	Сохраняет новый экземпляр StockEntity в базу данных
findById	public	StockEntity	int id	Находит сущность StockEntity в БД с переданным идентификатором

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛД				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## RU.17701729.04.01-01 81 01-1

Описание полей, методов и свойств класса TestSecurityController

Поля				
Имя	Модификатор доступа	Тип	Аннотация	Назначение
userService	private	UserService	@Autowired	Экземпляр класса для работы с БД
jwtProvider	private	UserService	@Autowired	Класс отвечающий за генерацию и валидацию токенов
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
getAdmin	public	ResponseEntity<UserEntity>	@RequestHeader("Authorization") String bearer	Обрабатывает запросы, приходящие на "/admin/get"
getUser	public	ResponseEntity<UserEntity>	@RequestHeader("Authorization") String bearer	Обрабатывает запросы, приходящие на "/user/get"

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛУ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## RU.17701729.04.01-01 81 01-1

## Описание полей, методов и свойств класса UserEntity

Класс имеет аннотацию lombok.Data => все “геттеры”, “сеттеры” и т.д. генерируются автоматически.

Поля				
Имя	Модификатор доступа	Тип	Аннотация	Назначение
id	private	Integer	@Id, @GeneratedValue, @Column	Id сущности в таблице БД
login	private	String	@Column	Логин
password	private	String	@Column	Пароль
role	private	RoleEntity	@ManyToOne, @JoinColumn	Роль пользователя
visitActions	private	List<VisitActionEntity>	@JsonManagedReference, @OneToMany	Все посещения, которые совершал пользователь
shops	private	List<ShopEntity>	@JsonManagedReference, @OneToMany	Магазины, которые принадлежат пользователю
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
addVisit	public	void	VisitActionEntity newVisit	Добавляет переданную сущность VisitActionEntity в List visitActions
addShop	public	void	ShopEntity newShop	Добавляет переданную сущность ShopEntity в List shops

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛД				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## RU.17701729.04.01-01 81 01-1

## Описание полей, методов и свойств интерфейса UserRepository

Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
findByLogin		UserEntity	String login	Поиск сущности UserEntity по полю "login" в базе данных

## Описание полей, методов и свойств класса UserService

Поля				
Имя	Модификатор доступа	Тип	Аннотация	Назначение
userRepository	private	UserEntityRepository	@Autowired	Экземпляр интерфейса для работы с БД
roleRepository	private	RoleRepository	@Autowired	Экземпляр интерфейса для работы с БД
passwordEncoder	private	PasswordEncoder	@Autowired	Экземпляр интерфейса для работы с БД
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
saveUser	public	UserEntity	UserEntity user	Сохраняет новый экземпляр UserEntity с ролью "USER_ROLE" в базу данных

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-Л1У				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



**RU.17701729.04.01-01 81 01-1**

saveAdmin	public	UserEntity	UserEntity user	Сохраняет новый экземпляр UserEntity с ролью “ADMIN_ROLE” в базу данных
findByLogin	public	UserEntity	String login	Поиск сущности UserEntity в БД по полю “login”
findByLoginAndPassword	public	UserEntity	String login, String password	Поиск сущности UserEntity в БД по полю “login” и полю “password”

## Описание полей, методов и свойств класса VisitActionEntity

Класс имеет аннотацию lombok.Data => все “геттеры”, “сеттеры” и т.д. генерируются автоматически.

Поля				
Имя	Модификатор доступа	Тип	Аннотация	Назначение
id	private	Integer	@Id, @GeneratedValue, @Column	Id сущности в таблице БД
user	private	UserEntity	@Column, @JsonBackReference, @ManyToOne, @JsonView	Пользователь - посетитель
shop	private	ShopEntity	@Column, @JsonBackReference, @ManyToOne, @JsonView	Магазин, который посетили
count	private	Integer	@Column, @JsonView	Кол-во посещений

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛЮ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**RU.17701729.04.01-01 81 01-1**

lastVisit	private	java.sql.Timestamp	@Column, @JsonView	Дата, время последнего посещения
-----------	---------	--------------------	--------------------	----------------------------------

Описание полей, методов и свойств интерфейса VisitActionRepository

Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
findById		VisitActionEntity	int id	Поиск сущности VisitActionEntity по полю “id” в базе данных

Описание полей, методов и свойств класса VisitActionService

Поля				
Имя	Модификатор доступа	Тип	Аннотация	Назначение
visitActionRepository	private	VisitActionRepository	@Autowired	Экземпляр интерфейса для работы с БД
Методы				
Имя	Модификатор доступа	Тип	Аргументы	Назначение
saveVisitAction	public	VisitActionEntity	VisitActionEntity newVisitAction	Сохраняет новый экземпляр VisitActionEntity в базу данных

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1-ЛД				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## 6. ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]