

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук

Кафедра Программирования и информационных технологий

Курсовой проект

## Корпоративный портал для обучения

09.03.04 Программная инженерия

Обучающийся \_\_\_\_\_ Кубленко П.В., 3 курс,

Обучающийся \_\_\_\_\_ Григорьев В.О., 3 курс

Обучающийся \_\_\_\_\_ Толстов М.В., 3 курс

Руководитель \_\_\_\_\_ А.В. Нужных, преподаватель

Воронеж 2020

# Содержание

<b>Содержание</b>	<b>2</b>
<b>Введение</b>	<b>3</b>
<b>Постановка задачи</b>	<b>5</b>
<b>Анализ предметной области</b>	<b>6</b>
Обзор существующих решений	6
Обобщение обзора существующих решений	7
Воронки	8
Организация программы	14
Обоснование архитектуры системы и выбора стека технологий	15
Описание диаграмм	17
<b>Реализация</b>	<b>34</b>
Скриншоты реализации приложения	34
Реализация серверной части приложения	48
<b>Тестирование</b>	<b>50</b>
<b>Заключение</b>	<b>53</b>

## Введение

Развитие бизнеса в условиях рыночной экономики предъявляет новые требования к системе образования. Если раньше высшее или среднее специальное образование было гарантией успешной деятельности, то на сегодняшний день так называемое «полезное знание» (useful knowledge) становится более востребованным, чем знание академическое, т.к. оно нацелено на конечный результат и приносит экономическую выгоду. Это, в свою очередь, приводит к росту спроса на специалистов, обладающих именно такими знаниями. Огромным преимуществом и фактором конкурентной борьбы для любой компании становятся ее сотрудники, а уровень ее развития определяется компетентностью персонала. Для компании становится все более важным не только повышение квалификации персонала, но и обучение его конкретным знаниям, умениям и навыкам, необходимым для осуществления профессиональной деятельности на конкретном рабочем месте. Подготовка персонала нового типа для постоянно развивающихся компаний потребовала новых форм обучения, одной из которых стало **корпоративное обучение**.

Корпоративное обучение может осуществляться в форме лекций, семинаров, тренингов, деловых игр или в комбинации этих форм. Если трудовая деятельность сотрудников предполагает постоянные изменения, то обучение носит систематический характер. К этой категории работников можно отнести, например, бухгалтеров, операторов программного обеспечения. Для решения единовременных конкретных задач проводятся одноразовые тренинги.

По формату корпоративное обучение может быть выездным или проводиться непосредственно на месте работы в офисе или на предприятии. В последнее время все большей популярностью пользуется дистанционная форма обучения с использованием мультимедийных технологий.

Крупные компании уже оценили преимущества корпоративного обучения посредством специального портала. Главное – это то, что больше не надо отправлять сотрудников в командировки на учебу, отрывать от производства, искать замену, нести дополнительные расходы. Персонал может эффективно обучаться не выходя из дома или прямо на рабочем месте, выделив для этого специальное время.

В данной курсовой работе рассматривается проблема создания простого, легковесного и не нагруженного излишней функциональностью, но обладающего необходимыми функциями веб-приложения для корпоративного обучения.

## **Глава 1. Постановка задачи**

Цель курсовой работы - реализовать веб-приложение для организации удаленного процесса обучения внутри одной компании, позволяющее сотрудникам развивать свои профессиональные навыки прямо на рабочем месте.

Приложение должно позволять сотрудникам, ответственным за корпоративное обучение, создавать обучающие курсы, публиковать туда полезные материалы и домашние задания, которые позволят рядовым сотрудникам компании получить и закрепить знания по какой-либо теме.

Приложение должно быть простым, ненагруженным функционалом, понятным пользователю, но при этом позволяющим наладить процесс обучения в компании.

Приложение должно отвечать некоторым требованиям безопасности, например, чтобы зарегистрироваться в нем могли только сотрудники компании.

Приложение должно иметь простой, не перегруженный интерфейс, но при этом все должно быть доступно и хорошо читаемо.

## **Глава 2. Анализ предметной области**

В данной главе рассматриваются уже существующие решения для организации корпоративного обучения, а также представлено описание предметной области.

### **2.1. Обзор существующих решений**

Для успешного и эффективного функционирования компании необходимо, чтобы её работники непрерывно повышали свою квалификацию. Для этого бывает разумно привлекать своих же работников, которые имеют обширный опыт работы в некоторой сфере, для того чтобы они передали свои знания неопытным коллегам.

#### **2.1.1. Moodle**

Бесплатная система электронного обучения. Одна из наиболее популярных платформ электронного обучения. На ее базе можно создать специализированную платформу для развития сотрудников.

Но у Moodle непростая организация обучения. Неопытный пользователь может потратить сутки на создание одной лекции. Система обладает редактором для создания лекций, тестов, заданий. Лекция - основной модуль в Moodle, в ней рассказывается теория.

В Moodle курс состоит из тем, которые содержат контент. Контент создается или загружается извне. Помимо курсов есть возможность организации тестирования.

Moodle поддерживает текстовый контент (doc, pdf, xls, csv), изображения (jpeg, png, gif), видео (flv, f4v, f4p, mp4, m4v, m4a, 3gp, mov) и аудио (mp3, aac, flac, m4a, oga, ogg, wav).

За счет интеграции с другими платформами при помощи Moodle можно создать платформу для продажи электронного контента.

#### **2.1.2. Learn DataArt**

Система электронного обучения, в которой можно создать курсы для развития сотрудников. В ней курс состоит из уроков, которые содержат контент и задание, необходимое для выполнения, поле отправки выполненного задания и/ комментария, которые будут доступны для проверки инструкторам.

Оценка производится инструктором вручную, система не предоставляет возможности для проведения тестирования. Инструктор также может оставить комментарий к работе слушателя.

Аналогично вышеупомянутой системе контент создается или загружается извне.

Круг лиц имеющих доступ к системе ограничивается администрацией системы.

## **2.2. Обобщение обзора существующих решений**

Проведя обзор существующих электронных систем для обучения, можно сделать вывод о том, что все они содержат некоторые курсы, содержащие контент. Обе рассмотренные системы схожи в способе подачи материала, но различны в проверке знаний. В первой системе есть возможность автоматизации проверки знаний. Во второй такой возможности нет, так как она зачастую используется для обучения в формате один инструктор - один ученик (или небольшая группа).

	Moodle	Learn Dataart	Наш портал
Создание курсов	+	+	+
Загрузка контента	+	+	+

Автоматизация контроля обучения	+	-	-
Открытость системы	+	-	-
Продажа курсов	+	-	-
Формат	Лектор - слушатели	Ментор - менти	Ментор - менти

Таблица 1. Обзор существующих решений

### 2.3. Воронки

Реализованы 3 воронки, которые будут использоваться для анализа использования системы.

Воронка отслеживающая заинтересованность слушателей курсов состоит из 5 шагов:

1. Посещение личного кабинета (вход в систему).
2. Просмотр всех курсов, на которых числится слушателем.
3. Просмотр конкретного курса.
4. Просмотр урока.
5. Отправка домашней работы (прикрепление файла и/или написание комментария).



Шаг 1

Личный кабинет

Условие

url: содержит

/user/me

+

Шаг 2

Курсы

Условие

url: содержит

/user/me/course

+

Шаг 3

Курс

Условие

url: регулярн...

.\*user/me/course/d+

+

Шаг 4

Урок

Условие

url: регулярн...

.\*user/me/course/d+/lessons/d+

+

Шаг 5

Домашняя работа отправлена

Условие

событие: и...

SEND\_COMMENT

Код цели для сайта

ym(61893537,'reachGoal','SEND\_COMMENT')

Скопировать код

или

событие: и...

DOWNLOAD\_MATERIAL

Код цели для сайта

ym(61893537,'reachGoal','DOWNLOAD\_MATERIAL')

Скопировать код

+

Рисунок 1. Настройки воронки отслеживающей заинтересованность слушателей курсов

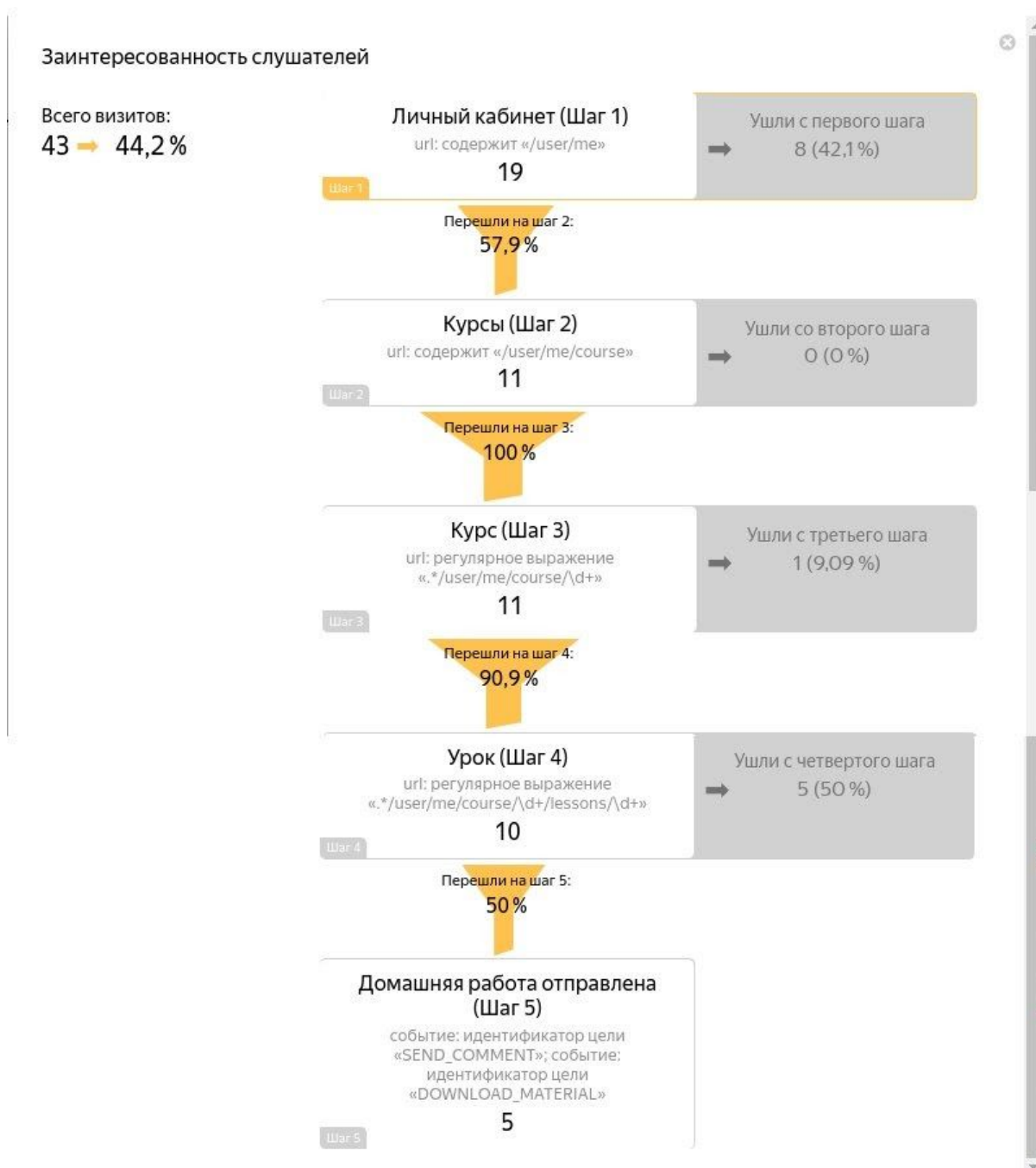


Рисунок 2. Воронка отслеживающая заинтересованность слушателей курсов

Воронка отслеживающая заинтересованность инструкторов курсов состоит из 5 шагов:

1. Просмотр всех курсов, на которых числится инструктором.
2. Посещение страницы курса.

3. Просмотр домашних работ к уроку.
4. Просмотр конкретной домашней работы.
5. Оценивание домашней работы (Поставить отметку или оставить комментарий).

<b>Шаг 1</b>	<input style="width: 90%;" type="text" value="Курсы"/> <span style="float: right;">✕ ✕</span>	
Условие	<div style="display: flex; align-items: center;"> <div style="border: 1px solid #ccc; padding: 2px; margin-right: 5px;">url: регулярн...</div> <div style="border: 1px solid #ccc; padding: 2px; flex-grow: 1;">.* /courses</div> <div style="margin-left: 10px;">✕</div> </div>	+

<b>Шаг 2</b>	<input style="width: 90%;" type="text" value="Курс"/> <span style="float: right;">✕ ✕</span>	
Условие	<div style="display: flex; align-items: center;"> <div style="border: 1px solid #ccc; padding: 2px; margin-right: 5px;">url: регулярн...</div> <div style="border: 1px solid #ccc; padding: 2px; flex-grow: 1;">.* /course /.*</div> <div style="margin-left: 10px;">✕</div> </div>	+

<b>Шаг 3</b>	<input style="width: 90%;" type="text" value="Домашние работы к уроку"/> <span style="float: right;">✕ ✕</span>	
Условие	<div style="display: flex; align-items: center;"> <div style="border: 1px solid #ccc; padding: 2px; margin-right: 5px;">url: регулярн...</div> <div style="border: 1px solid #ccc; padding: 2px; flex-grow: 1;">.* /lesson /homeworks ?lessonId = \d +</div> <div style="margin-left: 10px;">✕</div> </div>	+

<b>Шаг 4</b>	<input style="width: 90%;" type="text" value="Конкретная домашняя работа"/> <span style="float: right;">✕ ✕</span>	
Условие	<div style="display: flex; align-items: center;"> <div style="border: 1px solid #ccc; padding: 2px; margin-right: 5px;">url: регулярн...</div> <div style="border: 1px solid #ccc; padding: 2px; flex-grow: 1;">/lesson /homework ?lessonId = \d + &amp; userId = .*</div> <div style="margin-left: 10px;">✕</div> </div>	+

<b>Шаг 5</b>	<input style="width: 90%;" type="text" value="Оценка др"/> <span style="float: right;">✕ ✕</span>	
Условие	<div style="display: flex; align-items: center;"> <div style="border: 1px solid #ccc; padding: 2px; margin-right: 5px;">событие: и...</div> <div style="border: 1px solid #ccc; padding: 2px; flex-grow: 1;">SEND_COMMENT</div> <div style="margin-left: 10px;">✕ ✕</div> </div>	
	<div style="display: flex; align-items: center;"> <div style="flex-grow: 1;">           Код цели для сайта  <div style="background-color: #f9f9f9; padding: 5px; border: 1px solid #f5f5f5;">ym(61893537,'reachGoal','SEND_COMMENT')</div> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-left: 10px; flex-grow: 0 0 100px;">Скопировать код</div> </div>	
или	<div style="display: flex; align-items: center;"> <div style="border: 1px solid #ccc; padding: 2px; margin-right: 5px;">событие: и...</div> <div style="border: 1px solid #ccc; padding: 2px; flex-grow: 1;">SET_MARK</div> <div style="margin-left: 10px;">✕ ✕</div> </div>	
	<div style="display: flex; align-items: center;"> <div style="flex-grow: 1;">           Код цели для сайта  <div style="background-color: #f9f9f9; padding: 5px; border: 1px solid #f5f5f5;">ym(61893537,'reachGoal','SET_MARK')</div> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-left: 10px; flex-grow: 0 0 100px;">Скопировать код</div> </div>	
		+

Рисунок 3. Настройка воронки отслеживающей заинтересованность  
инструкторов курсов

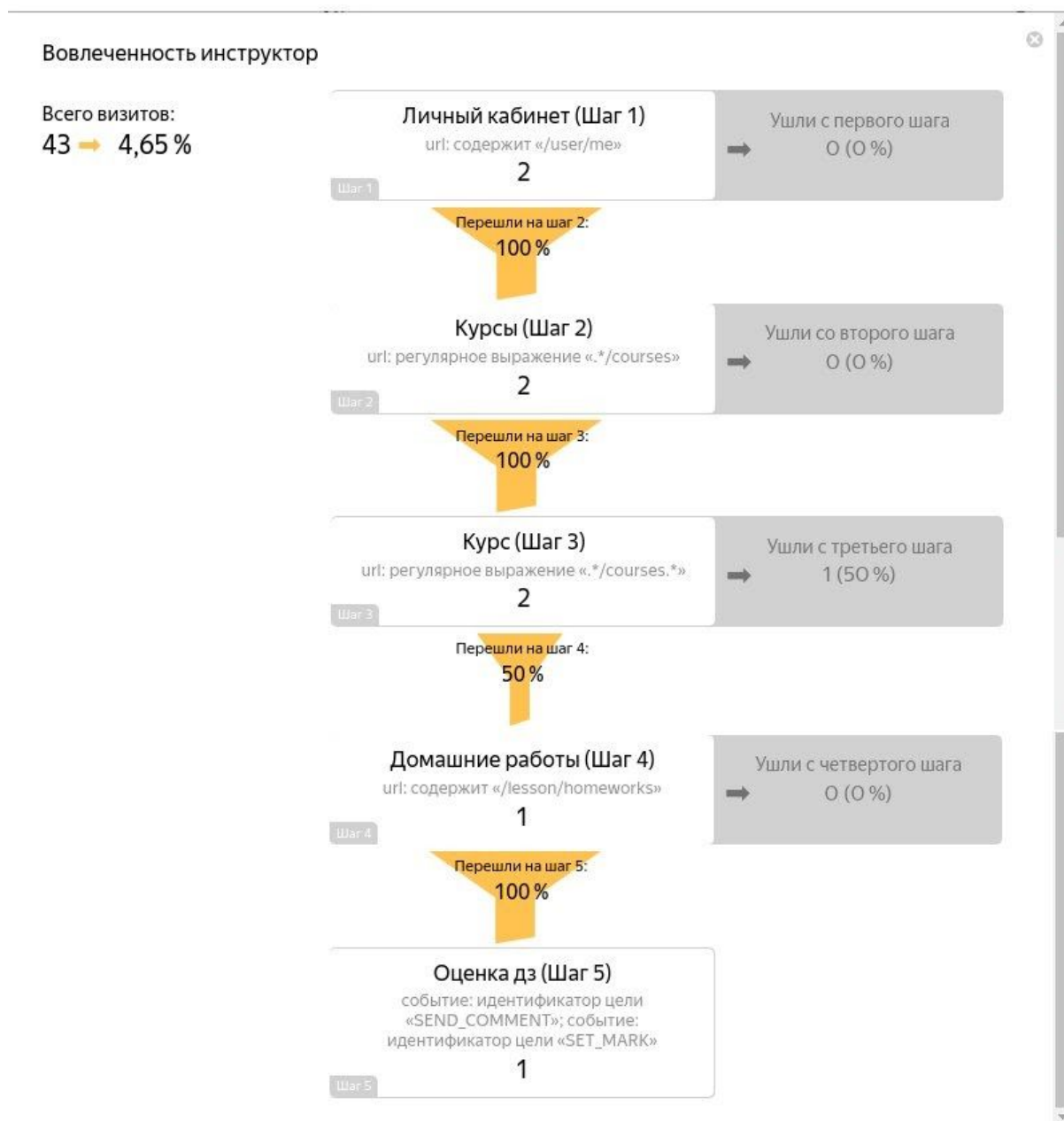


Рисунок 4. Воронка отслеживающая заинтересованность инструкторов  
курсов

Воронка отслеживающая действия администраторов состоит из 4 шагов:

1. Посещение личного кабинета (вход в систему).

2. Просмотр списка всех курсов.
3. Просмотр конкретного курса.
4. Добавление или удаление инструктора.

Шаг 1

Личный кабинет

Условие

url: содержит

/user/me

+

Шаг 2

Курсы

Условие

url: регулярн...

./courses

+

Шаг 3

Курс

Условие

url: регулярн...

./courses.\*

+

Шаг 4

Домашние работы

Условие

url: регулярн...

./lesson/homeworks?lessonId=id+

+

Шаг 5

Оценка дз

Условие

событие: и...

SEND\_COMMENT

Код цели для сайта

ym(61893537,'reachGoal','SEND\_COMMENT')

Скопировать код

или

событие: и...

SET\_MARK

Код цели для сайта

ym(61893537,'reachGoal','SET\_MARK')

Скопировать код

+

Добавить шаг

Рисунок 5. Настройка воронки отслеживающей действия администраторов

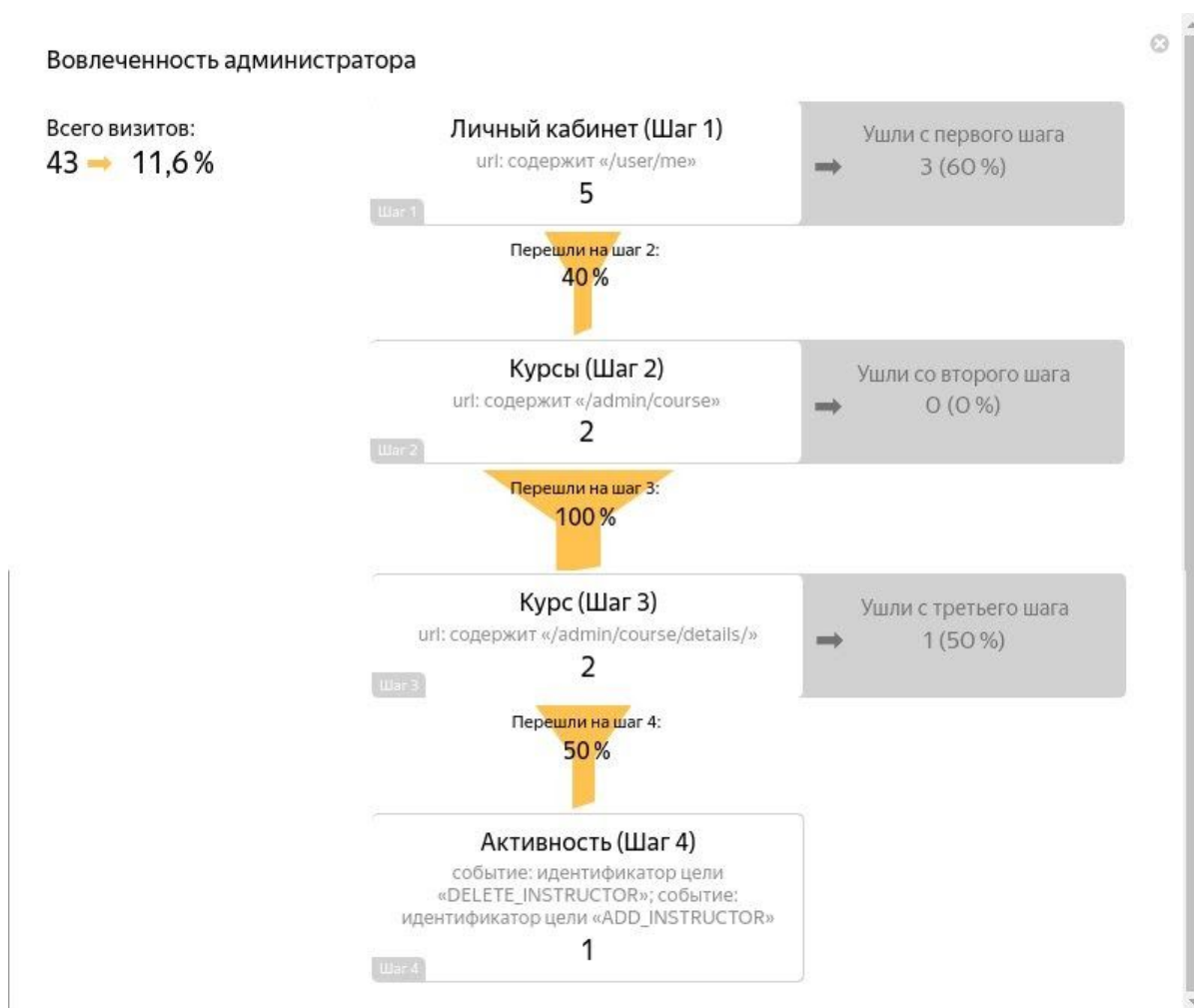


Рисунок 6. Воронка отслеживающая действия администраторов

## 2.4. Организация программы

### 2.4.1. Словесное описание

В разрабатываемую систему будет введена ролевая система, делящая пользователей на категории ответственных за обучение и обучающихся.

Ответственные за обучение (инструкторы) будут создавать учебные курсы, добавлять в них учебные материалы и домашние задания. Обычным пользователям (слушателям) раздается доступ к конкретным курсам, на которых они могут изучать учебные материалы и выполнять домашние задания. Общение инструкторов и слушателей осуществляется

путем комментариев к домашним заданиям. Таким образом при разработке системы были учтены все плюсы систем-аналогов.

В будущем система может быть расширена, например, добавление возможности нескольким организациям проводить свои курсы на портале по аналогии с Moodle. Также планируется ввод системы платных подписок для пользования приложением. Таким образом, планируется совместить в системе как можно больше сильных сторон продуктов-аналогов для получения качественного продукта.

## **2.5. Обоснование архитектуры системы и выбора стека технологий**

Поскольку предполагалось то, что в будущем возможности приложения будут расширяться, и оно дорастет до уровня достаточно сложного Enterprise-приложения, было принято решение использовать язык программирования, который используют для разработки таких платформ. То есть выбор остановился между C# и Java.

Дальнейший выбор сводился к выбору фреймворка. Выбор производился из следующих:

ASP.NET / C#

Spring Framework / Java

Выбор был сделан в пользу Spring из-за его огромнейшей экосистемы, в ее состав входит даже фреймворк для фреймворков - Spring Boot. Spring - легковесный каркас для построения приложений на джаве. Его легковесный характер сводится к тому, что для его использования нужно внести минимум изменений в прикладной код, если вообще придется. То есть если будет принято решение больше не использовать Spring, то реализовать это будет достаточно просто.

Система представляет собой RESTful API, позволяющий получить доступ к сервису с использованием http/https. Поскольку REST - это не стандарт и не протокол, а архитектурный подход, что в нашем случае упрощает разработку. Основной его конкурент - SOAP - это целое семейство протоколов, что делает его хорошо подходящим для интеграции двух крупных Enterprise-систем.

В будущих версиях продукта допускается возможность реализации приложений для мобильных устройств, что является дополнительным поводом выбрать REST, поскольку SOAP плохо подходит для предоставления веб-сервисов для мобильных приложений.

Таким образом, было решено, что использование SOAP нецелесообразно и излишне сложно для поставленной задачи

Итак, RESTful архитектура была выбрана потому что она позволяет:

- Написать легко масштабируемое приложение
- Легко изменять старый и добавлять новый функционал
- Легко тестировать функционал сервера (например с помощью программы Postman)
- Данные, отправляемые серверу, передаются в таком виде, в каком они есть (например, не оборачиваются в xml как в SOAP)

Для клиентского приложения был выбран Angular. Один из его плюсов заключается в том, что он полностью написан на TypeScript

Система представляет собой RESTful API, включающая в себя все основные свойства этой архитектуры, например такие как: модель “клиент-сервер”, отсутствие состояния клиента на сервере.

Данная архитектура была выбрана потому что она позволяет:

- Написать легко масштабируемое приложение



- Легко изменять старый и добавлять новый функционал
- Легко тестировать функционал сервера (например с помощью программы Postman)
- Данные, отправляемые серверу, передаются в таком виде, в каком они есть (например, не оборачиваются в xml как в SOAP)

## **2.6. Описание диаграмм**

### **2.6.1. Диаграмма прецедентов**

На диаграмме показаны все актеры системы, и варианты использования для них.

1. Для зарегистрированного пользователя:
  - Зарегистрироваться
2. Для зарегистрированного пользователя
  - Войти в систему
  - Выйти из системы
  - Изменить профиль
3. Для администратора
  - Удалить пользователя
  - Сгенерировать инвайт-токен для регистрации пользователя
  - Посмотреть информацию о всех курсах
  - Наделить и забрать привилегии у пользователя
  - Добавить/удалить инструкторов с курса
4. Для инструктора
  - Создать и редактировать учебный курс
  - Добавить/удалить урок в конкретном курсе
  - Заполнить конкретный урок учебными материалами и заданиями

- Дать слушателю доступ к курсу
- Просматривать и проверять домашние задания

5. Для слушателя

- Просмотреть доступные для него курсы
- Просмотреть уроки на этом курсе
- Просмотреть задания к нему
- Отправить ответ на задание. Расширяется вариантами:

Оставить комментарий

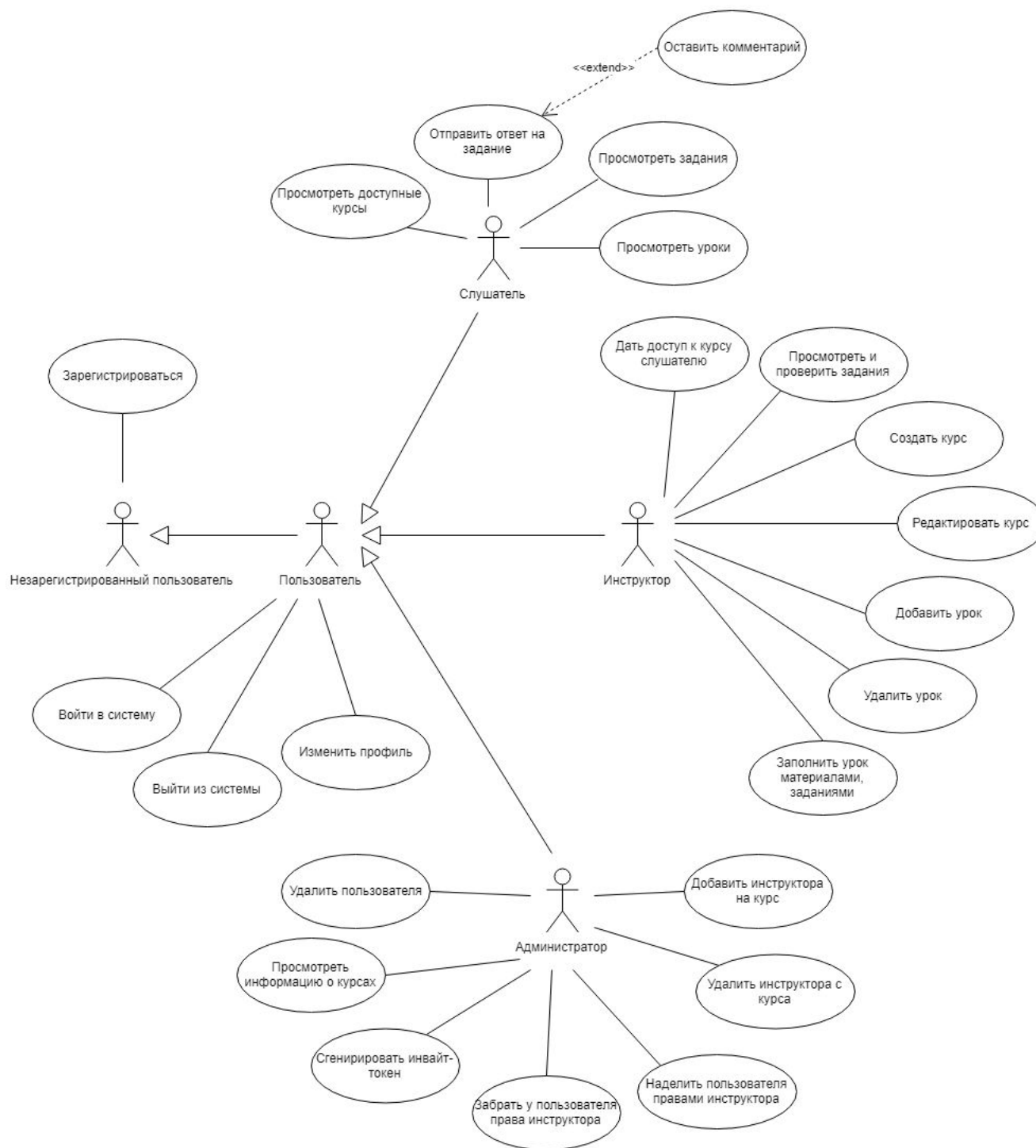


Рисунок 7. Диаграмма прецедентов

## 2.6.2. Диаграмма классов

Бекенд составляющая состоит из 8 пакетов

- `config` - содержит классы, в которых конфигурируется приложение;

- `controllers` - классы, взаимодействующие с запросами;

- `exceptions` - классы собственных исключений;

- `model` - классы, использованные для объектно-ориентированного отображения данных, содержащихся в базе данных;

- `payload` - классы, отображающие часть данных запросов и ответов;

- `repositories` - интерфейсы, при помощи которых происходит взаимодействие с базой данных;

- `security` - классы, использующиеся для авторизации и аутентификации;

- `services` - классы, содержащие бизнес логику приложения.

Далее приведены диаграммы классов некоторых пакетов.

На рисунке представлена диаграмма классов пакета `model`. Эти классы используются для отображения данных из базы данных. Отображение реляционной модели на объектно-ориентированную и наоборот происходит при помощи `Hibernate` и аннотаций, использованных в этих классах.

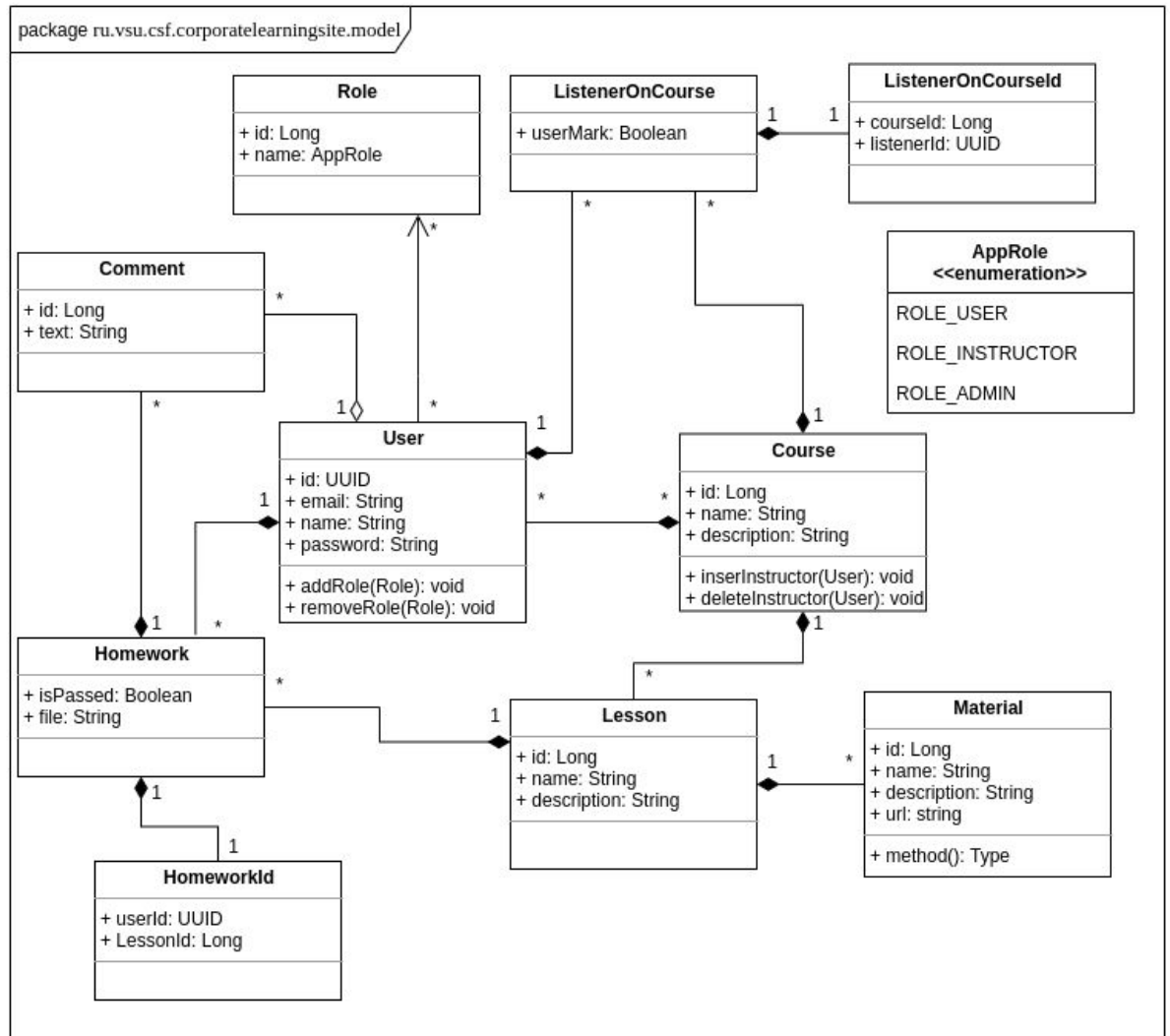


Рисунок 8. Диаграмма классов пакета model

На рисунке представлена диаграмма классов пакета repositories. В нем содержатся интерфейсы, расширяющие интерфейс JpaRepository. JpaRepository содержит основные CRUD-операции, но некоторые репозитории расширяют его, добавляя операции, необходимые для работы приложения.

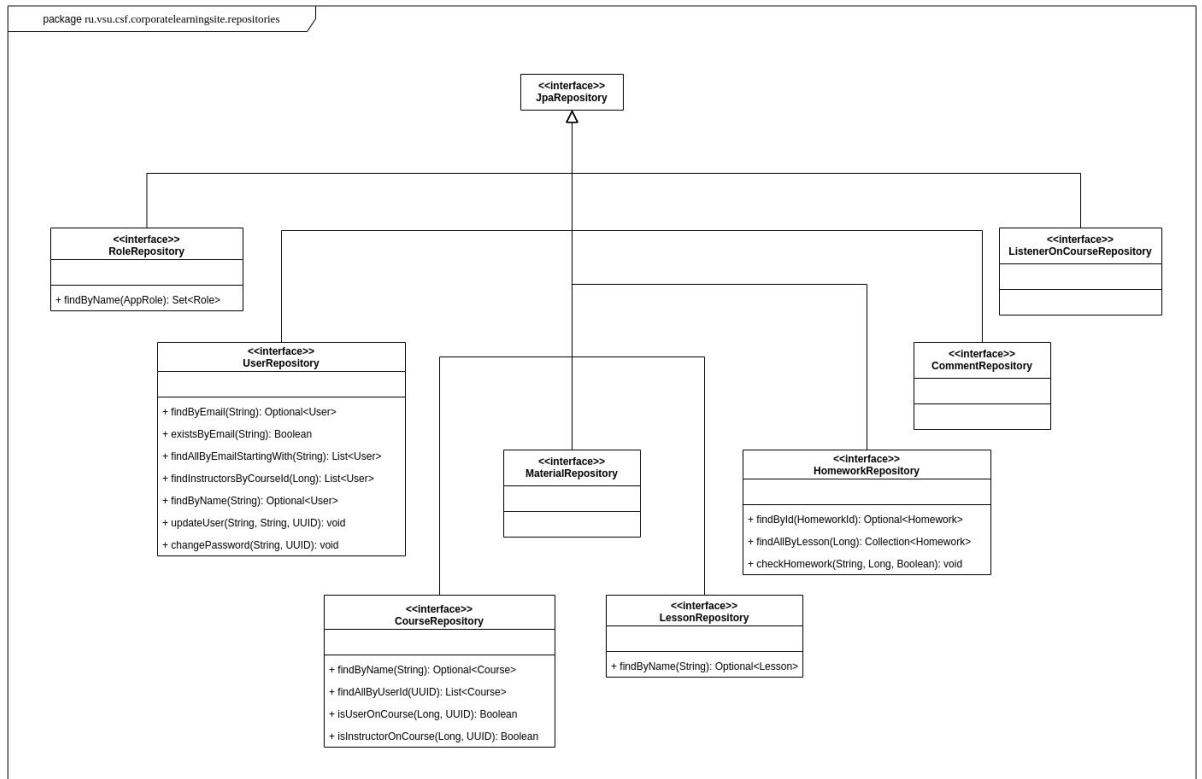


Рисунок 9. Диаграмма классов пакета repositories

### 2.6.3. Диаграмма объектов

На данной диаграмме изображены все объекты системы в конкретный момент времени. В данный момент в системе зарегистрированы 2 пользователя: обычный пользователь USER\_1 и USER\_2 с ролями инструктора и администратора. Также существует один курс COURSE\_1, на котором есть один урок LESSON\_1, который содержит один материал MATERIAL\_1. USER\_1 числится слушателем на этом курсе, а USER\_2 инструктором. USER\_1 сделал домашнюю работу HOMEWORK\_1 к уроку LESSON\_1 и оставил к ней комментарий COMMENT\_1. USER\_2 не зачел эту домашнюю работу и оставил к ней комментарий COMMENT\_2.

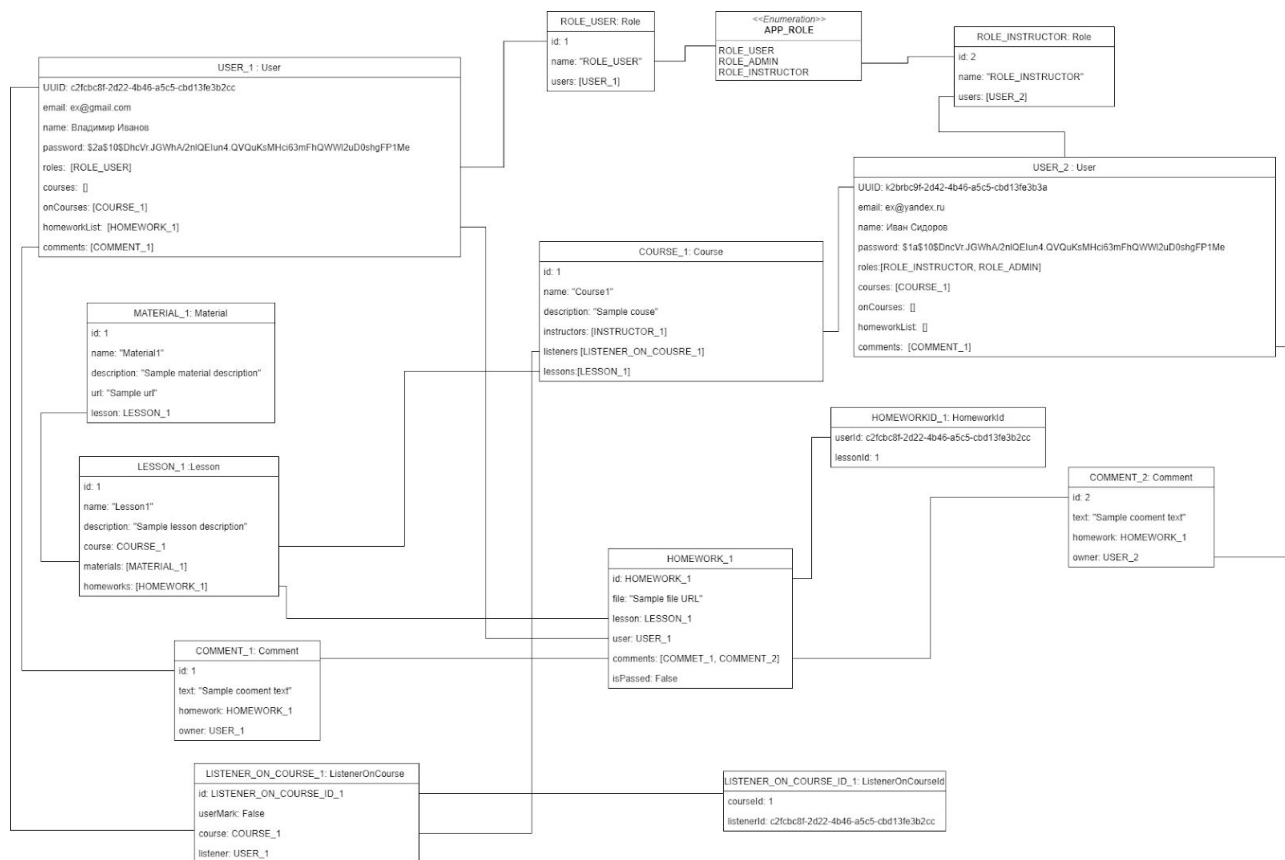


Рисунок 10. Диаграмма объектов

## 2.6.4. Диаграммы последовательностей

### 2.6.4.1. Проверка домашнего задания

На диаграмме показана последовательность действий при проверке домашнего задания инструктором. Сначала инструктор отправляет запрос на сохранение статуса проверки дз. На сервере сначала проходит проверка на наличие у отправившего запрос прав на выполнение этой операции, и если все корректно сохраняет результат проверки в БД и сообщает об успешном выполнении операции, иначе возвращается сообщение об ошибке.

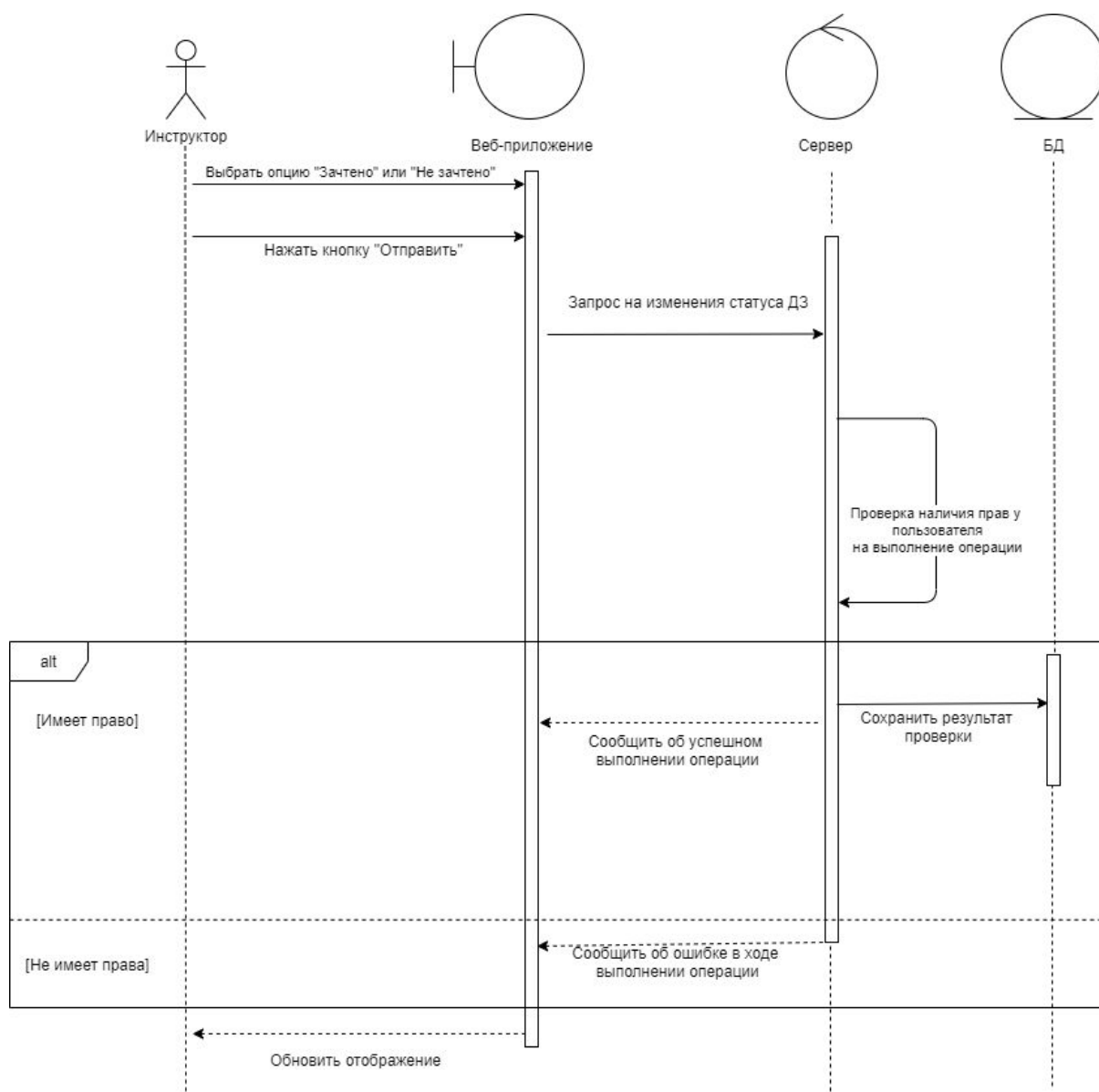


Рисунок 11. Диаграмма последовательностей проверки домашнего задания

#### 2.6.4.2. Добавление материала инструктором

На диаграмме показана последовательность действий при добавлении материала инструктором. Сначала инструктор выбирает нужный файл для загрузки и отправляет запрос на загрузку материала. На сервере сначала проходит проверка размера материала, и если размер файла не превышает допустимого значения, сохраняются данные



материала в БД и сообщается об успешной загрузке материала, иначе возвращается сообщение об превышении размера материала.

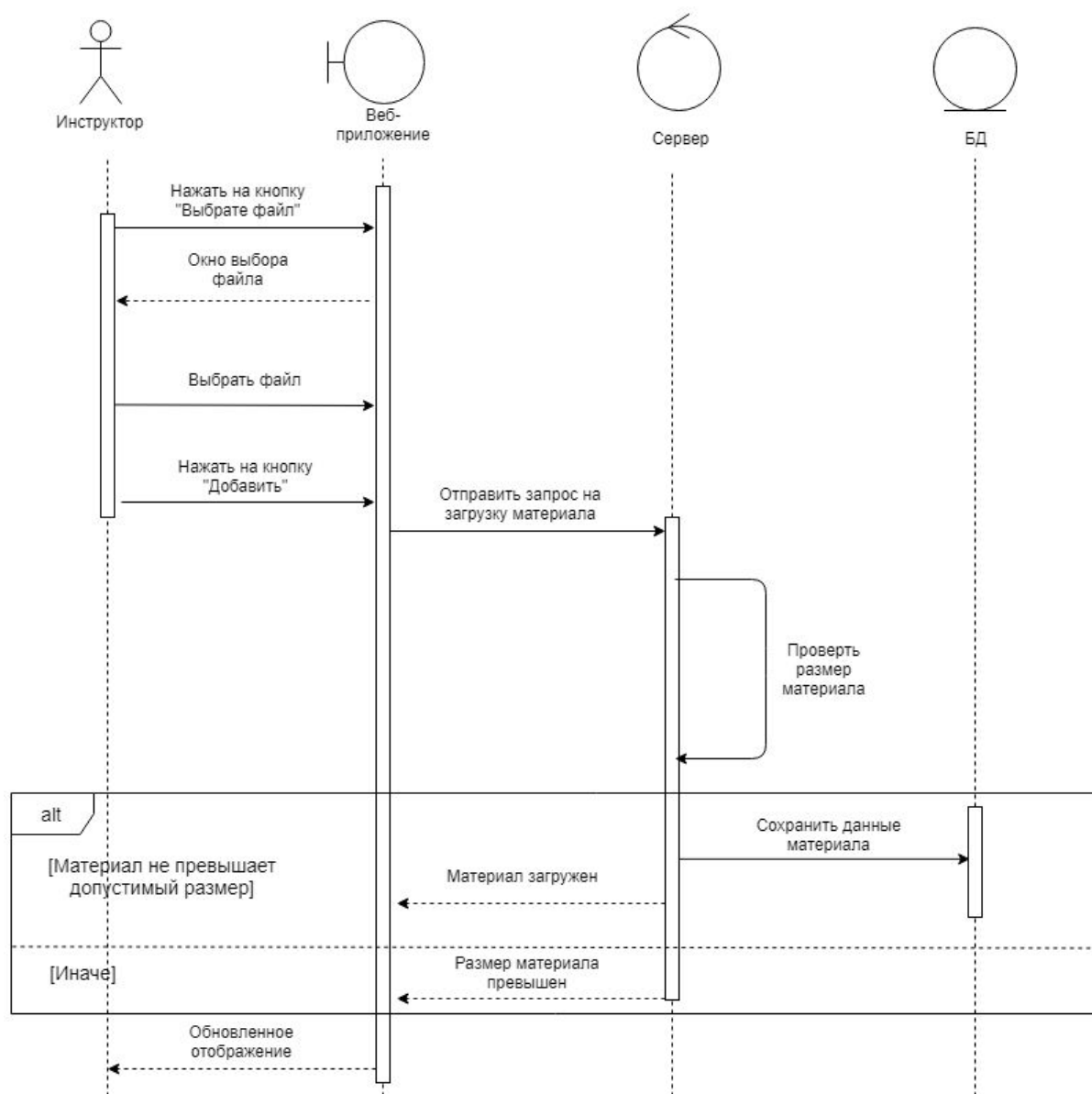


Рисунок 12. Диаграмма последовательностей добавления материала инструктором

#### 2.6.4.3. Добавление комментария

На диаграмме показана последовательность действий пользователя при добавлении комментария к домашней работе. Сначала пользователь вводит комментарий и отправляет запрос на его добавление. На сервере происходит проверка права пользователя на комментирование этой

домашней работы. Пользователь имеет право комментировать домашнюю работу только в двух случаях:

- он является её владельцем;
- он является инструктором на курсе, содержащему урок, к которому была выполнена домашняя работа;

Если добавление комментария произошло успешно, то возвращается сообщение об успехе, иначе сообщение ошибки.

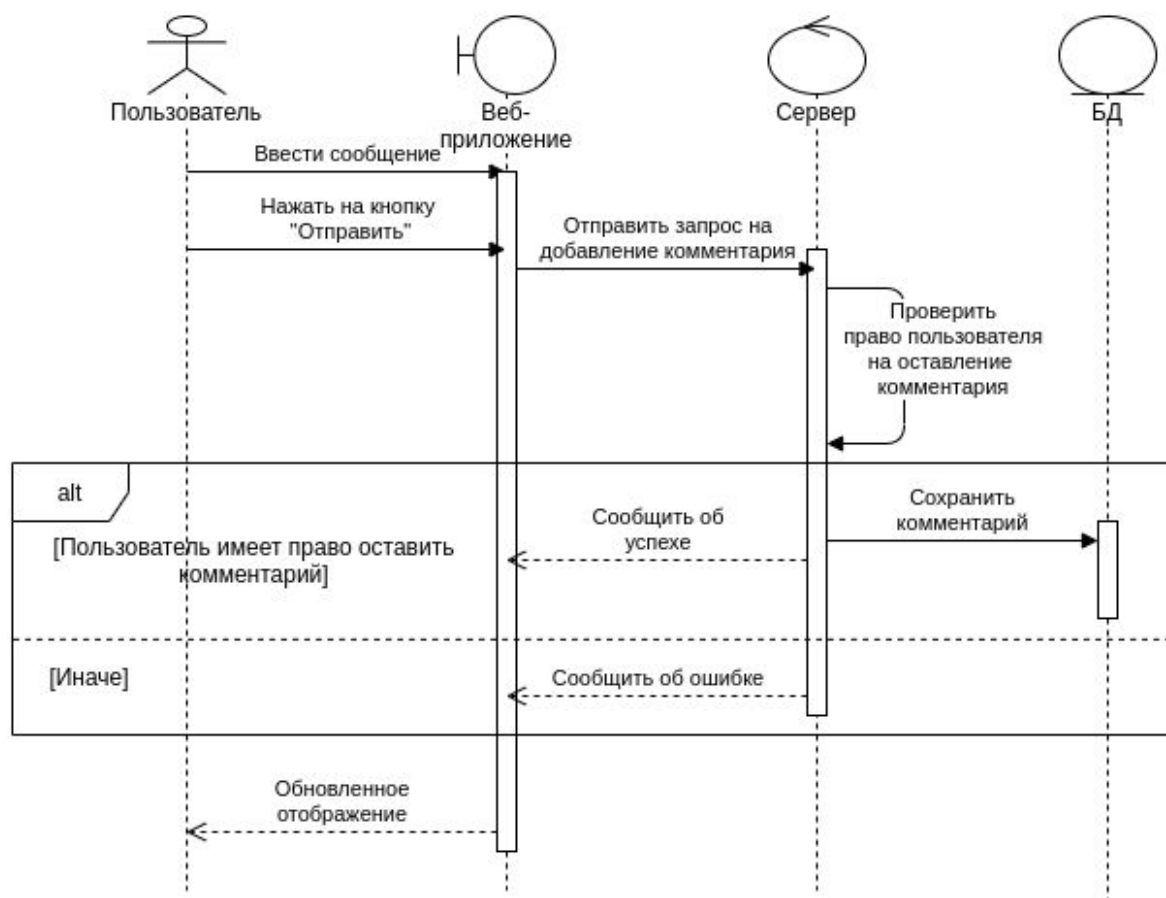


Рисунок 13. Диаграмма последовательностей добавления комментария

## 2.6.5. Диаграммы взаимодействия

### 2.6.5.1. Проверка домашнего задания

На рисунке представлена диаграмма взаимодействия при проверке домашнего задания инструктором.



Рисунок 14. Диаграмма взаимодействия проверки домашнего задания

### 2.6.5.2. Добавление материала инструктором

На рисунке представлена диаграмма взаимодействия добавления материала инструктором.

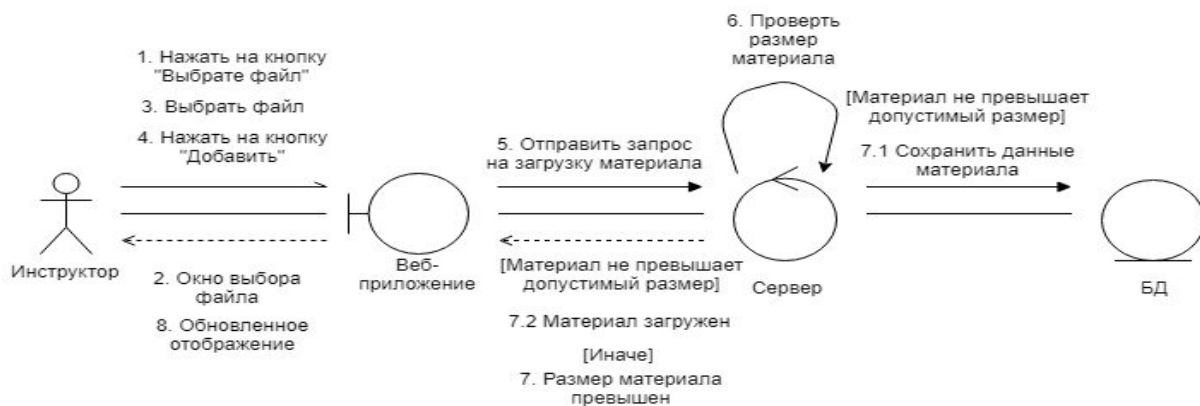


Рисунок 15. Диаграмма взаимодействия добавления материала инструктором

### 2.6.5.3. Добавление комментария

На рисунке представлена диаграмма взаимодействия добавления комментария к домашней работе.

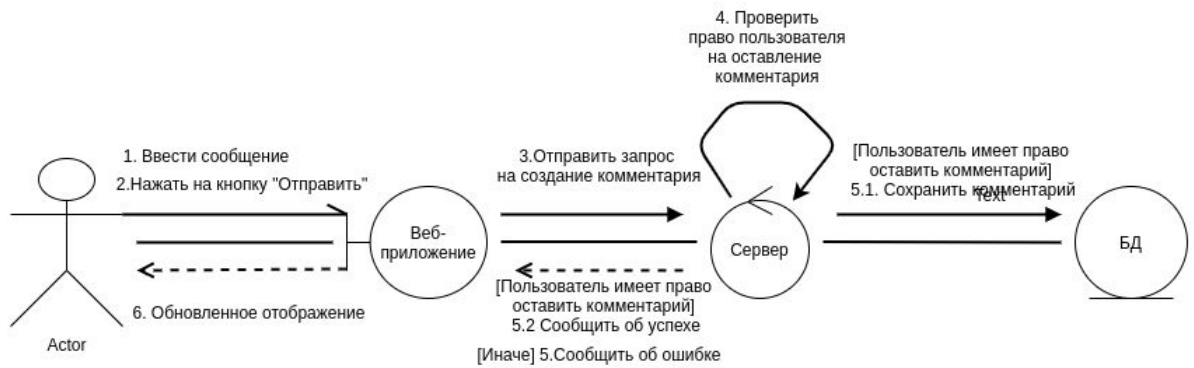


Рисунок 16. Диаграмма взаимодействия добавления комментария

### 2.6.6. Диаграмма состояний

На рисунке представлена диаграмма состояний системы. На данной диаграмме показаны все возможные состояния системы в которых может находиться система.

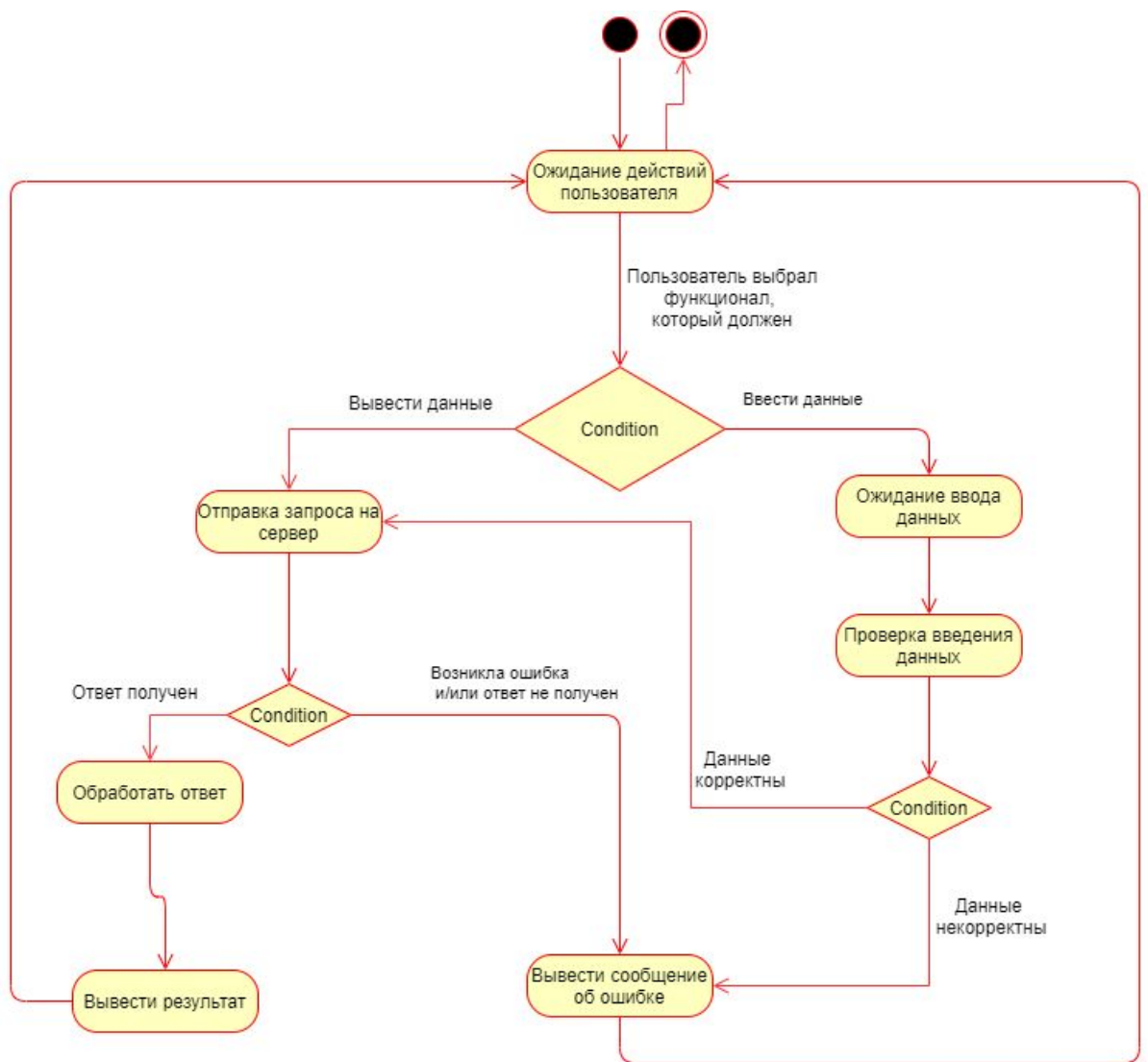


Рисунок 17. Диаграмма состояний

## 2.6.7. Диаграммы активностей

### 2.6.7.1. Проверка домашнего задания

На рисунке изображена диаграмма активности при проверке домашнего задания инструктором

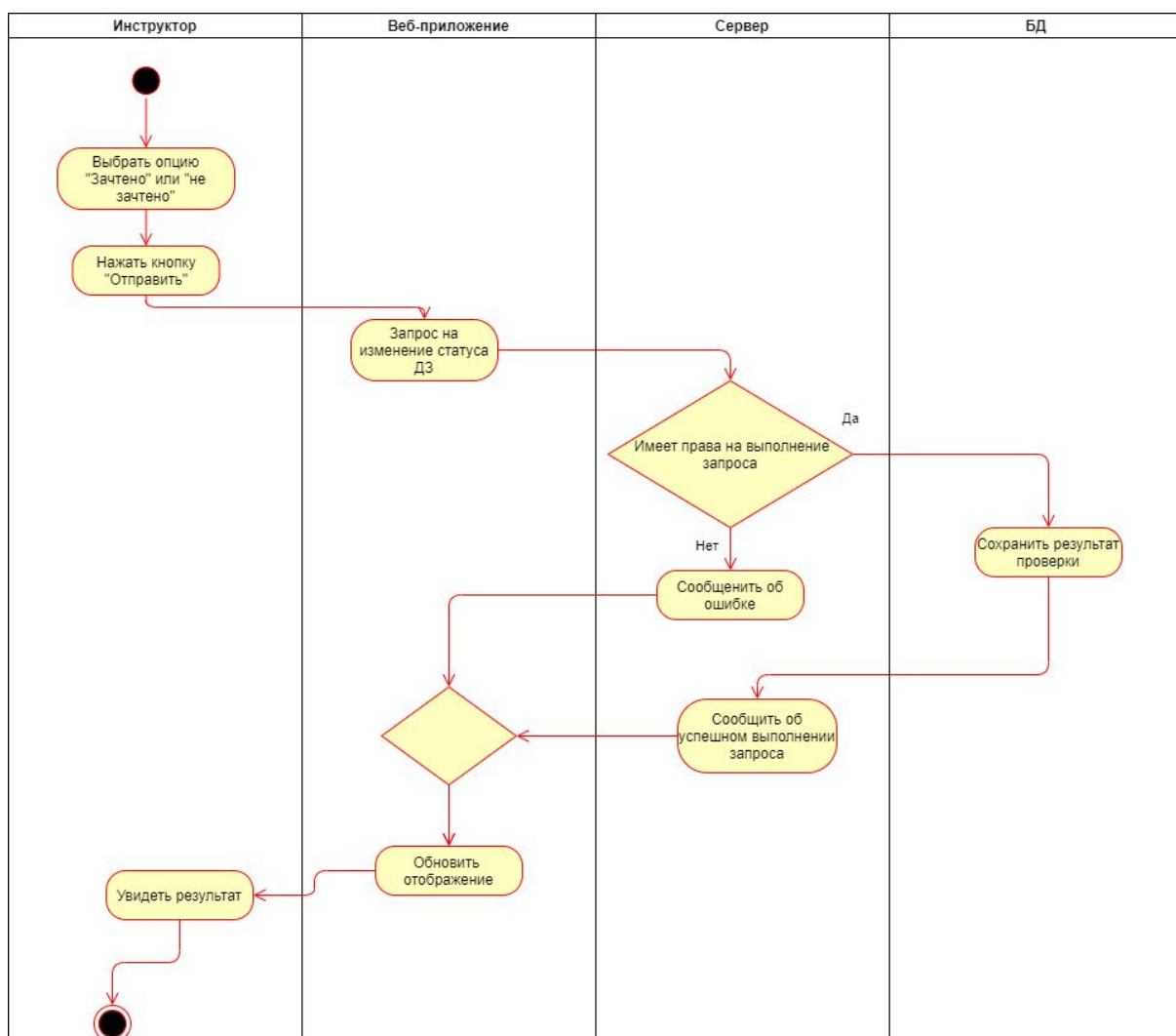


Рисунок 18. Диаграмма активностей проверки домашнего задания

#### 2.6.7.2. Добавление материала инструктором

На рисунке изображена диаграмма активности добавления материала инструктором.

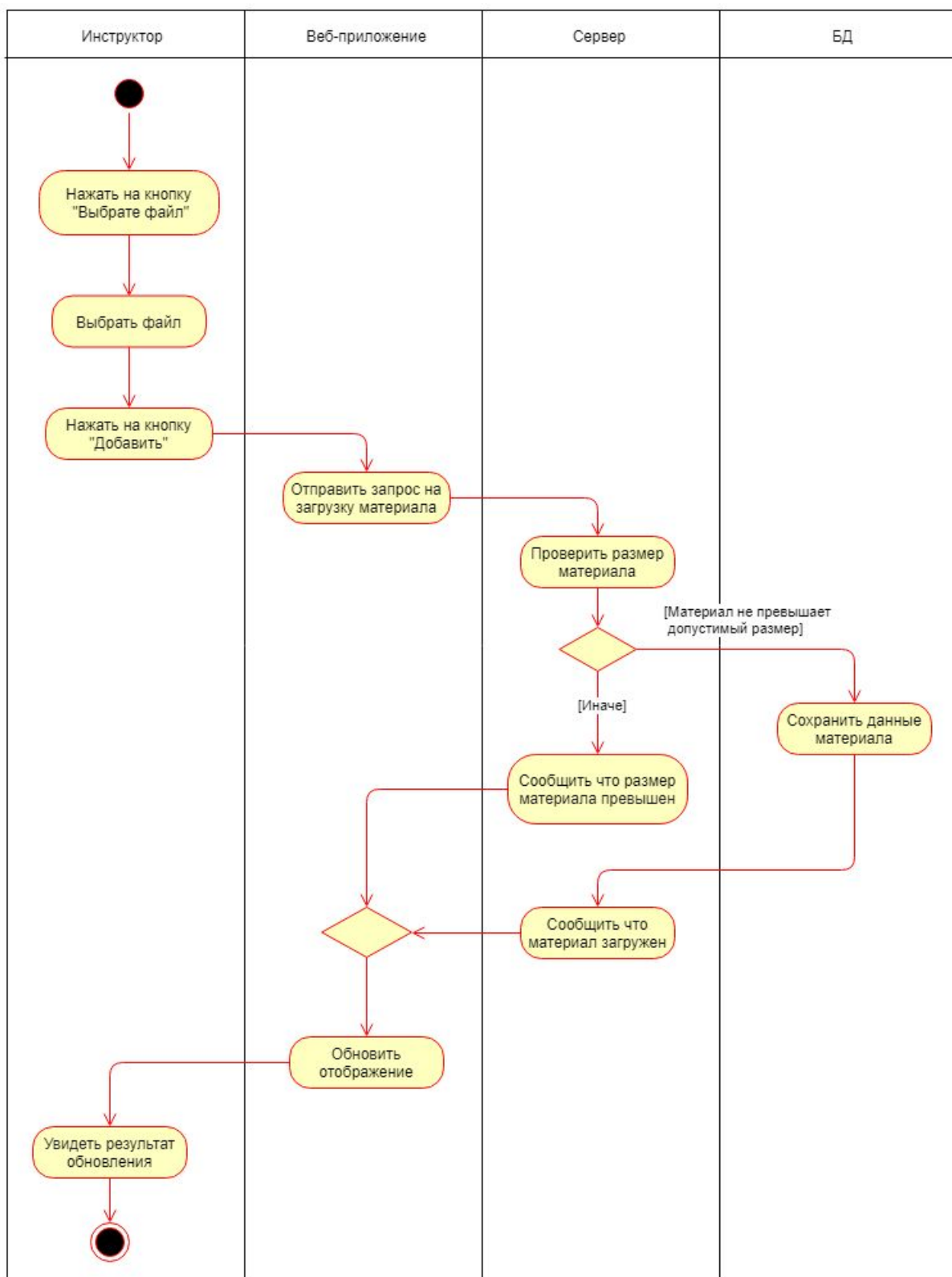


Рисунок 19. Диаграмма активностей добавления материала инструктором

### 2.6.7.3. Добавление комментария

На рисунке изображена диаграмма активности добавления комментария к домашней работе.

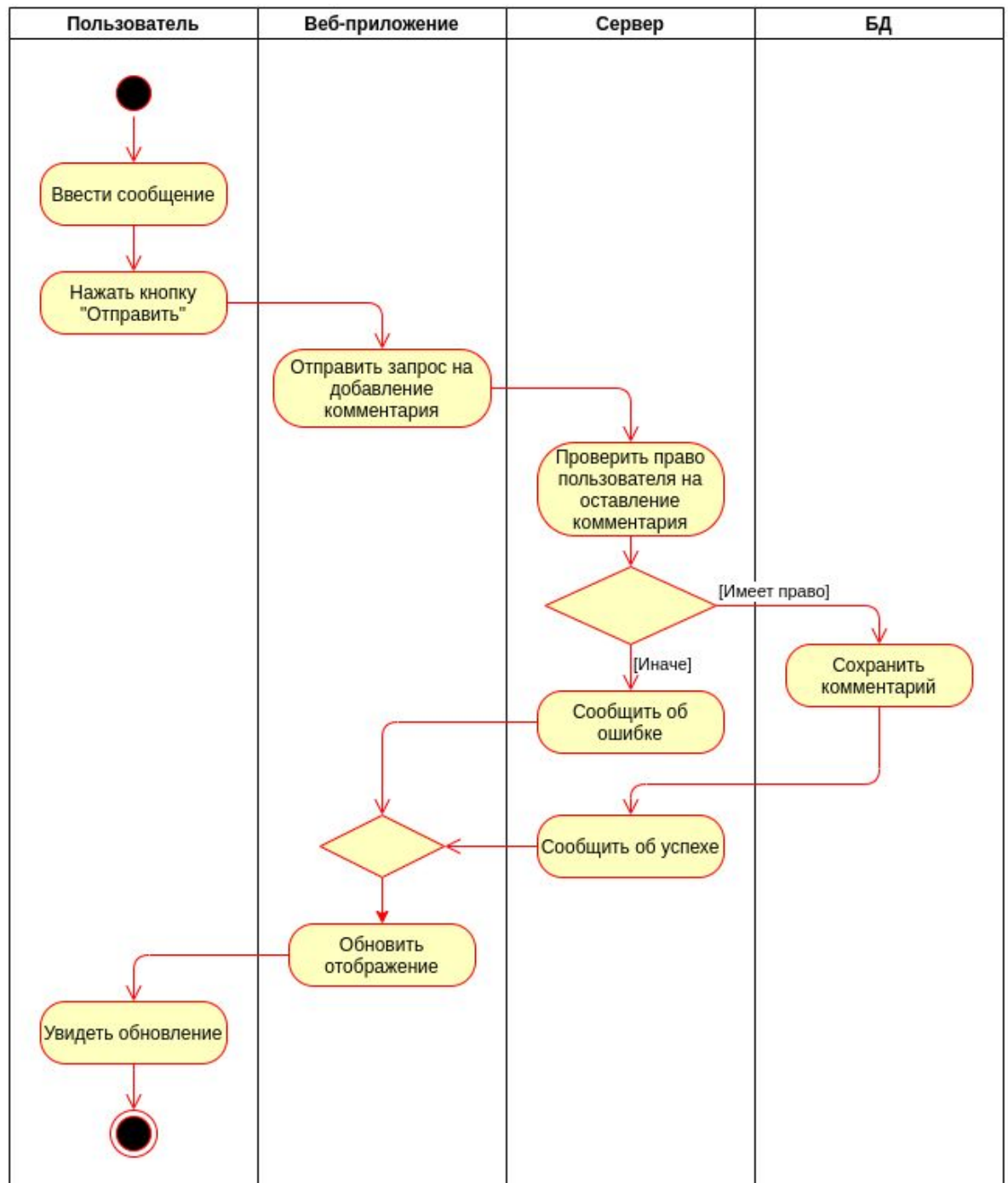


Рисунок 20. Диаграмма активностей добавления комментария



### 2.6.8. Диаграмма развертывания

На рисунке представлена диаграмма развертывания. Клиент - браузер любого устройства. Из него могут посылаться запросы на сервер. На сервере приложения за обработку запросов отвечает сервлет контейнер tomcat, на котором расположено приложение. Приложение может обращаться к СУБД (MySQL server).

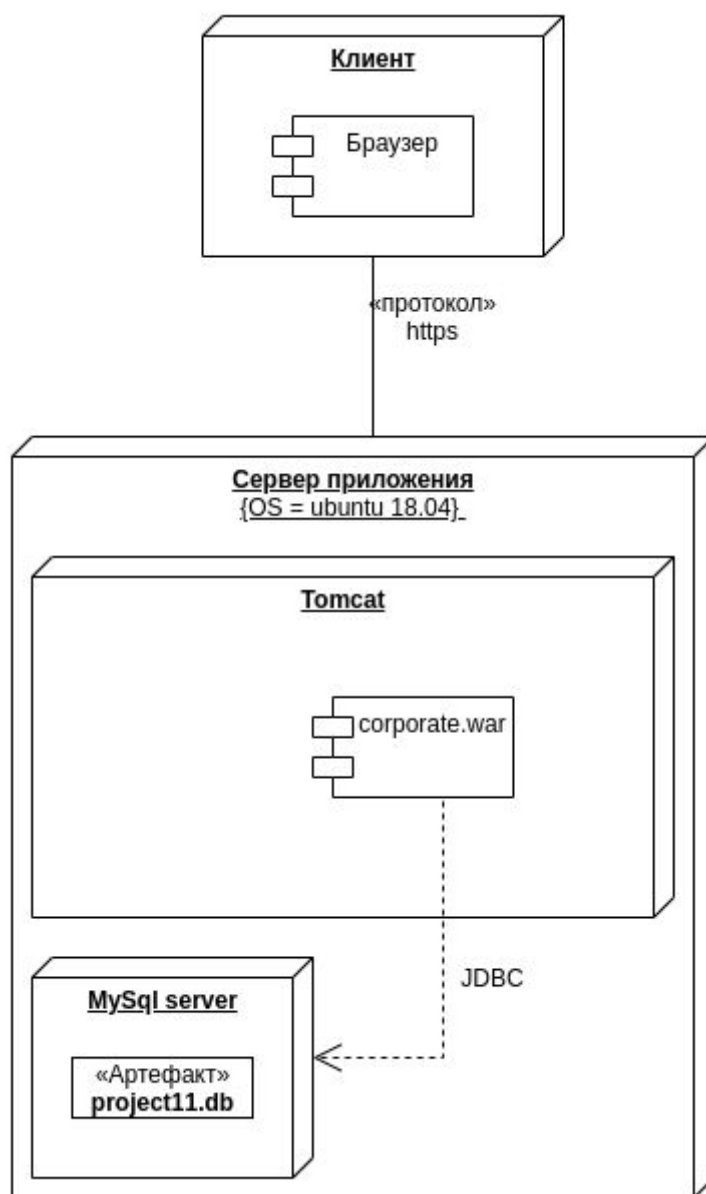


Рисунок 21. Диаграмма развертывания

## Глава 3.Реализация

### 3.1. Скриншоты реализации приложения

- Страница входа на сайт

The screenshot shows the login page of a website. At the top, there is a dark header bar with the text "Сайт корпоративного обучения" on the left and "Зарегистрироваться Войти" on the right. Below the header, the main content area is white. It features a login form with the following elements: a "Войти" button at the top right, an "Email:" label above a text input field, a "Ваш email" placeholder text inside the input field, a "Пароль:" label above another text input field, a "Пароль" placeholder text inside the input field, and a blue "Login" button at the bottom center.

Рисунок 22. Страница входа на сайт

На этой странице пользователю предлагается заполнить поля логина и пароля для входа в систему или же перейти на страницу регистрации.

- Страница регистрации

The screenshot shows the registration page of a website. At the top, there is a dark header bar with the text "Сайт корпоративного обучения" on the left and "Зарегистрироваться Войти" on the right. Below the header, the main content area is white. It features a registration form with the following elements: a "Регистрация" button at the top right, an "Инвайт-код:" label above a text input field, an "Email:" label above another text input field, an "Email" placeholder text inside the input field, an "Имя:" label above a text input field, a "Ваше имя" placeholder text inside the input field, a "Пароль:" label above a text input field, a "Пароль" placeholder text inside the input field, a "Подтверждение пароля:" label above another text input field, a "Подтверждение пароля" placeholder text inside the input field, and a blue "Зарегистрироваться" button at the bottom center.

Рисунок 23. Страница регистрации

На этой странице пользователю предлагается заполнить поля логина, электронной почты, пароля, подтверждения пароля и токена доступа, выданного администратором для регистрации в системе.

- Страница личного кабинета пользователя

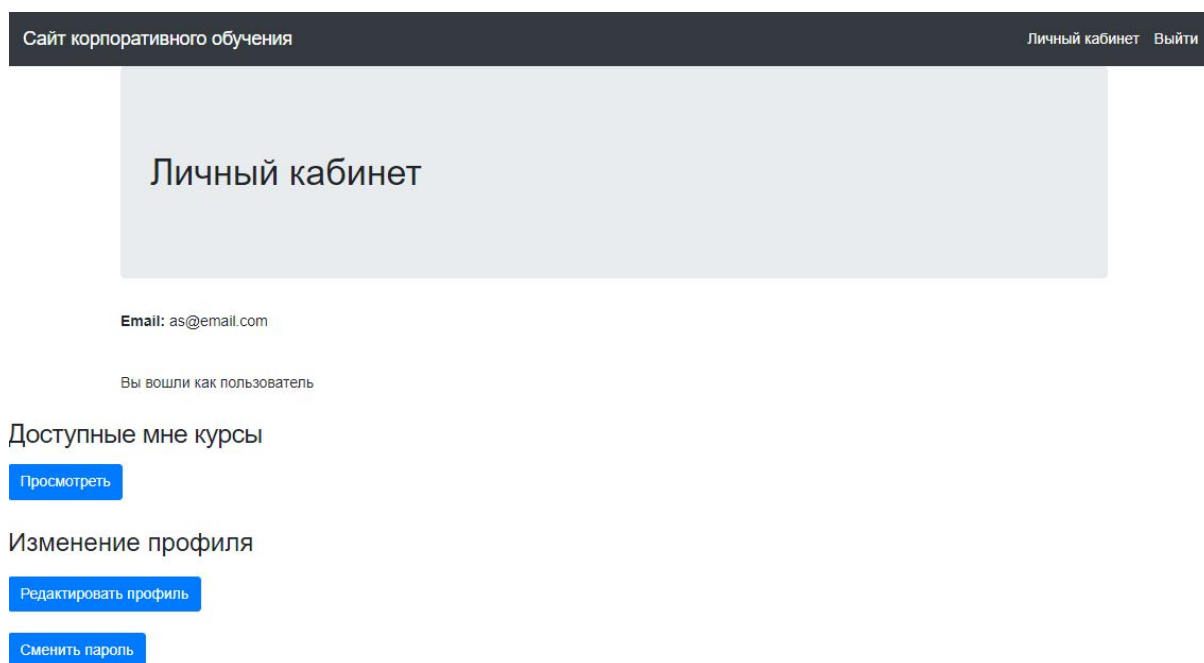


Рисунок 24. Страница личного кабинета пользователя

На данной странице слушатель может перейти на список курсов, на которых числится слушателем и также перейти на страницу редактирования профиля.

- Страница личного кабинета инструктора

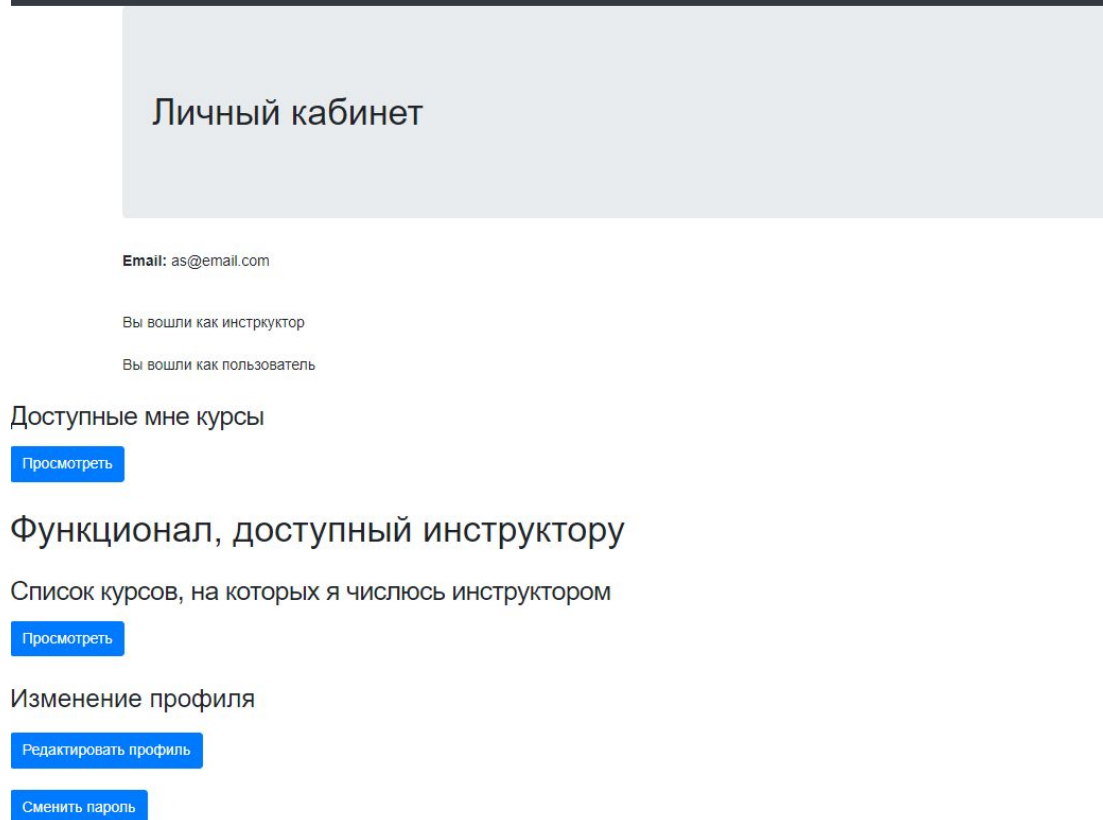


Рисунок 25. Страница личного кабинета инструктора

На данной странице пользователь может перейти на список курсов, на которых числится инструктором и также перейти на страницу редактирования профиля.

- Страница личного кабинета инструктора

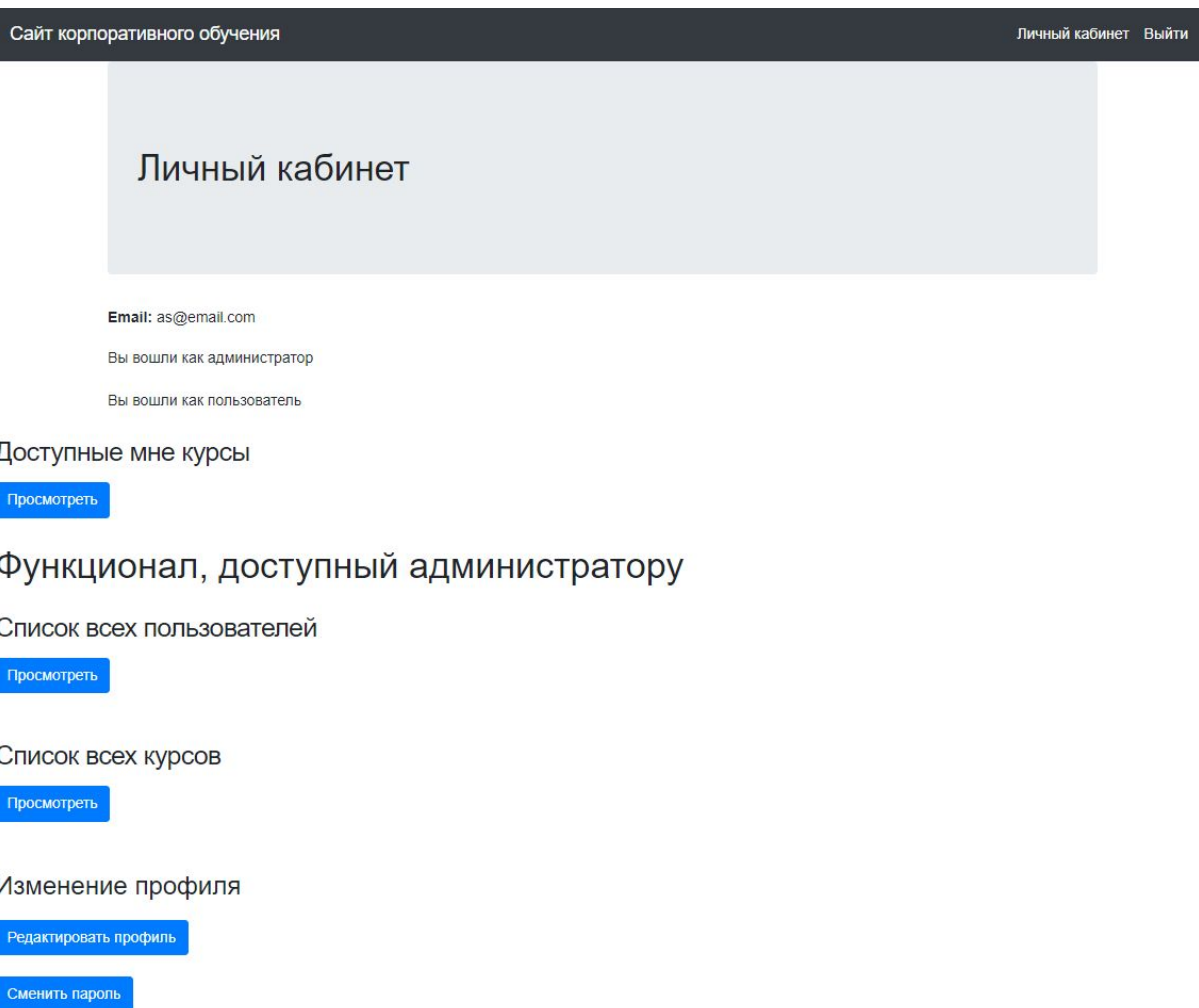


Рисунок 26. Страница личного кабинета администратора

На данной странице администратор может перейти на список курсов на сайте или же на список всех пользователей сайта и также перейти на страницу редактирования профиля.

- Страница добавления курса

## Добавление курса

Название курса

Краткое описание курса

Paragraph	▼	<b>B</b>	<i>I</i>	<u>U</u>	⋮
<input type="text"/>					

Добавить

Назад

Рисунок 27. Страница добавления курса

На данной странице инструктор может создать новый курс, введя его название и описание.

- Страница добавления урока

## Добавление урока

Название урока

Краткое описание урока

Добавить

Назад

Рисунок 28. Страница добавления урока

На данной странице инструктор может добавлять урок к конкретному курсу.

### Динамические страницы

- Страница редактирования данных профиля

Сайт корпоративного обучения

Личный кабинет Выйти

Редактировать данные профиля

Email:

sample@email.com

Имя:

Павел

Сохранить

Назад

Рисунок 29. Страница редактирования данных профиля

На этой странице пользователь может изменить свою почту или имя пользователя.

- Страница смены пароля

Сайт корпоративного обучения

Личный кабинет Выйти

Редактировать пароль

Пароль:

Новый пароль

Подтверждение пароля:

Подтверждение пароля

Сохранить

Назад

Рисунок 30. Страница смены пароля

На этой странице пользователь может поменять свой пароль.

- Список курсов слушателя

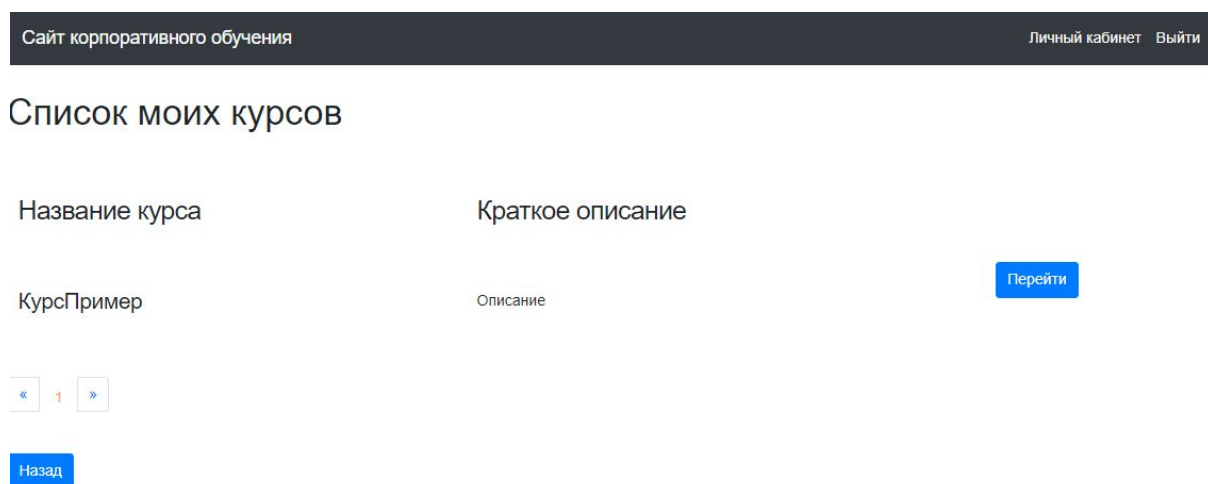


Рисунок 31. Страница списка курсов слушателя

На этой странице слушатель получает список курсов, на которых числится, и может переходить к конкретному курсу. Также он может вернуться в личный кабинет по кнопке “Назад”.

- Главная страница курса для слушателя

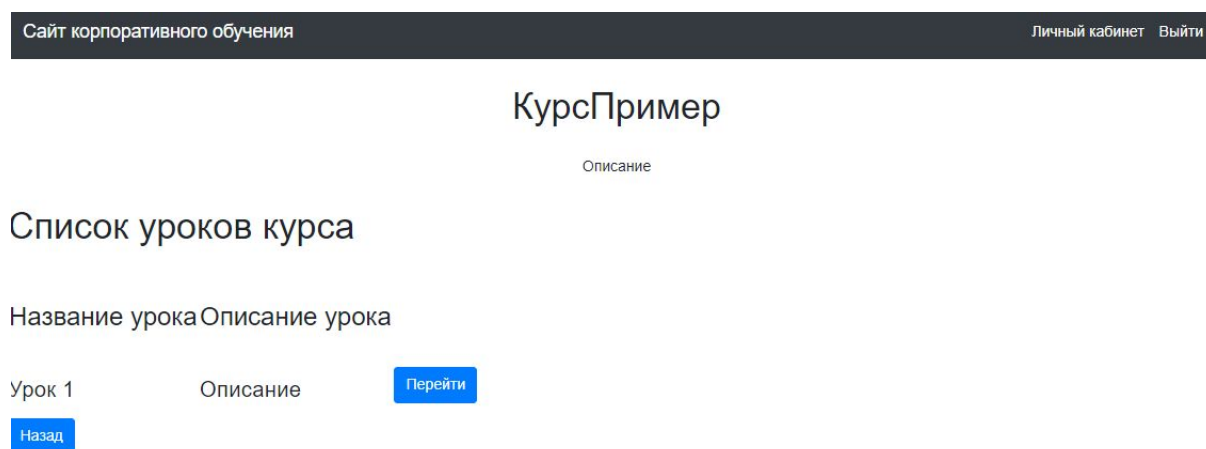


Рисунок 32. Страница курса для слушателя



На данной странице представлены название курса, его описание и список уроков, на каждый из которых можно перейти. Также есть кнопка “Назад”, которая возвращает пользователя на список курсов.

- Страница урока для слушателя

Сайт корпоративного обучения

Личный кабинет Выйти

## Урок 1

Описание

### Материалы к уроку

Урок\_6\_Урок\_4\_Урок\_4\_Урок\_2\_ClassDiagram (1).png

### Домашняя работа

Файл не прикреплен  
Домашняя работа не зачтена

### Комментарии

К домашней работе пока не оставлено ни одного комментария

### Прикрепить файл

Выберите файл Выберите файл Файл не выбран

Добавить

Ваш комментарий:

Отправить

Назад

Рисунок 33. Страница урока для слушателя

На данной странице слушатель видит название урока, его тему + вопрос к домашнему заданию. Ниже находятся материалы для ознакомления, список комментариев и поле для отправки домашнего задания (файл) и поле для отправки комментария.

- Страница с подопечными курсами инструктора

# Список подопечных курсов

Название курса	Краткое описание	
Новый курс	Описание .....	<a href="#">Перейти</a>
Курс 1	sample	<a href="#">Перейти</a>
Курс 1	описание курса	<a href="#">Перейти</a>
Пример 1	Описание примера 1	<a href="#">Перейти</a>
КурсПример	Описание	<a href="#">Перейти</a>

[«](#) [1](#) [»](#)

[Назад](#) [Добавить курс](#)

Рисунок 34. Страница списка курсов инструктора

На данной странице инструктор может просмотреть список курсов, на которых преподает и перейти к конкретному курсу. Также он перейти на страницу добавления курса, нажав на “Добавить курс”.

- Страница управления курсом

Название курса

КурсПример

Краткое описание курса

Описание

Список уроков

Название урока	Краткое описание	
Урок 1	Описание	<div>Редактировать</div> <div>Просмотреть домашние работы</div> <div>Удалить урок</div>

Добавить урок

Добавить слушателя на курс по email

Email

Искать

Найти слушателя курса по email

Email

Искать

Назад

Рисунок 35. Страница управления курсом

На данной странице инструктор может увидеть список уроков на данном курсе, перейти к их изменению или удалению какого-либо урока. На этой странице он может добавить слушателя на курс по email или же получить слушателя по email. С этой страницы инструктор может перейти к странице создания урока.

- Страница управления уроком

## Урок 1

Описание

### Материалы

Урок\_6\_фото.jpg

Удалить

Выберите файл

Выберите файл

Файл не выбран

Добавить

[Назад](#)

Рисунок 36. Страница управления уроком

На данной странице инструктор может дозаполнить урок материалами. Или же он может вернуться назад к курсу.

- Список домашних работ к уроку

## Список домашних работ к уроку

Домашняя работа пользователя Павел (sample@email.com)

[Перейти](#)[Назад](#)

Рисунок 37. Страница списка домашних работ к уроку

На данной странице инструктор видит список домашних работ к уроку от слушателей. Он может перейти к любой работе или вернуться назад.

- Конкретная домашняя работа

Сайт корпоративного обучения

Личный кабинет Выйти

# Домашняя работа

Прикрепленный файл

Пользователь не прикрепил файл

Ваш комментарий:

Ваш комментарий

Оставить комментарий

Комментарии к работе

## Отметка о зачете

☐ Зачтено

☐ Не зачтено

Сохранить

Рисунок 38. Страница проверки домашней работы

На данной странице инструктор может просмотреть материалы по домашней работе, присланной пользователем, если они есть, увидеть список комментариев, если они есть, оставить свой комментарий по этой работе и зачесть или не зачесть работу. Может сохранить результат или вернуться назад.

- Страница со списком курсов для администратора

## Список всех учебных курсов на сайте

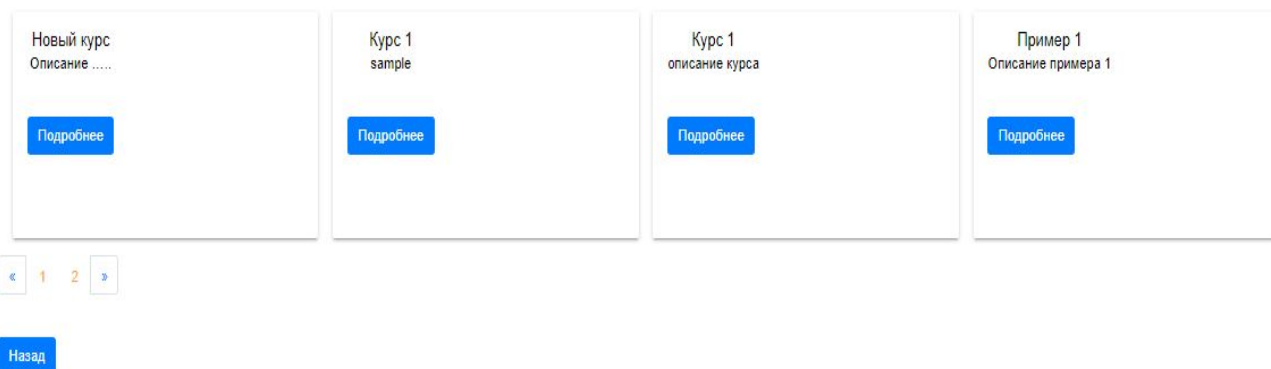


Рисунок 39. Страница списка всех учебных курсов

На данной странице администратор видит список всех существующих курсов, на каждый из которых он может перейти, либо вернуться в личный кабинет.

- Страница управления курсом для администратора

# Курс 1

[описание курса](#)

## Список слушателей курса

Email	Имя
-------	-----

## Список инструкторов курса

Email	Имя инструктора	
sample@email.com	Павел	<a href="#">Удалить</a>

Добавить инструктора на курс по email

[Искать](#)

Email Имя

[Назад](#)

Рисунок 40. Страница управления курсом для администратора

На данной странице администратор может просмотреть список слушателей и инструкторов на конкретном курсе, а также удалять инструкторов с курса или же, наоборот, добавить инструктора по email.

- Страница управления пользователями

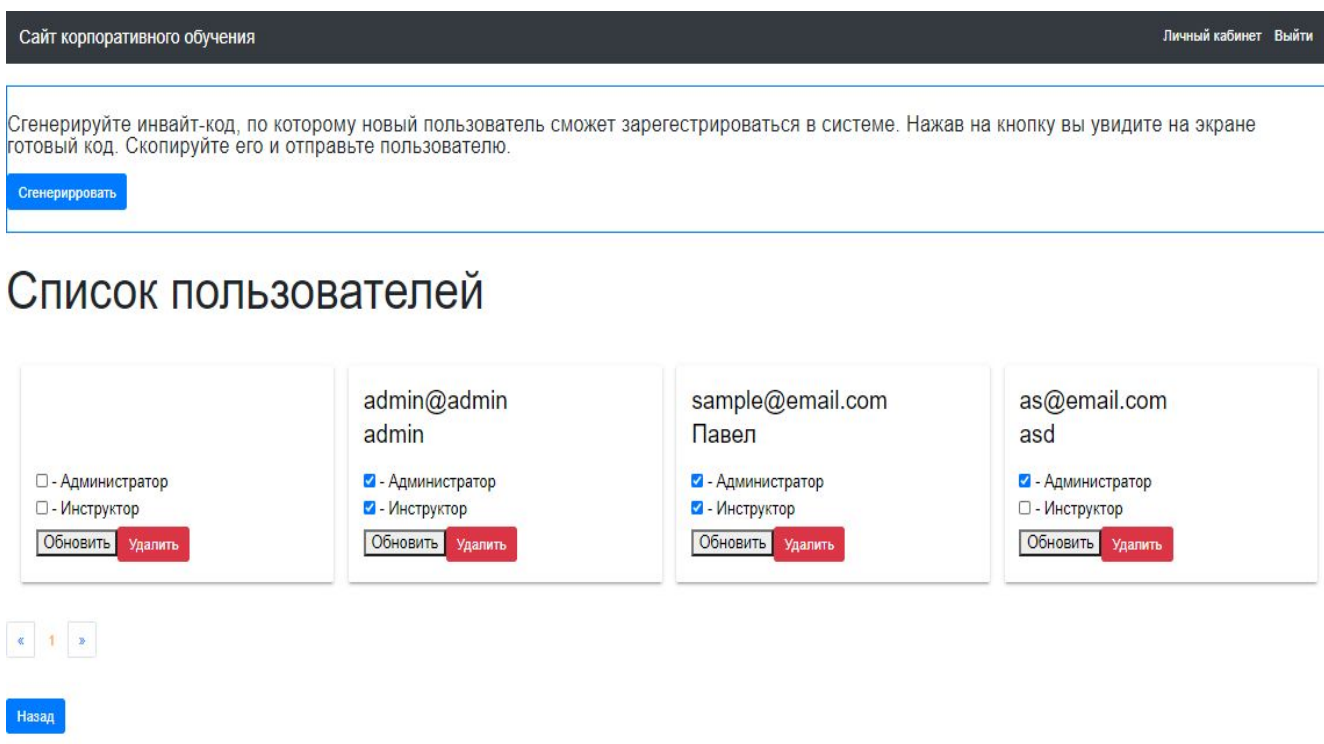


Рисунок 31. Страница управления пользователями

На этой странице администратор может видеть список всех пользователей на сайте, удалять их и выставлять их права. Также на этой странице можно сгенерировать инвайт-токен для регистрации нового пользователя.

### 3.2. Реализация серверной части приложения

Для реализации конечных точек, которые соответствуют относительно простым `crud`-командам, был использован Spring Data REST, который предоставляет REST API для выбранных моделей домена. Более того это позволяет добавлять нестандартные конечные точки, которые, например, возвращают более сложные выборки.

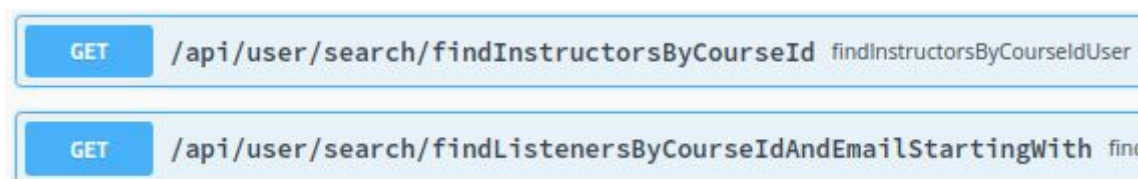


Рисунок 41. Пример конечных точек предоставленных Spring Data REST

Так точка доступа `findInstructorsByCourseId` возвращает коллекцию, содержащую инструкторов, закрепленных на указанном курсе, а



findListenersByCourseIdAndEmailStartedWith - слушателей курса, соответствующих критериям. Для реализации подобных точек достаточно добавить метод в соответствующий интерфейс.

Листинг 1. Пример методов, опираясь на которые, создаются конечные

ТОЧКИ

```
@RestResource()
@Query("select u from User u inner join u.courses c where c.id=:courseId")
List<User> findInstructorsByCourseId(@Param("courseId") Long courseId);

@RestResource
@Query("select u from User u inner join u.onCourses onC inner join onC.course c where c.id=:courseId and u.email like concat(:email, '%')")
List<User> findListenersByCourseIdAndEmailStartingWith(@Param("courseId") Long courseId, @Param("email") String email);
```

Для действий требующих какой-то более сложной аутентификации или действий (например, создание курса инструктором требует автоматического добавления инструктора на курс) были реализованы в Rest контроллерах.

<b>auth-controller</b> Auth Controller	>
<b>comment-controller</b> Comment Controller	>
<b>course-controller</b> Course Controller	>
<b>homework-controller</b> Homework Controller	>
<b>lesson-controller</b> Lesson Controller	>
<b>material-controller</b> Material Controller	>
<b>profile-controller</b> Profile Controller	>
<b>user-controller</b> User Controller	>

Рисунок 42. REST контроллеры

## Глава 4. Тестирование

В рамках разработки приложения были проведены smoke, sanity, usability, negative, unit, integration тестирования, покрывающие основные варианты использования приложения и часть дополнительной функциональности, в соответствии с поставленными целями тестирования, которые были описаны в плане тестирования:

1. Smoke testing – поверхностное тестирование системы на предмет работоспособности. В рамках этого тестирования были проверены следующие возможности:

- открытие сайта,
- проверка соединения с БД и API,
- проверка возможности просмотра личного кабинета,
- просмотр списка подопечных курсов для инструктора,
- просмотр подопечного курса для инструктора,
- поиск слушателя по email,
- переход на страницу добавления урока на курс,
- переход на форму редактирование урока,
- просмотр списка домашних работ,
- переход к конкретной домашней работе,
- просмотр всех курсов на сайте,
- просмотр конкретного курса,
- поиск инструктора по email,
- просмотр всех пользователей сайта,
- просмотр доступных курсов,
- просмотр уроков слушателем.

2. Sanity testing – тестирование работы конкретных функций системы подтверждающих работу системы согласно требованиям в Техническом Задании. Были протестированы функциональные возможности системы:

- проверка возможности редактирования данных профиля,
- проверка возможности изменения пароля,
- добавление слушателя на курс,
- добавление урока на курс,
- удаление урока с курса,
- загрузка материала к уроку,
- удаление материала из урока,
- добавление комментария к домашней работе,
- загрузить прикрепленный к домашней работе файл,
- поставить отметку о зачете к домашней работе,
- добавление инструктора на курс,
- удаление инструктора с курса,
- обновление прав пользователя,
- удаление пользователя из системы,
- сгенерировать инвайт-токен,
- отправка домашнего задания,
- оставление комментария,
- скачивание материала к уроку,
- просмотр уроков слушателем.

3. Negative testing – тестирование поведения системы, при попытке ввода некорректных данных. Было изучено поведение системы при выполнении следующих функций:

- поиск слушателя по несуществующему email,

- авторизация с некорректными данными.
4. Usability testing – тестирование того, насколько легко конечный пользователь сможет освоить систему. Было изучено поведение системы при выполнении следующих функций:
    - добавление курса с пустым полем названия,
    - добавление урока с пустым полем имени,
    - добавление урока с пустым полем описания.
  5. Unit testing – тестирование, позволяющее проверить на корректность отдельные модули исходного кода программы, наборы из одного или более программных модулей вместе с соответствующими управляющими данными, процедурами использования и обработки.
  6. Integration testing – тестирование, при котором отдельные программные модули объединяются и тестируются в группе.

В рамках проведения составленных тестовых кейсов для системы, можно утверждать, что поведение системы является корректным для всех описанных тестовых сценариев smoke, sanity, usability, negative, unit, integration тестирований.

## Заключение

Подводя итог проделанной работе, можно сказать, что разработан продукт, готовый к эксплуатации и облегчающий работу пользователя. Теперь любая компания может без труда организовать процесс обучения своих сотрудников, которым будут предоставлены все функциональные возможности программы, а именно: просмотр всех необходимых курсов, скачивание необходимых материалов. Не вызывает никаких проблем пользование приложением, весь интерфейс достаточно прост и интуитивно понятен.

Проблема, заключающаяся в существовании многих нюансов связанных с бумажной работой в составлении накладных и ведении всех списков учета, была решена путем оптимизации и автоматизации данной деятельности.

Цель работы, поставленная в самом начале, достигнута путем выполнения ряда задач:

1. Изучены приложения-аналоги.
2. Сформулированы основные методы, доступные для администраторов и инструкторов системы.
3. Разработан программный продукт для автоматизации процесса обучения.

Таким образом, была реализована программа, решающая реальные задачи, стоящие перед корпоративным обучением. Тем самым подтверждается практическая значимость работы.

В ходе курсового проекта в команде было реализовано легковесное клиент-серверное приложение с простым и понятным функционалом для организации удаленного процесса обучения.

В ходе работы были выполнены следующие задачи:

- разработана клиентская часть приложения;
- разработана серверная часть приложения;

- осуществлено хранение данных в базе данных, расположенной на сервере;
- осуществлено взаимодействие клиентского приложения и сервера с помощью RESTfull API.