

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

<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	7
Организация программы	11
Обоснование архитектуры системы	11
Описание диаграмм	12

## 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

можно эффективно обучать 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).

<b>Шаг 1</b>	<input style="width: 95%;" type="text" value="Личный кабинет"/>	✕ ✕	
Условие	<input style="width: 95%;" type="text" value="url: содержит"/> <input style="width: 95%;" type="text" value="/user/me"/>	✕	+
<b>Шаг 2</b>	<input style="width: 95%;" type="text" value="Курсы"/>	✕ ✕	
Условие	<input style="width: 95%;" type="text" value="url: содержит"/> <input style="width: 95%;" type="text" value="/user/me/course"/>	✕	+
<b>Шаг 3</b>	<input style="width: 95%;" type="text" value="Курс"/>	✕ ✕	
Условие	<input style="width: 95%;" type="text" value="url: регулярн..."/> <input style="width: 95%;" type="text" value=".*user/me/course/ld+"/>	✕	+
<b>Шаг 4</b>	<input style="width: 95%;" type="text" value="Урок"/>	✕ ✕	
Условие	<input style="width: 95%;" type="text" value="url: регулярн..."/> <input style="width: 95%;" type="text" value=".*user/me/course/ld+/lessons/ld+"/>	✕	+
<b>Шаг 5</b>	<input style="width: 95%;" type="text" value="Домашняя работа отправлена"/>	✕ ✕	
Условие	<input style="width: 95%;" type="text" value="событие: и..."/> <input style="width: 95%;" type="text" value="SEND_COMMENT"/>	✕ ✕	
	Код цели для сайта <div style="display: flex; align-items: center;"> <span style="background-color: #f0f0f0; padding: 2px 5px;">ym(61893537,'reachGoal','SEND_COMMENT')</span> <span style="margin-left: 10px; border: 1px solid #ccc; padding: 2px 5px;">Скопировать код</span> </div>		
или	<input style="width: 95%;" type="text" value="событие: и..."/> <input style="width: 95%;" type="text" value="DOWNLOAD_MATERIAL"/>	✕ ✕	
	Код цели для сайта <div style="display: flex; align-items: center;"> <span style="background-color: #f0f0f0; padding: 2px 5px;">ym(61893537,'reachGoal','DOWNLOAD_MATERIAL')</span> <span style="margin-left: 10px; border: 1px solid #ccc; padding: 2px 5px;">Скопировать код</span> </div>		
			+

Рисунок 1. Настройки воронки отслеживающей заинтересованность слушателей курсов



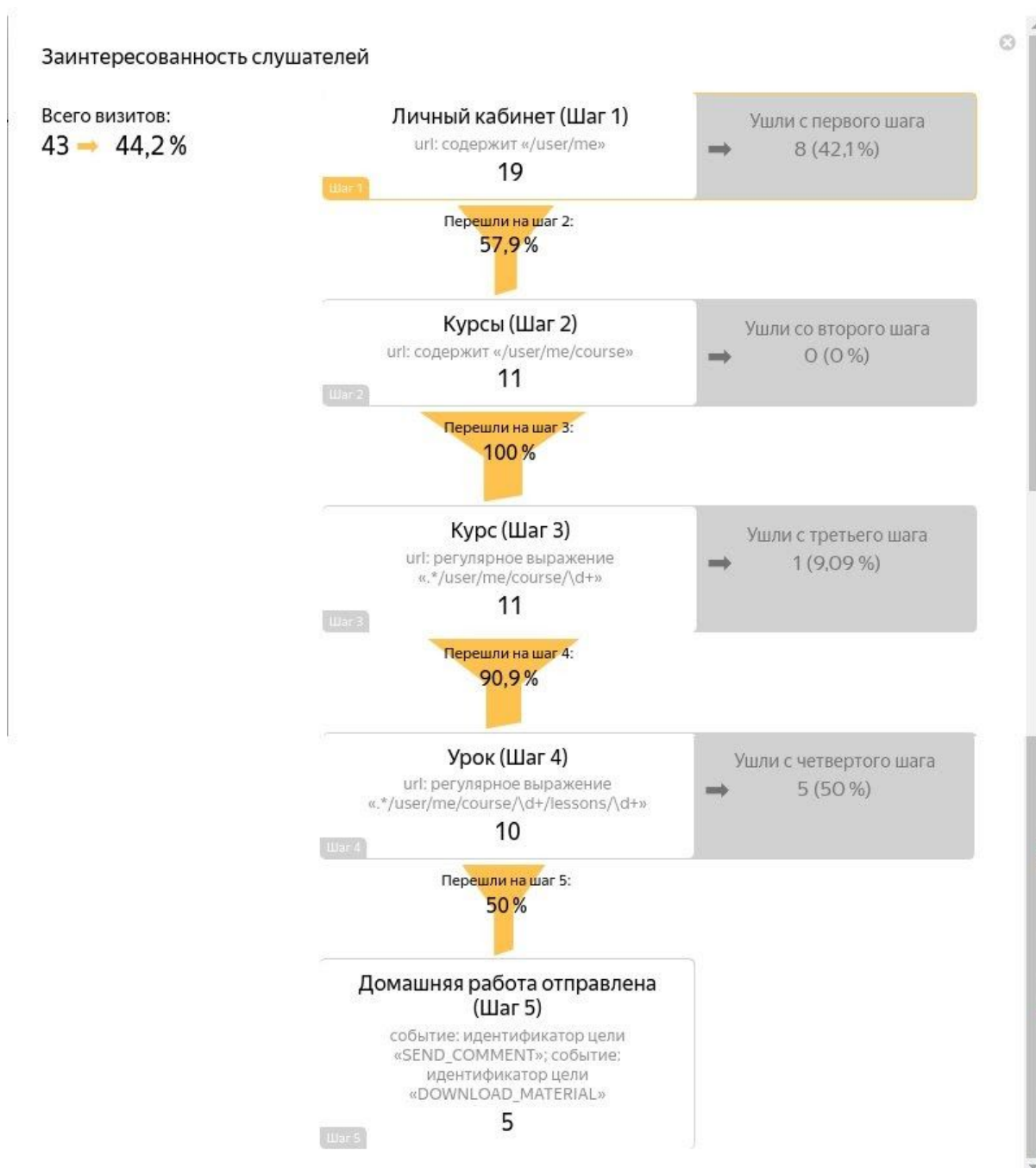


Рисунок 2. Воронка отслеживающая заинтересованность слушателей курсов

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).

**Шаг 1**

Курсы

Условие

url: регулярн...

.\* /courses

**Шаг 2**

Курс

Условие

url: регулярн...

.\* /course /.\*

**Шаг 3**

Домашние работы к уроку

Условие

url: регулярн...

.\* /lesson /homeworks ?lessonId =\d +

**Шаг 4**

Конкретная домашняя работа

Условие

url: регулярн...

/lesson /homework ?lessonId =\d +&userId =.\*

**Шаг 5**

Оценка др

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

Рисунок 3. Настройка воронки отслеживающей заинтересованность инструкторов курсов

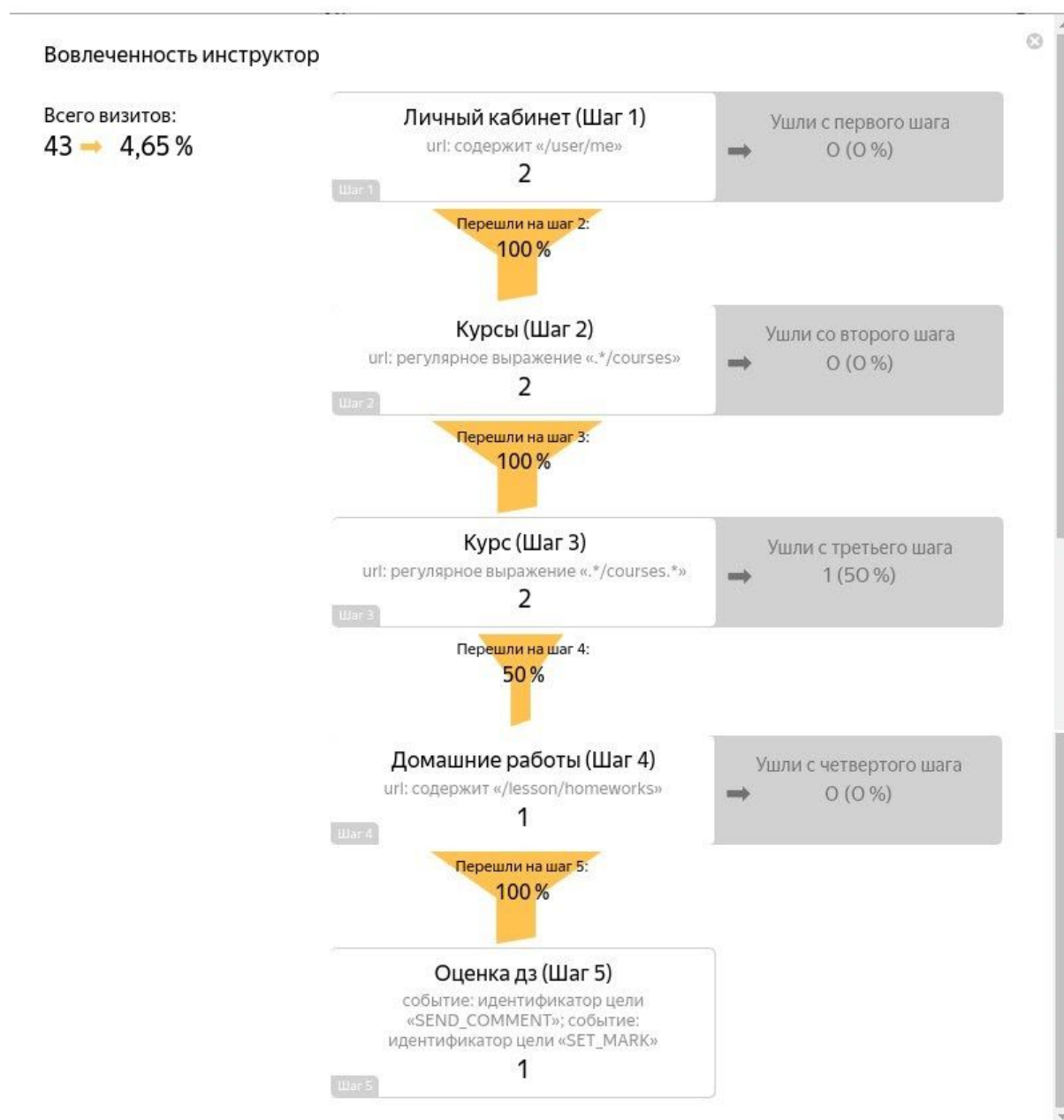


Рисунок 4. Воронка отслеживающая заинтересованность инструкторов курсов

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=\d+

+

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

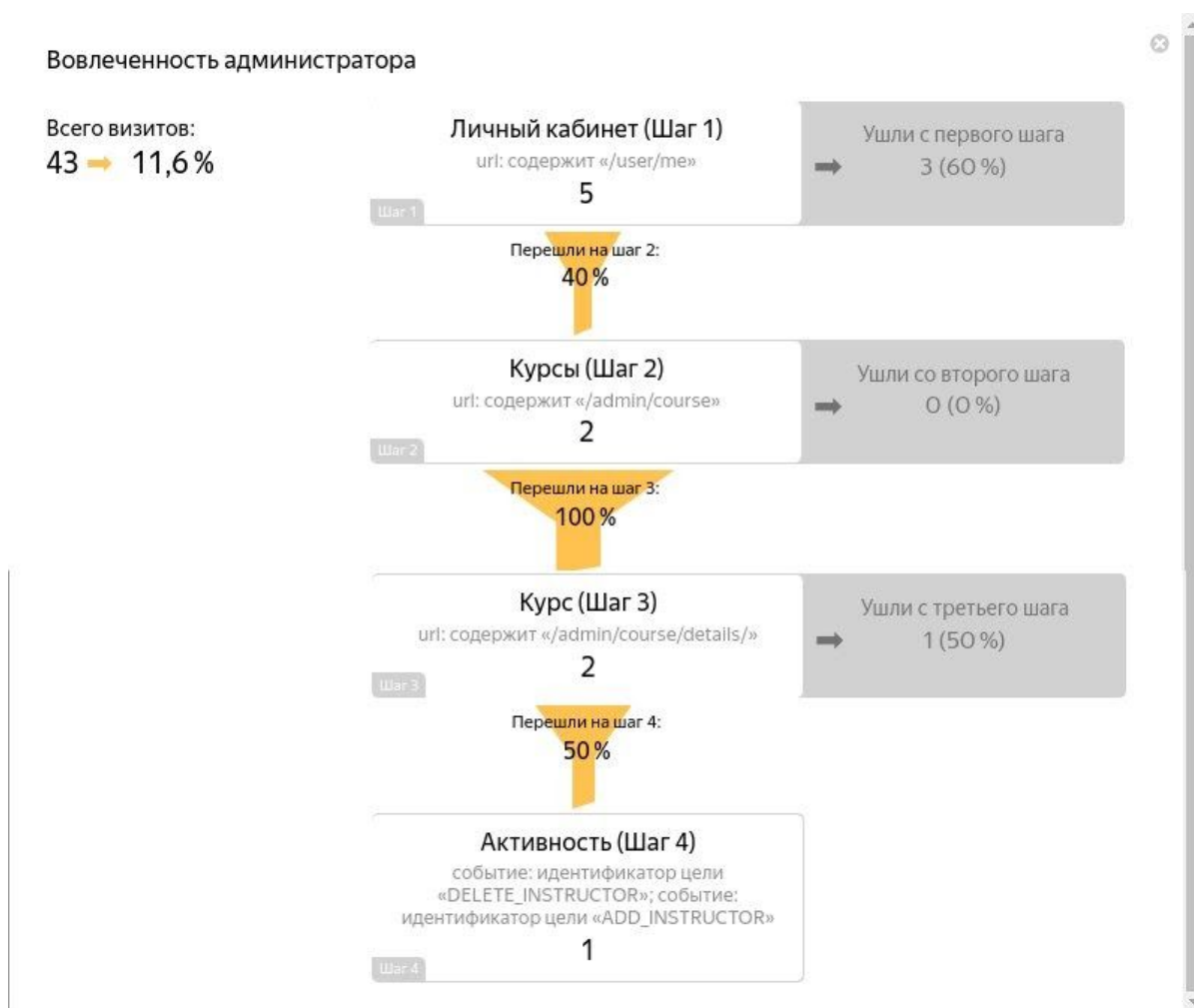


Рисунок 6. Воронка отслеживающая действия администраторов

## 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**

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)

В качестве средства разработки серверной части приложения был выбран Spring Framework, представляет собой IoC контейнер, что позволяет писать гораздо меньше кода серверной части приложения. Нам достаточно лишь написать компонент и внедрить его с помощью аннотации. Spring позволяет легко управлять приложением, поскольку все его конфигурации можно вынести в отдельный файл.

Также Spring JPA позволяет легко работать с базой данных, требуя написания небольшое количество дополнительного кода.

## **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

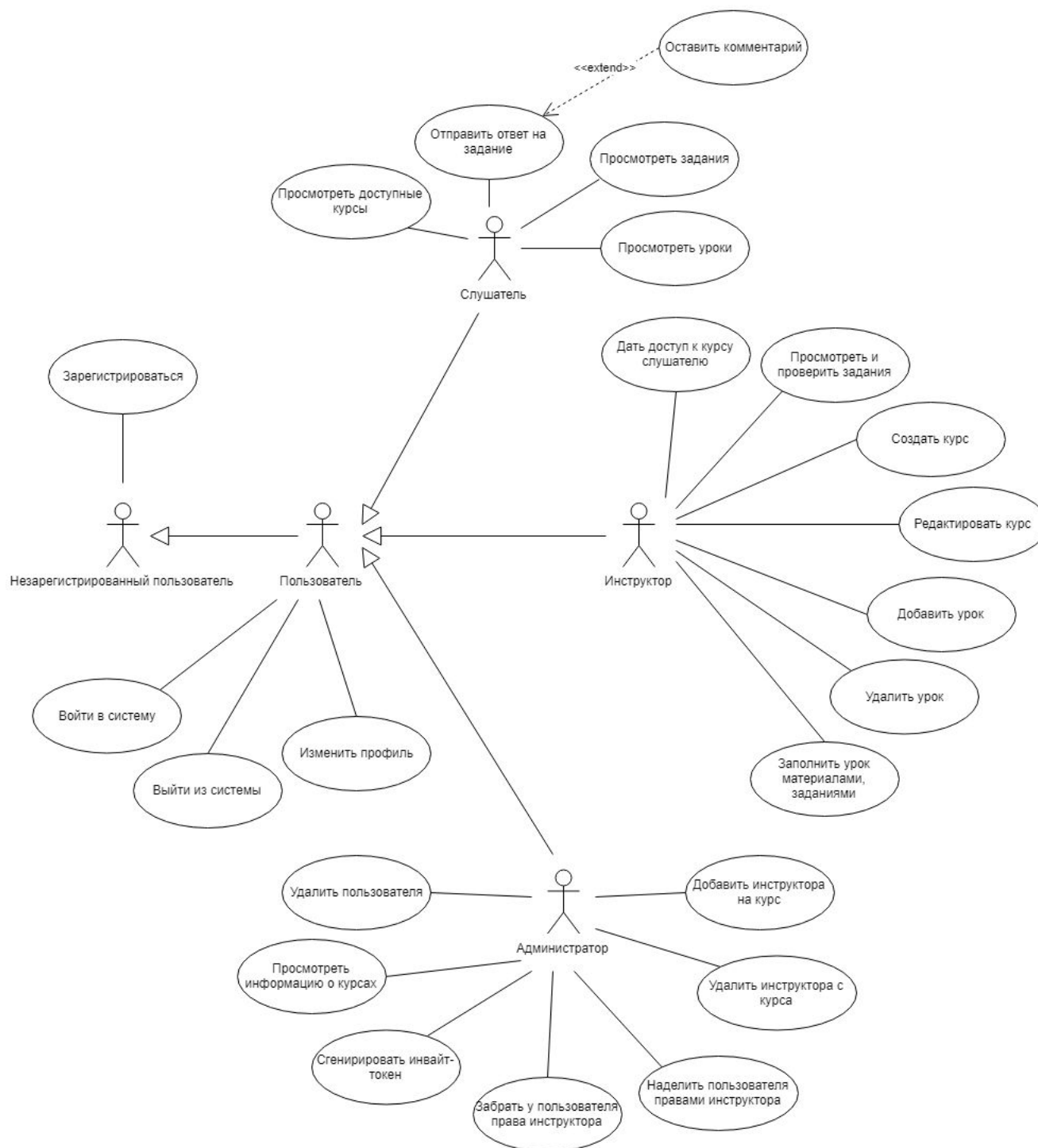


Рисунок 7. Диаграмма прецедентов

### 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, используемые 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.

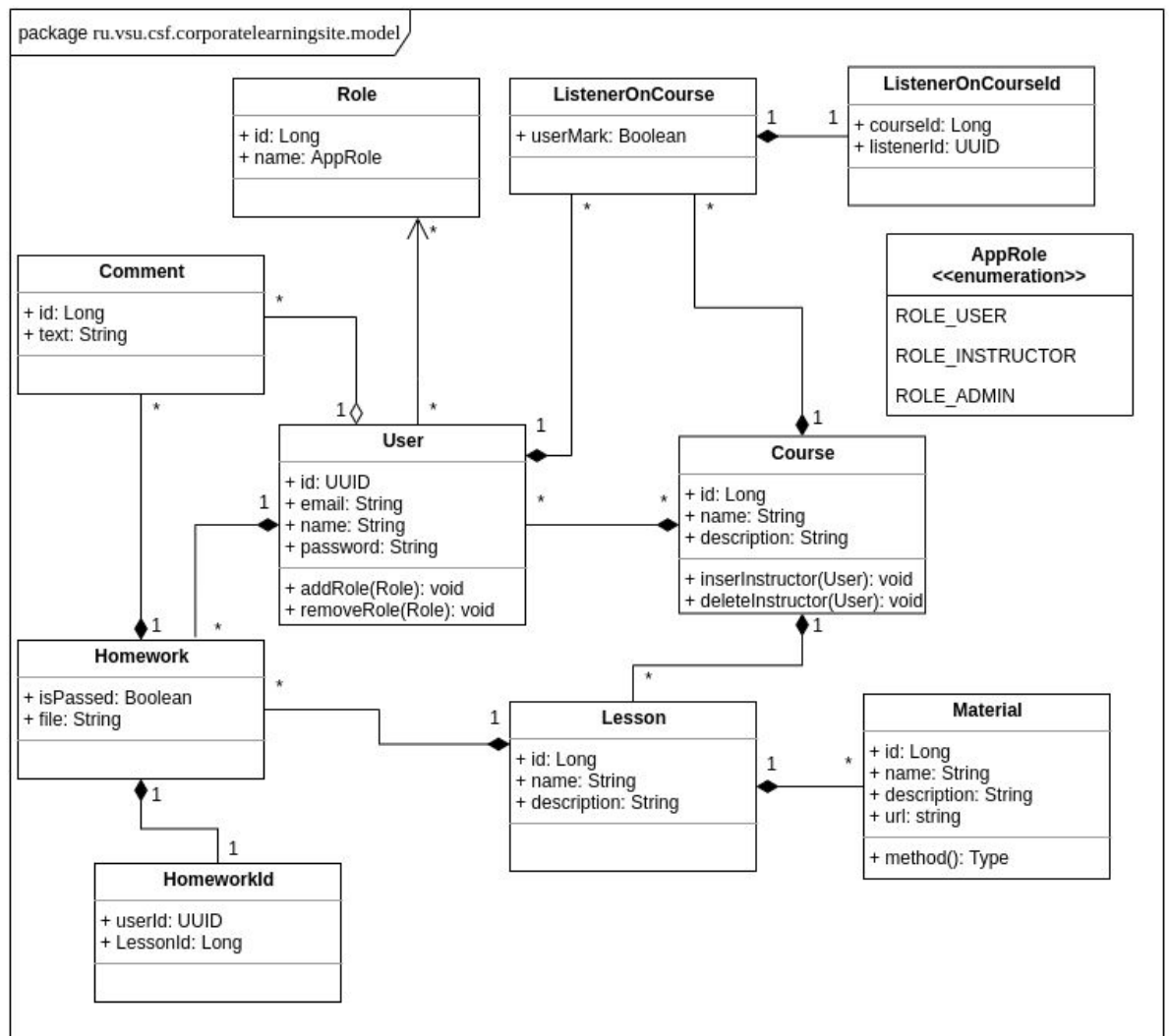


Рисунок 8. Диаграмма классов пакета model

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.

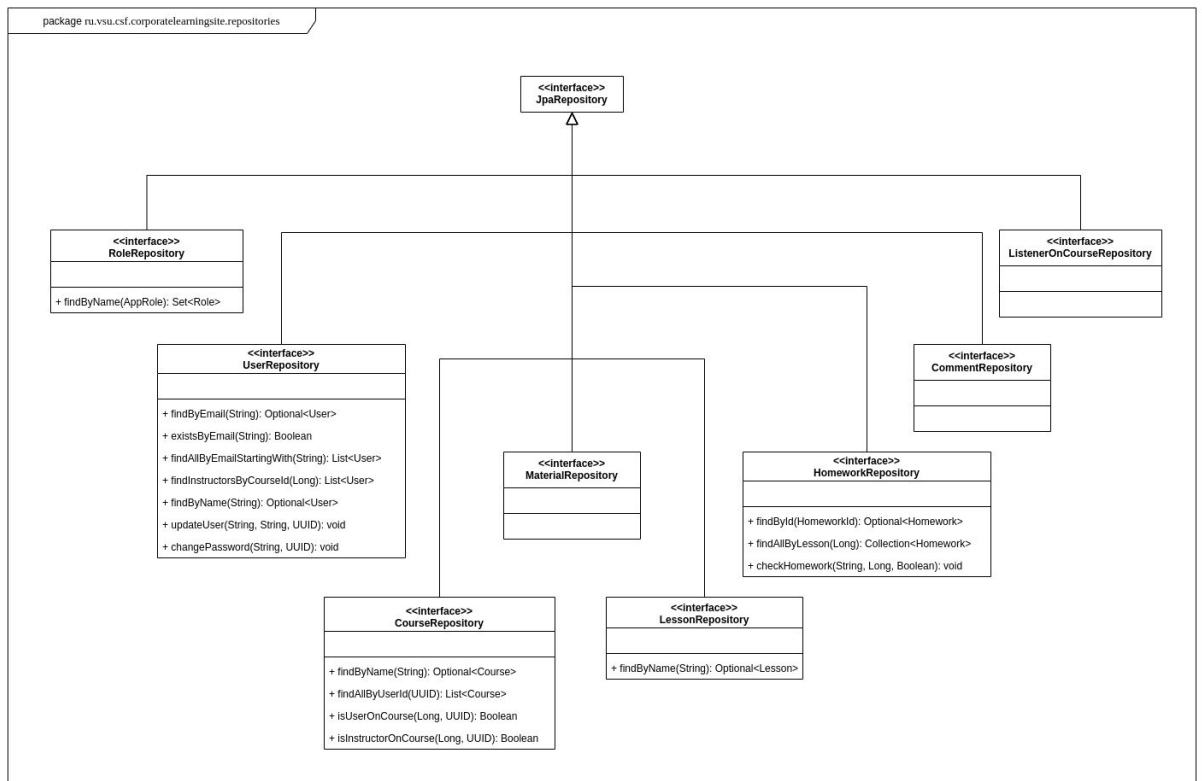


Рисунок 9. Диаграмма классов пакета repositories

### 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.

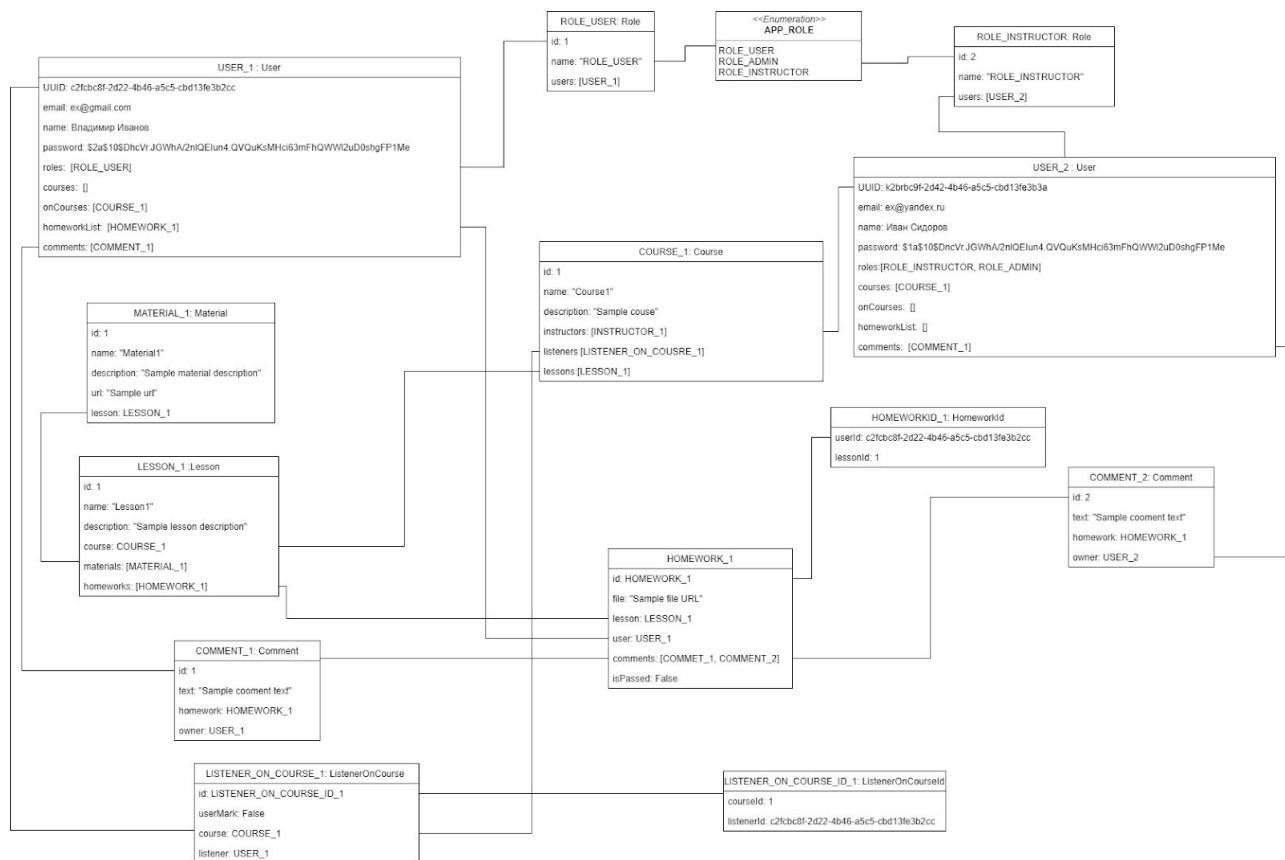


Рисунок 10. Диаграмма объектов

## 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.

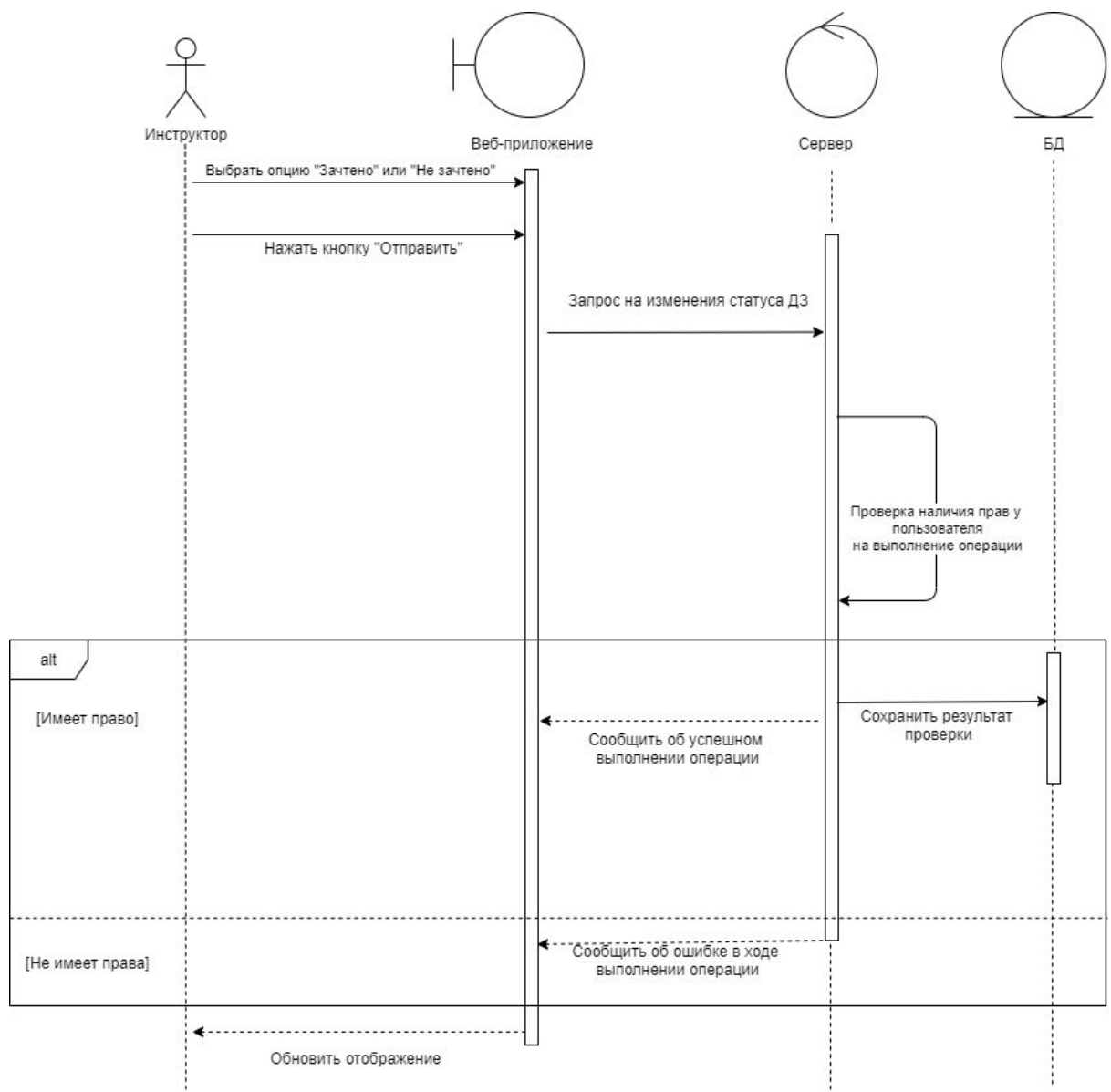


Рисунок 11. Диаграмма последовательностей проверки домашнего задания

#### 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.

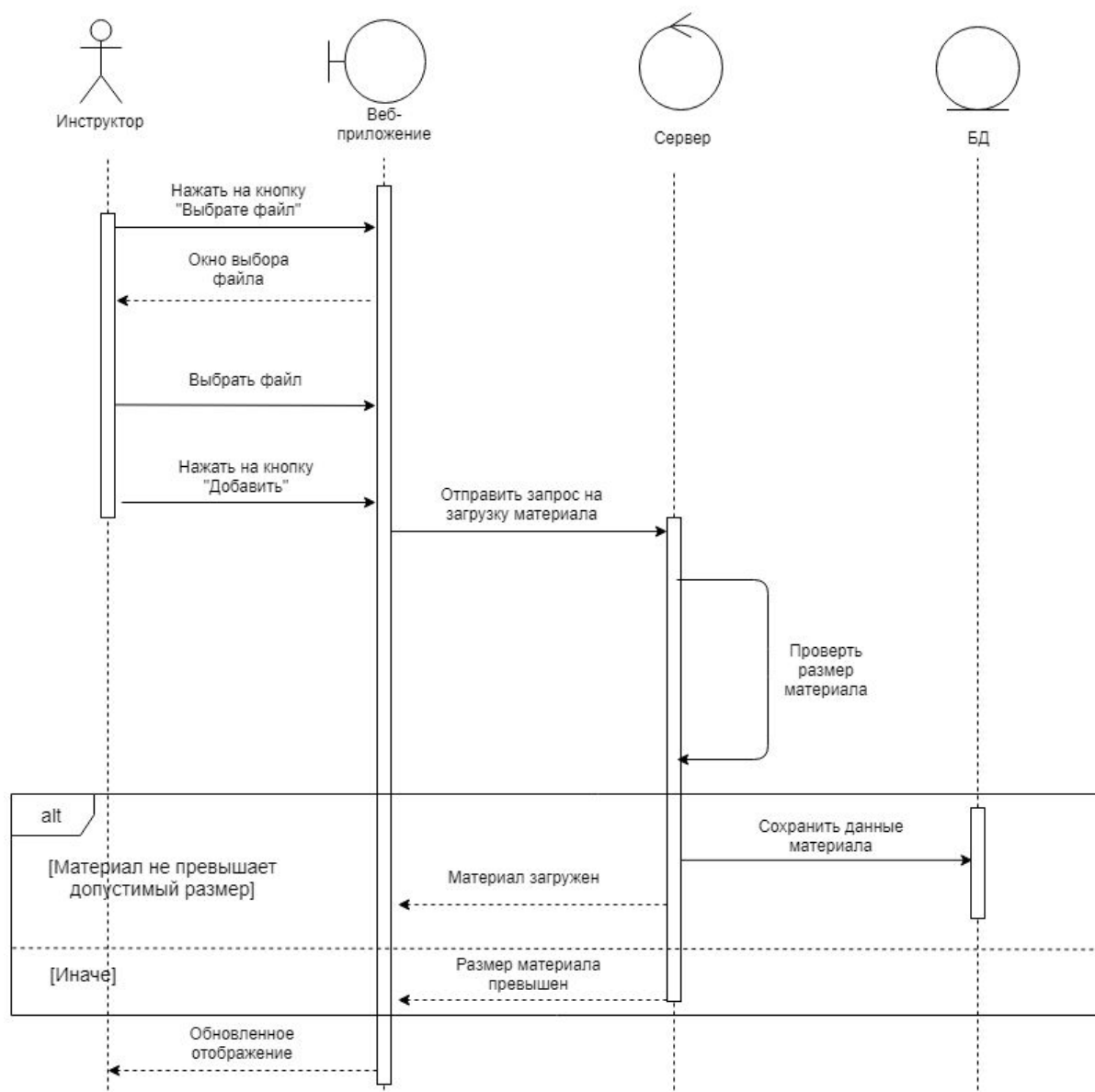


Рисунок 12. Диаграмма последовательностей добавления материала инструктором

#### 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.

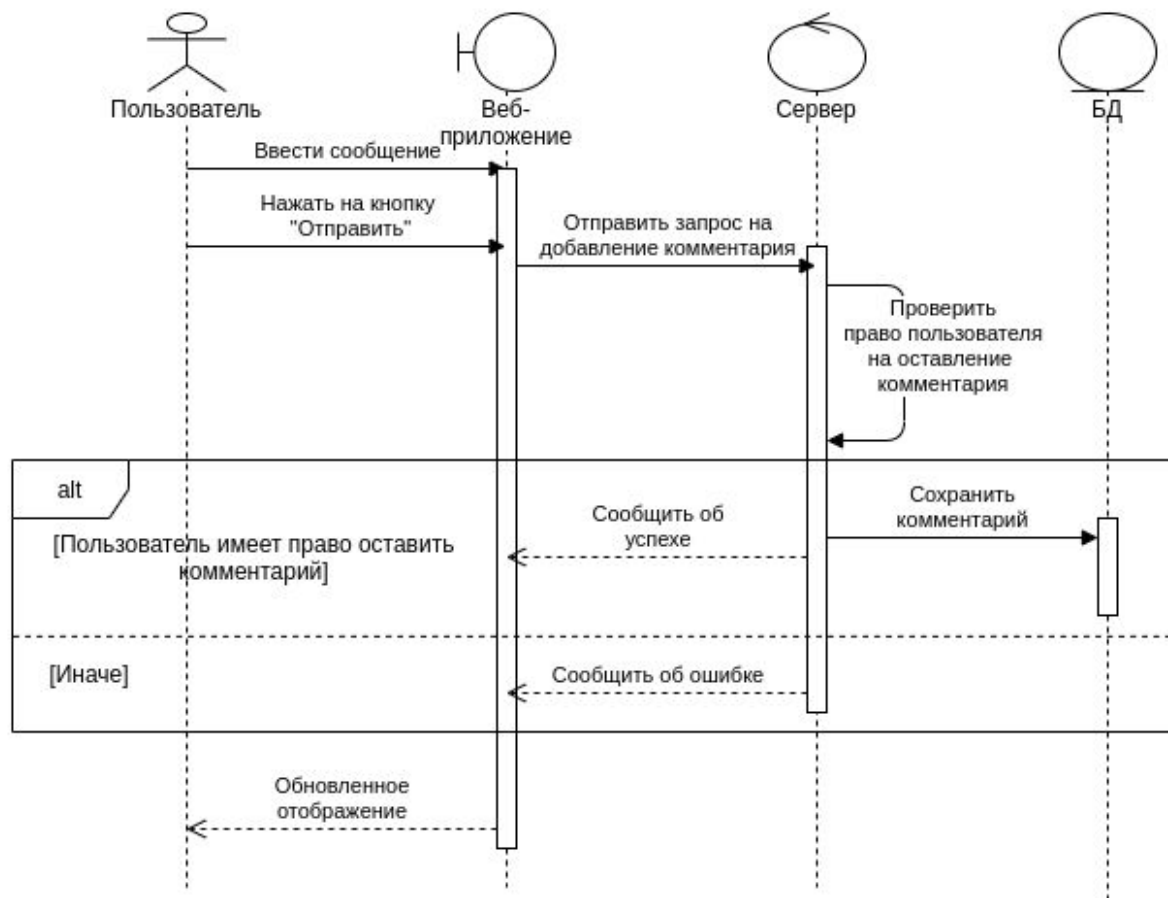


Рисунок 13. Диаграмма последовательностей добавления комментария

## 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.

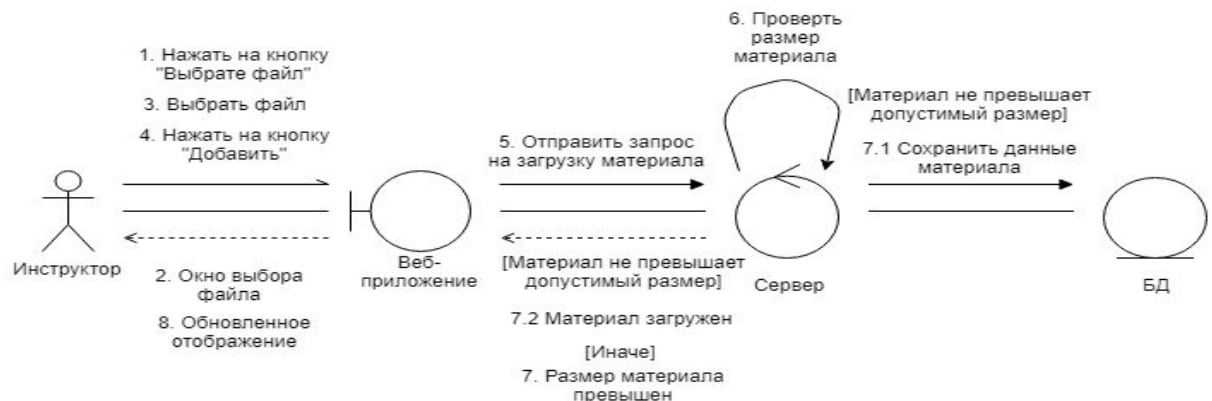


Рисунок 15. Диаграмма взаимодействия добавления материала инструктором

### 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.



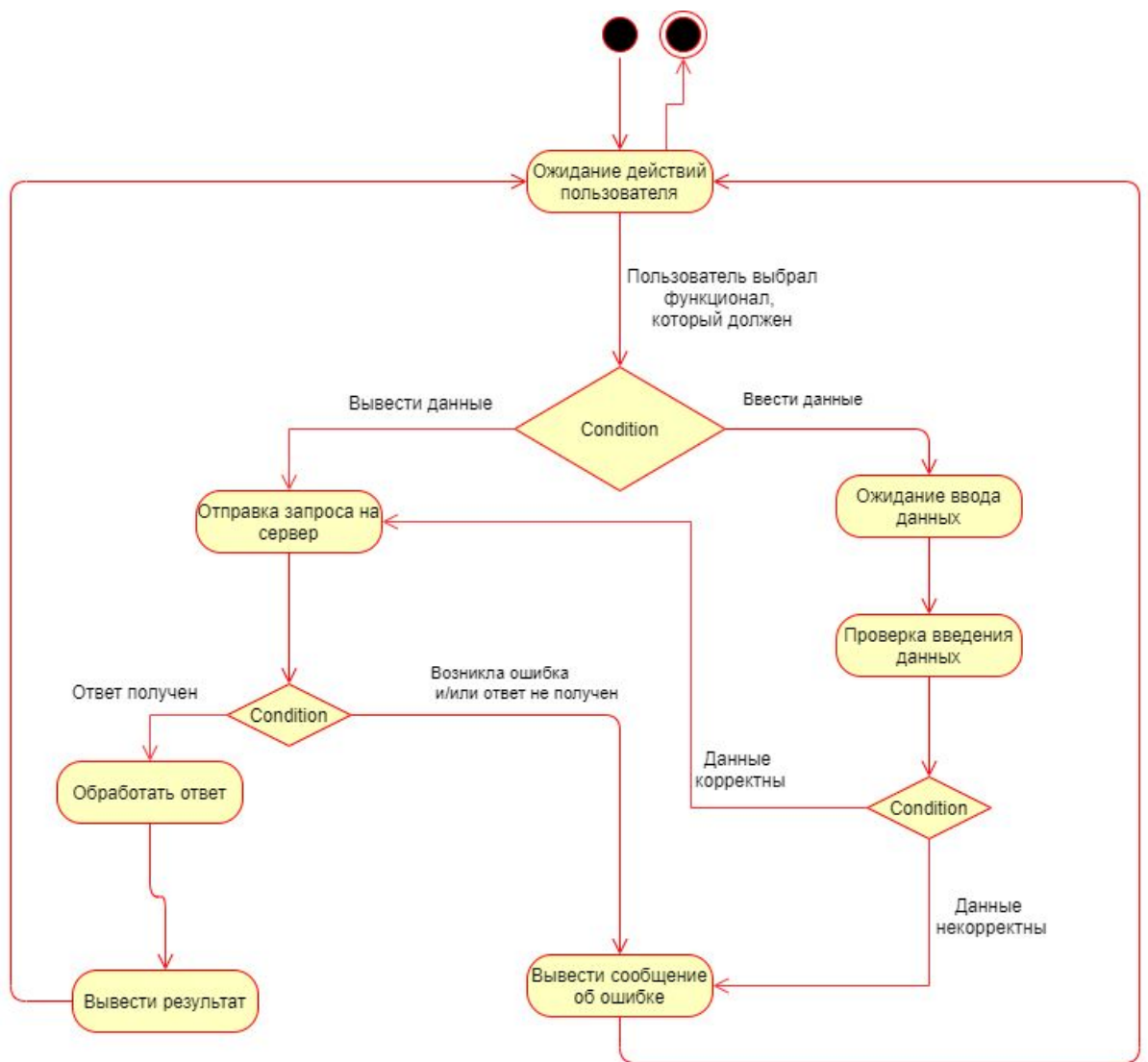


Рисунок 17. Диаграмма состояний

## 2.6.7.Диаграммы активностей

### 2.6.7.1. Проверка домашнего задания

На рисунке изображена диаграмма активности при проверке домашнего задания инструктором

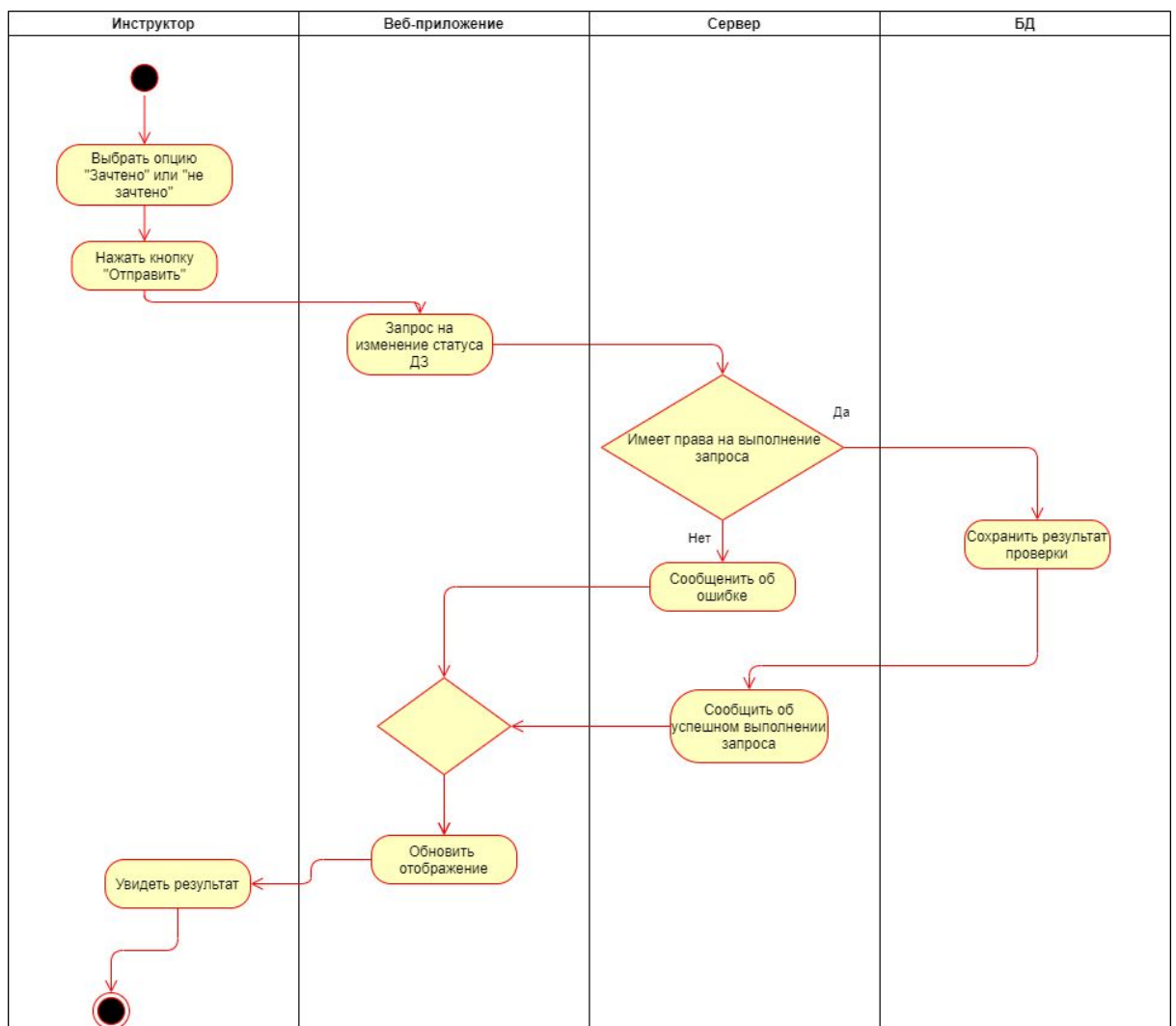


Рисунок 18. Диаграмма активностей проверки домашнего задания

#### 2.6.7.2. Добавление материала инструктором

На рисунке изображена диаграмма активности добавления материала инструктором.

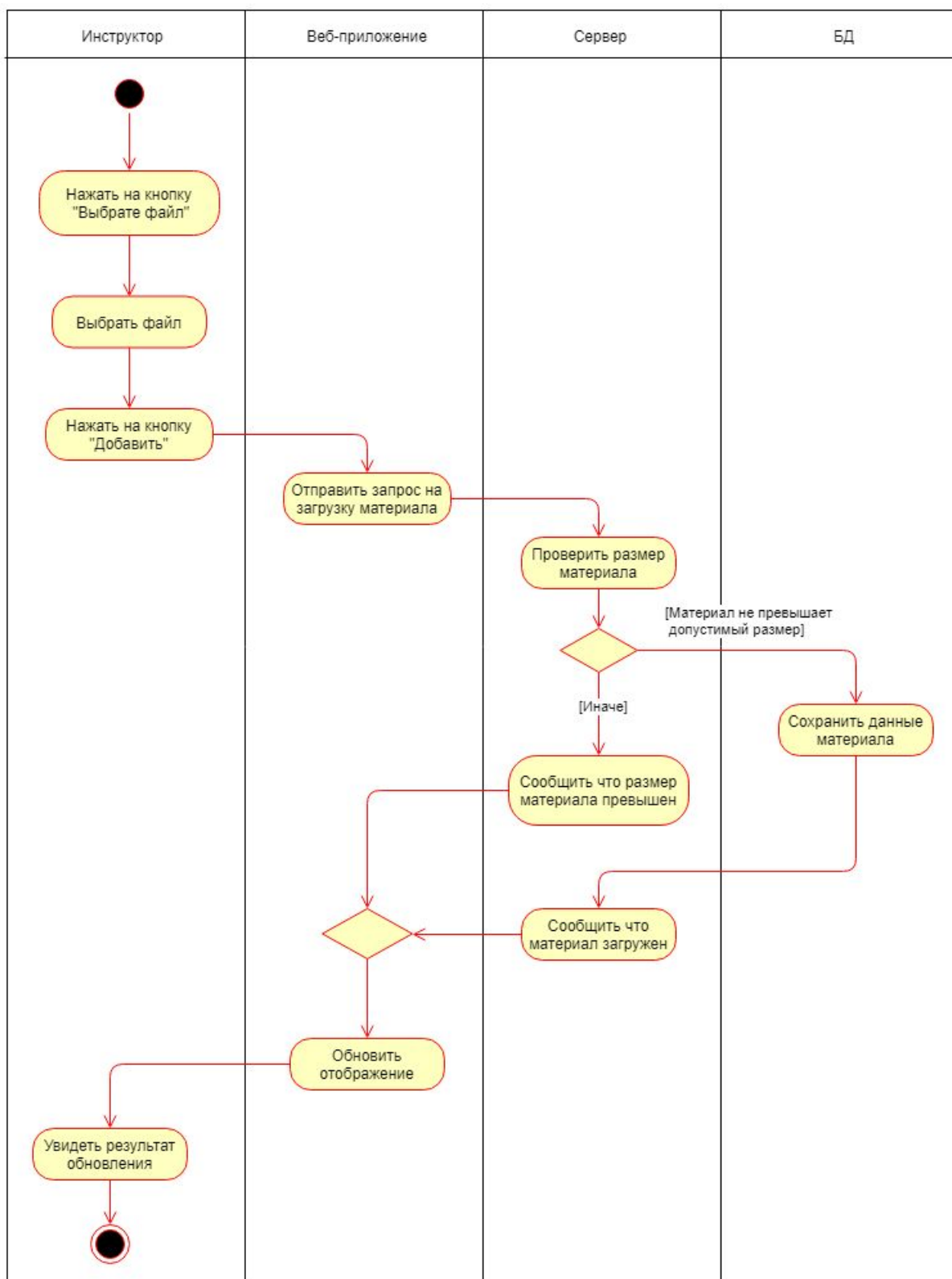


Рисунок 19. Диаграмма активностей добавления материала инструктором

### 2.6.7.3. Добавление комментария

На рисунке изображена диаграмма активности добавления комментария к домашней работе.

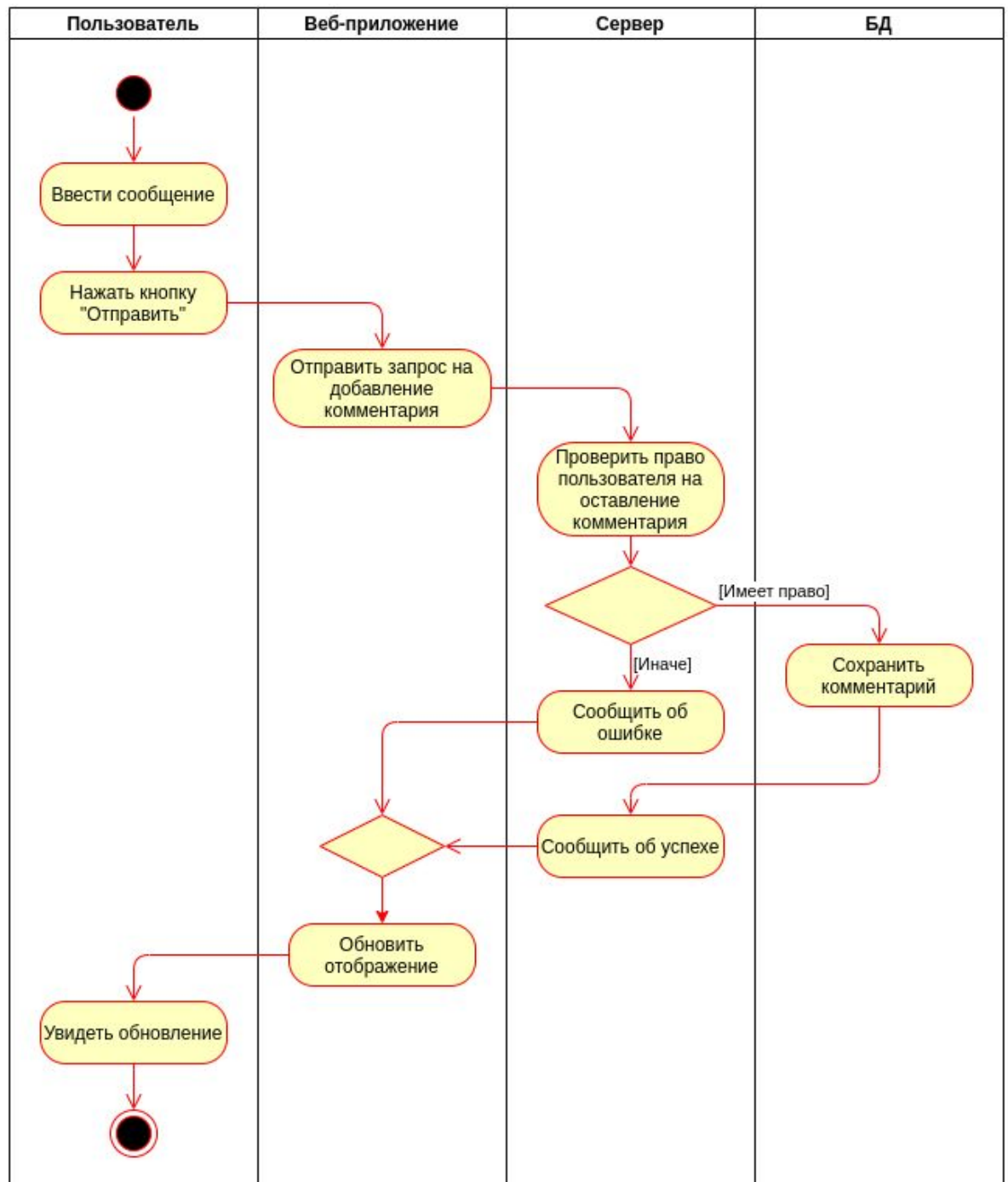


Рисунок 20. Диаграмма активностей добавления комментария

### 2.6.8.Диаграмма развертывания

На рисунке представлена диаграмма развертывания. Клиент - браузер любого устройства. Из него могут посылаются запросы на сервер. На сервере приложения за обработку запросов отвечает сервлет контейнер

tomcat, на котором расположено приложение. Приложение может обращаться к СУБД (MySQL server).

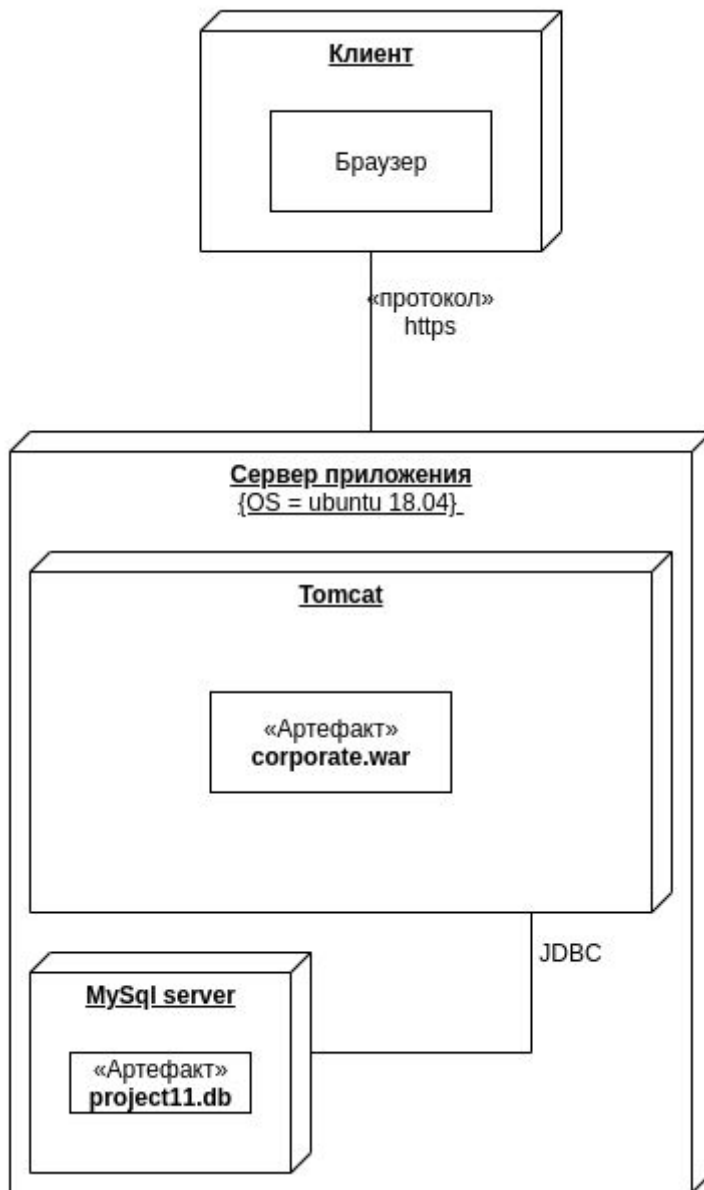


Рисунок 21. Диаграмма развертывания