МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

«МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

СВЯЗИ И ИНФОРМАТИКИ»

(МТУСИ)

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ**

На тему

**«Разработка платформы для онлайн-курсов программирования для начинающих»**

Выполнили студентки группы БАС2101 факультета КиИБ:

Беспалова А.В.

Блинкова А.А.

Ярцева С.С.

Проверил ассистент кафедры «Информационная Безопасность»:

Маторин Ф.А.

Москва 2023

**СОДЕРЖАНИЕ**

[ЗАДАНИЕ НА ПРОЕКТ 3](#_Toc152770196)

[ВВЕДЕНИЕ 4](#_Toc152770197)

[Анализ выбранной темы 4](#_Toc152770198)

[ОСНОВНАЯ ЧАСТЬ 8](#_Toc152770199)

[Проектирование серверной части 11](#_Toc152770200)

[Проектирование структуры базы данных 11](#_Toc152770201)

[Разработка 11](#_Toc152770202)

[Тестирование 11](#_Toc152770203)

[Описание клиентской части 11](#_Toc152770204)

[Демонстрация функционала 11](#_Toc152770205)

[ЗАКЛЮЧЕНИЕ 11](#_Toc152770206)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 11](#_Toc152770207)

[Приложения 11](#_Toc152770208)

# ЗАДАНИЕ НА ПРОЕКТ

Создать сервис, состоящий из серверной и клиентской части. Клиентское приложение может быть представлено в виде веб-сайта, мобильного или desktopприложения.  
Тема для сервиса выбирается самостоятельно, опираясь на требования к сервису, и должна быть уникальна.

Требования к серверной части:

1. Реализация серверной части должна быть создана на языке программирования Java с использованием фреймвока Spring Boot;
2. В сервисе должны быть предусмотрены возможности аутентификации и авторизации. Рекомендуется использовать инструменты библиотеки Spring Security, однако запрещено использовать авторизацию по умолчанию;
3. В сервисе должно присутствовать не меньше 3 ролей для пользователей. Доступные страницы и функционал должны быть разделены по ролям;
4. Важные события должны быть залогированы (например, вход пользователя в систему или важное изменение в сущностях);
5. Серверная часть обязательно должна иметь базу данных. К базе данных вводятся следующие требования:
   1. База данных должна иметь как минимум 3 таблицы, отображающие бизнес-сущности (таблицы используемые для механизмов аутентификации не учитываются);
   2. Между некоторыми таблицами должны существовать связи (один ко многим, много ко многим, один к одному);
   3. Для хранения сущностей должна использоваться реляционная база данных (MySQL, PostgreSQL).
6. При написании кода должны соблюдаться принципы S.O.L.I.D, и, по возможности, некоторые из шаблонов проектирования (минимум 2);
7. Для методов, реализующих логику, должны быть написаны модульные тесты, которые должны выполняться во время сборки проекта. Если хотя бы один тест провалился, сборка должна быть прекращена;
8. Взаимодействие с базой данных должно происходить при помощи фреймворка Hibernate и языка запросов HQL (простые запросы могут быть выполнены средствами самого фреймворка);
9. Исходный код должен быть логически разбит по пакетам;

Требования к отчету:

1. Отчет должен быть оформлен согласно требованиям к ВКР;
2. Отчет должен содержать следующие обязательные пункты:
   1. титульный лист;
   2. оглавление;
   3. анализ выбранной темы;
   4. проектирование серверной части;
   5. проектирование структуры базы данных;
   6. разработка;
   7. тестирование;
   8. описание клиентской части;
   9. демонстрация функционала.

Требования к презентации:

* + 1. Оформление студенты выбирают сами, но оно должно соответствовать требованием к ВКР;
    2. Количество слайдов не ограничено, но выступление не должно превышать 5-7 минут;
    3. В презентации должно содержаться только та часть, что реализовали студенты, не должно быть слайдов с теорией или обзоров других продуктов, использованных в ходе разработки.

# ВВЕДЕНИЕ

## Анализ выбранной темы

В настоящее время тенденции развития системы образования способствуют повышению спроса на дистанционные электронные системы обучения для студентов. Пропорционально увеличивается необходимость в разработке интерактивных онлайн-курсов, направленных на освоение конкретных профессиональных компетенций, способствующих дальнейшему более успешному освоению образовательной программы и конкурентоспособности на рынке труда.

Мы рассмотрим создание онлайн-платформы с курсами по программированию для начинающих. Этот анализ поможет определить цели, задачи и актуальность данной работы.

**Актуальность проекта:**

* Рост интереса к программированию. Современный мир становится все более цифровым, и программирование становится ключевым навыком. Начинающие программисты нуждаются в доступных ресурсах для обучения.
* Онлайн-образование. Онлайн-курсы по программированию предоставляют возможность учиться из любой точки мира, а также онлайн-курсы позволяют прослушивать лекционный материал и выполнять задания в любое удобное для учащегося время.
* Потребность в учебных ресурсах. Множество людей хочет начать карьеру в программировании, и им нужны ресурсы для обучения и развития.

**Цели проекта:**

* Предоставить качественное и доступное образование в области программирования для новичков.
* Создать сообщество, где ученики могут обмениваться знаниями и опытом.
* Улучшить навыки программирования и работы с современными технологиями.

**Задачи проекта:**

* Создание образовательного контента.
* Создание серверной части с базой данных для хранения курсов и пользовательских данных.
* Разработка клиентской части с удобным интерфейсом и системой навигации.
* Реализация системы аутентификации пользователей.
* Управление доступом к курсам и контенту в зависимости от уровня пользователя.

Чтобы эффективно функционировать и обеспечить комфортное обучение, платформа определяет различные роли пользователей, каждая из которых имеет свои уникальные функциональные обязанности.

**Описание ролей.**

Администратор:

* Управление контентом: Добавление, редактирование и удаление курсов, лекций и материалов.
* Управление пользователями: Создание и управление аккаунтами пользователей, назначение ролей и прав.
* Мониторинг активности: Отслеживание активности пользователей, статистика по курсам.
* Решение конфликтов и поддержка: Реагирование на жалобы и запросы от пользователей.

Преподаватель:

* Создание и редактирование курсов: Загрузка материалов, создание лекций, тестов и домашних заданий.
* Оценка студентов: Выставление оценок за домашние задания и тесты.
* Модерация форума: Ответы на вопросы студентов и обсуждение тем курса.

Студент:

* Регистрация и вход: Создание аккаунта, вход на сайт.
* Просмотр курсов: Просмотр списка доступных курсов, их описаний и преподавателей.
* Участие в курсах: Запись на курсы, просмотр лекций и материалов, выполнение домашних заданий и тестов.
* Участие в форумах: Обсуждение вопросов, связанных с курсом, обмен опытом с другими студентами.

Функциональные возможности:

* Полный контроль и управление сайтом с помощью интернет-браузера без дополнительного клиентского программного обеспечения.
* Веб-публикации - интерактивное создание и управление структурой сайта, публикация и редактирование текстовых и графических материалов пользователем, у которого есть на это права, с любого компьютера, подключенного к сети интернет.
* Регистрация и авторизация посетителей на сайте, поддержка пользователей, различных по назначению и привилегиям.
* Управление правами доступа. Возможно разграничение прав посетителей как на изменение, так и на просмотр информации на уровне подразделов, страниц и даже отдельных блоков на страницах сайта.
* Управление настройками и внешним видом сайта, включая создание отдельных страниц и подсайтов со своим дизайном и системой навигации, с учетом прав доступа.
* Создание и ведение своих собственных баз данных, управляющих формированием динамических блоков на страницах сайта.

# ОСНОВНАЯ ЧАСТЬ

Для визуализации серверной части сервиса, мы составили реляционную таблицу. Таблица сущностей является фундаментальным элементом базы данных этого сервиса. В ней отражены различные аспекты образовательных курсов, студентов, преподавателей и другие важные детали, которые обеспечивают правильное функционирование платформы, а также показывает их связи.

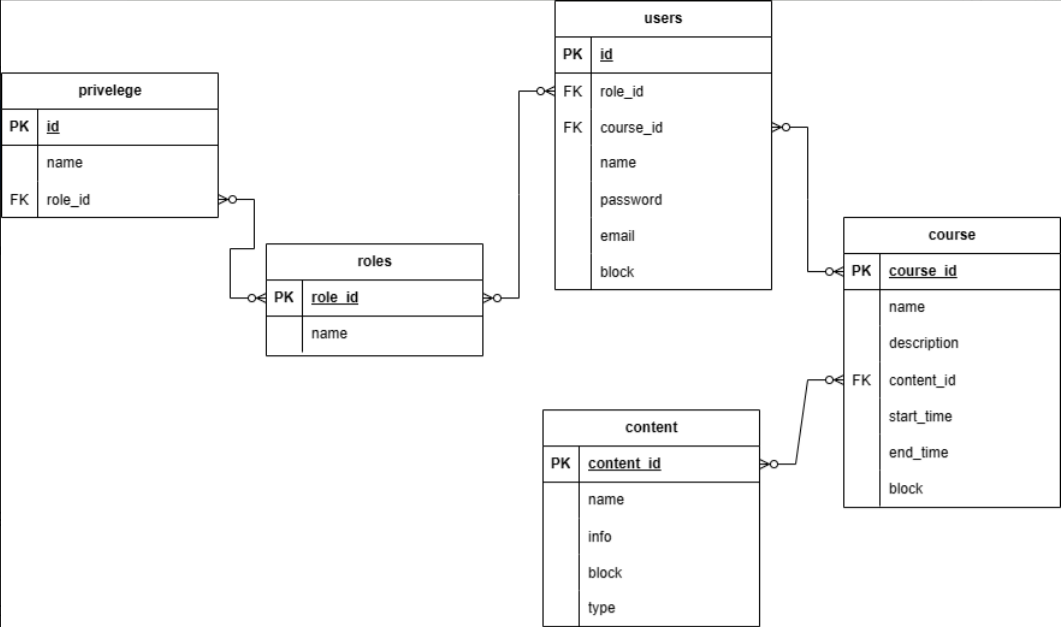


Рис.1. Реляционная таблица сущностей сервиса

Создание онлайн-платформы с курсами по программированию для начинающих представляет собой перспективный и образовательно-ориентированный проект. Он будет способствовать росту навыков программирования у новичков и может успешно конкурировать на рынке онлайн-образования при качественной реализации. Кроме того, этот проект позволит нам приобрести ценный опыт в разработке сервисов и улучшить свои навыки программирования.

В проекте используются следующие технологии:

* Версия Java 17
* Версия Spring Boot 3.1.4
* Версия Spring Security 6.1.4
* Maven
* mySQL
* Lombok

Реализация проекта выглядит следующим образом.

**1. Структура проекта:**

├── src

│ ├── main

│ │ ├── java

│ │ │ └── com

│ │ │ └── example

│ │ │ └── online\_courses

│ │ │ ├── config

│ │ │ ├── controllers

│ │ │ ├── dto

│ │ │ ├── models

│ │ │ ├── exceptions

│ │ │ ├── repositories

│ │ │ ├──service

│ │ │ └── util

│ │ │ └── привет

│ │ ├── resources

│ │ │ └── templates

│ │ │ └── привет

│ └── test

│

└── pom.xml

**2. Модули проекта:**

- `config`: Модуль, содержащий классы для настройки безопасности приложения с использованием Spring Security.

- `controllers`: Модуль, содержащий классы контроллеров для обработки HTTP-запросов и взаимодействия с клиентом.

- `dto`: Здесь определены классы для передачи данных между клиентом и сервером.

- `exceptions`: Модуль, содержащий классы исключений для обработки ошибок.

- `models`: В этом модуле находятся классы, представляющие модели данных, такие как курсы, пользователи и т.д.

- `repositories`: Здесь определены интерфейсы репозиториев для взаимодействия с базой данных, используя Spring Data JPA.

- `service`: Здесь определены сервисы, которые содержат бизнес-логику приложения и взаимодействуют с репозиториями.

- `util`: Здесь определены методы для обработки запросов dto

**4. Реализация кода:**

- Создание классов моделей, такие как `Course`, `User`, `Content`, и так далее, с помощью аннотаций Lombok

- Определение интерфейсов репозиториев с использованием Spring Data JPA, аннотируя их аннотацией `@Repository`.

- Реализация сервисов с бизнес-логикой приложения, аннотируя их аннотацией `@Service`.

- Создание контроллеров для обработки HTTP-запросов, используя аннотацию `@Controller` и реализуя необходимые методы обработки запросов.

- Настройка аутентификации и авторизации с помощью Spring Security, настраивая классы конфигурации для определения прав доступа и настройки безопасности.

Также важно продумать архитектуру и дизайн приложения, обеспечить безопасность и организовать соответствующие механизмы авторизации, работу с пользовательскими данными, курсами и другими необходимыми функциями для платформы онлайн-курсов.

## Проектирование серверной части

Проектирование серверной части является ключевым этапом создания эффективной и функциональной платформы для онлайн-курсов программирования. Для того, чтобы обеспечить безопасность, эффективно управлять данными и структурированно взаимодействовать с проектом, мы выбрали архитектурную модель проектирования программного обеспечения MVC. MVC — это сокращение, которое обозначает Model-View-Controller (Модель-Представление-Контроллер).

**Модель** представляет собой основные данные и бизнес-логику приложения. Отвечает за хранение данных, их обработку и обновление в ответ на внешние события.

**Представление** отвечает за отображение данных пользователю и взаимодействие с пользователем. Получает данные из модели и отображает их пользователю.

**Контроллер** обрабатывает входные данные от пользователя (например, нажатие кнопок, ввод текста). Отправляет команды модели или представлению в зависимости от входных данных.

**Преимущества использования архитектуры MVC включают:**

* Каждая компонента выполняет свою специфическую задачу, что упрощает поддержку, модификацию и тестирование кода.
* Компоненты можно повторно использовать в других частях приложения или даже в других проектах.
* Изменения в одной компоненте обычно не затрагивают другие, что упрощает сопровождение и добавление новых функций.
* Структура MVC делает код более организованным, понятным и легким для понимания другими разработчиками.

Данная архитектура очень удобна для командной разработки, поэтому для нас это лучший вариант.

## Проектирование структуры базы данных

В представленном коде были созданы пять сущностей: user, course, content, role и privilege.

Основной сущностью является user, которая содержит информацию о всех пользователях и связана с role и course с помощью отношения "многие-ко-многим".

Сущность role, в свою очередь, определяет доступные роли для назначения пользователю (admin, teacher, user) и связана с privilege с помощью отношения "многие-ко-многим".

Сущность privilege описывает возможности пользователя или допустимые действия на сайте.

Сущность course содержит информацию о созданных курсах и связана с content с использованием отношения "многие-ко-многим".

Сущность content содержит учебные материалы для курсов. Всего в базе данных присутствует восемь таблиц, причем пять из них соответствуют сущностям, а остальные три служат для реализации отношения "многие-ко-многим".

## Разработка

## Тестирование

## Описание клиентской части

Ниже представлены основные возможности, которые доступны нашим клиентам:

**Регистрация и вход.** Зарегистрируйтесь на сайте, чтобы получить полный доступ к функционалу. Если у вас уже есть аккаунт, просто войдите, используя свои учетные данные.

**Запись на курс.** После успешной регистрации вы сможете легко записываться на интересующие вас курсы. Просто выберите желаемый курс из нашего каталога и следуйте указаниям для завершения процесса записи. (в разработке)

**Просмотр курсов.** Исследуйте наше предложение образовательных программ. У нас есть широкий спектр курсов, охватывающих различные языки программирования. Вы можете просматривать все доступные курсы.

**Просмотр определенного курса.** Получите подробную информацию о каждом курсе, включая его описание, программу обучения и многое другое. Это поможет вам выбрать наиболее подходящий для ваших потребностей и интересов курс. (в разработке)

## Демонстрация функционала

# ЗАКЛЮЧЕНИЕ

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

# ПРИЛОЖЕНИЯ