



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА _____ КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ _____

НАПРАВЛЕНИЕ ПОДГОТОВКИ _09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА_

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ

НА ТЕМУ:

ВЕБ-ПРИЛОЖЕНИЕ

ПЛАТФОРМА ОНЛАЙН-ТРЕНИРОВОК

Студент ИУ6-51Б
(Группа)

Мед 01.03.2022 Л.С. Эльмукаев
(Подпись, дата) (И.О. Фамилия)

Руководитель курсовой работы

Б 1.01.2022 Н.О. Кузнецов
(Подпись, дата) (И.О. Фамилия)

Прогр. 405
Б

2021 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой ИУ6
(Индекс)

А.В. Пролетарский
(И.О. Фамилия)

« 11 » сентября 2021 г.

ЗАДАНИЕ
на выполнение курсовой работы

по дисциплине Технология разработки программных систем

Студент группы ИУ6-51Б

Эльнукаев Лом-Али Сайд-Муратович
(Фамилия, имя, отчество)

Тема курсовой работы Платформа онлайн-тренировок

Направленность КР (учебная, исследовательская, практическая, производственная, др.)
учебная

Источник тематики (кафедра, предприятие, НИР) кафедра

График выполнения КР: 25% к 4 нед., 50% к 7 нед., 75% к 11 нед., 100% к 14 нед.

Техническое задание см. техническое задание в приложении А

Оформление курсовой работы:

1. Расчетно-пояснительная записка (РПЗ) на 25-30 листах формата А4.
2. Техническое задание на 5-9 листах формата А4 – оформляется в качестве приложения А к РПЗ.
3. Руководство пользователя на 6-8 листах формата А4 – оформляется в качестве приложения Б к РПЗ (если предусмотрено в техническом задании).
4. Графический и иллюстративный материал оформляется в виде рисунков и помещается в РПЗ.
5. Все материалы и исходный текст программы загрузить на страницу дисциплины на сайте кафедры.

Дата выдачи задания « 1 » сентября 2021 г.

Руководитель курсовой работы

Н.О. Курнечов
(Подпись, дата) (И.О. Фамилия)

Студент

Л.С. Эльнукаев
(Подпись, дата) (И.О. Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

Реферат

Записка 66 страниц, 30 рисунков, 8 таблиц, 4 источника, 3 приложения.

Объектом разработки является веб-приложение платформа онлайн-тренировок, предназначенное для создания индивидуального плана тренировок, их отображения, хранимых в базе данных, а также администрирования базы данных.

Цель работы – веб-приложение платформа онлайн-тренировок, позволяющее:

- создание курсов, модулей и тренировок;
- редактирования курсов, модулей и тренировок;
- удаление курсов, модулей и тренировок;
- просмотр выбранного курса;
- выбор курса из списка предложенных пользователю;
- создание нового пользователя;
- авторизация пользователя;
- вход в личный кабинет;
- редактирование личных данных пользователя;
- редактирование прав доступа пользователей и запись пользователей на курсы;
- просмотр личных данных пользователей;
- удаление пользователей.

В результате разработки было спроектировано и реализовано веб-приложение платформа онлайн-тренировок.

Актуальность разрабатываемого приложения обусловлено тем, что каждому человеку необходимо заниматься спортом.

Пользователями платформы могут быть люди, желающие продвинуть свои услуги тренера в области спорта и те, кто потребляет представленную на платформе онлайн-тренировок информацию.

Содержание

Введение	5
1 Анализ технических требований и уточнение спецификация	6
1.2 Анализ и выбор технологии, языка и среды разработки	6
1.3 Выбор модели жизненного цикла программного обеспечения	7
1.4 Разработка диаграммы вариантов использования	8
1.5 Разработка функциональной диаграммы программного продукта	11
1.6 Разработка диаграмм деятельности	14
2 Проектирование структуры и компонентов программного продукта.....	16
2.1 Разработка структурной схемы программного продукта	16
2.2 Разработка информационной модели программного продукта	17
2.3 Построение графа состояний интерфейса	18
2.4 Проектирование диалогов и разработка форм интерфейса.....	20
2.5 Проектирование базы данных	30
2.6 Проектирование классов	32
3 Выбор стратегии тестирования и разработка тестов.....	35
3.1 Функциональное тестирование	35
3.2 Модульное тестирование	37
3.3 Оценочное тестирование.....	38
Заключение.....	40
Список использованных источников.....	41
ПРИЛОЖЕНИЕ А.....	42
ПРИЛОЖЕНИЕ Б	51
ПРИЛОЖЕНИЕ В	61

Введение

Настоящая курсовая работа посвящена разработке веб-приложения платформа онлайн-тренировок. Настоящая работа является веб-приложением, которое создается для записи, отображения и хранения данных об образовательных инструментах и данных пользователей, хранящихся в базе данных. Главная задача – реализовать возможности создания, редактирования и просмотра курсов тренировок, запись пользователей на определенные курсы тренировок из всех представленных и разделение прав пользователей. Проектируемый интерфейс должен позволять оптимизировать администрирование всей платформы, создавать и редактировать курсы тренировок, личные данные авторизованного пользователя. Каждый курс тренировок может быть изменен и удален вместе с зависимыми от него модулями и тренировками. При этом при удалении модуля выбранного курса тренировок удаляется не только модуль, но и зависимые от него тренировки. Таким образом выстраивается иерархия для работы веб-приложения, предотвращающая хранение избыточных данных в базе данных. В веб-приложении реализованы 15 веб-страниц, позволяющие использовать разработанные функции.

На сегодняшний день существует множество приложений, предназначенных для осуществления помощи в занятии спортивной деятельностью. Однако было выявлено, что существующие приложения обладают следующими недостатками: медленная работа веб-сайта, немобильная архитектура, а также высокая стоимость использования. В итоге был составлен список необходимых функций, которыми должна обладать разрабатываемая программа.

Актуальность разрабатываемого веб-приложения также заключается в том, что он создается как свободно распространяемый аналог подобных решений, который объединяет в себе удобный интерфейс для уже имеющихся функций, а также возможность для расширения.

Предполагается, что потребителями настоящего программного продукта должны быть пользователи, желающие продвинуть свои персональные услуги тренера, и их потребители.

1 Анализ технических требований и уточнение спецификации

1.2 Анализ и выбор технологии, языка и среды разработки

Для того, чтобы упростить манипуляцию данными, а также обеспечить модульную структуру системы был выбран смешанный подход программирования. Данное решение обусловлено тем, что предметная область программы – объектная, которая содержит явные взаимодействия между сущностями, а подход к проектированию интерфейса – структурный.

Объектно-ориентированное программирование определяется как технология создания сложного программного обеспечения, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы, в свою очередь, образуют иерархию с наследованием свойств. Взаимодействие программных объектов в системе осуществляется путём передачи сообщений.

Объектно-ориентированное программирование предполагает декомпозицию программного обеспечения, которая существенно облегчает процесс его разработки [1]. Это приводит к интегрированию с другими подпрограммами обработки, что позволяет вести практически независимую разработку отдельных частей программы, а также приводит к более полной локализации данных. Кроме того, механизмы наследования, наполнения, композиции, полиморфизма позволяют конструировать сложные объекты из более простых.

Для реализации поставленных задач был выбран фреймворк Spring, являющийся расширением для объектно-ориентированного языка программирования Java, поддерживающего структурный подход. Достоинствами данного фреймворка являются:

- 1) удобство разработки – объекты слабо связаны за счет использования внедрения зависимостей, настройки компонентов отделены от программного кода;
- 2) высокая скорость разработки;
- 3) высокая надежность и сопровождаемость решений;
- 4) отчуждаемость – соблюдение заложенных во фреймворк соглашений и стандартов кодирования делает программный код сопровождаемым не только изначальными разработчиками, но и любыми другим специалистами;
- 5) возможность программирования в декларативном стиле с помощью аннотаций уменьшает количество кода в приложении;
- 6) поддержка и интеграция с технологиями доступа к данным;
- 7) наличие подробной документации

В качестве текстового редактора для написания исходного кода программы была выбрана полнофункциональная интегрированная среда разработки (IDE) от JetBrains IntelliJ IDEA. IntelliJ IDEA – это комплексная кроссплатформенная IDE. К преимуществам данной среды можно отнести:

- 1) Удобная навигация, инструменты для анализа качества кода, расширенные рефакторинги и форматирование для Java, HTML, CSS, JavaScript, XML и многих других языков.
- 2) Инструменты для работы с базами данных и SQL файлами.
- 3) Интеграция с серверами приложений, включая Tomcat, TomEE, GlassFish, JBoss, WebLogic, WebSphere, Geronimo, Resin, Jetty и Virgo.
- 4) Интеграция с коммерческими системами управления версиями Perforce, Team Foundation Server, ClearCase, Visual SourceSafe.
- 5) Инструменты для запуска тестов и анализа покрытия кода, включая поддержку всех популярных фреймворков для тестирования.

Для разработки базы данных была выбрана СУБД MySQL. Это современная система управления данными, обеспечивающая хранилище данных для веб-приложений и быстрый доступ к ним.

1.3 Выбор модели жизненного цикла программного обеспечения

При разработке веб-приложений хорошо широко применяется метод прототипирования, который базируется на создании прототипов. Прототипом называют действующий программный продукт, реализующий отдельные функции и внешние интерфейсы разрабатываемого программного обеспечения.

В связи с этим, при разработке использована спиральная модель жизненного цикла создаваемого программного продукта, которая основывается на методе прототипирования. Спиральная модель позволяет быстро спроектировать, реализовать и протестировать первый рабочий прототип, который можно показать заказчику для дальнейшего уточнения требуемых условий и характеристик проекта.

Это позволяет:

- заинтересовать большое количество пользователей, а следовательно, обеспечить быстрое продвижение на рынке;
- сократить время до появления первых версий программного продукта;
- ускорить формирование и уточнение спецификаций за счет появления практики использования продукта;

Все принимаемые решения апробируются путем создания прототипов. На завершающем шаге каждого витка спирали создается версия, на которой уточняются все

требуемые условия и характеристики проекта и планируется следующая итерация. Также, спиральная модель облегчает дальнейшее создание новых версий программного продукта.

1.4 Разработка диаграммы вариантов использования

Спецификации веб-приложения платформы онлайн-тренировок рассматриваются диаграммой вариантов использования. Она призвана описать взаимодействие с действующим лицом – зарегистрированным пользователем, незарегистрированным пользователем, тренером и администратором. В результате анализа технического задания были выделены следующие варианты использования:

- 1) просмотр списка доступных курсов тренировок;
- 2) просмотр тренировок;
- 3) регистрация аккаунта;
- 4) авторизация аккаунта;
- 5) изменение личных данных;
- 6) просмотр списка всех курсов тренировок;
- 7) создание курсов тренировок;
- 8) редактирование курсов тренировок;
- 9) удаление курсов тренировок;
- 10) просмотр списка пользователей;
- 11) изменение прав пользователей;
- 12) запись пользователей на курс тренировок;
- 13) редактирование логина пользователя;
- 14) изменение прав пользователей.

Некоторые варианты использования рассмотрены более подробно в таблицах 1–6.

Таблица 1 – Описание варианта использования "просмотр тренировок"

Название варианта	Просмотр тренировок
Цель	<i>Просмотр и выбор курса тренировок из списка предложенных</i>
Действующие лица	<i>Зарегистрированный пользователь</i>
Краткое описание	<i>Выбирает нужный курс тренировок и переходит к просмотру предложенной информации</i>
Тип	<i>Основной</i>

Таблица 2 – Вариант использования "просмотр тренировок"

Действие исполнителя	Отклик системы
<p>1. Авторизованный пользователь, находясь на странице главного меню, нажимает кнопку «Войти».</p> <p>3. Пользователь заканчивает просмотр выбранного курса тренировок, нажимая любую кнопку, представленную на панели навигации.</p>	<p>2. Приложение открывает страницу с предложенными в курсе модулями и тренировками, а также со всей информацией к этим тренировкам.</p> <p>4. Приложение переходит на выбранную страницу. Просмотр предложенного курса завершается.</p>

Таблица 3 – Описание варианта использования "регистрация аккаунта"

Название варианта	Регистрация аккаунта
Цель	Создать новый аккаунт
Действующие лица	Незарегистрированный пользователь, администратор
Краткое описание	Создается новый аккаунт с необходимыми данными
Тип	Основной

Таблица 4 – Вариант использования "регистрация аккаунта"

Действие исполнителя	Отклик системы
<p>1. Незарегистрированный пользователь нажимает кнопку «Регистрация» или администратор на панели администратора нажимает кнопку «Создать пользователя»</p> <p>3. Пользователь активирует поле ввода нажатием указателя мыши и вводит запрашиваемые данные. Затем он нажимает кнопку «Зарегистрироваться»</p>	<p>2. Приложение открывает страницу с формой регистрации, в которой есть поля для ввода: логин (текст), пароль (текст), имя (текст), фамилия (текст), отчество (текст), мобильный телефон (текст), электронную почту (текст) и дату рождения (дата).</p> <p>4. Приложение добавляет пользователя с личными данными в базу данных и переводит его на страницу авторизации.</p>

Таблица 5 – Описание варианта использования "создание курса тренировок"

Название варианта	Создание курса тренировок
Цель	<i>Создать новый курс тренировок</i>
Действующие лица	<i>Тренер</i>
Краткое описание	<i>Создание курса тренировок путем ввода необходимой информации</i>
Тип	<i>Основной</i>

Таблица 6 – Вариант использования "создание курса тренировок"

Действие исполнителя	Отклик системы
<p>1. Пользователь с ролью «Тренер» нажимает на кнопку навигационной панели «Редактор курсов тренировок»</p> <p>3. Пользователь активирует поле ввода нажатием указателя мыши и вводит запрашиваемые данные. Затем он нажимает кнопку «Добавить»</p>	<p>2. Приложение открывает страницу редактора курсов тренировок с таблицей всех курсов тренировок и формой «Добавление курса тренировок», в которой присутствуют поля ввода: название курса тренировок (текст), тег курса (текст).</p> <p>4. Приложение добавляет новый курс тренировок в базу данных и выводит обновленный список всех курсов тренировок.</p>

Для наглядного представления всех возможностей пользователя программы на рисунке 1 представлена диаграмма вариантов использования.

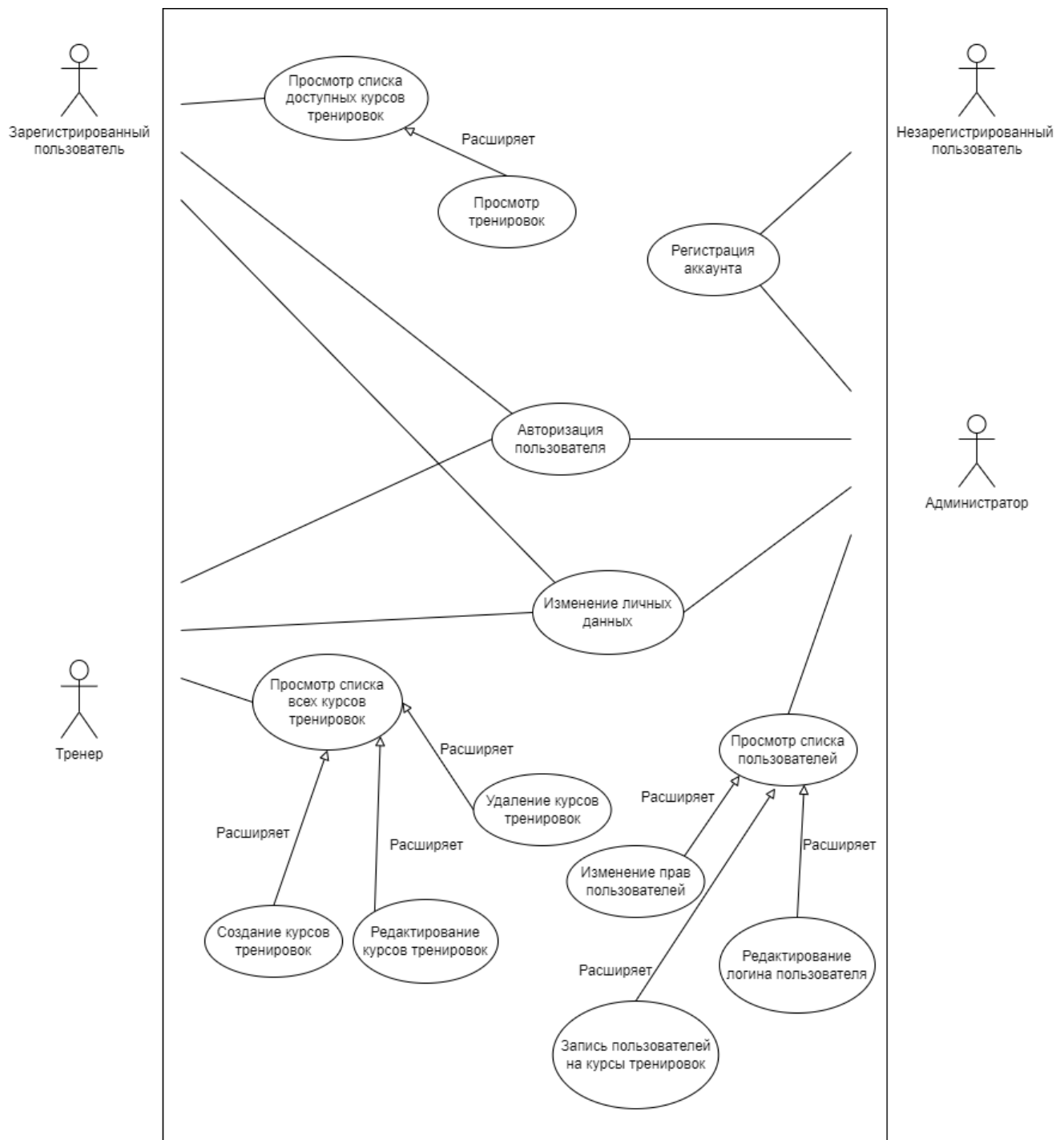


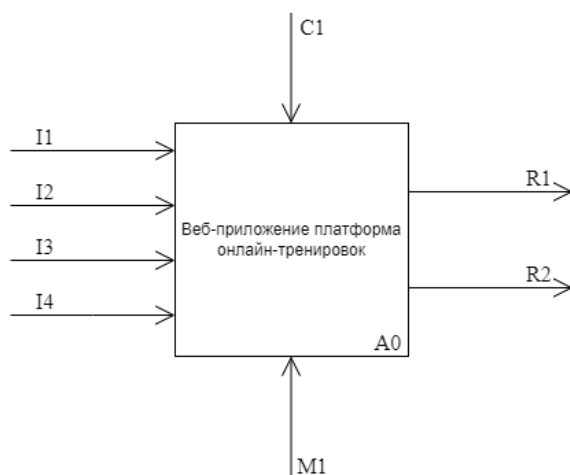
Рисунок 1 – Диаграмма вариантов использования

Таким образом, основными вариантами использования стали: «Просмотр списка всех курсов тренировок», «Просмотр списка доступных курсов тренировок», «Просмотр списка пользователей», «Регистрация аккаунта», «Авторизация аккаунта», «Изменение личных данных».

1.5 Разработка функциональной диаграммы программного продукта

В соответствии с техническим заданием на вход программы должны поступать имя, пароль и запрос пользователя, а также исходная база данных, на выходе должны формироваться изменённая база данных и веб-страница с результатом запроса

пользователя. Это можно увидеть на функциональной диаграмме верхнего уровня, изображённой на рисунке 2.



- I1 - логин пользователя
- I2 - пароль пользователя
- I3 - исходная база данных
- I4 - запрос пользователя
- C1- уровни доступа
- M1- верификация данных со стороны системы
- R1- измененная база данных
- R2- веб-страница с результатом запроса пользователя

Рисунок 2 – Функциональная диаграмма верхнего уровня

Для уточнения функций веб-приложения была разработана детализирующая функциональная диаграмма. Она изображена на рисунке 3.

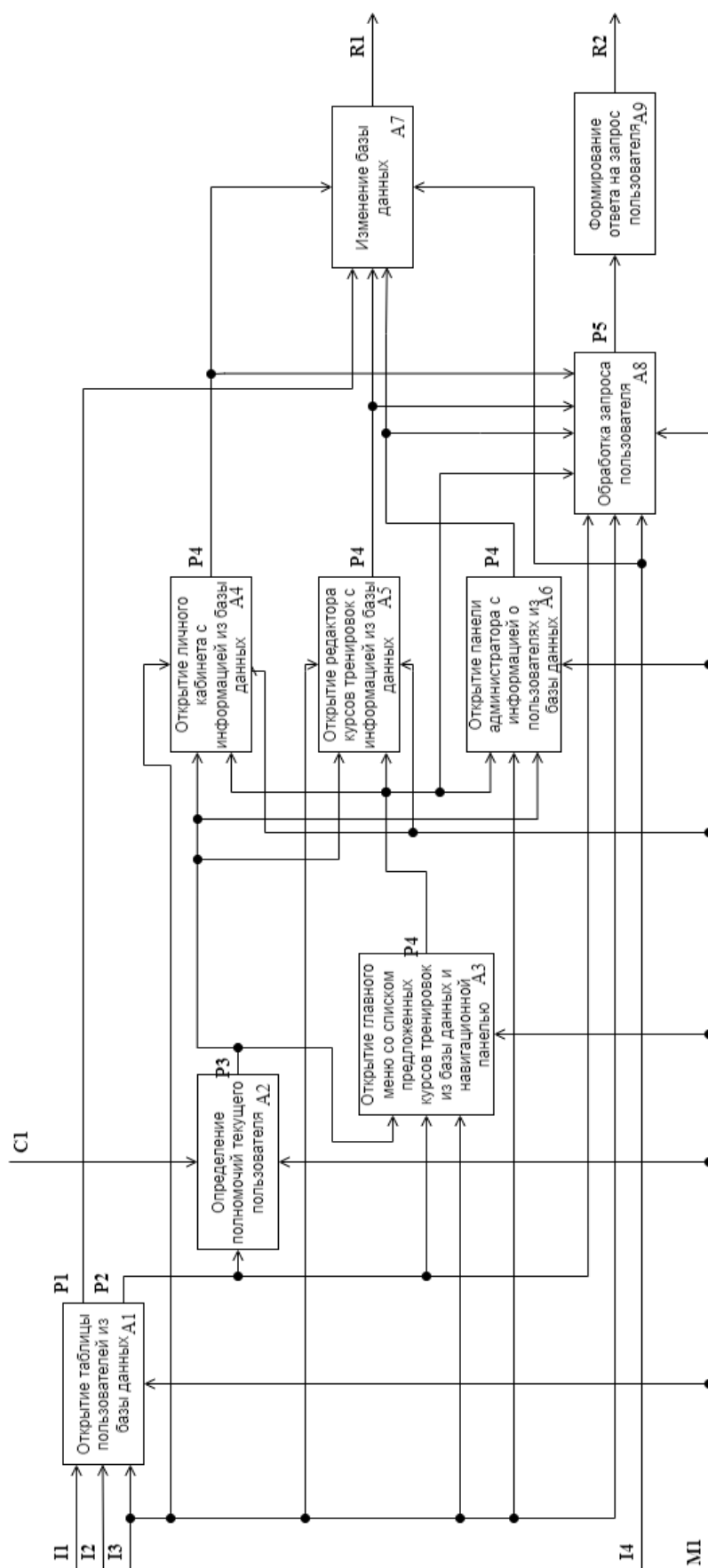


Рисунок 3 – Детализирующая функциональная диаграмма

На детализирующей функциональной диаграмме показано 9 блоков: открытие таблицы пользователей из базы данных, определение полномочий текущего пользователя, открытие главного меню со списком предложенных курсов тренировок из базы данных, открытие личного кабинета с информацией из базы данных, открытие редактора курсов тренировок с информацией из базы данных, открытие панели администратора с информацией о всех пользователях из базы данных, изменение базы данных, обработка запроса пользователя, формирование запроса на ответ пользователя.

1.6 Разработка диаграмм деятельности

С учетом описания предметной области в виде детализирующей функциональной диаграммы были проанализированы описания вариантов использования – «Просмотр списка пользователей», «Регистрация аккаунта», «Авторизация аккаунта», «Изменение личных данных», «Просмотр списка всех курсов тренировок», «Просмотр списка доступных курсов тренировок». Процесс был разбит на отдельные операции. Часть из них представлена на диаграммах деятельности, изображенных на рисунках 4 и 5.

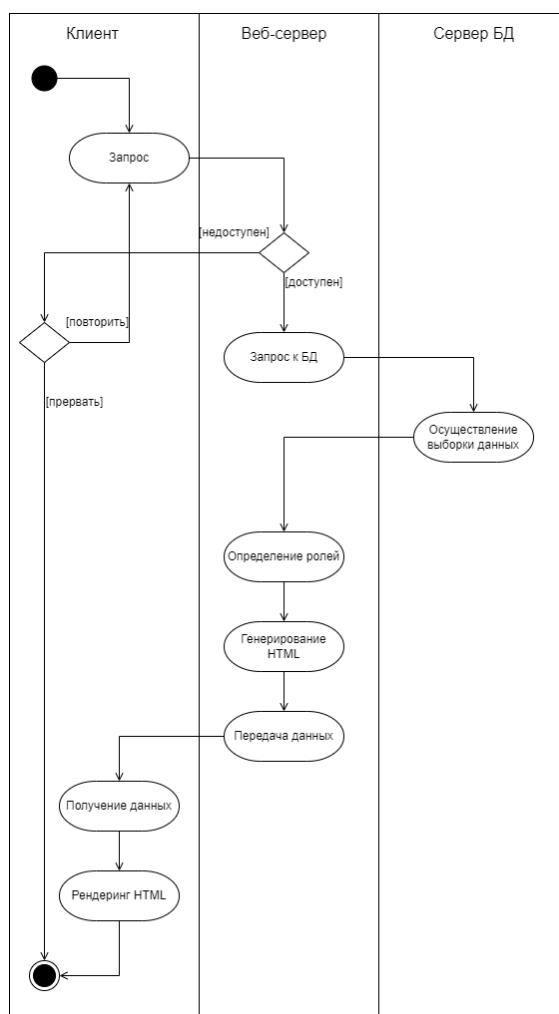


Рисунок 4 – Диаграмма деятельности для авторизации пользователя

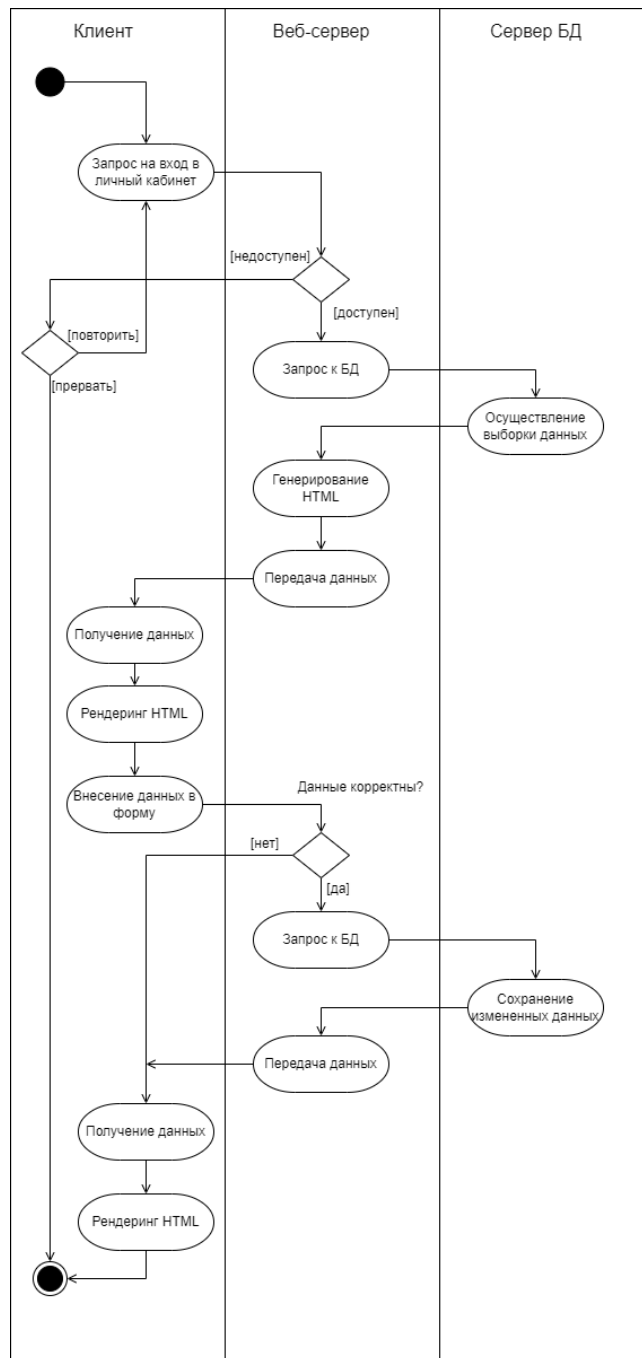


Рисунок 5 – Диаграмма деятельности для изменения личных данных

В зависимости от степени детализации диаграммы деятельности так же, как диаграммы классов, используют на разных этапах разработки. На этапе анализа требований и уточнения спецификаций диаграммы деятельности позволяют конкретизировать основные функции разрабатываемого программного обеспечения. Под деятельностью в данном случае понимают задачу (операцию), которую необходимо выполнить вручную или с помощью средств автоматизации. Каждому варианту использования соответствует своя последовательность задач.

2 Проектирование структуры и компонентов программного продукта

2.1 Разработка структурной схемы программного продукта

Процесс проектирования программного обеспечения начинается с уточнения его структуры, т. е. определения структурных компонентов и связей между ними. Результат уточнения структуры может быть представлен в виде структурной схемы программного продукта и описания (спецификаций) компонентов. Разработку структурной схемы программного продукта выполняют методом пошаговой детализации.

Структурная схема программного продукта показывает наличие подсистем и других структурных компонентов. В отличие от программного комплекса, отдельные подсистемы интенсивно обмениваются данными между собой и, возможно, с основной программой. Структурная схема программного продукта этого обычно не показывает.

Исходя из задач и функций, выполняемых системой в целом и её подсистемами, получим структурную схему веб-приложения платформа онлайн-тренировок. Структурная схема изображена на рисунке 6.

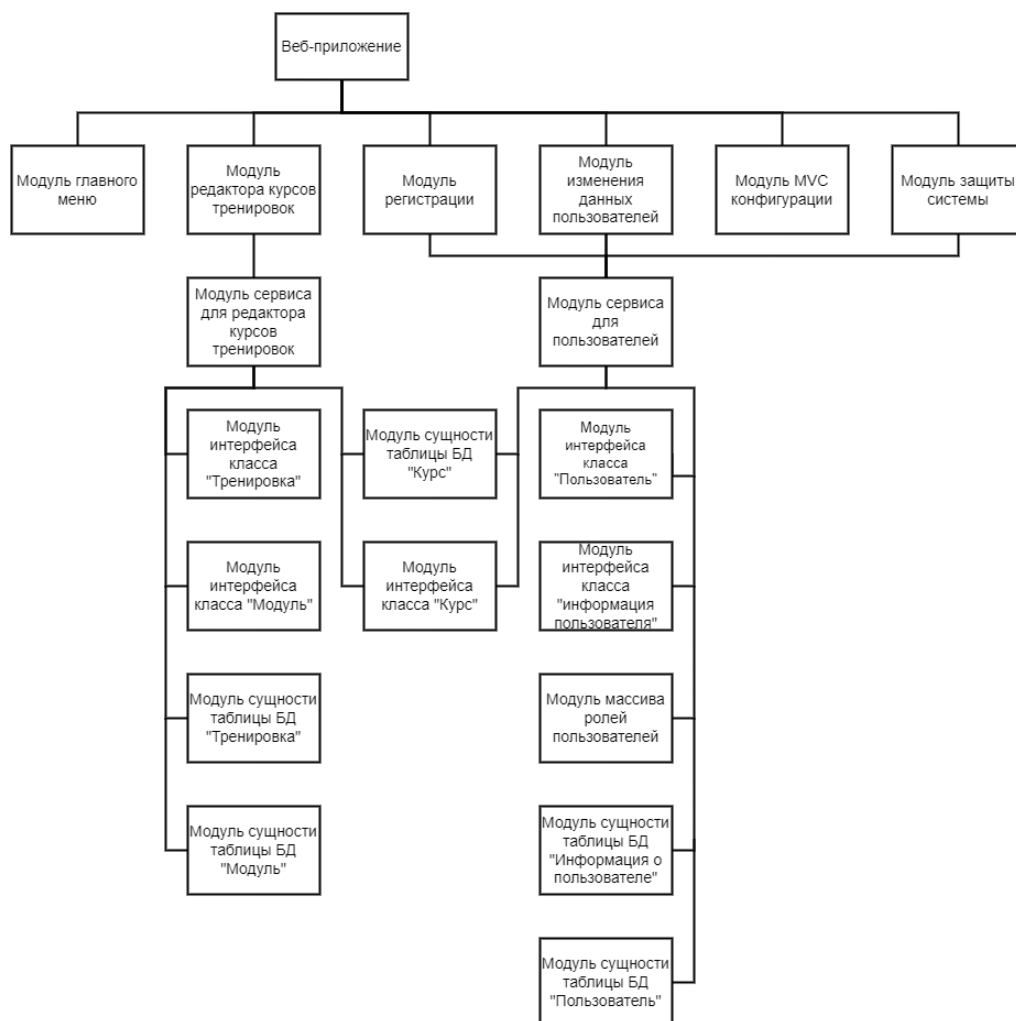


Рисунок 6 – Структурная схема программного продукта

Модуль главного меню позволяет переходить на страницу главного меню и всех доступных страниц на панели навигации, а также вывод информации о доступных курсах тренировок.

Модуль редактора курсов тренировок предназначен для отображения страниц редактора курсов тренировок, а также для отправки данных из форм. Модуль сервиса для редактора курсов тренировок содержит в себе основные алгоритмы для создания, редактирования и удаления курсов тренировок.

Модуль регистрации содержит в себе отображение главных страниц регистрации, а также принимает данные из формы регистрации.

Модуль изменения данных пользователей отображает страницы личного кабинета, списка пользователей из панели администратора, и принимает данные из форм с этих страниц. Он принимает данные из форм редактирования пользователей как в личном кабинете, так и в панели администратора, который может изменять логин, роли и запись на курсы любого пользователя.

Модуль защиты системы содержит конфигурации безопасности и подпрограмму шифрования пароля пользователя.

Модуль сервиса для пользователей содержит основные алгоритмы для создания пользователей, редактирования, смены ролей, записи на курсы тренировок.

Модуль MVC конфигурации содержит подпрограмму вызова страницы авторизации для незарегистрированных пользователей.

Модули сущностей базы данных содержат все поля таблиц базы данных. В проекте используется технология ORM, которая связывает базы данных с концепциями ООП. В результате разработчику необходимо лишь создать класс с полями и вызываемыми методами под специальной аннотацией «@Entity». Технология ORM преобразует классы в таблицы базы данных. А полями таблиц являются объекты этого класса. Модули интерфейсов содержат в себе алгоритмы работы с соответствующими таблицами базы данных. Создаются методы на языке Java, которые транслируются в SQL. С помощью интерфейсов можно сохранять и удалять объекты в базе данных, а также выполнять поиск по требуемым параметрам без написания кода на SQL.

2.2 Разработка информационной модели программного продукта

Информационная модель программного продукта может быть представлена диаграммой потоков данных [2]. Поток данных - процесс передачи некоторой информации от источника к приемнику. Диаграммы потоков данных позволяют специфицировать функции разрабатываемого программного продукта и обрабатываемые им данные, а также позволяют показать управляющие процессы в программном продукте. Программный

продукт представляется в виде иерархии диаграмм потоков данных, описывающих асинхронный процесс преобразования информации с момента ввода в систему до выдачи пользователю. Результатом такого представления является модель хранения/обработки информации. Диаграмма потоков данных разрабатываемого программного продукта, представленная на рисунке 7.

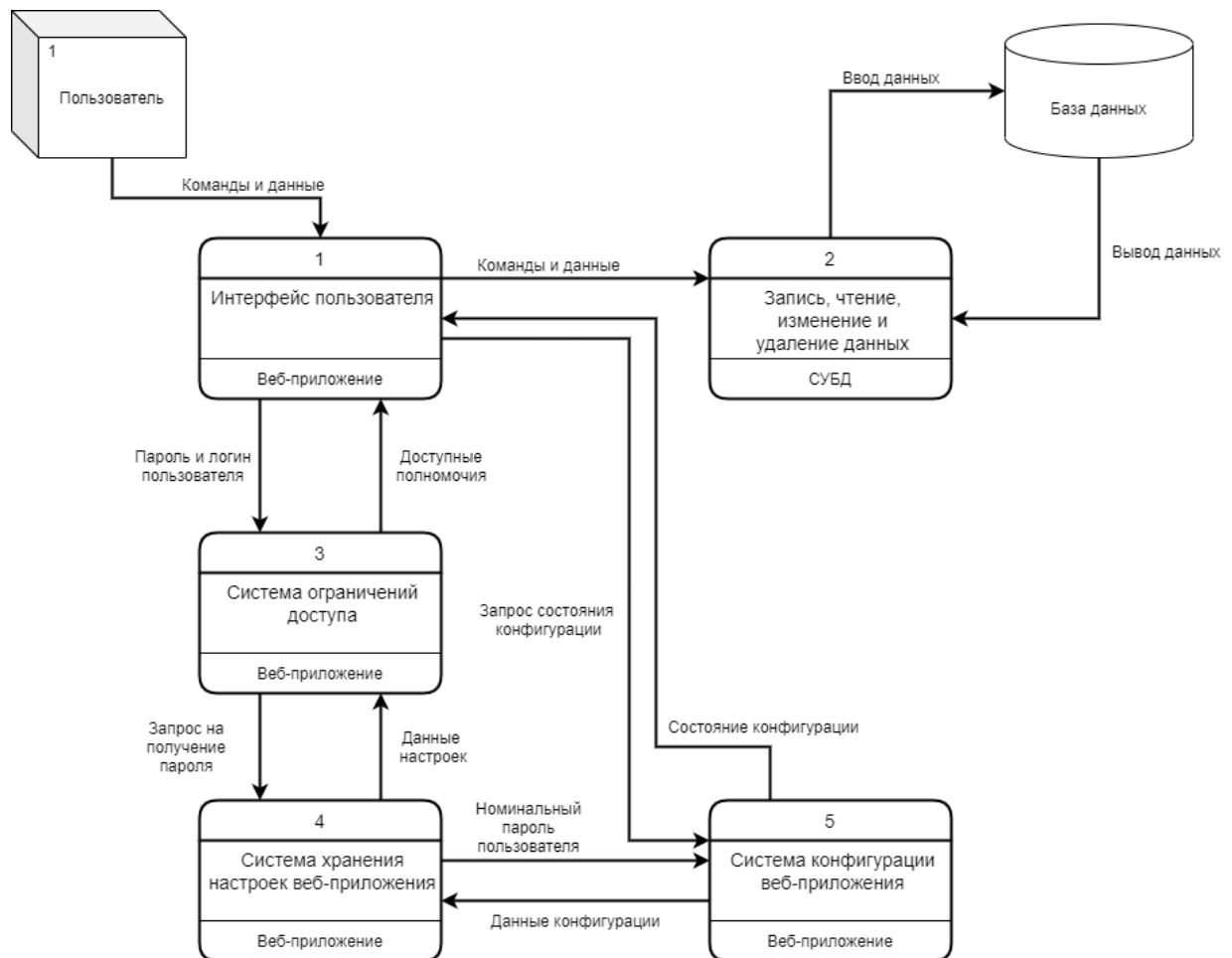


Рисунок 7 – Диаграмма потоков данных

В соответствии с разработанной схемой, было выполнено дальнейшее проектирование структур и компонентов программного продукта.

2.3 Построение графа состояний интерфейса

В соответствии с вариантами использования, можно выделить следующие состояния интерфейса:

- веб-страница главного меню;
- веб-страница создания курсов тренировок;
- веб-страница создания модулей;
- веб-страница создания тренировок;
- веб-страница просмотра выбранной тренировки;
- веб-страница просмотра доступного курса тренировок;

- веб-страница панели администратора;
- веб-страница списка всех пользователей и их данных;
- веб-страница авторизации пользователя;
- веб-страница регистрации пользователя.

Разрабатываемая программа является веб-приложением. Все команды, которые воспринимает приложение, являются событиями. Поэтому был разработан граф состояний интерфейса [2], представленный на рисунке 8. События, которыми вызваны переходы между состояниями, указанными на графе, описаны ниже.

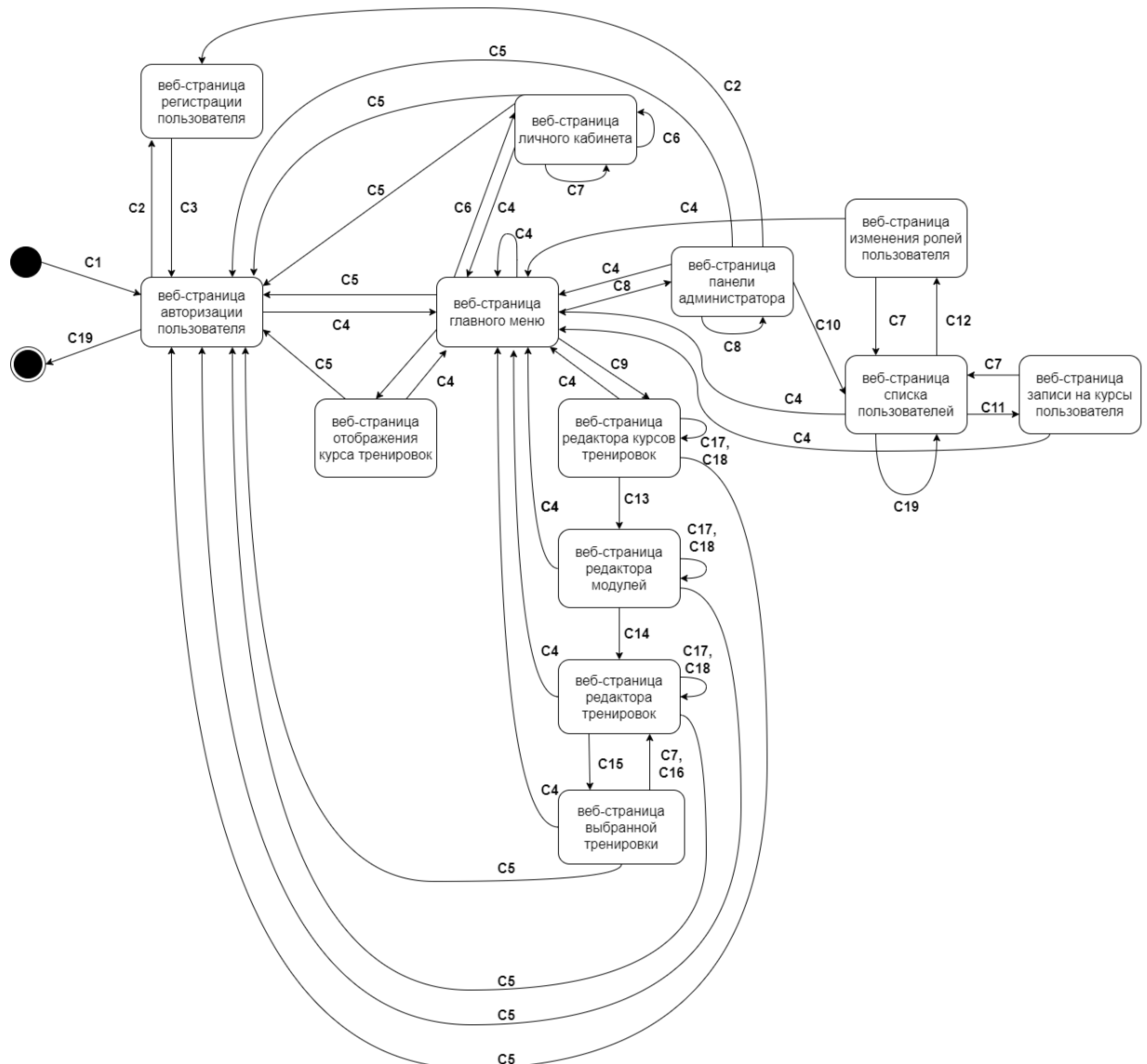


Рисунок 8 – Граф состояний интерфейса

Описание событий:

- C1 – открытие веб-страницы;
- C2 – нажатие кнопки «Создать пользователя»;
- C3 – нажатие кнопки «Зарегистрироваться»;
- C4 – нажатие кнопки «Войти»;

- C5 – нажатие кнопки «Выйти»;
- C6 – нажатие кнопки «Личный кабинет»;
- C7 – нажатие кнопки «Сохранить»;
- C8 – нажатие кнопки «Панель администратора»;
- C9 – нажатие кнопки «Редактор курсов тренировок»;
- C10 – нажатие кнопки «Список пользователей»;
- C11 – нажатие кнопки «Записать на курс тренировок»;
- C12 – нажатие кнопки «Роли»;
- C13 – нажатие на «Название курса тренировок»;
- C14 – нажатие на «Название модуля»;
- C15 – нажатие на «Название тренировки»;
- C16 – нажатие на кнопку «Отменить»;
- C17 – нажатие кнопки «Удалить»;
- C18 – нажатие кнопки «Добавить»;
- C19 – закрытие веб-страницы.

2.4 Проектирование диалогов и разработка форм интерфейса

Веб-приложение включает в себя 13 веб-страниц помимо страниц авторизации и регистрации:

- главное меню;
- личный кабинет;
- информация о текущей тренировке;
- панель администратора;
- роли пользователя;
- таблица пользователей;
- запись на курсы тренировок.
- просмотр курса тренировок;
- редактор курсов тренировок;
- редактор модулей;
- редактор тренировок.

На веб-страницах «Редактор курсов тренировок», «Редактор модулей», «Редактор тренировок» пользователь может выполнять следующие действия:

- создание курса тренировок;
- удаление курса тренировок;
- редактирование курса тренировок;
- создание модуля;

- удаление модуля;
- редактирование модуля;
- создание тренировки;
- удаление тренировки;
- редактирование тренировки.

Веб-страница «Информация о текущей тренировке» необходима для полного редактирования выбранной тренировки. Также с помощью нее можно протестировать корректность и отображение видеоурока по использованной ссылке. Основными полями тренировки являются:

- название тренировки;
- описание тренировки;
- текст тренировки;
- ссылка на видеоурок.

Веб-страницы «Роли пользователя» и «Запись на курсы тренировок» отображает роли выбранного пользователя и доступ к курсам из базы данных соответственно.

Указанные выше экранные формы похожи по своей структуре, в верхней части веб-страницы находится навигационная панель - меню, в котором расположены основные кнопки для переключения между веб-страницами. Все страницы несут в себе разные функции, хотя некоторые из них схожи по функционалу и структуре, например, страницы редактора курсов тренировок, модулей и тренировок.

Рассмотрим страницу создания нового пользователя. На веб-странице создания нового пользователя расположены поля для ввода логина пользователя, электронной почты пользователя, пароля, имени, фамилии, отчества и даты рождения. Также на веб-странице расположена кнопка перехода на веб-страницу входа в систему. При вводе логина, который уже используется в веб-приложении, пользователь увидит предупреждение о том, что аккаунт с таким логином уже создан. Диалог формы веб-страницы создания нового аккаунта пользователя отображен на графе абстрактного диалога, представленном на рисунке 9. На рисунке 10 отображена экранная форма веб-страницы создания нового аккаунта пользователя.

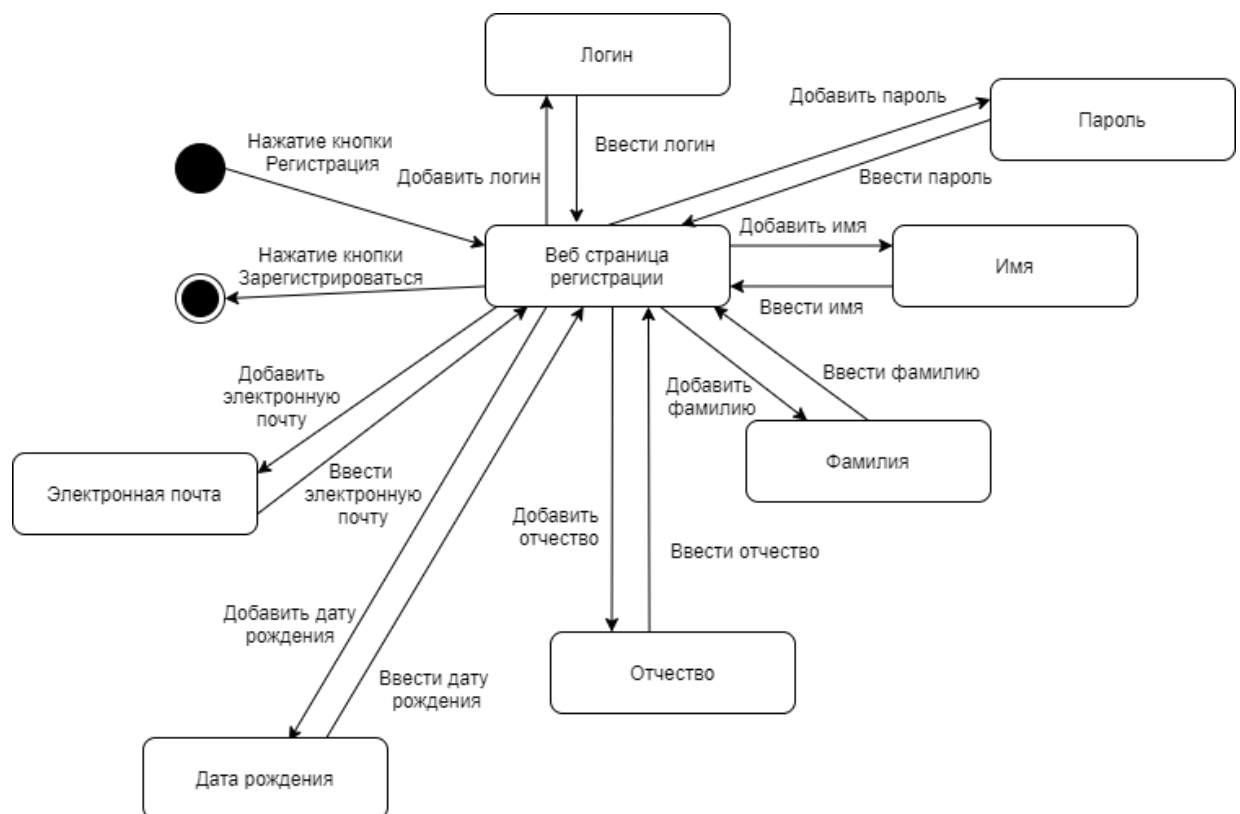


Рисунок 9 – Граф абстрактного диалога веб-страницы создания нового пользователя

Регистрация

Имя пользователя:

Пароль:

Электронная почта:

Номер телефона:

Имя:

Фамилия:

Отчество:

Дата рождения:

test1

.....

ДД.ММ.ГГГГ

поле ввода логина

поле ввода пароля

поле ввода электронной почты

поле ввода номера телефона

поле ввода имени

поле ввода фамилии

поле ввода отчества

поле ввода даты рождения

Кнопка сохранения данного пользователя

Зарегистрироваться

Рисунок 10 – Экранная форма веб-страницы создания нового аккаунта пользователя

На веб-странице входа в систему расположены поля для ввода логина пользователя и пароля пользователя. Также на веб-странице расположены кнопки для подтверждения входа в систему и для перехода на веб-страницу создания нового аккаунта. Диалог формы веб-страницы входа в систему отображен на графе абстрактного диалога, представленном на рисунке 11. На рисунке 12 отображена экранная форма веб-страницы входа в систему.

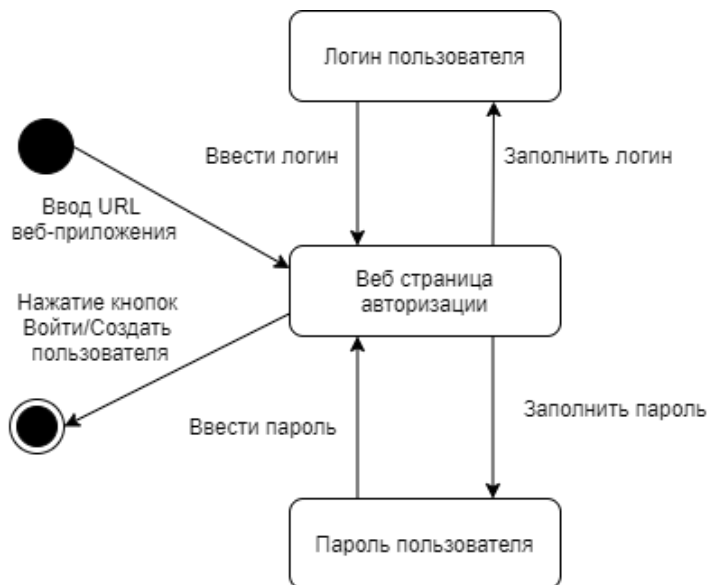


Рисунок 11 – Граф абстрактного диалога веб-страницы входа в систему

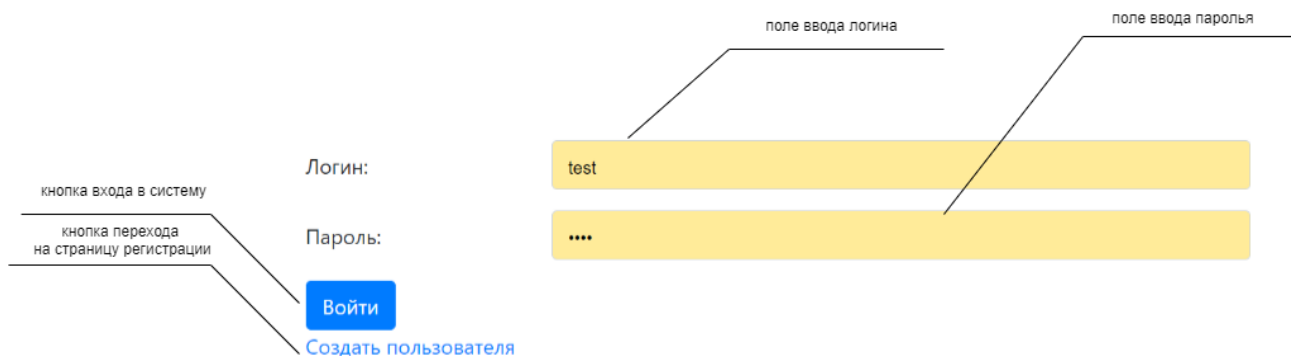


Рисунок 12 – Экранная форма веб-страницы входа в систему

Веб-страница главного меню является основной страницей, на которую пользователь попадает после авторизации. На веб-странице расположена навигационная панель сверху и список курсов, на которые у данного пользователя есть запись. Граф абстрактного диалога и экранная форма веб-страницы представлены на рисунках 13 и 14 соответственно.

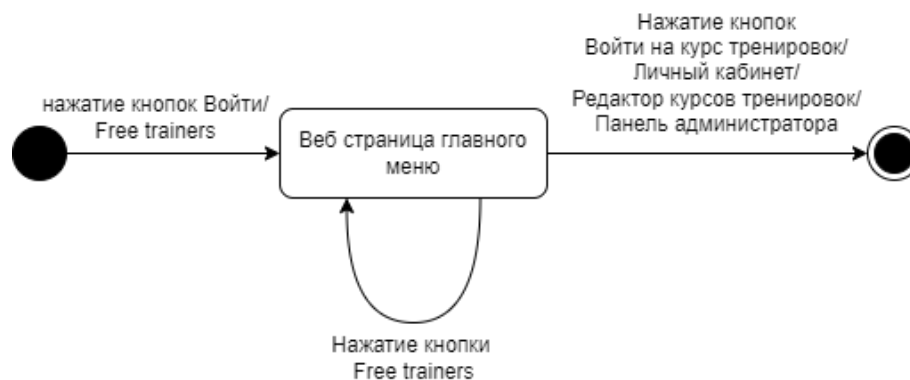
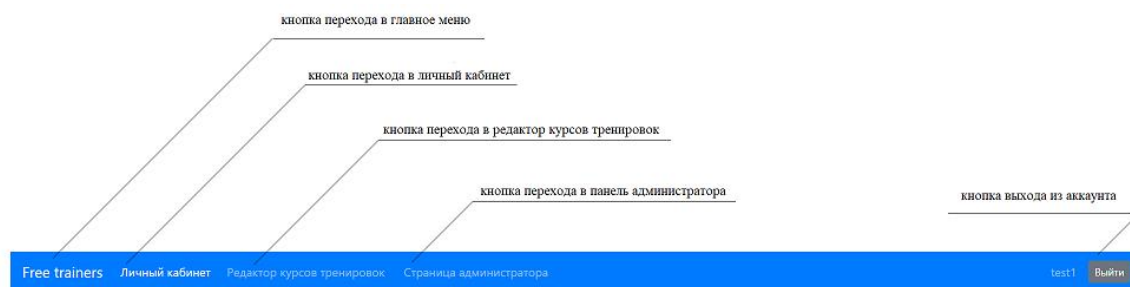


Рисунок 13 – Граф абстрактного диалога веб-страницы главного меню



Мои курсы тренировок

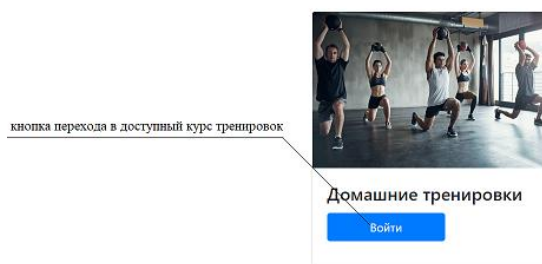


Рисунок 14 – Экранная форма веб-страницы главного меню

Веб-страница просмотра курсов тренировок содержит в себе информацию всего доступного курса тренировок. Экранная форма представлена на рисунке 15.

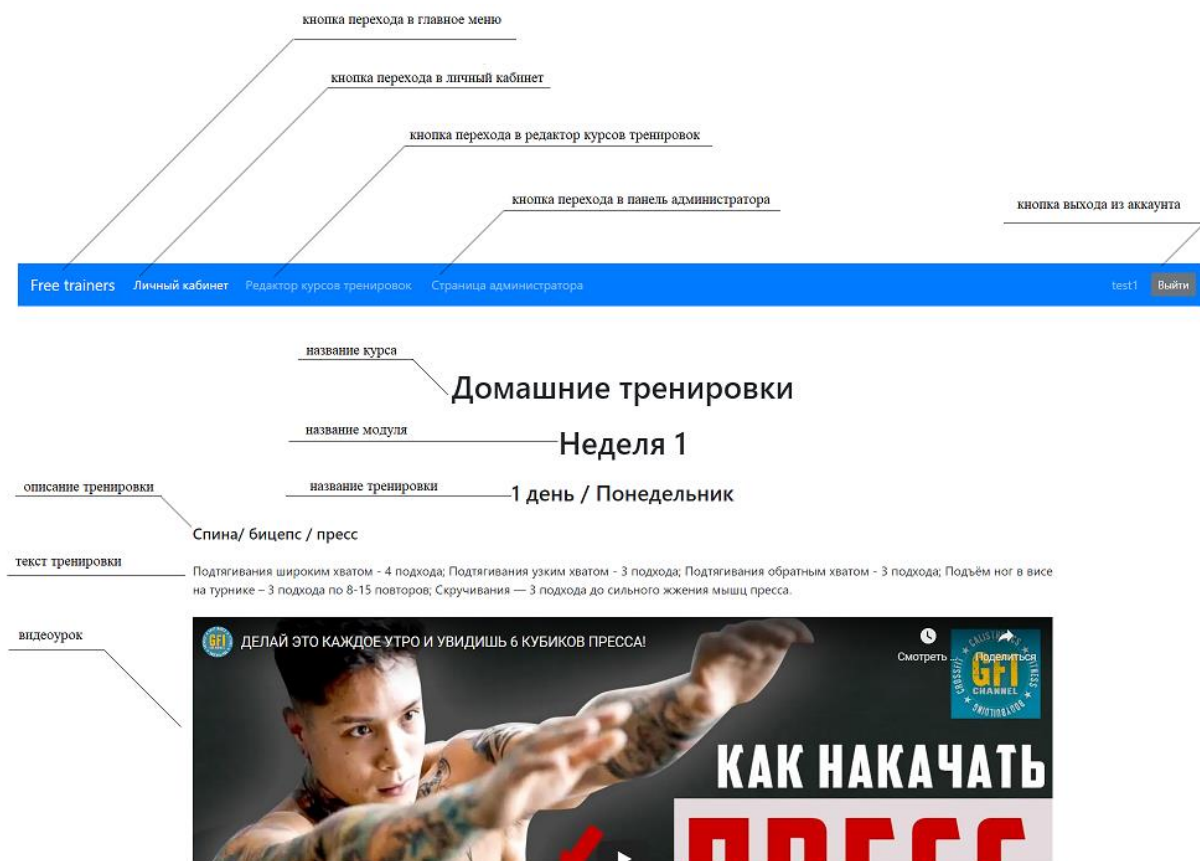


Рисунок 15 – Экранная форма веб-страницы просмотра курса тренировок

Далее на рисунке 16 отображена экранная форма страницы личного кабинета с теми же полями, что представлены на веб-странице регистрации нового аккаунта.

Рисунок 16 – Экранная форма веб-страницы личного кабинета

Далее рассмотрим систему редактора курсов тренировок. Веб страница создания курсов тренировок является основной в редакторе. Здесь расположены поля «Название курса тренировок» и «Тег курса», а также кнопки добавления нового курса тренировок в систему, удаления существующих курсов тренировок и таблица с информацией о созданных курсах тренировок. При нажатии на название существующего курса тренировок можно перейти на страницу редактирования этого курса тренировок и создания в нем модулей. Граф абстрактного диалога и экранная форма представлены на рисунках 17 и 18 соответственно.

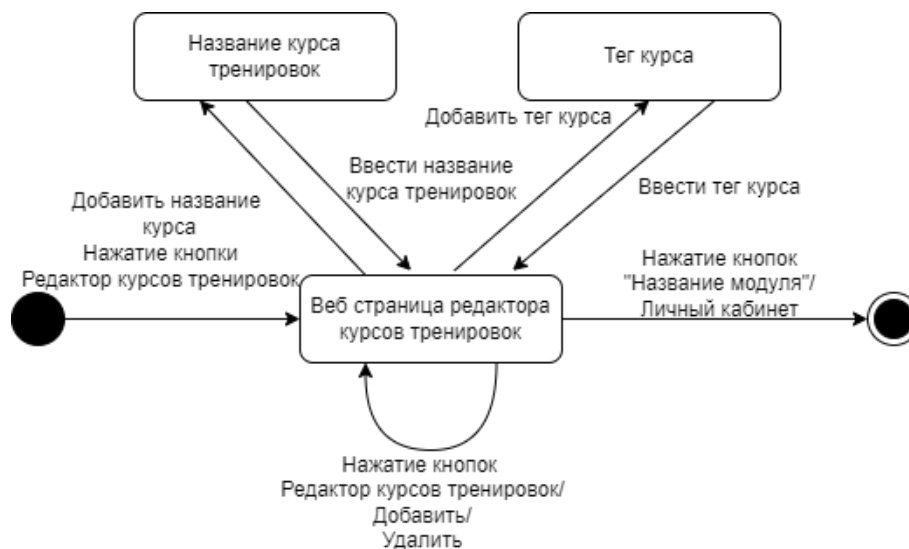


Рисунок 17 – Граф абстрактного диалога веб-страницы создания курсов тренировок

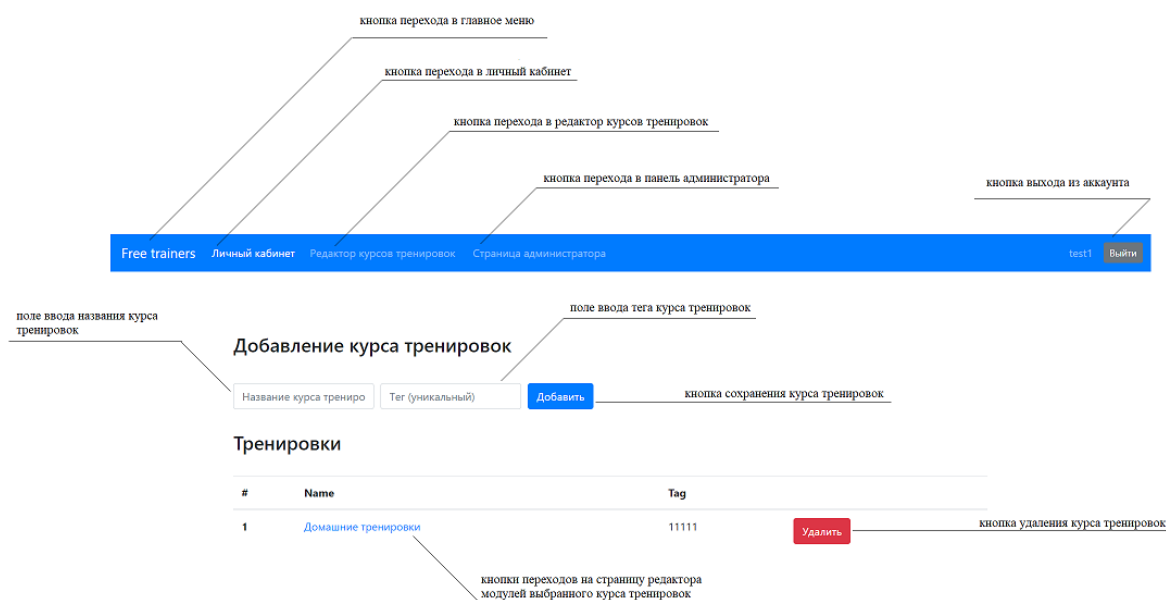


Рисунок 18 – Экранная форма веб-страницы создания курсов тренировок

Так как структура и функционал других веб-страниц редактора курсов тренировок очень похожи, далее на рисунках 19-22 будут рассмотрены лишь экранные формы.

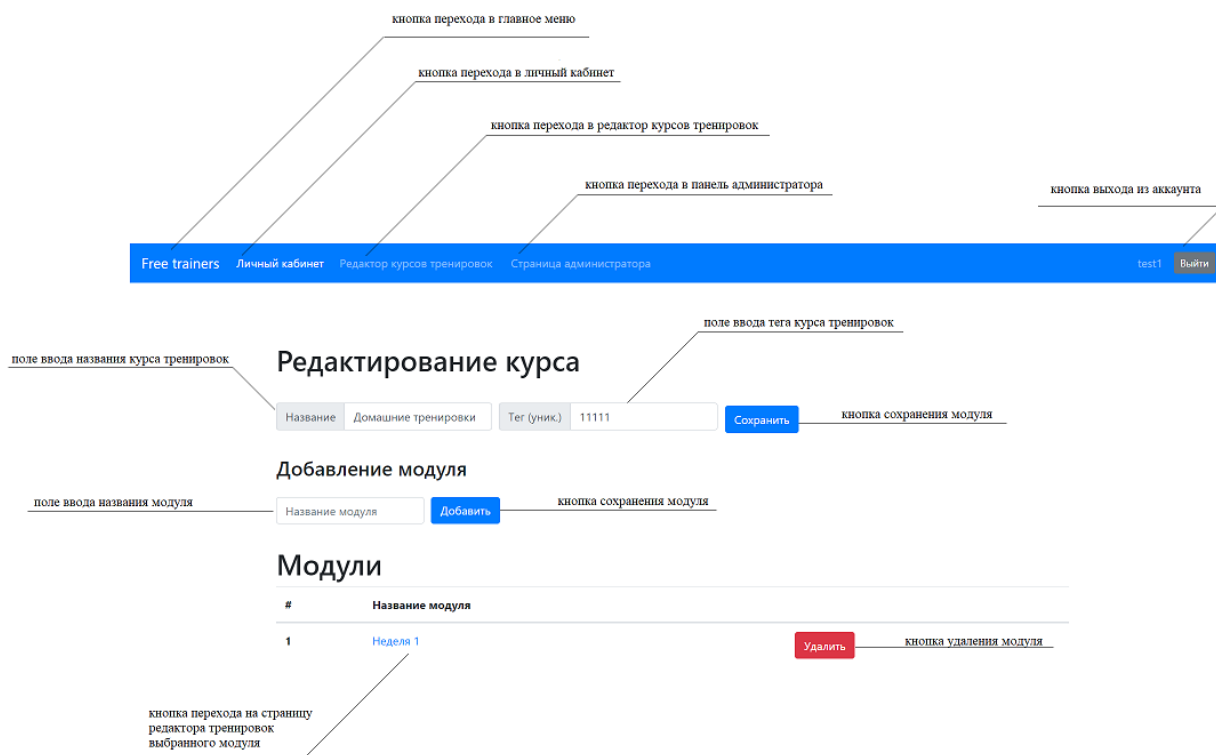


Рисунок 19 – Экранная форма веб-страницы создания модулей и редактирования курса

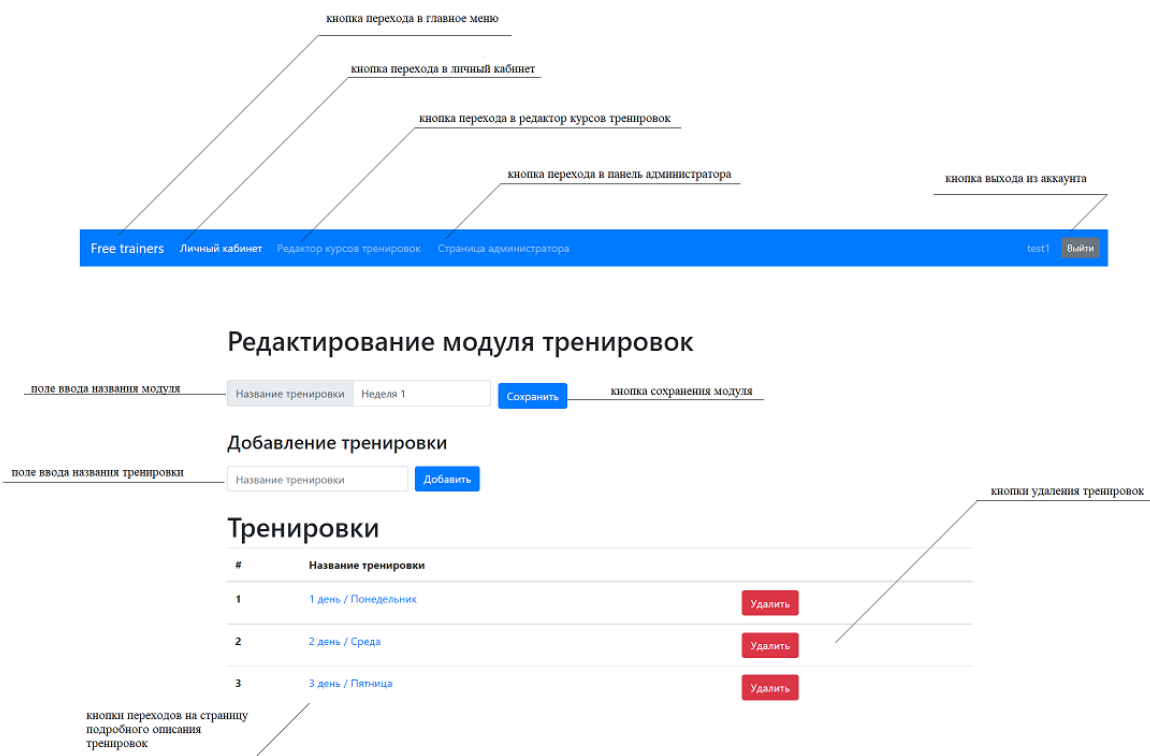


Рисунок 20 – Экранная форма веб-страницы создания тренировок и редактирования модуля

Веб-страница редактирования выбранной тренировки сформирована другим образом. Граф абстрактного диалога и экранная форма этой веб-страницы представлены на рисунках 20–21.

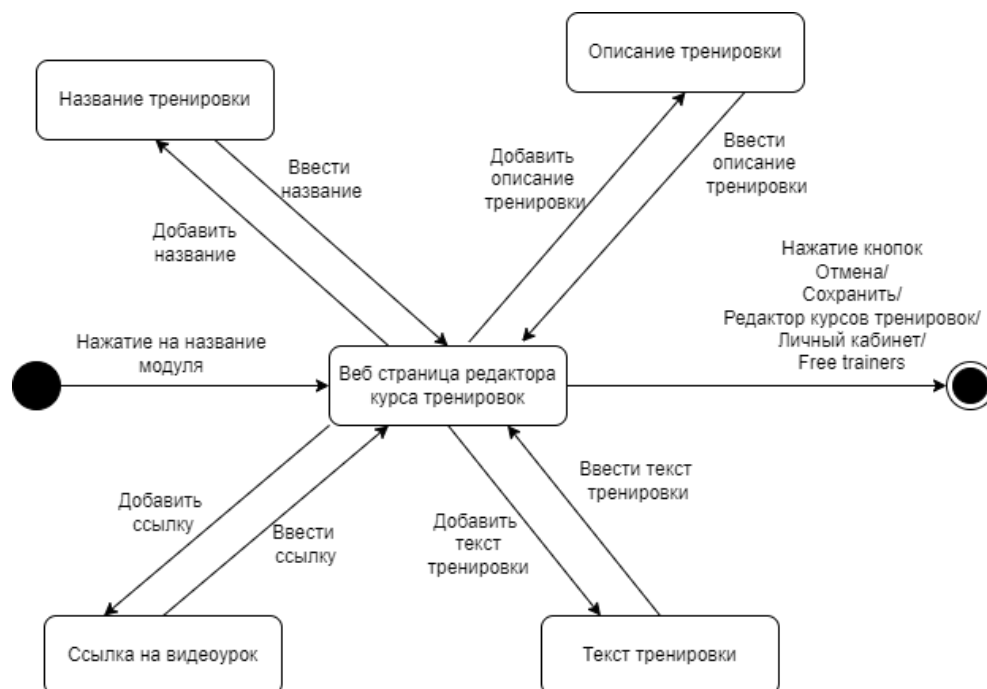


Рисунок 21– Граф абстрактного диалога веб-страницы редактирования тренировки

Рисунок 22– Экранная форма веб-страницы редактирования тренировки

После системы редактирования курсов будет рассмотрена панель администратора, которая включает в себя две функции: просмотр списка всех пользователей и создание нового пользователя. При переходе на страницу списка всех пользователей отображается таблица с записями о пользователях, а также кнопки изменения ролей и доступа на курсы в каждой записи. Все экранные формы представлены на рисунках 23-26.

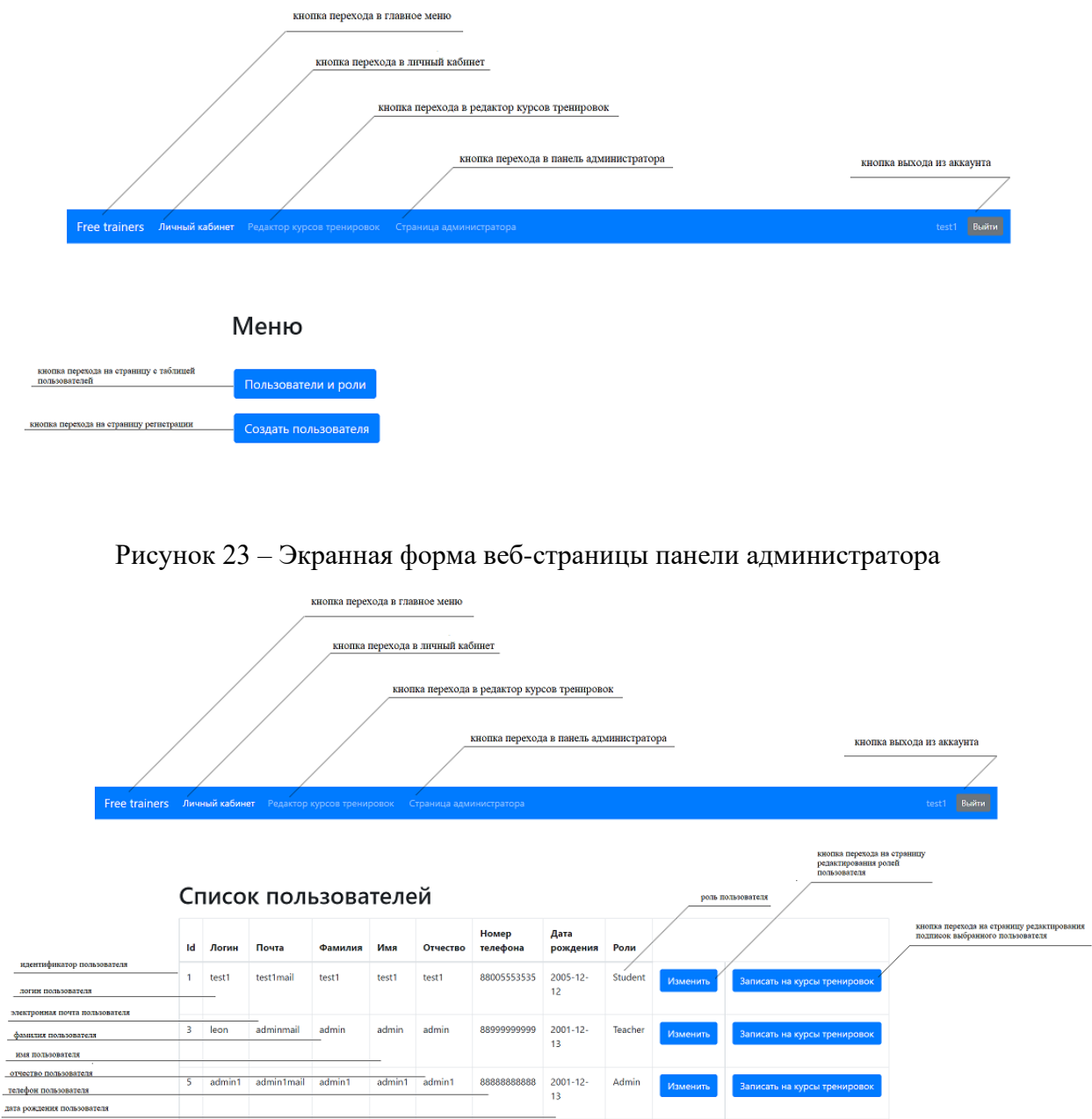


Рисунок 24 – Экранная форма веб-страницы списка всех пользователей

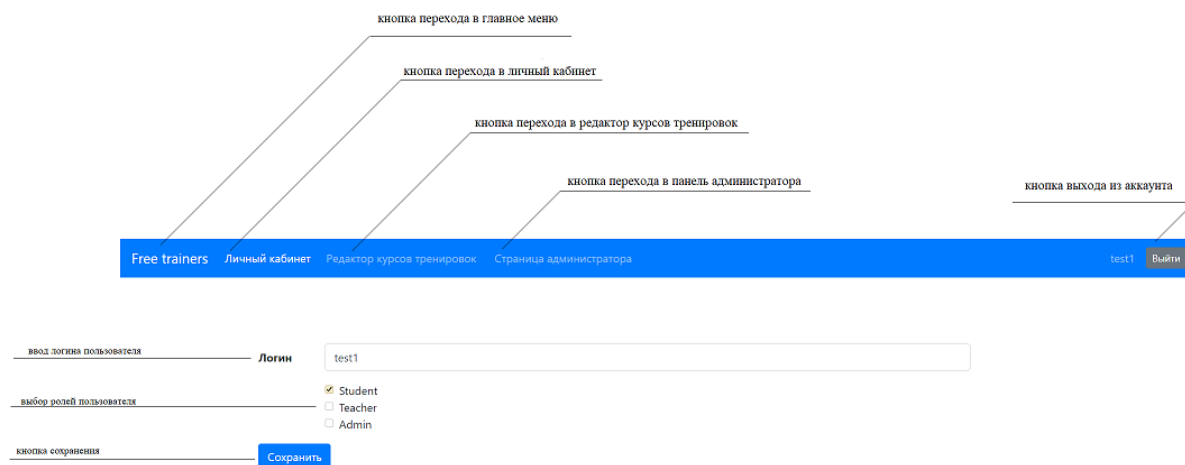


Рисунок 25 – Экранная форма веб-страницы изменения ролей и логина выбранного пользователя

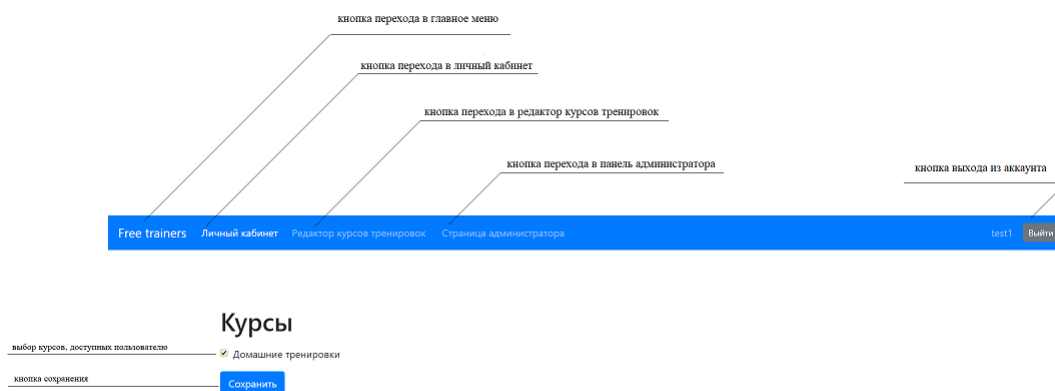


Рисунок 26 – Экранная форма веб-страницы записи на курсы выбранного пользователя

2.5 Проектирование базы данных

Разрабатываемая система работает с различными данными о пользователях и курсах тренировок. Для их хранения используется реляционная база данных и СУБД MySQL. Это обеспечивает быстрое чтение, запись, изменение и удаление данных. На рисунке 27 приведена даталогическая модель базы данных.

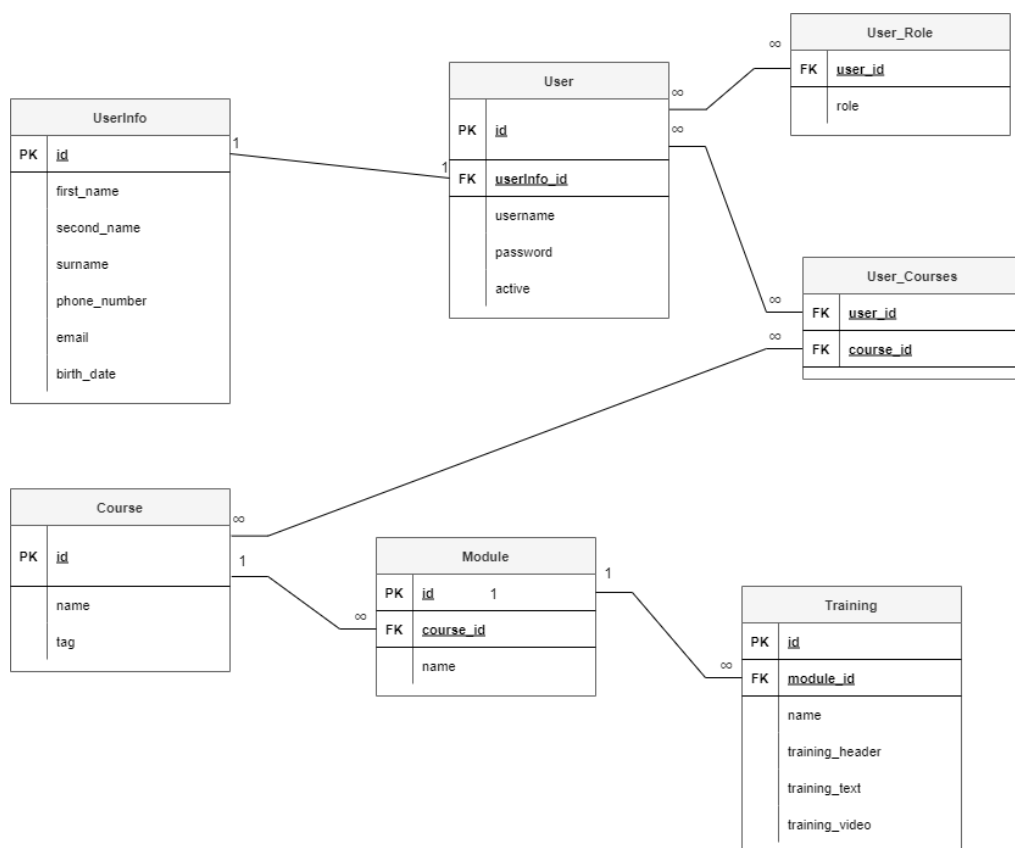


Рисунок 27 – Дatalogическая модель базы данных

Далее модель, приведенная на рисунке 27 рассматривается более подробно. В БД представлены таблицы: пользователи «User», информация о пользователях «UserInfo», роли «User_Role», записи за курсы «User_Courses», курсы «Course», модули «Module» и тренировки «Training». У всех таблиц есть свои поля, за исключением таблицы записей на курсы. Таблицы пользователей и информацией о них взаимодействуют с помощью первичных и внешних ключей со связью «Один к одному» [3]. Таблицы пользователей и ролей и пользователей и записями на курсы взаимодействуют со связями «Многие ко многим». Таблицы курсов и записей на курсы взаимодействуют со связью «Многие ко многим». Таблицы курсов и модулей взаимодействуют со связью «Один ко многим». Таблицы модулей и тренировок взаимодействуют со связями «Один ко многим».

В базе данных присутствуют следующие таблицы, записи которых содержат поля:

1) user – таблица пользователей:

- id – идентификатор пользователя, первичный ключ;
- username – логин пользователя;
- password – пароль пользователя;
- active – поле логического типа для контроля сессий пользователей;
- userInfo_id – идентификатор личной информации, внешний ключ.

2) UserInfo – таблица с личной информацией:

- id – идентификатор личной информации;
- first_name – имя пользователя;
- second_name – отчество пользователя;
- surname – фамилия пользователя;
- phone_number – мобильный телефон пользователя;
- email – электронная почта пользователя;
- birth_date – дата рождения пользователя.

3) User_Role – таблица ролей пользователей:

- user_id – идентификатор пользователя, внешний ключ;
- role – роль, которую имеет пользователь.

4) User_Courses – таблица записей пользователей на курсы:

- user_id – идентификатор пользователя, внешний ключ;
- course_id – идентификатор курса, внешний ключ.

5) Course – таблица курсов:

- id – идентификатор курса, первичный ключ;
- name – название курса;
- tag – тег курса.

6) Module – таблица модулей:

- id – идентификатор модуля, первичный ключ;
- course_id – идентификатор курса, внешний ключ;
- name – название модуля.

7) Training – таблица тренировок:

- id – идентификатор тренировок, первичный ключ;
- module_id – идентификатор модуля, внешний ключ;
- name – название тренировки;
- training_header – описание тренировки;
- training_text – текст тренировки;
- training_video – ссылка на видеоурок.

2.6 Проектирование классов

UML предлагает использовать три уровня диаграмм классов в зависимости от степени их детализации:

1) концептуальный уровень, на котором диаграммы классов, называемые в этом случае контекстными, демонстрируют связи между основными понятиями предметной области;

2) уровень спецификаций, на котором диаграммы классов отображают интерфейсы классов предметной области, т. е. связи объектов этих классов;

3) уровень реализации, на котором диаграммы классов непосредственно показывают поля и операции конкретных классов.

Каждую из перечисленных моделей используют на конкретном этапе разработки программного обеспечения: концептуальную модель - на этапе анализа; диаграммы классов уровня спецификации - на этапе проектирования; диаграммы классов уровня реализации - на этапе реализации.

Диаграмма классов представлена на рисунке 29.

Шаблон проектирования MVC говорит о необходимости существования моделей, представлений и контроллеров [4]. Представления являются HTML кодом, поэтому выносить их в диаграмму классов не корректно. Они являются лишь отображением данных, переданных с модели через методы контроллера.

3 Выбор стратегии тестирования и разработка тестов

Для тестирования программы было выбрано функциональное тестирование, так как большая часть взаимодействий с пользователем происходит посредством интерфейса и небольшого количества входных данных, и модульное тестирование, обеспечивающее автономное тестирование, для которого просто генерировать тестовые последовательности.

3.1 Функциональное тестирование

Данный вид тестирования производится вручную и часто называется тестированием с управлением по данным, т. е. при тестировании неизвестно, как работает программа, но важно, чтобы входные данные при этом корректно обрабатывались и приводили к ожидаемому результату. Данный вид тестирования поможет выявить ошибки, поскольку работа с некорректными данными может испортить пользовательский опыт использования приложением.

Воспользуемся методом причинно-следственной связи, т. к. методы эквивалентного разбиения и анализа граничных значений предназначены для программ, занимающихся, например, расчетами и вычислениями или отрисовкой объектов. Разработанная программа выполняет более простые функции. Разработанные методом причинно-следственной связи тесты приведены в таблице 7.

Таблица 7 - Тестирование методом анализа причинно-следственных связей

Тест	Ожидаемый результат	Полученный результат	Вывод
Ввод пустого значения названия курса тренировок при создании нового курса тренировок	Добавления курса тренировок не происходит	Добавления курса тренировок не происходит	Успешно
Ввод пустого значения названия модуля при создании нового модуля	Добавления модуля не происходит	Добавления модуля не происходит	Успешно

Ввод логина пользователя в окне регистрации при условии, что такой пользователь уже существует	Создания пользователя не происходит	Создания пользователя не происходит	Успешно
Ввод некорректных логина и пароля на странице входа в систему	Авторизации пользователя не происходит	Авторизации пользователя не происходит	Успешно
Ввод пустых значений электронной почты, логина, пароля	Регистрации нового пользователя не происходит	Регистрации нового пользователя не происходит	Успешно
Ввод пустого значения в поле для ссылки на видеоурок на странице редактирования тренировки	При просмотре курса окно с плеером не отображается	При просмотре курса окно с плеером не отображается	Успешно
Попытка пользователя, не являющегося администратором, воспользоваться функциями администратора	Возможность воспользоваться данными функциями отсутствует, выводится соответствующее сообщение	Возможность воспользоваться данными функциями отсутствует, выводится соответствующее сообщение	Успешно
Попытка зайти в курс тренировок, к которому у текущего пользователя нет доступа	Возможность получить доступ отсутствует, выводится страница с ошибкой	Возможность получить доступ отсутствует, выводится страница с ошибкой	Успешно

При тестировании программного продукта методом анализа причинно-следственных связей ошибок не выявлено.

3.2 Модульное тестирование

Данный вид тестирования предназначен для проверки корректности функционирования веб-приложения и удобен тем, что является автоматизированным. Тестирование заключается в имитации действия пользователя — кликов, переходов по ссылкам, добавлении и удалении элементов и других действий; в проверке взаимодействия компонентов друг с другом. Поддержка модульного тестирования (unit тестирования) встроена в SpringBoot изначально. Были разработаны двадцать тестов, проверяющих работоспособность основных компонентов.

В качестве примера тестирования работы веб-приложения были приведены следующие тесты: @Test

```
public void loginTest() throws Exception {    this.mockMvc.perform(get("/main"))
    .andDo(print())
    .andExpect(status().is3xxRedirection())
    .andExpect(redirectedUrl("http://localhost/login"));
}

@Test      public void correctLoginTest() throws Exception {
this.mockMvc.perform(formLogin().user("test").password("test"))
    .andDo(print())
    .andExpect(status().is3xxRedirection())
    .andExpect(redirectedUrl("/main"));
}

@Test  public void badCredentials() throws Exception {
    this.mockMvc.perform(post("/login").param("user", "Ivan"))
        .andDo(print())
        .andExpect(status().isForbidden());
}

@Test      public void mainPageTest() throws Exception {
this.mockMvc.perform(get("/main"))
    .andDo(print())
    .andExpect(authenticated())
    .andExpect(xpath("//div[@id='navbarSupportedContent']/div").string("test"));

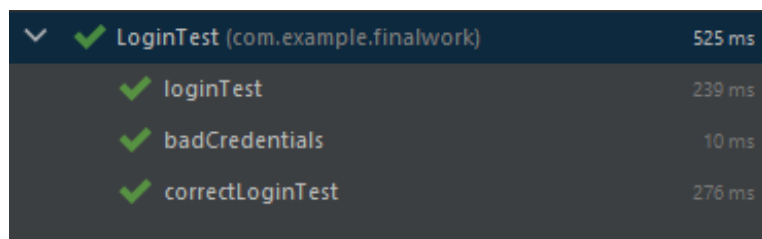
}
```

```

@Test
public void subscriptionsTest() throws Exception {
    this.mockMvc.perform(get("/main"))
        .andExpect(authenticated())
        .andExpect(xpath("//div[@id='subscriptionsList']/div").nodeCount(1));
}

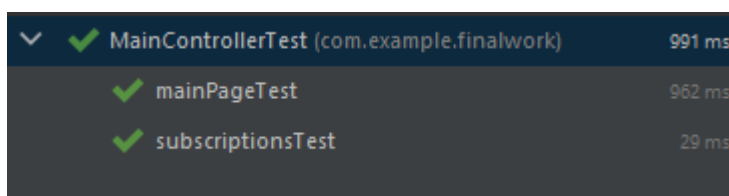
```

Результаты тестирования показаны на рисунках 29-30.



✓ LoginTest (com.example.finalwork)	525 ms
✓ loginTest	239 ms
✓ badCredentials	10 ms
✓ correctLoginTest	276 ms

Рисунок 29 – Результат UI-тестирования (1)



✓ MainControllerTest (com.example.finalwork)	991 ms
✓ mainPageTest	962 ms
✓ subscriptionsTest	29 ms

Рисунок 30 – Результат UI-тестирования (2)

3.3 Оценочное тестирование

После пройденных тестирований приложение может быть протестировано с помощью фокус-группы, каждый из участников которой оценит приложение по нескольким параметрам с помощью десятибалльной шкалы. Целью оценочного тестирования является тестирование программы на соответствие основным требованиям. В результате проектирования и реализации приложения были выявлены следующие критерии, по которым необходимо оценить программный продукт:

1. Удобство использования, так как интерфейс приложения должен быть интуитивно-понятен для пользователя.
2. Адаптивность. Приложение должно подстраивать размеры кнопок и их расположение в зависимости об окна браузера.
3. Работоспособность. Приложение не должно выводить пользователю сообщения о системных ошибках во время работы.
4. Правильность. Все действия пользователя должны быть корректно обработаны и получен ожидаемый результат.

Результаты опроса фокус-группы продемонстрированы в таблице 8.

Таблица 8 – Результаты опроса фокус-группы

Участник №	Удобство использования	Адаптивность	Работоспособность	Правильность
1	9	10	10	10
2	9	9	10	10
3	7	8	10	10
4	8	9	10	10
5	9	10	10	10
Средняя оценка:	8,4	9,2	10	10

По результатам тестирования, можно сделать вывод, что не всем участникам был интуитивно понятен интерфейс данного приложения. В дальнейших версиях планируется доработать его с целью улучшения и увеличения простоты понимания для будущих пользователей.

Заключение

В результате выполнения курсовой работы было спроектировано и реализовано веб-приложение, полностью удовлетворяющее всем требованиям технического задания. Разработанная система предоставляет возможности создания и редактирования курсов тренировок, потребления информации и администрирования всей системы.

Система может быть расширена в последующих версиях. В качестве усовершенствования рассматривается поиск курсов тренировок в таблице, отправка запроса записи на курс тренировок, добавление вкладки «Прогресс» в навигационной панели для отслеживания прогресса выполнения курса тренировок.

В ходе разработки программы был приобретён опыт проектирования программных продуктов с использованием фреймворка Spring. Также был приобретен опыт работы с базой данных, составлением технической документации. Кроме того, получены навыки тестирования и отладки приложения.

Список использованных источников

- 1 Иванова Г.С., Технология программирования. Изд-во МГТУ им. Н.Э. Баумана, 2002 - 241 с.
- 2 Иванова Г.С., Ничушкина Т.Н., Пугачёв Е.К., Самарёв Р.С., Фетисов М.В. Методические указания по выполнению курсовой работы по дисциплине «Технология разработки программных систем»: Электронное учебное издание. - МГТУ им. Н.Э. Баумана, 2019
- 3 Дейт К. Дж., Введение в системы баз данных / К. Дж. Дейт, Введение в системы баз данных, 8-е издание - Издательский дом "Вильямс", 2005, 1316с. - ISBN 58459-0788-8;
- 4 Документация по Spring [Электронный ресурс]. URL: <https://spring.io/guides>
(дата обращения: 29.12.2021)

ПРИЛОЖЕНИЕ А
Техническое задание
Листов 7



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И СИСТЕМЫ УПРАВЛЕНИЯ
КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ _____

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

ВЕБ-ПРИЛОЖЕНИЕ ПЛАТФОРМА ОНЛАЙН-ТРЕНИРОВОК

Техническое задание на курсовую работу по дисциплине Технология
разработки программных систем

Листов 7

Студент ИУ6-51Б
(Группа)

И.И.И. 01.03.2022
(Подпись, дата)

Л.С. Эммукаев
(И.О. Фамилия)

Руководитель курсовой работы

И.И.И. 10.03.2022
(Подпись, дата)

Н.О. Кузнецов
(И.О. Фамилия)

2021 г.

1 ВВЕДЕНИЕ

Настоящее техническое задание распространяется на разработку программной системы «Платформа онлайн-тренировок» [ПОТ], используемой для просмотра онлайн тренировок и предназначенной для обучения людей, заинтересованных в спортивной деятельности.

Данная разработка актуальна ввиду необходимости каждого человека заниматься спортом.

2 ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

Программный продукт ПОТ разрабатывается в соответствии с учебным планом кафедры по личной инициативе автора.

3 НАЗНАЧЕНИЕ РАЗРАБОТКИ

Разработка приложения ведется на основе учебного плана кафедры ИУ6 «Компьютерные системы и сети», утвержденного советом МГТУ им. Н. Э. Баумана.

4 ТРЕБОВАНИЯ К ПРОГРАММНОМУ ИЗДЕЛИЮ

4.1 Требования к функциональным характеристикам

4.1.1 Выполняемые функции

4.1.1.1 Для обучающегося:

- выбор курса из списка предложенных ему курсов;
- просмотр учебных материалов выбранного курса;
- вход в личный кабинет;
- редактирование личных данных;
- регистрация;
- авторизация;

4.1.1.2 Для преподавателя:

- добавление курсов тренировок, модулей и тренировок;
- удаление курсов тренировок, модулей и тренировок;
- редактирование курсов тренировок, модулей и тренировок;
- авторизация.

4.1.1.3 Для администратора:

- редактирование ролей пользователей и доступов на курс тренировок;
- просмотр личных данных пользователей;
- создание новых пользователей;
- авторизация.

4.1.2 Исходные данные:

- логин пользователя;
- пароль пользователя;
- исходная база данных;
- запрос пользователя;
- уровни доступа;
- верификация данных со стороны системы.

4.2 Требования к надежности

4.2.1 Предусмотреть контроль вводимой информации.

4.2.2 Предусмотреть блокировку некорректных действий пользователя.

4.3 Условия эксплуатации

4.3.1 Условия эксплуатации в соответствии с СанПиН 2.2.2/2.4.1340-03.

4.4 Требования к составу и параметрам технических средств

4.4.1 Программное обеспечение должно функционировать на IBM-совместимых персональных компьютерах.

4.4.2 Минимальная конфигурация технических средств:

4.4.2.1 ОС Windows:

а) Версия Windows 8, Windows 10, Windows 11

б) Наличие доступа в интернет.

4.4.2.2 ОС Linux:

а) Версия.....Ubuntu 14.04 (64-разрядная версия)

б) Наличие доступа в интернет.

4.4.2.3 Mac OS:

а) Версия.....OS X Yosemite 10.10

б) Наличие доступа в интернет.

4.5 Требования к информационной и программной совместимости

4.5.1 Программное обеспечение должно работать под управлением операционных систем семейств WIN32, WIN64, Linux (Ubuntu), Mac OS.

4.5.2 Результаты должны быть представлены в следующем формате: отображение страниц с данными.

5 ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

5.1 Разрабатываемые программные модули должны быть самодокументированы, т.е. тексты программ должны содержать все необходимые комментарии.

5.2 Разрабатываемое программное обеспечение должно включать справочную систему.

5.3 В состав сопровождающей документации должны входить:

5.3.1 Расчетно-пояснительная записка на 25-30 листах формата А4 (без приложений 5.3.2, 5.3.3 и 5.3.4).

5.3.2 Техническое задание (Приложение А).

5.3.3 Руководство пользователя (Приложение Б).

5.4 Графическая часть должна быть включена в расчетно-пояснительную записку в качестве иллюстраций:

5.4.1 Диаграмма вариантов использования.

5.4.2 Диаграммы деятельности.

5.4.3 Структурная схема.

5.4.4 Диаграмма потоков данных.

5.4.5 Граф состояний интерфейса.

5.4.6 Графы абстрактного диалога.

5.4.7 Схема структур базы данных.

5.4.8 Диаграмма классов уровня реализации.

5.4.9 Таблицы тестов.

6 СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Этап	Содержание этапа	Сроки и объем	Представляемые результаты	
			Спецификации и программный продукт	Документы
1.	Выбор темы, составление задания, решение организационных вопросов	1..2 недели (10 %)	-	Заполненный бланк задания на курсовую работу – вывешивается на сайт кафедры для получения утверждающей подписи заведующего кафедрой

Этап	Содержание этапа	Сроки и объем	Представляемые результаты	
			Спецификация и программный продукт	Документы
2.	Анализ предметной области, разработка ТЗ. Исследование методов решения, выбор основных проектных решений	3..4 недели	Результаты декомпозиции предметной области. Эскизный проект: интерфейс, часть программы (выбранные готовые решения).	Фрагмент расчетно-пояснительной записки с обоснованием выбора средств и подходов к разработке
3.	Сдача ТЗ	4 неделя (25 %)	-	Техническое задание – утверждается руководителем
4.	Проектирование и реализация основных компонентов – ядра программы	5..7 недели	Технический проект основной части: структура программы, алгоритмы программ, описания структур данных, диаграмма классов. Программный продукт, реализующий основные функции (демонстрируется руководителю)	Фрагмент расчетно-пояснительной записки с обоснованием разработанных спецификаций Тексты части программного продукта, реализующего основные функции.
5.	Сдача прототипа программного продукта	7 неделя (50 %)	Прототип программного продукта – демонстрируется руководителю	

Этап	Содержание этапа	Сроки и объем	Представляемые результаты	
			Спецификации и программный продукт	Документы
6.	Разработка компонентов, обеспечивающих функциональную полноту	8..10	Рабочий проект программы. Готовая программа	Черновик расчетно-пояснительной записки. Тексты программного продукта.
7.	Сдача программного продукта	11 неделя (75 %)	Готовая программа – оценивается руководителем в баллах	-
8.	Тестирование программы и подготовка документации	12..14	Тесты и результаты тестирования.	РПЗ и Руководство пользователя.
9.	Оформление и сдача документации	14 неделя (90 %)	–	Расчетно-пояснительная записка и Руководство пользователя – проверяются и подписываются руководителем
10.	Защита курсовой работы	15..16 недели (100%)	–	Доклад (3-5 минут). Защита курсовой работы. Подписанная документация – вывешивается на сайт кафедры

7 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

7.1 Порядок контроля

Контроль выполнения осуществляется руководителем еженедельно.

7.2 Порядок защиты

Защита осуществляется комиссии преподавателей кафедры.

7.3 Срок защиты

Срок защиты: 15-16 недели.

8 ПРИМЕЧАНИЕ

В процессе выполнения работы возможно уточнение отдельных требований технического задания по взаимному согласованию руководителя и исполнителя.

ПРИЛОЖЕНИЕ Б

Руководство пользователя

Листов 8



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И СИСТЕМЫ УПРАВЛЕНИЯ
КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ _____
НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

ВЕБ-ПРИЛОЖЕНИЕ ПЛАТФОРМА ОНЛАЙН- ТРЕНИРОВОК

Руководство пользователя

Листов 8

Студент ИУ6-51Б
(Группа)

01.03.2022
(Подпись, дата)

Л.С. Эльмухамед
(И.О. Фамилия)

Руководитель курсовой работы

01.03.2022
(Подпись, дата)

Н.О. Кузнецов
(И.О. Фамилия)

2021 г.

Содержание

1	Общие сведения о программном продукте.....	54
2	Начало работы с приложением.....	54
3	Инструкция по работе с приложением.....	55
3.1	Просмотр курсов тренировок.....	55
3.2	Изменение личных данных.....	56
3.3	Редактирование курсов тренировок.....	56
3.4	Администрирование системы.....	59
4	Общие рекомендации.....	60

1 Общие сведения о программном продукте

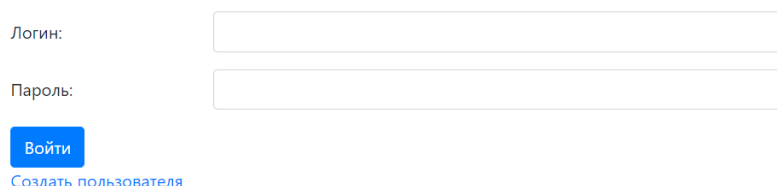
Задача программного продукта заключается в предоставлении пользователю возможности просмотра курсов тренировок и их создания. Главная задача – предоставить пользователю возможность потребления информации и корректное создание курсов тренировок. Проектируемый интерфейс должен позволять корректный вывод и ввод данных, и иметь удобный, понятный (легко читаемый) вид. Данные об аккаунте должны быть редактируемыми.

В приложении реализована система авторизации, позволяющая создавать новых пользователей, входить в систему существующим пользователям и редактировать свои личные данные. Администратор системы может регистрировать новых пользователей и удалять имеющихся.

Работа с веб-приложением «Платформа онлайн-тренировок» может осуществляться с помощью персонального компьютера, имеющего доступа к сети Интернет.

2 Начало работы с приложением

Для начала работы с настоящим приложением достаточно иметь стабильное интернет-соединение и перейти на одну из веб-страниц, реализованных в данном приложении. После запуска веб-приложение перенаправит на веб-страницу входа в систему. Пользователь может переходить на вкладку с регистрацией, используя кнопку, расположенную под кнопкой «Войти». Страницы авторизации и регистрации представлены на рисунках Б.1 и Б.2.



Логин:

Пароль:

[Создать пользователя](#)

Рисунок Б.1 – Страница входа в систему

Регистрация


Имя пользователя:	<input type="text"/>
Пароль:	<input type="password"/>
Электронная почта:	<input type="text"/>
Номер телефона:	<input type="text"/>
Имя:	<input type="text"/>
Фамилия:	<input type="text"/>
Отчество:	<input type="text"/>
Дата рождения:	<input type="text" value="ДД.ММ.ГГГГ"/> 
<input type="button" value="Зарегистрироваться"/>	

Рисунок Б.2 – Страница регистрации

После успешного входа в систему пользователю будет показана страница главного меню с навигационной панелью, которая представлена на рисунке Б.3.



Рисунок Б.3 – Навигационная панель

На навигационной панели присутствуют кнопки возврата на главную страницу «Free trainers», кнопки входа в личный кабинет, редактор курсов тренировок и страницу администратора. Для отображения последних двух кнопок необходимо обладать правами тренера и администратора соответственно.

3 Инструкция по работе с приложением

3.1 Просмотр курсов тренировок

Для просмотра выбранного курса пользователю необходимо нажать на кнопку «Войти» на карточке необходимого курса. Выйти из курса можно путем нажатия на кнопки навигационной панели. Главное меню изображено на рисунке Б.4.

Мои курсы тренировок

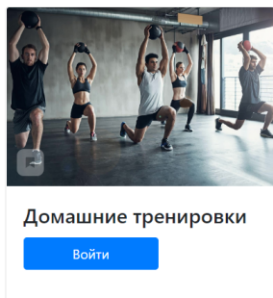


Рисунок Б.4 – Страница главного меню

3.2 Изменение личных данных

Для редактирования личных данных необходимо перейти в личный кабинет и изменить нужные пользователю поля. Личный кабинет представлен на рисунке Б.5.

Скриншот веб-страницы 'Профиль пользователя'. Вверху навигационная панель с 'Личный кабинет' и 'Редактор курсов тренировок'. В центре форма с полями: Почта (elnukaev.leon@yandex.ru), Телефон (88999999999), Имя (Lom-ALI), Фамилия (Elnukaev), Отчество (Said-Muradovich), Дата рождения (13.12.2001). Внизу синяя кнопка 'Сохранить'.

Рисунок Б.5 – Страница личного кабинета

3.3 Редактирование курсов тренировок

Для использования редактора курсов тренировок необходимо иметь права тренера. После нажатия на кнопку редактора курсов тренировок пользователь попадает на страницу, представленную на рисунке Б.6.

Free trainers Личный кабинет Редактор курсов тренировок leon Выйти

Добавление курса тренировок

Тренировки

#	Name	Tag	
1	Домашние тренировки	11111	<input type="button" value="Удалить"/>

Рисунок Б.6 – Страница редактора курсов тренировок

На этой странице можно создать новый курс путем ввода необходимых данных и нажать кнопку «Добавить». Можно удалить выбранный курс, нажав в строке таблицы кнопку «Удалить» напротив этого курса. Также можно перейти в редактор модулей, нажав на название курса. Страница редактора модулей представлена на рисунке Б.7.

Free trainers Личный кабинет Редактор курсов тренировок leon Выйти

Редактирование курса

Домашние тренировки
 11111

Добавление модуля

Модули

#	Название модуля	
1	Неделя 1	<input type="button" value="Удалить"/>

Рисунок Б.7 – Страница редактора модулей выбранного курса

На представленной выше странице можно редактировать выбранный ранее курс путем изменения данных в форме и нажатия кнопки «Сохранить», создать новые модули путем ввода необходимых данных и нажать кнопку «Добавить», удалить уже созданные модули путем нажатия кнопки «Удалить» напротив названия модуля. Также можно перейти в редактор тренировок нажатием на название модуля. Страница редактора тренировок представлена на рисунке Б.8.

Free trainers Личный кабинет Редактор курсов тренировок leon Выйти

Редактирование модуля тренировок

Добавление тренировки

Тренировки

#	Название тренировки	
1	1 день / Понедельник	<input type="button" value="Удалить"/>
2	2 день / Среда	<input type="button" value="Удалить"/>
3	3 день / Пятница	<input type="button" value="Удалить"/>

Рисунок Б.8 – Страница редактора тренировок выбранного модуля

На представленной выше странице можно редактировать выбранный ранее модуль путем изменения данных в форме и нажатия кнопки «Сохранить», создать новые тренировки путем ввода необходимых данных и нажать кнопку «Добавить», удалить уже созданные тренировки путем нажатия кнопки «Удалить» напротив названия тренировки. Также можно перейти в подробный редактор выбранной тренировки нажатием на название тренировки. Страница подробного редактирования тренировок представлена на рисунке Б.9.

Free trainers Личный кабинет Редактор курсов тренировок leon Выйти

Информация о тренировке

Название тренировки

Описание тренировки

Текст тренировки

Рисунок Б.9 – Страница подробного редактирования выбранной тренировки

На странице редактирования выбранной тренировки можно изменить название тренировки, добавить описание, текст тренировки и ссылку на видеоурок. После чего необходимо нажать кнопку «Сохранить». При нажатии кнопки «Отменить» пользователь перейдет на предыдущую страницу. Также можно посмотреть, корректно ли работает

ссылка на видеоурок, зайдя на страницу подробного редактирования и опустившись вниз страницы.

3.4 Администрирование системы

Для администрирования системы была создана панель администратора, попасть в которую можно только в том случае, если пользователь имеет права администратора. Меню панели администратора представлено на рисунке Б.10.

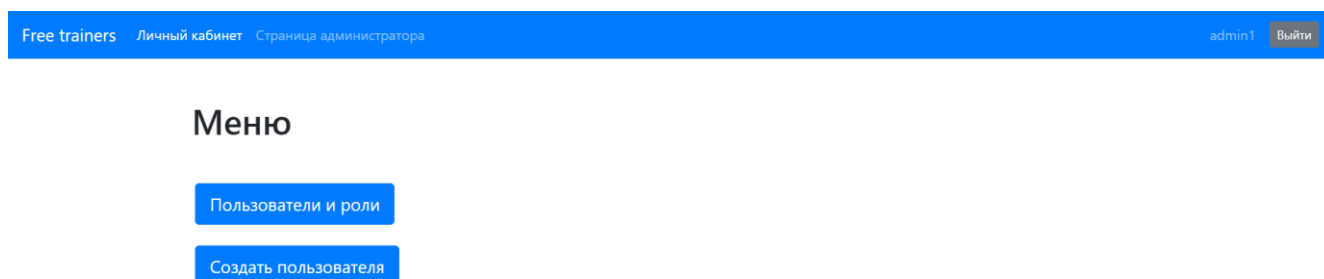


Рисунок Б.10 – Страница панели администратора

На странице имеются кнопки для создания нового пользователя и отображения пользователей и их данных. Создание пользователя было рассмотрено в пункте 2. Страница со списком всех пользователей представлена на рисунке Б.11.

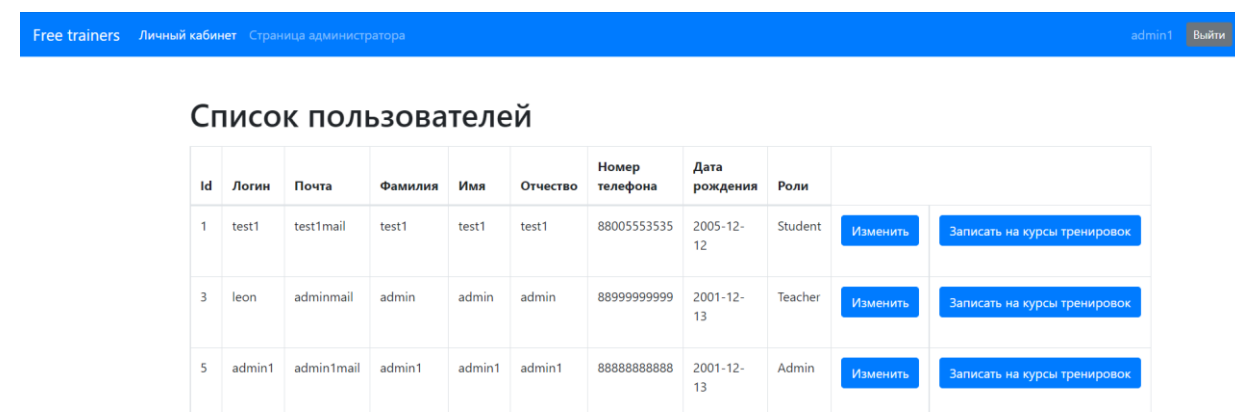


Рисунок Б.11 – Страница со списком всех пользователей

На данной странице отображаются личные данные всех пользователей, а также их роли. При нажатии на кнопку «Изменить» пользователь попадает на страницу редактирования ролей и логина выбранного аккаунта. При нажатии на кнопку «Записать на курсы тренировок» пользователь попадает на страницу редактирования доступа данного аккаунта к курсам. Страницы изменения ролей и доступа на курсы представлены на рисунках Б.12 и Б.13.

Free trainers Личный кабинет Страница администратора admin1 Выйти

Логин test1

☒ Student
☐ Teacher
☐ Admin

Сохранить

Рисунок Б.12 – Страница изменения аккаунта администратором

Free trainers Личный кабинет Страница администратора admin1 Выйти

Курсы

☒ Домашние тренировки

Сохранить

Рисунок Б.13 – Страница изменения доступа к курсам администратором

4. Общие рекомендации

В заключении следует рассмотреть возможные ошибки приложения и общие рекомендации по использованию приложения. Поскольку приложение требует работы через Интернет, оно не будет функционировать без подключения к сети. В этом случае при попытке входа или работы с данными будут получены ошибки; данные о курсах и пользователях не будут отображаться. В таком случае следует проверить сетевое подключение на своем устройстве. При регистрации рекомендуется не указывать в качестве пароля личные данные и не использовать простые пароли. В случае утери пароля пользователь может обратиться к администрации, которая поможет восстановить утерянный пароль.

ПРИЛОЖЕНИЕ В

Листинг фрагмента программного продукта

Листов 4



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И СИСТЕМЫ УПРАВЛЕНИЯ
КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ _____
НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

ВЕБ-ПРИЛОЖЕНИЕ ПЛАТФОРМА ОНЛАЙН-ТРЕНИРОВОК

Листинг фрагмента программного продукта

Листов 4

Студент ИУ6-51Б
(Группа)

Ильин 01.03.2022
(Подпись, дата)

Л.С. Эльмухамедов
(И.О. Фамилия)

Руководитель курсовой работы

Кузнецов 01.01.2022
(Подпись, дата)

Н.О. Кузнецов
(И.О. Фамилия)

2021 г.

В качестве примера фрагмента программы приведен код файла userService, в котором реализуются основные алгоритмы для контроллера UserController, отвечающего за создание пользователя, редактирование его личных данных, записей на курсы, изменения ролей и логина и т. д.

```
package com.example.finalwork.service; //Необходимые импорты

import com.example.finalwork.actions.CourseAction;
import com.example.finalwork.actions.UserAction;
import com.example.finalwork.model.Course;
import com.example.finalwork.model.Role;
import com.example.finalwork.model.User;
import com.example.finalwork.model.UserInfo;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.core.userdetails.UsernameNotFoundException;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.stereotype.Service;

import java.util.*;
import java.util.function.Function;
import java.util.stream.Collectors;

@Service
public class UserService implements UserDetailsService {
    private final UserAction userAction; //Используемые инжекты
    private final CourseAction courseAction;
    private final PasswordEncoder passwordEncoder;

    public UserService(UserAction userAction, CourseAction courseAction,
PasswordEncoder passwordEncoder) {
        this.userAction = userAction;
        this.courseAction = courseAction;
        this.passwordEncoder = passwordEncoder;
    }
}
```

```

@Override //Поиск пользователя по логину
public UserDetails loadUserByUsername(String username) throws
UsernameNotFoundException {
    return userAction.findByUsername(username);
}

public List<User> findAllUsers() {
    return userAction.findAll();
} //список со всеми пользователями

public List<Course> findAllCourses() {
    return courseAction.findAll();
} //список со всеми курсами

public boolean addUser(User user, UserInfo userInfo){ //создание пользователя и
//проверка наличия такого логина в БД
    User userFromDb = userAction.findByUsername(user.getUsername());
    if (userFromDb != null){
        return false;
    }
    else {
        user.setActive(true);
        user.setRoles(Collections.singleton(Role.ROLE_ADMIN));
        user.setPassword(passwordEncoder.encode(user.getPassword()));
        user.setUserInfo(userInfo);
        userAction.save(user);
        return true;
    }
}

public void saveUser(User user, String username, Map<String, String> form) {
//изменение ролей пользователя
    user.setUsername(username);

```



```

Set<String> roles = Arrays.stream(Role.values())
    .map(Role::name)
    .collect(Collectors.toSet());
user.getRoles().clear();
for (String key : form.keySet()) {
    if (roles.contains(key)) {
        user.getRoles().add(Role.valueOf(key));
    }
}
userAction.save(user);
}

```

```

public void updateProfile(User user,          //Обновление личных данных в личном
кабинете

```

```

        String email,
        String firstName,
        String secondName,
        String surname,
        String phoneNumber,
        String birthDate) {

```

```

String userEmail = user.getUserInfo().getEmail();
boolean isEmailChanged = (email != null && !email.equals(userEmail)) ||
    (userEmail != null && !userEmail.equals(email));

```

```

if (isEmailChanged){
    user.getUserInfo().setEmail(email);
}

```

```

user.getUserInfo().setFirst_name(firstName);
user.getUserInfo().setSecond_name(secondName);
user.getUserInfo().setSurname(surname);
user.getUserInfo().setPhone_number(phoneNumber);

```

```

        user.getUserInfo().setBirth_date(birthDate);
        userAction.save(user);
    }

```

public void saveSubscriptions(User user, Map<String, Course> form) { //организация доступа на курсы пользователю

```

        Map<String, Course> courseMap = findAllCourses().
            stream().
            collect(Collectors.toMap(Course::getTag, Function.identity()));
        Set<Course> courseSet = new HashSet<>();
        user.getSubscriptions().clear();
        for (String key : form.keySet()) {
            if (courseMap.containsKey(key)) {
                courseSet.add(courseMap.get(key));
            }
        }
    }

```

```

        user.setSubscriptions(courseSet);
        userAction.save(user);
    }

```

```

}

```

Весь проект расположен в репозитории по адресу:
https://github.com/1dkrly/TRPS_Finalwork