

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт космических и информационных технологий  
Кафедра систем искусственного интеллекта

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ Г. М. Цибульский  
« \_\_\_\_\_ » \_\_\_\_\_ 2018 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.02 «Информационные системы и технологии»

Разработка web-платформы для учебного курса по допечатной подготовке

Руководитель	_____	ст. преп. каф. СИИ	М. А. Аникьева
	подпись, дата		С. А. Приловский
Выпускник	_____		
	подпись, дата		
Нормоконтролер	_____		М. А. Аникьева
	подпись, дата		

Красноярск 2018

## СОДЕРЖАНИЕ

Введение.....	5
Глава 1. Обзор теоретических данных для понимания контекста разработки платформы .....	7
1.1 Основные понятия, определения и принципы организации электронного образовательного процесса .....	7
1.2 Обзор существующих разработок.....	13
1.3 Общие сведения и практическая значимость разрабатываемого продукта .....	16
1.4 Выводы по главе 1 .....	17
Глава 2. Проектирование платформы .....	18
2.1 Основные требования.....	18
2.2 Классификация пользователей .....	19
2.3 Схема работы и взаимодействие компонент разрабатываемой платформы .....	19
2.4 Обзор прецедентов.....	21
2.5 Описание вариантов использования .....	21
2.6 Структуризация вариантов использования .....	25
2.7 Диаграмма деятельности .....	27
2.8 Диаграмма классов.....	28
2.9 Диаграмма последовательностей .....	30
2.10 Диаграммы компонентов и развертывания.....	31
2.11 Концептуальная схема сайта.....	33
2.12 Средства разработки .....	34
2.13 Выводы по главе 2 .....	36
Глава 3. Разработка платформы.....	38
3.1 Разработка платформы .....	38

3.1.1 Установка CMS и создание сайта.....	38
3.1.2 Настройка сайта .....	39
3.1.3 Настройка безопасности сайта .....	41
3.1.4 Создание страниц сайта .....	43
3.1.5 Настройка дизайна сайта.....	44
3.2 Разработка редактора БД.....	45
3.3 Выводы по главе 3 .....	49
Заключение .....	50
Список сокращений .....	52
Список используемых источников.....	53
Приложение А Техническое задание .....	55
Приложение Б Исходный код редактора БД.....	61
Приложение В Презентационный материал .....	79

## ВВЕДЕНИЕ

В последнее время большое число исследований направлено на автоматизацию и интеллектуализацию различных сфер деятельности. Однако в автоматизации обучения на сегодняшний день существует много нерешенных проблем. Такое положение дел связано с тем, что обучение представляет собой сложный интеллектуальный процесс, плохо поддающийся формализации.

Заметной тенденцией в автоматизации обучения является разработка методов, методик и средств, реализующих индивидуальный подход к процессу обучения. К актуальным проблемам в этой области можно отнести составление программ обучения, проверку и оценку знаний, индивидуальный подбор учебного материала и др. [18].

Для решения данной проблемы в бакалаврской работе разработана платформа учебно-методического обеспечения учебного курса по допечатной подготовке.

Темой выпускной квалификационной работы является разработка веб-платформы для учебного курса по допечатной подготовке.

Целью данного проекта является разработка автоматизированной информационной системы для автоматизации процесса обучения. Данная система выполняет функции:

- описания предметной области учебной дисциплины;
- формирования на основе предметной области тематических разделов учебного плана;
- формирование на основе предметной области индивидуального графика обучения, анализа успеваемости студентов;
- визуализации понятий предметной области с помощью графовой модели представления для отражения содержания каждого модуля в учебной программе, его объем и логическую сложность.

Для достижения поставленной цели в рамках работы решаются следующие задачи: обзор теоретических данных для понимания контекста разработки платформы, проектирование платформы, разработка веб-платформы.

## **Глава 1. Обзор теоретических данных для понимания контекста разработки платформы**

### **1.1 Основные понятия, определения и принципы организации электронного образовательного процесса**

В настоящее время продолжается стремительная информатизация сферы образования, в частности, разработка электронных средств обучения. В связи с этим перед учебными заведениями стоит задача создания электронных образовательных ресурсов (ЭОР): электронных курсов лекций, электронных учебных пособий, контрольно-измерительных материалов с постепенным переходом к созданию полных электронных модулей и электронных учебно-методических комплексов по дисциплинам, поддерживающих проведение всех видов занятий для получения соответствующих специальностей. В связи с этим одним из этапов развития любого современного вуза является формирование такой системы образования, которая способна дать специалистов, подготовленных для осуществления инновационных преобразований в экономике и социальной сфере. При такой постановке вопроса главной целью вузов становится формирование у студентов способности эффективно применять знания и умения на практике при создании новой конкурентоспособной, наукоемкой продукции.

Современные тенденции развития системы высшего образования предъявляют новые требования к уровню подготовки специалистов различных направлений. Меняются цели и задачи, а вместе с ними и технологии, используемые в процессе обучения. В настоящий момент «становится востребованной не модель узкопрофессиональной подготовки выпускника вуза, а модель интегрального типа, в которой цели, содержание и результаты обучения формулируются в комплексном виде с учетом изменений в профессиональной деятельности и направлены на формирование у студента широкой социально-профессиональной компетентности». В этой связи особое значение приобретает формирование основополагающих знаний и опыта будущего

квалифицированного работника — информационной компетентности как интегративного качества личности, системного образования знаний, умений и способности субъекта в сфере информационно-коммуникационных технологий (ИКТ) и опыта их использования в меняющихся условиях на основе новых технологических средств [5].

При реализации образовательных программ с применением учебно-методического обеспечения в вузе должны быть созданы условия для эффективного функционирования информационно-образовательной среды, включающей в себя электронные ресурсы учебного назначения, совокупность информационных и телекоммуникационных технологий, компьютерных и технологических средств обучения.

Внедрение систем электронного обучения в вузе требует использование программно-технических и психолого-педагогических средств и условий, обеспечивающих эффективное сопровождение и функционирование образовательного процесса, в связи с чем происходит возникновение новых требований к его учебно-методическому обеспечению. Из этого следует, что изменение формы предоставления учебного материала должно определять состав и структуру нового поколения учебно-методического обеспечения, где главная роль отводится образовательным электронным изданиям и средствам [6]. Кроме этого, с учетом специфики времени, сегодня современное учебно-методическое обеспечение должно позволять реализовывать концептуальные подходы федеральных государственных образовательных стандартов высшего профессионального образования (ФГОС ВПО).

Электронное образование предоставляет ряд возможностей в сфере образования.

1. Электронное обучение позволяет включать в образовательный процесс те категории населения, которые в силу возрастных, физических, социальных, национальных, региональных, географических особенностей не имели возможности воспользоваться традиционными образовательными услугами.

2. Электронное обучение приводит к повышению экономической эффективности образования. С одной стороны, оно позволяет снизить часть расходов традиционных учебных заведений, оптимизировать и улучшить управление образовательным процессом, повысить уровень оплаты труда преподавателей и специалистов, создающих учебные ресурсы. С другой стороны, для потребителя, уменьшаются ограничения финансового характера, как вследствие снижения затрат учебной организации, так и вследствие снижения затрат обучаемого, связанных с необходимостью приобретения литературы, поездок и т. д.

3. Электронное обучение дает возможность реализовать индивидуальную траекторию обучения в соответствии с когнитивными особенностями личности обучающегося, его знаниями и имеющейся подготовкой, наличием времени для обучения, материальными возможностями и т. д.

4. Электронное обучение может способствовать повышению качества учебного процесса. Как правило, внедрение информационных технологий, сопровождается повышением качества курсов и программ учебных заведений, что связано с тем, что курсы и программы создаются при участии целой команды специалистов. Качество учебного процесса может быть повышено с точки зрения представления учебных материалов, педагогической поддержки студентов и т. д.

5. Электронное обучение способствует развитию новых образовательных концепций и новых моделей преподавания. В основе таких моделей и концепций находится индивидуально-ориентированное обучение, с ориентацией на обучаемом и его индивидуальных особенностях. При этом обучение становится активным, ориентированным на практическую деятельность.

6. Электронное обучение открывает возможность перехода к открытой модели образования, позволяет реализовать концепцию непрерывного образования. Рост образовательного уровня личности в течение жизни является общественной потребностью, особенно в связи с задачей компенсации утрачиваемых функций и возможностей.



7. Электронное обучение дает возможность всем участникам учебного процесса развиваться в соответствии с требованиями времени. И преподаватели, и обучаемые в системе электронного обучения развивают свои знания, умения и навыки в соответствии с новейшими технологиями и стандартами. Кроме того, использование электронных технологий позволяет быстро актуализировать содержание учебных материалов и оперативно обновлять форму их представления в соответствии с современными требованиями.

8. Электронное обучение предоставляет обучающимся потенциально равные возможности обучения. Вне зависимости от качества преподавания в конкретном учебном заведении, обучающийся может получать учебную информацию и заниматься практически в любом оборудованном месте [7].

Электронное обучение осуществляется через компьютерное средство обучения (КСО). КСО — это программный комплекс, предназначенный для решения определенных педагогических задач, имеющий предметное содержание и ориентированный на взаимодействие с обучаемым [8].

Существует огромное количество различных видов КСО, но по решаемым учебно-педагогическим задачам их можно разбить на группы теоретической, практической подготовки, контроля знаний, учебной и производственной практики, самостоятельной работы.

Также, могут быть представлены комплексные системы, в которых интегрированы функции различных средств. Но для эффективного внедрения КСО в образовательный процесс необходимо учитывать ряд общих педагогических требований при их проектировании и разработке [9]:

- обеспечение дидактического и психолого-педагогического сопровождения учебного процесса, применение различных методик и теорий усвоения знаний;

- создание условий для самостоятельного изучения учебного материала (самообразования), позволяющих студенту выбрать удобные для него время и место работы с КСО;

– предоставление средств для вариативности и адаптивности обучения, способных учитывать и настраиваться на индивидуальные особенности студента, такие как: текущий уровень подготовки, темп изучения, восприятие учебного материала, мыслительная деятельность, память и др.;

– возможность автоматизированного контроля и объективного оценивания знаний, умений и навыков;

– предоставление полных методических рекомендаций и указаний, помощи по работе как с электронным ресурсом, так и с учебным материалом.

Внедрение новых информационных и коммуникационных технологий в высшее профессиональное образование приводит к существенной перестройке процесса обучения и, как следствие, к необходимости разработки соответствующего учебно-методического обеспечения [11], которые строятся на веб-платформах. Веб-платформа — это многоцелевая платформа для разработки веб-проектов и управления содержимым. Она представляет собой набор комплексных решений, направленных на легкую и успешную разработку веб-сайта и его поддержку. Благодаря модульной конструкции, веб-платформу можно настроить для выполнения задач любого интернет-проекта. Под модулем подразумевается набор страниц сайта, которые реализуют одну общую функцию веб-сайта, например, форум, статьи, новости и т. д. Модуль включает в себя административную и пользовательскую части. Модульная конструкция легка и функциональна, что требует минимум времени для изучения и начала применения.

Сегодня существуют несколько десятков платформ электронного обучения, построенных по принципу открытых источников. Девять наиболее популярных открытых платформ (Atutor, Dokeos, dotLRN, ILIAS, LON-CAPA, Moodle, OpenUSS, Sakai, Spaghetti Learning) были проанализированы американскими исследователями по 34 параметрам, сгруппированным в восемь блоков [11]:

- инструменты управления учебным курсом;
- возможности администрирования;

- технические аспекты;
- возможности адаптации;
- удобство использования платформы;
- управление данными пользователя;
- объекты обучения;
- средства общения.

Учёные пришли к выводу, что по большинству критериев лучшей среди открытых платформ является система Moodle (Модульная объектно-ориентированная динамическая обучающая среда). Система Moodle изначально была ориентирована на университетское образование. Сегодня эта платформа используется в 214 странах и более чем на 78 языках. В России на начало 2013 года зарегистрировано более тысячи установок Moodle на компьютеры учебных заведений и преподавателей, при этом количество пользователей достигает пятисот тысяч человек

Согласно выше изложенному можно сказать, что компьютерные средства обучения как функциональные составляющие учебно-методического обеспечения предоставляют широкие возможности по развитию эффективности процесса обучения студентов.

Многочисленные исследования возможностей подобных методов обучения показали, что, используя сетевые технологии, можно эффективно реализовать все составляющие образовательного процесса: информационную, практическую, коммуникационную и оценочную [9].

Таким образом, проектирование, разработка и реализация модели учебно-методического обеспечения образовательного процесса в контексте электронного обучения в высшем учебном заведении предоставляет широкий спектр возможностей.

## 1.2 Обзор существующих разработок

Обучение с использованием возможностей ИКТ применяют многие образовательные учреждения. В 2012 году был подписан и вступил в силу Федеральный закон «О внесении изменений в Закон Российской Федерации «Об образовании» в части применения электронного обучения, дистанционных образовательных технологий» [12]. Согласно положениям документа при реализации основных образовательных программ (ООП), независимо от форм получения образования, может применяться электронное обучение, а также должны быть созданы все условия для функционирования информационно-образовательной среды, обеспечивающей освоение обучающимися ООП в полном объеме, независимо от их мест нахождения. Зачет результатов обучения осуществляется в порядке и формах, установленных организацией самостоятельно, при представлении обучающимся документов, подтверждающих пройденное им обучение. Уточнено также, что образовательные организации самостоятельно определяют порядок оказания учебно-методической помощи обучающимся, в том числе в форме индивидуальных консультаций, оказываемых дистанционно с использованием информационных и телекоммуникационных технологий.

При реализации образовательных программ или их частей с применением исключительно электронного обучения, дистанционных образовательных технологий образовательная организация самостоятельно и (или) с использованием ресурсов иных организаций:

- создает условия для функционирования электронной информационно-образовательной среды, обеспечивающей освоение обучающимися образовательных программ или их частей в полном объеме независимо от места нахождения обучающихся;

- обеспечивает идентификацию личности обучающегося, выбор способа которой осуществляется организацией самостоятельно, и контроль соблюдения условий проведения мероприятий, в рамках которых осуществляется оценка результатов обучения.

Онлайн-образование – это возможность в домашних условиях если не научиться практически всему, то хотя бы получить примерное представление: от психологии и биологии до языков программирования и квантовой механики. Подавляющее число курсов ведется на английском языке, но в последнее время и в России стали появляться программы интерактивного обучения, наиболее популярные из которых:

1. Uniweb — это платформа, где собраны онлайн-курсы нескольких крупных университетов, среди которых МГИМО, РАНХиГС и МГУ. Здесь, в основном, представлены программы по бизнес образованию продолжительностью от одной до восьми недель. За это время учащиеся должны просмотреть курс небольших видеолекций, изучить сопутствующие материалы и литературу, а также выполнить домашние задания. Слушатели могут сами планировать свое расписание, однако, у курса есть дедлайн и на выполнение заданий дается строго определенное количество времени. По окончании занятий слушатели получают сертификаты.

2. «Универсариум», часть курсов ориентирована на конкретных работодателей и дает возможность трудоустройства для самых успешных выпускников. Обучение в «Универсариуме» бесплатное и может продолжаться от семи до десяти недель. Программы, разработанные преподавателями МГУ и МФТИ, состоят из модулей, а модули, в свою очередь, включают в себя видеолекцию, самостоятельную работу, домашнее задание и тест. Сейчас на сайте доступны программы по химии, математике и микроэкономике. В ближайшее время появятся курсы для гуманитариев.

Одна из главных особенностей «Универсариума» — система кросс-проверки домашних работ. Каждое выполненное задание просматривают не только преподаватели, но и другие учащиеся, а итоговая оценка рассчитывается как средний балл. За проверку чужих заданий также можно получить дополнительные баллы.

3. Университет без границ. Главное отличие «Университета без границ» от других систем онлайн-обучения — это то, что курсы проходят

в режиме реального времени, поэтому учащиеся могут не только послушать лекцию, но и задать вопросы, спросить совет или устроить дискуссию. Разработчики учебных программ делали основной упор на медиа, историю, политику, глобализацию, урбанистику, социологию. Курсы ведут преподаватели американских, европейских и российских вузов, включая Tufts University, Georgetown University, George Mason University, Max Planck Institute, МВШСЭН. По окончании обучения студентам выдается электронный сертификат.

4. HTML Academy. Интерактивные курсы HTML Academy посвящены обучению HTML и CSS. Основной упор делается на практических занятиях, поэтому программа состоит из многочисленных упражнений, в ходе которых ученику необходимо изменять HTML и CSS код в онлайн редакторах, следуя указаниям системы. Учебная программа состоит из восьми тем, изучению которых посвящен один или несколько курсов. В конце ученикам необходимо выполнить итоговое испытание — сверстать макет, специально разработанный с учетом пройденного материала.

5. Онлайн-МФТИ. Учебный центр Онлайн-МФТИ предназначен в основном для абитуриентов и студентов. Здесь проводятся разборы олимпиад, консультации, подготовка к письменным и устным экзаменам, интерактивные дни открытых дверей и курсы повышения квалификации. Все занятия бесплатные и предусматривают возможность живого двустороннего общения, то есть учащиеся могут задавать вопросы и участвовать в обсуждениях.

Сегодня в мире существует около 400 различных обучающих платформ. Наибольшее распространение получили системы Blackboard и Moodle. Из TOP-200 лучших университетов мира по версии The World University Rankings за 2011 году в настоящее время более 70 % используют системы и решения компании Blackboard. Среди них: Гарвардский, Стэнфордский и Оксфордский университеты. Некоторые российские вузы тоже используют данную платформу. В их числе: Санкт-Петербургский государственный университет (СПбГУ), Дальневосточный федеральный университете (ДФУ),

Ленинградский Государственный Университет им. Пушкина. Недостатками этого продукта применительно к российским условиям является её высокая стоимость и отсутствие русскоязычной версии.

С 2007 года по настоящее время Сибирский федеральный университет активно внедряет систему Moodle, которая является важнейшим звеном во взаимодействии студентов с преподавателями. LMS Moodle является одной из наиболее известных и распространенных систем управления дистанционным обучением (модульная объектно-ориентированная динамическая учебная среда). LMS Moodle представляет собой свободное веб-приложение, представляющее возможность создавать сайты для онлайн обучения. Основными недостатками связанным с работой в системах электронного обучения, в частности LMS Moodle, является сложность вариативности и адаптивности обучения, т.е. нельзя учитывать и настраиваться на индивидуальные особенности обучающихся.

### **1.3 Общие сведения и практическая значимость разрабатываемого продукта**

Рассмотрев существующие разработки электронных обучающих систем, можно сделать вывод, что в настоящее время нет действующих электронных систем, которые бы имели функциональные возможности ориентироваться на пользователя системы, учитывая его индивидуальные особенности обучения.

В связи с этим разрабатываемая система даст возможность формировать график работы обучаемого по освоению материалов учебной дисциплины, а также адаптивного управления деятельностью обучаемого при реализации запланированного учебного процесса с использованием электронных обучающих систем, что в теории должно решить проблемы реализации метода индивидуального подхода обучения.

## **1.4 Выводы по главе 1**

В первой главе выпускной квалификационной работы рассмотрены теоретические аспекты для разработки веб-платформы, и сформулирована следующая проблема.

Заметной тенденцией в автоматизации обучения является разработка методов, методик и средств, реализующих индивидуальный подход к процессу обучения [18]. Но в настоящий момент, рассмотрев существующие электронные обучающие системы, можно сделать вывод, что большинство систем образовательного процесса направлены на общие массы обучающихся и существует проблема отсутствия индивидуального подхода к обучению, не учитывая индивидуальные особенности обучающихся.

В ходе решения поставленной задачи были рассмотрены основные понятия, определения и принципы организации электронного образовательного процесса, существующие разработки, общие сведения и практическая значимость разрабатываемого продукта. При ознакомлении с существующими разработками сделан вывод, что в настоящий момент существует проблема реализации индивидуального подхода к обучению.

Одним из способов решения данной проблемы может быть разработка веб-платформы учебно-методического обеспечения, а также редактор базы данных для работы с понятийным аппаратом изучаемой учебной дисциплины, что и представлено в практической части выпускной квалификационной работы.



## Глава 2. Проектирование платформы

### 2.1 Основные требования

Для предоставления полных функциональных возможностей разрабатываемой платформы необходимо обозначить основные требования, которые применяются к ней. В первую очередь пользователям необходима регистрация, так как доступ к функциям платформы возможен только после регистрации и подтверждения администратором учетной записи пользователя. Так же в систему требуется внедрить модули для работы с учебным курсом. Необходимо предоставить возможность доработки или изменения системы в соответствии с требованиями заказчика.

Платформа должна состоять из следующих основных элементов:

- редактор базы данных для работы с предметной областью;
- модуль визуализации понятий предметной области;
- модуль формирования тематических разделов;
- модуль индивидуального графика обучения;
- информационная страницы «Справка».

Пользовательский интерфейс сайта должен обеспечивать наглядное, легко понятное отображение структуры информации, размещенной на нем, быстрый и логичный переход к разделам и страницам. Навигационные элементы должны обеспечивать однозначное понимание пользователем их смысла: ссылки на страницы должны быть снабжены заголовками, условные обозначения соответствовать общепринятым. Графические элементы навигации должны быть с дополнительными подписями.

Система должна обеспечивать навигацию по всем доступным пользователю ресурсам и отображать соответствующую информацию.

При выборе какого-либо из пунктов меню пользователем должна загружаться соответствующая ему информационная страница.

В систему необходимо внедрить элемент «Справка». Он содержит пояснения всех используемых терминов, описания основных вариантов исполь-

зования, включая альтернативные сценарии, а также подробный обзор интерфейса программы. Справка направлена для разрешения возникших во время работы вопросов.

## 2.2 Классификация пользователей

В системе присутствует два типа пользователя:

- Пользователем платформы является эксперт (преподаватель), он выполняет работу по составлению базы данных описывающую предметную область для последующего её применения к учебному курсу;
- Администратор осуществляет контроль над работой сайта и редактирует его содержимое.

## 2.3 Схема работы и взаимодействие компонент разрабатываемой платформы

Основную структуру взаимодействия элементов платформы учебно-методического обеспечения можно выразить следующей схемой (рисунок 1):

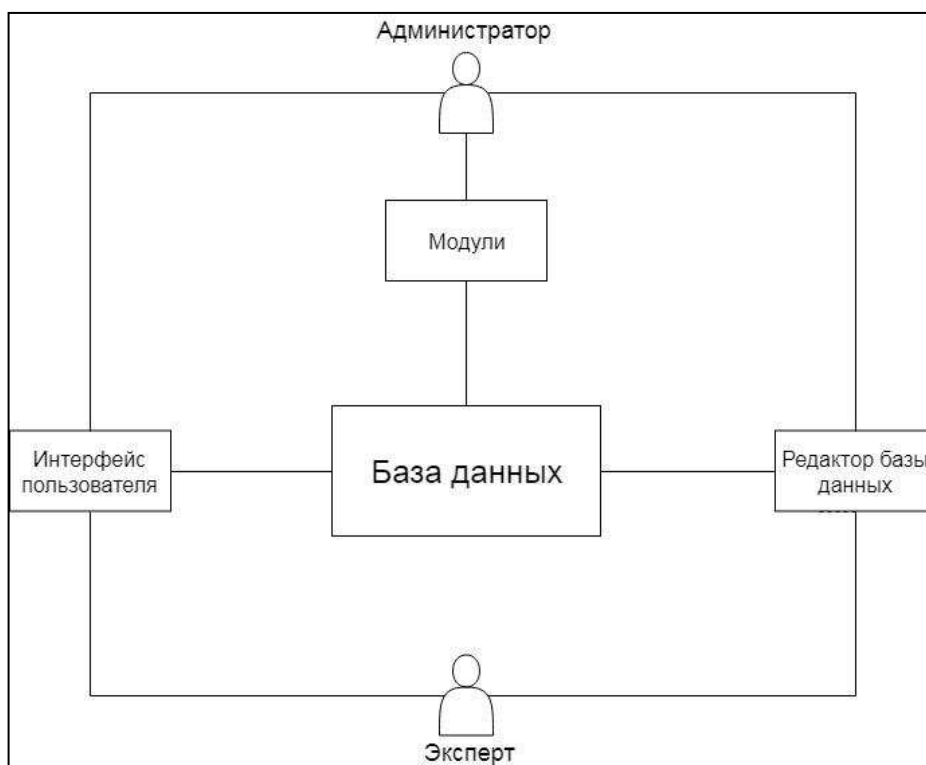


Рисунок 1 – Схема взаимодействия элементов разрабатываемой системы

Главным элементом системы является база данных (БД). Она описывает предметную область изучаемой дисциплины. Предметная область состоит из системы понятий необходимых для получения знаний при изучении дисциплины. В свою очередь БД создается, удаляется, редактируется экспертом, знающим ее структуру и правильность заполнения с помощью редактора БД.

Модули, представляют собой программу осуществляющие определенные действия на основе БД. Наименования и функциональные возможности отражены в таблице 1.

Таблица 1 – Функции и возможности модулей платформы

Наименование	Функции
Модуль визуализации	Модуль предназначен для осуществления работы с системой понятий, учебной дисциплины и представляет возможность для анализа элементов графа и их характеристик, таких как, наследственность и время, необходимое для изучения понятия. Модуль помогает реализовать наглядное представление системы понятий и структурировать их наиболее эффективным способом для анализа и преобразований. Это позволит преподавателям формировать сложные системы понятий и быстро в них ориентироваться по сравнению с текстовыми документами.
Модуль формирования тематических разделов	Группирует понятия в тематические разделы, учитывая связи и время обучения.
Модуль формирования индивидуального графика обучения	Для каждого студента в случае не совпадения скорости его обучения с учебным графиком дает возможность переформировывать его график (сокращать/увеличивать).

Интерфейс пользователя – комплекс программных средств, реализующих диалог пользователя с АИС как для ввода информации, так и для получения результатов работы платформы.

Администратор – обладает полными правами доступа. Имеет возможность осуществлять манипуляции с такими элементами как: интерфейс пользователя, модули, база данных и ее редактор. Это необходимо для гибкости платформы, т. е. добавление или упреждения функционала данных элементов разрабатываемой платформы.

## 2.4 Обзор прецедентов

Для разрабатываемой системы предусмотрено две роли. Выявленные прецеденты представлены в таблице 2.

Таблица 2 – Функции и возможности модулей платформы

Код	Основной актер	Наименование	Формулировка
Д1	Эксперт	Работа с БД	Этот вариант использования позволяет эксперту, внести, удалить, изменить данные в БД
Д2	Эксперт	Поиск информации по БД	Может осуществить поиск необходимой информации в БД
Д3	Эксперт	Работа с модулями платформы	Данный вариант предусматривает использование функций встроенных модулей платформы
Д4	Администратор	Управление сайтом	Работа с наполнением сайта, добавление функциональных возможностей сайта
Д5	Администратор	Редактирование разделов	Редактирование содержимого и структуры разделов сайта
Д6	Администратор	Удаление разделов	Удаление разделов сайта
Д7	Администратор	Создание разделов	Создание новых разделов сайта
Д8	Администратор	Подтверждение пользователей	Подтверждение аккаунтов пользователей после регистрации

Для более детального выявления требований к разрабатываемой платформе, необходимо более рассмотреть каждый прецедент.

## 2.5 Описание вариантов использования

Краткое описание прецедента «Работа с БД» отражено в таблице 3.

Таблица 3 – Работа с БД

Код	Основной актер	Наименование	Формулировка
Д1	Эксперт	Формирование БД	Этот вариант использования позволяет эксперту, внести, удалить, изменить данные в БД

Основное действующее лицо: Эксперт.

Другие участники прецедента: отсутствуют.

Связи с другими вариантами использования: расширяется прецедентами «Поиск информации по БД», «Работа с модулями платформы» так как по БД осуществляется поиск нужной информации в БД, а работа с модулями осуществляется за счет данных сформированной БД.

Краткое описание: данный вариант использования позволяет эксперту внести, удалить, изменить некоторую информацию в БД.

Краткое описание прецедента «Поиск информации в БД» отражено в таблице 4.

Таблица 4 – Поиск информации в БД

Код	Основной актер	Наименование	Формулировка
Д2	Эксперт	Поиск информации БД	Может осуществить поиск необходимой информации в БД

Основное действующее лицо: Эксперт.

Другие участники прецедента: отсутствуют.

Связи с другими вариантами использования: поиск происходит по БД, поэтому связан с прецедентом «Работа с БД».

Краткое описание: данный вариант использования позволяет эксперту осуществить поиск нужной информации в таблице БД, для получения необходимой информации.

Краткое описание прецедента «Работа с модулями платформы» отражено в таблице 5.

Таблица 5 – Работа с модулями платформы

Код	Основной актер	Наименование	Формулировка
Д3	Эксперт	Работа с модулями платформы	Данный вариант предусматривает использование функции встроенных модулей платформы

Основное действующее лицо: Эксперт.

Другие участники прецедента: отсутствуют.

Связи с другими вариантами использования: работа модулей происходит на основе заполненной БД, поэтому связан с прецедентом «Работа с БД».

Краткое описание: данный вариант использования позволяет эксперту использовать функции встроенных модулей платформы.

Краткое описание прецедента «Управление сайтом» отражено в таблице 6.

Таблица 6 – Управление сайтом

Код	Основной актер	Наименование	Формулировка
Д4	Администратор	Управление сай- том	Работа с наполнением сайта, редактирование вспомога- тельных сервисов.

Основное действующее лицо: Администратор.

Другие участники прецедента: отсутствуют.

Связи с другими вариантами использования: включает в себя прецеденты «Редактирование разделов», «Создание разделов» и «Удаление разделов».

Краткое описание: данный вариант использования позволяет администратору работать с наполнением сайта, а также добавлять функциональные возможности сайта.

Краткое описание прецедента «Редактирование разделов» отражено в таблице 7.

Таблица 7 – Редактирование разделов

Код	Основной актер	Наименование	Формулировка
Д5	Администратор	Редактирование разделов	Редактирование содержимого и структуры разделов сайта

Основное действующее лицо: Администратор.

Другие участники прецедента: отсутствуют.

Связи с другими вариантами использования: включается прецедентом «Управление сайтом».

Краткое описание: данный вариант использования позволяет администратору редактировать содержимое и структуру разделов сайта.

Краткое описание прецедента «Удаление разделов» отражено в таблице 8.

Таблица 8 – Удаление разделов

Код	Основной актер	Наименование	Формулировка
Д6	Администратор	Удаление разделов	Удаление разделов сайта

Основное действующее лицо: Администратор.

Другие участники прецедента: отсутствуют.

Связи с другими вариантами использования: включается прецедентом «Управление сайтом».

Краткое описание: данный вариант использования позволяет администратору удалять разделы сайта.

Краткое описание прецедента «Удаление разделов» отражено в таблице 9.

Таблица 9 – Создание разделов

Код	Основной актер	Наименование	Формулировка
Д7	Администратор	Создание разделов	Создание новых разделов сайта

Основное действующее лицо: Администратор.

Другие участники прецедента: отсутствуют.

Связи с другими вариантами использования: включается прецедентом «Управление сайтом».

Краткое описание: данный вариант использования позволяет администратору создавать новые разделы сайта.

Таблица 10 – Подтверждение пользователей

Код	Основной актер	Наименование	Формулировка
Д8	Администратор	Подтверждение пользователей	Подтверждение аккаунтов пользователей после регистрации

Основное действующее лицо: Администратор.

Другие участники прецедента: отсутствуют.

Связи с другими вариантами использования: отсутствуют.

Краткое описание: данный вариант использования позволяет администратору подтверждать аккаунты пользователей регистрации.

## 2.6 Структуризация вариантов использования

Диаграмма прецедентов (Use case) — UML-диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне [13]. Вариант использования — «Работа с БД» включает в себя внесение, удаление и изменение информации в БД. На рисунке 2 отображена диаграмма для данного варианта использования.

На рисунке 3 показаны варианты использования актора под именем «Администратор». На нем показано совмещение вариантов использования «Управление сайтом» и «Подтверждение пользователей». На диаграмме также показано, что вариант использования «Управление сайтом» включает в себя создание, удаление, редактирование разделов сайта.



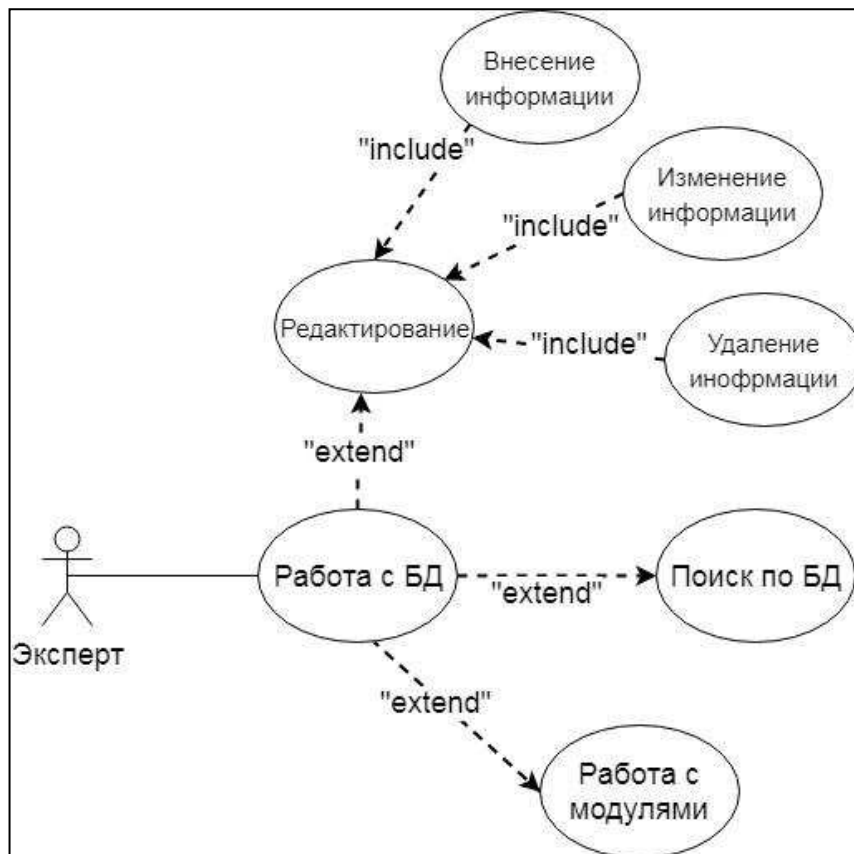


Рисунок 2 – Вариант использования «Работа с БД»

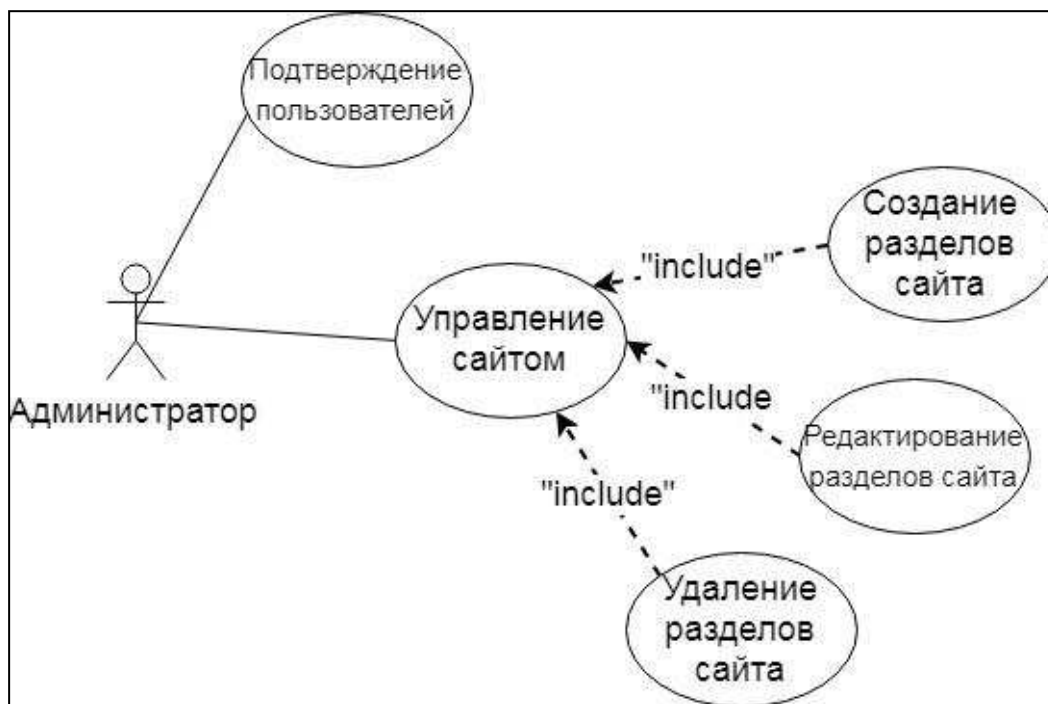


Рисунок 3 – Варианты использования «Управление сайтом» и «Подтверждение пользователей»

## 2.7 Диаграмма деятельности

Анализ вариантов использования показал, что с точки зрения потенциальных рисков и архитектурной значимости наиболее существенными являются прецедент, связанный с работой эксперта. Требуется построить диаграмму деятельности. Диаграмма деятельности — UML-диаграмма, на которой показано разложение некоторой деятельности на её составные части [14]. На рисунке 4 показана диаграмма деятельности прецедента «Работа с БД».

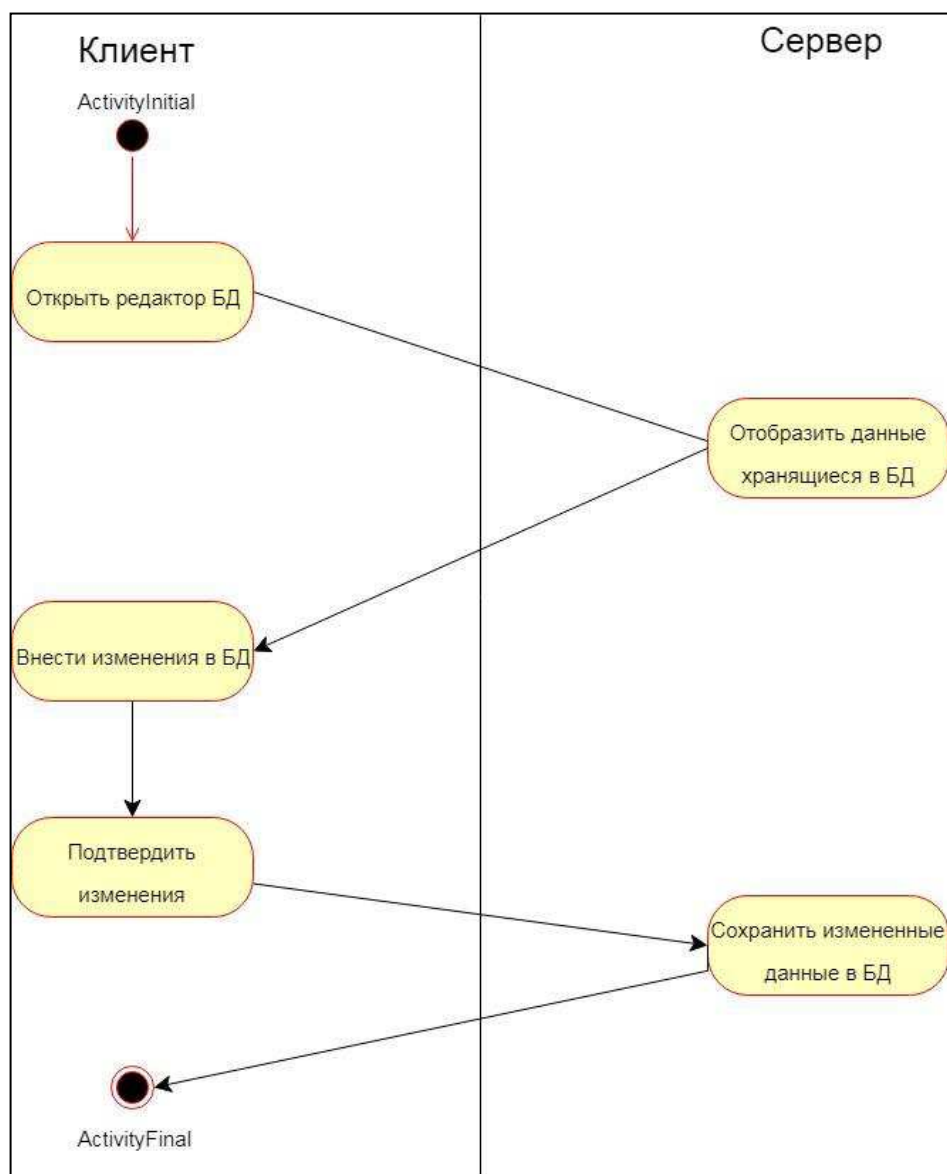


Рисунок 4 – Диаграмма деятельности прецедента «Работа с БД»

Прецедент начинается, когда эксперт выбирает пункт основного меню «Редактор БД» на главной странице сайта.

Перед тем как начинается этот прецедент, эксперт должен авторизоваться в платформе учебно-методического обеспечения, чтобы получить доступ к редактору БД.

Базовый поток — редактировать БД

1. Пользователь открывает «Редактор БД».
2. Система отображает интерфейс программы, где эксперт может внести изменения
3. Система сохраняет изменения, что является окончание прецедента.

Время сохранения изменений в БД не должно превышать 20 секунд.

После окончания прецедента эксперт завершает работу с БД и может работать с модулями платформы.

## 2.8 Диаграмма классов

Диаграмма классов — диаграмма, демонстрирующая классы системы, их атрибуты, методы и взаимосвязи между ними [15]. Диаграмма классов разрабатываемой системы представлена на рисунке 5.

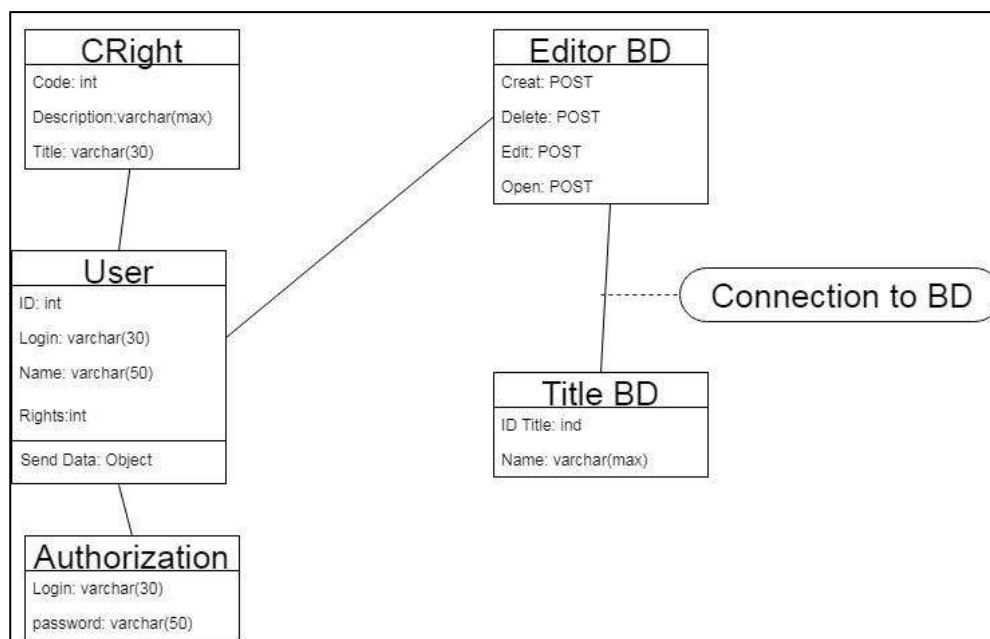


Рисунок 5 – Диаграмма классов платформы

На представленной диаграмме изображено 5 классов. Класс «User» содержит информацию о пользователях.

Таблица 11 – Класс User

Название	Тип	Описание
ID	Int	Код записи
Login	String	Логин пользователя
Name	String	Имя пользователя
Rights	Int	Код прав пользователя
SendData	Void	Отправляет запрос от пользователя

Основная функция данного класса – хранение данных о зарегистрированных пользователях платформы. Так же эти данные необходимы для авторизации пользователя.

Класс « Authorization» (таблица 12) обеспечивает авторизацию пользователей.

Таблица 12 – Класс Authorization

Название	Тип	Описание
Login	String	Логин пользователя
Password	String	Логин пользователя

Основная функция этого класса авторизировать пользователя и обеспечить доступ к функциям сайта. Авторизация происходит с помощью операции «Authentication» и атрибутов Login, Password.

Класс «CRight» описывает права пользователей (таблица 13). Разделяя пользователей информационной системы на администраторов и пользовате-

лей. Различие пользователей и администраторов осуществляется по атрибуту «Code».

Таблица 13 – Класс CRight

Название	Тип	Описание
Code	Int	Код права
Description	String	Описание прав
Title	String	Название

Классы «Editor BD» и «Title BD» необходимы для работы редактора БД. После авторизации и входа пользователя в редактор БД, создается запрос на открытие из БД таблицы находящуюся в классе «Title BD» и возвращает ответ на открытие необходимой таблицы. Аналогичная структура запросов формируется при редактировании, создании или удалении записи хранящейся в БД.

## 2.9 Диаграмма последовательностей

Диаграмма последовательности — диаграмма, на которой показано взаимодействие объектов (обмен между ними сигналами и сообщениями), упорядоченное по времени, с отражением продолжительности обработки и последовательности их проявления [16].

На рисунке 6 представлена последовательность выполняемых действий прецедента «Работа с БД». Manager — это эксперт, BigData — база данных. Сплошная стрелка показывает, какое действие совершает пользователь, а пунктирная возвращает ответ системы.

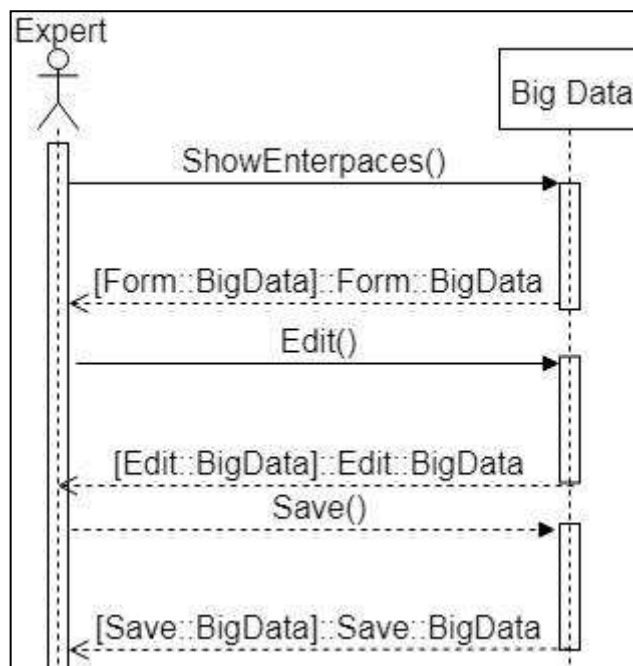


Рисунок 6 – Диаграмма последовательностей

Поток событий:

- 1) Эксперт запрашивает таблицу БД.
- 2) Система показывает таблицу.
- 3) Диспетчер изменяет данные в БД, а система сохраняет их.

Данная последовательность может использоваться бесконечное количество раз подряд при необходимости эксперта. Необходимо отметить, что в данной диаграмме вместо «Edit» могут быть операции «Create», «Delete», в зависимости от задачи пользователя системы.

## 2.10 Диаграммы компонентов и развертывания

Диаграмма компонентов — статическая структурная диаграмма, показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами [17]. На рисунке 7 представлена диаграмма компонентов разрабатываемой системы.

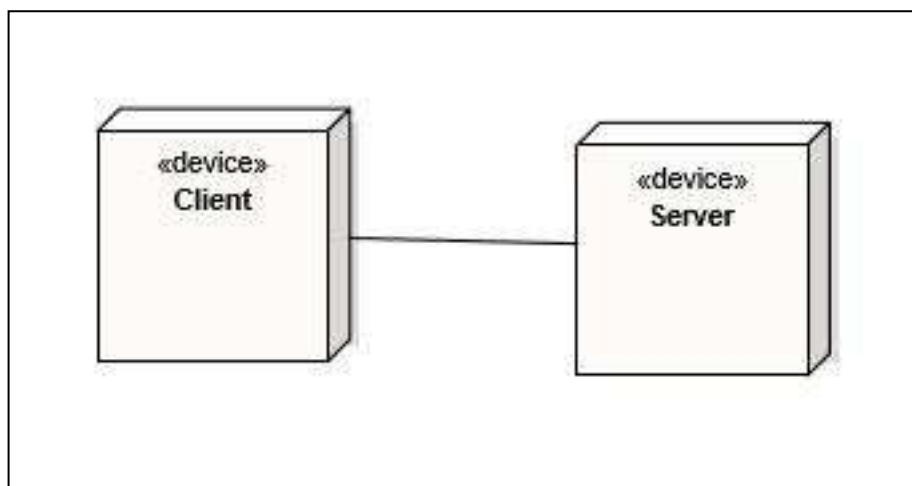


Рисунок 7 – Диаграмма компонентов

Элементы диаграммы:

- Client — компонент для программного представления пользователя, обеспечивает доступ к БД;
- Server — компонент реализующий взаимодействие с БД системы.

Между элементами существуют связь: ассоциации (сплошная линия).  
 Архитектура клиент-сервер — это концепция информационной сети, в которой основная часть ее ресурсов сосредоточена в серверах, обслуживающих своих клиентов [19]. Диаграмма развёртывания системы представлена на рисунке 8.

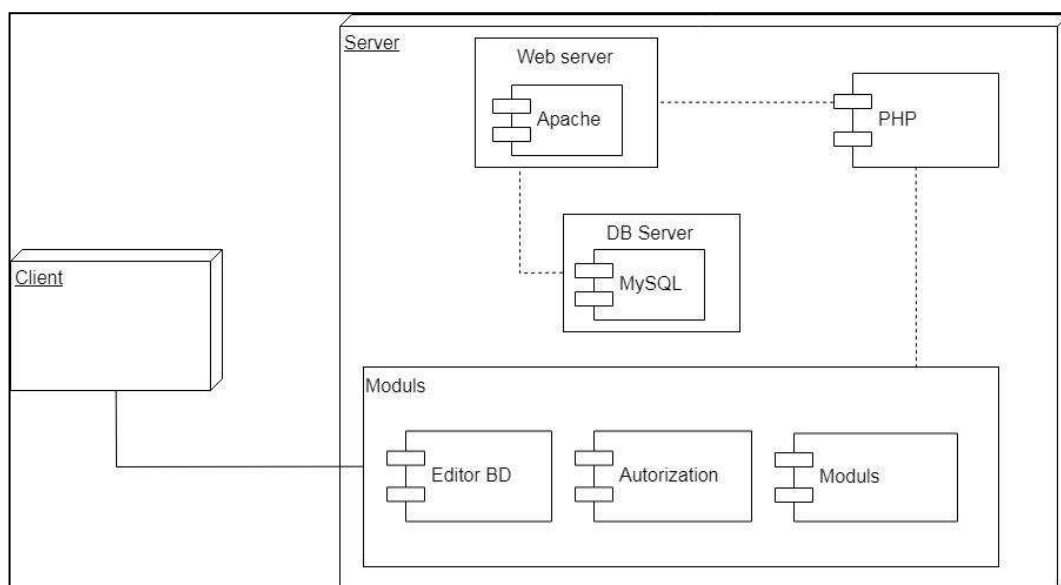


Рисунок 8 – Диаграмма развертывания

В отличие от диаграммы компонентов в диаграмме развертывания указано физическое расположение элементов. Диаграмма развертывания отображает способ взаимодействия компонентов с аппаратными средствами в физической системе, а также соединение аппаратных средств между собой. Основным элементом системы является узел. Узел – представление любого вычислительного ресурса.

## 2.11 Концептуальная схема сайта

Схема сайта включает в себя шесть элементов. Для наглядного представления нарисуем концептуальную схему сайта (рисунок 9).

Основной страницей реализуемого сайта является «Главная страница». Она включает все основные элементы и ссылки необходимые для работы эксперта с платформой. На ней содержатся основные ссылки для работы со всем модулями и элементами навигации.



Рисунок 9 – Концептуальная схема сайта



Страница «Редактор БД» содержит модуль работы с предметной областью учебного курса. Вставка модуля на страницу осуществляется с помощью фрейма.

Страница «Модуль визуализации» содержит модуль для визуализации предметной области. Структура построения страницы сайта аналогична содержанию «Редактор БД».

Страница «Модуль формирования тематических разделов» содержит ссылку на разработанный модуль.

Страница «Модуль формирования индивидуального графика обучения» содержит ссылку на разрабатываемый модуль.

Элемент справки отвечает за помощь в затрудненных ситуациях, объясняющий работу элементов платформы.

## **2.12 Средства разработки**

Система учебно-методического обеспечения будет разработана на базе веб-технологий, так как в соответствии с требованиями заказчика необходимо обеспечить непрерывную работу системы и доступ к ней независимо от территориального расположения пользователя с любого персонального компьютера, ноутбука или мобильного устройства. Поэтому рассмотрим информационные системы и технологии, которые будут применяться при реализации данной системы.

PHP — это распространенный язык программирования общего назначения с открытым исходным кодом. PHP сконструирован специально для ведения веб-разработок и его код может внедряться непосредственно в код HTML.

Популярность в области построения веб-сайтов определяется наличием большого набора встроенных средств для разработки веб-приложений:

- автоматическое извлечение POST и GET-параметров, а также переменных окружения веб-сервера в предопределённые массивы;

- взаимодействие с большим количеством различных систем управления базами данных в частности MySQL;
- автоматизированная отправка HTTP-заголовков;
- работа с HTTP-авторизацией;
- работа с cookies и сессиями;
- работа с локальными и удалёнными файлами;
- обработка файлов, загружаемых на сервер;

Для работы с БД будет использоваться СУБД под названием «phpMyAdmin». Она позволяет через браузер осуществлять администрирование сервера MySQL и просматривать содержимое таблиц и баз данных.

«HyperText Markup Language» — язык гипертекстовой разметки (HTML), стандартизированный язык разметки документов в среде Internet. Язык HTML интерпретируется браузерами, полученный в результате интерпретации форматированный текст отображается в рабочем окне приложения.

CSS (Cascading Style Sheets) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Язык используется создателями веб-страниц для задания цветов, шрифтов, расположения и других аспектов представления документа.

Основной целью разработки на языке CSS является разделение содержимого (написанного на HTML или другом языке разметки) и представления документа (написанного на CSS). Это разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печать, чтение голосом.

Для придания интерактивности веб-страницам будет применяться язык сценариев JavaScript. Данный язык программирования позволяет создавать интерактивные динамические веб-страницы, содержимое которых может изменяться в зависимости от действий пользователя. Сценарии JavaScript вы-

полняются браузером при наступлении различных событий: щелчок по ссылке, ввод текста, наведение указателя мыши на объект и т. д.

Для работы с HTML,JS и CSS будет использоваться свободный текстовый редактор для веб-разработчиков под названием Adobe Brackets.

Apache HTTP-сервер — полнофункциональный, расширяемый веб-сервер с открытым кодом. Apache является кроссплатформенным ПО, поддерживает большинство операционных систем существующих на рынке. Основными достоинствами Apache считаются надёжность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках и т.д.

Основное ядро разрабатываемого сайта будет система управления контентом (далее в тексте CMS) WordPress, данная система реализована на языке программирования PHP. В качестве базы данных используется реляционная система управления базами данных MySQL. Код CMS WordPress является открытым, а подключаемые модули позволяют увеличить возможности, изначально заложенные в систему.

### **2.13 Выводы по главе 2**

В ходе выполнения поставленной задачи вынесены основные требования предъявляемые в техническом задании (Приложение А) к платформе. Проанализировав сформулированные варианты использования, был сделан вывод, что с точки зрения потенциальных рисков и архитектурной значимости наиболее существенным прецедентом является «Работа с БД» принадлежащий актору «Эксперт».

В ходе решения поставленной задачи для проектирования модуля редактора БД предметной области учебной дисциплины использовались UML-диаграммы. Диаграммы отображают, каким образом будет работать и разрабатываться модуль.

Также, была проведена классификация пользователей платформы, рассмотрено взаимодействие ее компонент и схема работы, описана концептуальная схема разрабатываемого сайта, были выбраны средства для разработки платформ.

## Глава 3. Разработка платформы

### 3.1 Разработка платформы

Данный этап включает в себя такие моменты разработки платформы как:

- установка CMS и создание сайта;
- настройка сайта;
- настройка безопасности сайта;
- создание страниц сайта;
- настройка дизайна сайта.

#### 3.1.1 Установка CMS и создание сайта

Как было сказано ранее в перечне используемых средств разработки, основным ядром разрабатываемого сайта является CMS Wordpress. Для того, чтобы начать работу над сайтом необходимо установить CMS.

В первую очередь для установки CMS Wordpress на хостинг, нужно скачать дистрибутив на официальном сайте. Далее в файловом менеджере распаковать скаченный архив в автоматически созданную папку при создании нового сайта на используемом хостинге (рисунок 10).

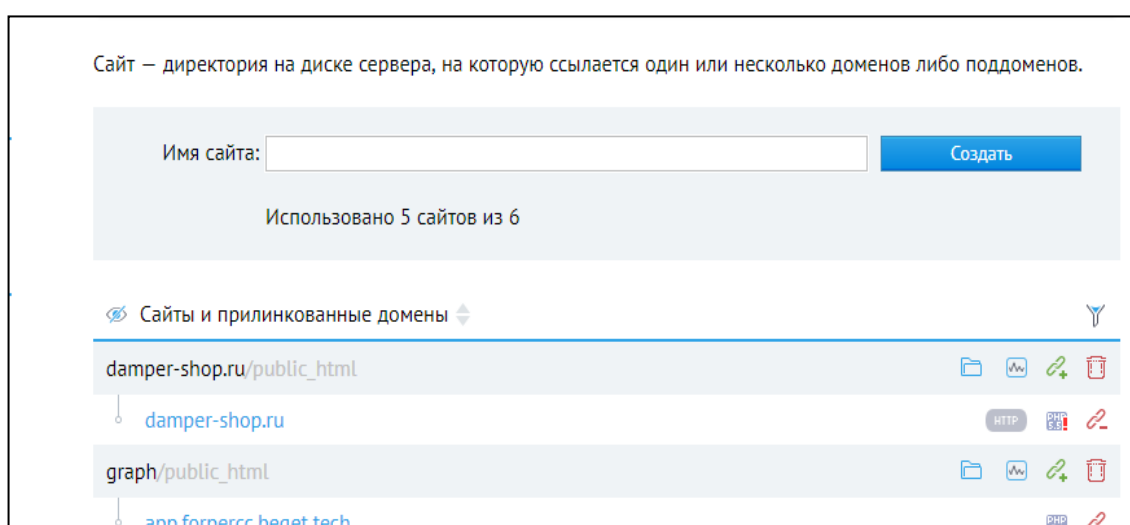


Рисунок 10 — Создание сайта на хостинге

Следующим шагом по установке CMS Wordpress является создание базы данных и подключения её к сайту. Подключение происходит во время установки CMS, после перехода по доменному имени на сайт. При переходе откроется меню установки, где необходимо ввести данные для подключения к БД, а так же для логин и пароль для входа в панель администратора (рисунок 11).

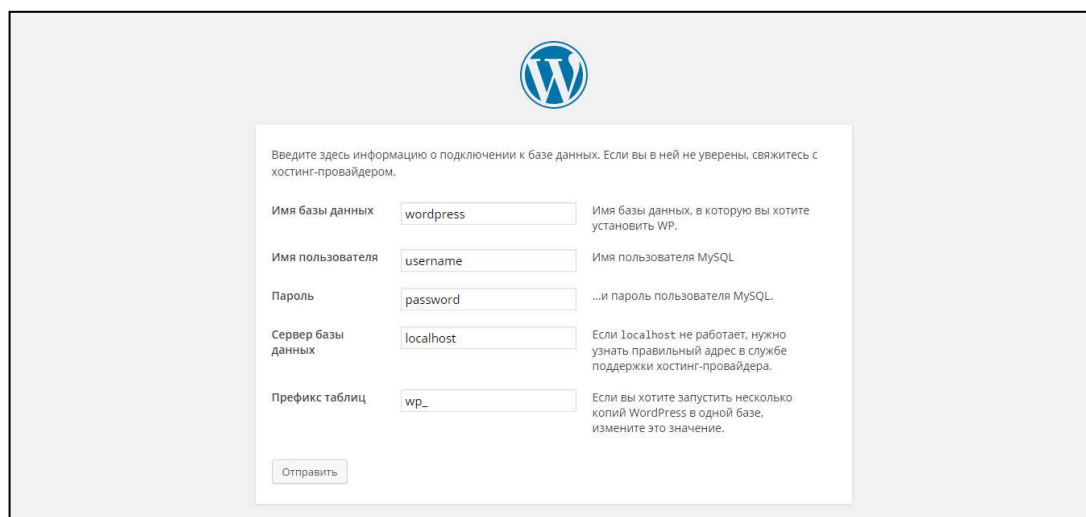
The image shows the WordPress installation database configuration screen. At the top center is the WordPress logo. Below it is a heading: "Введите здесь информацию о подключении к базе данных. Если вы в ней не уверены, свяжитесь с хостинг-провайдером." The form contains five rows of input fields with labels and explanatory text to the right. 1. "Имя базы данных" with the value "wordpress" and the note: "Имя базы данных, в которую вы хотите установить WP." 2. "Имя пользователя" with the value "username" and the note: "Имя пользователя MySQL." 3. "Пароль" with the value "password" and the note: "...и пароль пользователя MySQL." 4. "Сервер базы данных" with the value "localhost" and the note: "Если localhost не работает, нужно узнать правильный адрес в службе поддержки хостинг-провайдера." 5. "Префикс таблиц" with the value "wp\_" and the note: "Если вы хотите запустить несколько копий WordPress в одной базе, измените это значение." At the bottom left of the form is a button labeled "Отправить".

Рисунок 11 — Установка и подключение к БД

Заключительный этап по установке и созданию сайта — установка дополнительных расширений. Они направлены для реализации возможности видоизменять сайт и добавлять новые функции.

### 3.1.2 Настройка сайта

На данном шаге разработки платформы, необходимо настроить систему управления контентом, данные настройки открыты без установки дополнений.

Перейдя в панель администратора, необходимо выбрать в меню «Настройки» далее в «Общие» необходимо изменить название сайта, а также пункт меню «Членство», оно необходимо для добавления функции регистрации и входа на сайт (рисунок 12). Остальные пункты настроек остаются без изменений.

Название сайта: Платформа учебно-методического обеспечения

Краткое описание:   
*Объясните в нескольких словах, о чём этот сайт.*

Членство:  Любой может зарегистрироваться

Рисунок 12 — Настройка общих настроек

Помимо общих настроек так же необходимо указать в разделе «Чтение», какая из созданных страниц будет отображаться главной страницей сайта. Для этого требуется создать главную страницу сайта в разделе «Страницы». Далее, назначить её главной в пункте «На главную отображать» (рисунок 13).

На главной странице отображать

Ваши последние записи

Статическую страницу (выберите ниже)

Главная страница: Главная - навигация ▼

Страница записей: — Выбрать — ▼

Рисунок 13 — Настройка главной страницы сайта

Остальные разделы основных настроек CMS Wordpress, а именно «Написание», «Обсуждения», «Медиафайлы» не требует дополнительных настроек, и остаются по умолчанию.

В последнюю очередь перед реализацией общей структуры и дизайна сайта, необходимо настроить «Меню». Для этого, перейдя в данный раздел панели администратора, требуется его создать (рисунок 14).



Отредактируйте меню ниже или [создайте новое меню.](#)

Рисунок 14 — Создание нового меню

Далее создав меню, необходимо добавить элементы для навигации по сайту. Они направлены на облегчение понимания работы с системой. Основными элементами меню будут: «Переход на главную страницу», «Справка», «Выход из системы».

### 3.1.3 Настройка безопасности сайта

В техническом задании разрабатываемой системы прописано требование об ограничении пользователей пользующихся платформой. Для выполнения данного требования необходимо установить дополнения к CMS Wordpress в виде плагинов.

Для начала произведем ввод списка используемых плагинов для ограничения пользователей на сайте:

1. Dashboard Access Settings;
2. WP Security;
3. Login Press;
4. My Private Site;
5. Adminize.

В первую очередь необходимо осуществить ограничение веб-сайта для не зарегистрированных пользователей. Для этого необходимо установить и настроить плагин My Private Site. Он позволит администратору сайта ограничить сайт от случайных пользователей. При входе на сайт неопознанного пользователя, ему будет предложен вход на сайт. Также для того чтобы любой пользователь смог зарегистрироваться в системе удаленно, необходимо в настройках плагина выбрать пункт «Любой может зарегистрироваться».

Для повышения безопасности на сайте также необходимо добавить функцию подтверждения администратором регистрации пользователя на



сайте. С помощью плагина WP Security возможно включить функцию ручного одобрения новых регистраций.

Итоговой настройкой формы входа и регистрации будет с помощью плагина Login Press. Функциональные возможности данного плагина направлены на изменение визуальной составляющей формы входа и придание ей определенных стилей. С помощью плагина Login Press была добавлена визуальная составляющая формы регистрации и входа на сайте в корпоративной стилистике Сибирского Федерального Университета (рисунок 15).

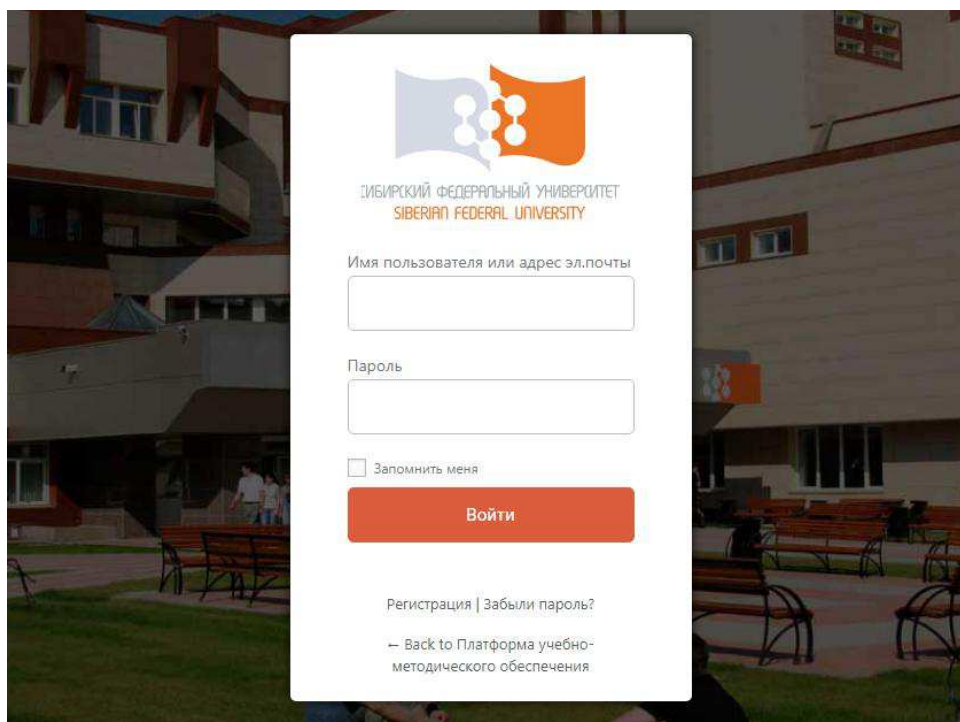


Рисунок 15 — Окно входа на сайт

Также необходимо ограничить простого пользователя от входа и настроек в панели и консоли администратора. Для этого необходимо использовать два плагина под названиями «Dashboard Access Settings» и «Adminize». Воспользовавшись плагином Dashboard Access Settings можно ограничить доступ к консоли администратора других пользователей, кроме администратора. Используя плагин Adminize можно ограничить по функциональным возможностям любого пользователя. С помощью него удалось максимально ограничить пользователя в функциональных возможностях работы с панелью и консолью администратор.

### 3.1.4 Создание страниц сайта

Основываясь на функциональных возможностях системы, прописанных в техническом задании, необходимо создать разделы сайта для работы с модулями платформы. В первую очередь необходимо их обозначить: «Главная страница», «Редактор БД», «Модуль визуализации», «Справка». Для того чтобы их создать необходимо перейти во вкладку страницы и создать по отдельности требуемые разделы (рисунок 16).

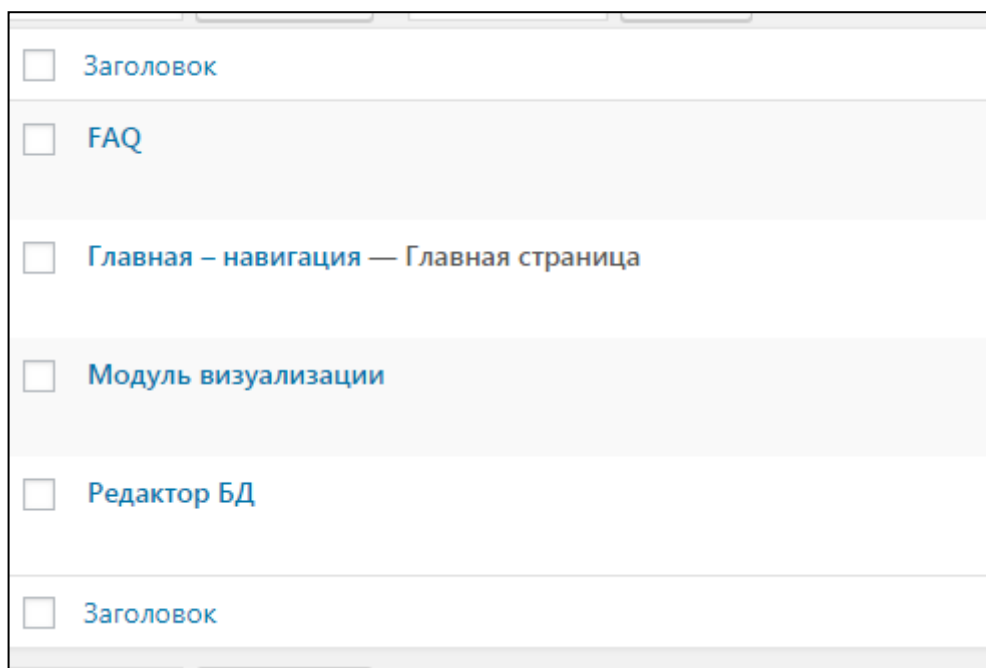


Рисунок 16 — Созданные страницы сайта

Для того, чтобы ограничить функциональные возможности созданных страниц, оговорим их содержание.

Главная страница является основной реализуемого сайта. Она включает все основные элементы необходимые для работы пользователя. С помощью нее осуществляется навигация по функциональным частям.

Страница «Справка» содержит пояснения всех используемых терминов, описание основных вариантов использования, включая альтернативные сценарии, а также подробный обзор интерфейса программы. Справка направлена для разрешения возникших во время работы вопросов.

Для работы с предметной областью на сайте размещены страницы «Редактор БД» и «Модуль визуализации». Данные страницы содержат программные модули соответствующие своим именам.

### **3.1.5 Настройка дизайна сайта**

Чтобы добавить целостности сайту, необходимо заполнить страницы необходимой информацией, а также добавить элементы навигаций для удобства пользования системой.

В первую очередь необходимо добавить единый стиль на сайте. Для этого в административной панели CMS Wordpress необходимо перейти во вкладку «Внешний вид» и выбрать меню «Настроить». Перечислим необходимые данные для настройки и приданию единообразия сайту:

- в разделе «Общий вид» необходимо выбрать блочный макет сайта, а также основной стиль кнопок и размер сетки сайта;
- в разделе «Логотип» добавим логотип сайта;
- в разделе «Шапка и подзаголовок» осуществим настройки «шапки» сайта выбрав стандартный вид;
- в разделе «Подвал» добавим общий фон нижней части и продублируем элементы навигации содержащиеся в меню сайта;
- в разделе «Цвета» с помощью «Генератор цвета» передадим единообразие цветовой гамме сайта.

Сформируем главную страницу сайта. Для этого необходимо воспользоваться плагином Visual composer. Функциональные возможности данного плагина направлены построение страниц сайта. С помощью него реализуем главную страницу с основными элементами необходимыми для работы и навигации по сайту (рисунок 17).

Как видно на рисунке главной страницы сайта, платформа выполнена в едином стиле. Таким же методом необходимо реализовать остальные страницы сайта.

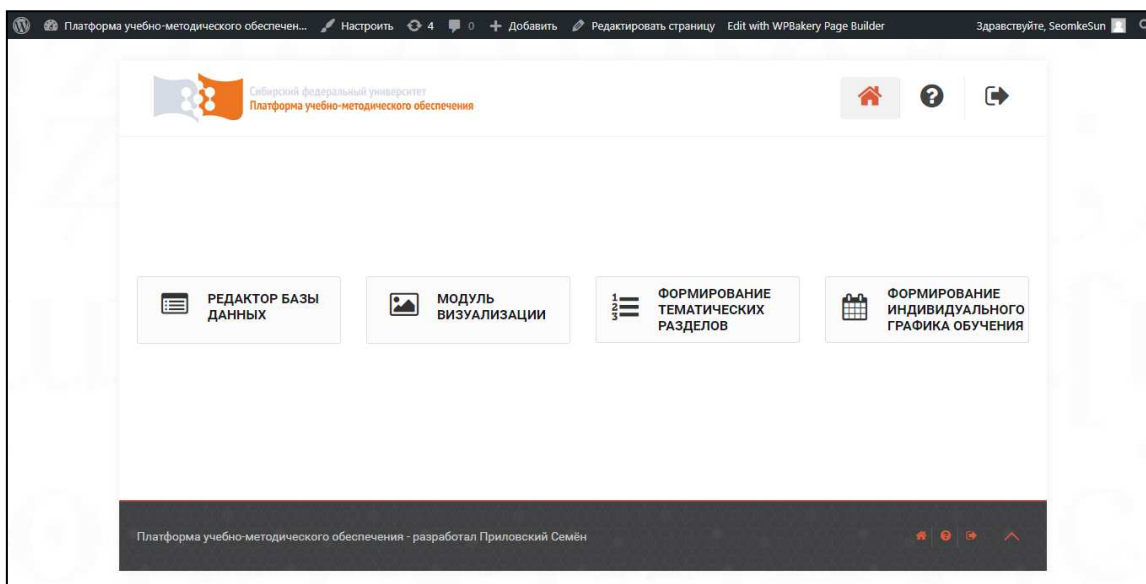


Рисунок 17 — Созданные страницы сайта

На данном этапе созданные страницы сайта имеют только оформление и единый стиль, далее необходимо добавить содержание страниц, т. е. добавить необходимые функции страницам, такие как работа с редактором БД и модулем визуализации предметной области. Для этого необходимо их разработать и внедрить в структуру сайта с помощью технологии фреймов.

### 3.2 Разработка редактора БД

Для того чтобы начать разработку программного модуля, необходимо создать БД, где в свою очередь будут храниться данные о работе модуля.

База данных состоит из четырех основных структурных элементов, которые отвечают за хранение данных работы своих модулей. На приведенной диаграмме (рисунок 18) наглядно представлены все таблицы, используемые в базе данных, их поля, а также взаимосвязи между таблицами. Подобное визуальное представление базы данных чрезвычайно удобно, так как позволяет визуально представить всю структуру хранения данных, что чрезвычайно важно при дальнейшей работе с ними. Далее будут подробно рассмотрены все элементы представленной ER-диаграммы — как сами таблицы, так и их поля.

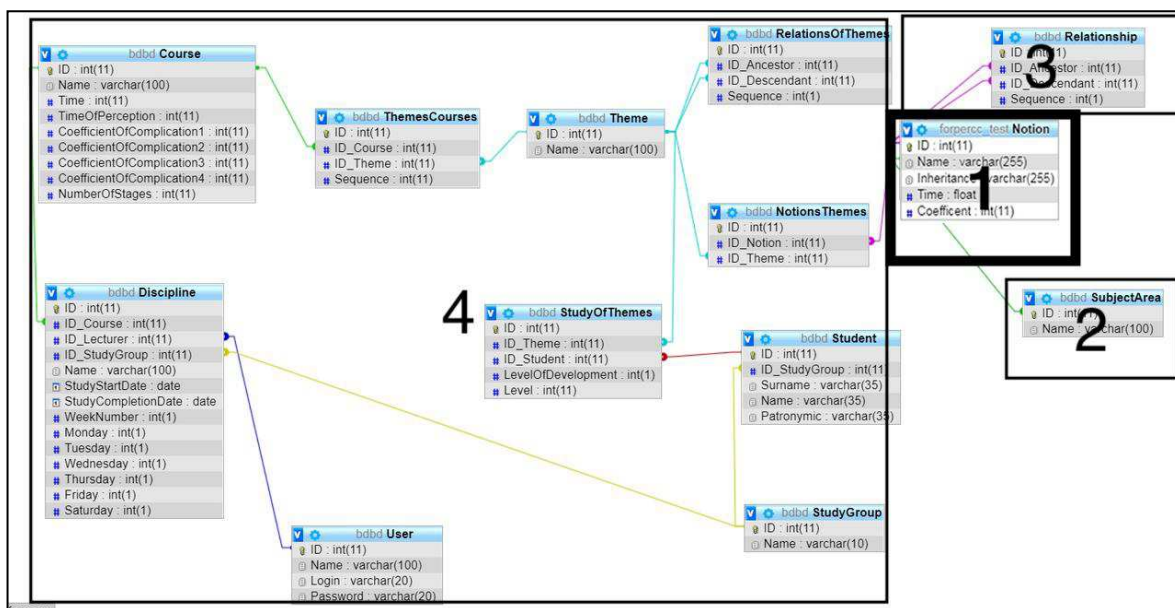


Рисунок 18 — Схема базы данных

Таблица под номером один является основной, так как она описывает предметную область необходимую для работы модулей. Она содержит такие поля как:

- «ID» является первичным ключом, тип данных – целый, значение не может быть равно нулю;
- «Name» содержит название . Тип данных – строка, значение не может быть равно нулю;
- «Inheritance» необходимо для указания наследственности при составлении csv файлов для визуализации;
- «Coeffcent» поле необходимое для работы модулей: формирования тематических разделов учебного плана, формирования индивидуального графика обучения, анализа успеваемости студентов.

Таблица под номером два содержит связанные поля с таблицей «Notion» необходимые для работы модуля визуализации, в свою очередь на основе данной таблице формируется CSV файл необходимый для построения графов.

Таблица под номер три содержит связанные поля с таблицей «Notion» необходимые для работы модуля формирования тематических разделов учебного курса на основе предметной области.

Таблицы под номер четыре содержат связанные поля с таблицей «Notion» необходимые для работы модуля формирования индивидуального графика обучения и анализа успеваемости студентов на основе предметной области.

Реализовав БД и добавив в нее необходимые таблицы, начинается процесс разработки редактора БД. Данный модуль построен на основе таких языков веб-программирования как HTML, PHP и JavaScript.

В первую очередь необходимо осуществить подключение к БД. Создав файл «DB.php» с последующим добавлением программного кода указанного в приложении Б.

После подключения к базе данных создадим основную страницу «index.php» разрабатываемого модуля, а также добавим файл «Stule.css» для описания внешнего вида, он необходим для визуальной составляющей и последующей доработке под общий стиль разрабатываемой платформы.

Для обработки и реагирования на действия пользователя редактора необходимо использовать язык программирования JavaScript. Чтобы добавить данные функциональные возможности требуется скачать библиотеку jQuery для возможности взаимодействия языков HTML и JavaScript. Создав файл «Script.js» добавим необходимые элементы взаимодействия пользователя и интерфейса, такие как поиск по понятиям, сортировки, добавление, удаление или редактирование БД. Программный код на языке JavaScript примененного к сайту указан в приложении Б.

Теперь необходимо осуществить обработку пользовательских действий на уровне сервера. Для этого необходимо создать необходимый набор файлов написанных на языке программирования PHP. Обозначим данные файлы с функциональными возможностями в виде перечисления (программный код содержится в Приложении Б):

- «Add.php» - запрос на добавление в таблицу «Notion» новых полей;
- «Drop.php» - запрос на удаление полей в таблице «Notion»;
- «Get.php» - запрос на отображение существующих полей в таблице «Notion»;

- «PageGet.php» - запрос осуществляющий передачу существующих полей в таблице «Notion» на странице «Index.html»;
- «Save.php» - запрос на сохранение добавленных изменений в поля таблицы «Notion»;
- «Search.php» - запрос на поиск необходимой информации по полям таблицы «Notion»;
- «CSV\_export.php» - запрос на формирование файла формата CSV для визуализации понятий предметной области.

На итоговой стадии проделанной работы необходимо добавить связь с вышеперечисленными файлами и их программным кодом для исполнения определенных действий при работе на главной странице сайта «Index.php» (Приложение Б). Также требуется прописать стили к элементам для корректного отображения в файле «Stule.css» (Приложение Б).

Итоговый вариант главной страницы разрабатываемого программного модуля содержит все необходимые элементы для работы с БД (рисунки 19, 20). Созданы основные структурные элементы для работы с базой данных: навигации, создания, удаление, редактирования, поиска.

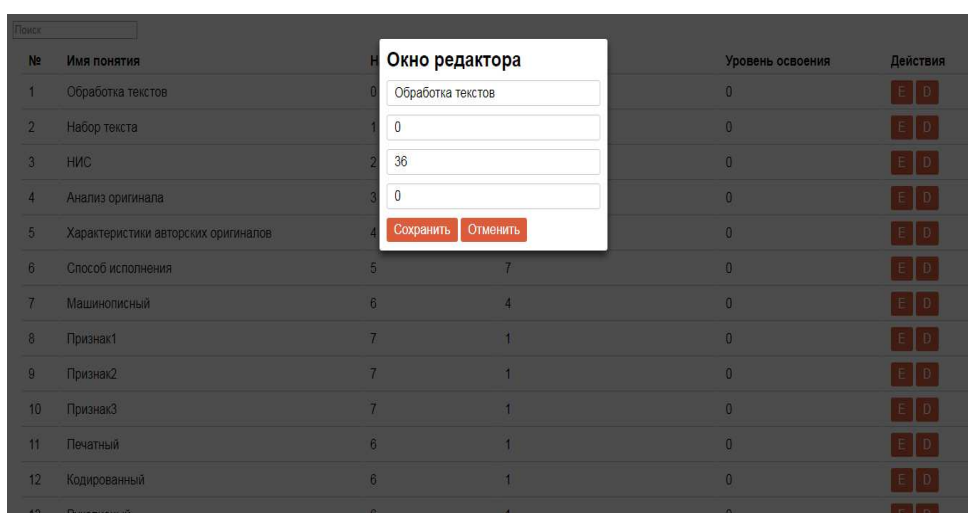


Рисунок 19 — Модальное окно редактора понятий предметной области

№	Имя понятия	Наследование	Время	Уровень освоения	Действия
1	Обработка текстов	0	36	0	E D
2	Набор текста	1	30	0	E D
3	НИС	2	24	0	E D
4	Анализ оригинала	3	10	0	E D
5	Характеристики авторских оригиналов	4	9	0	E D
6	Способ исполнения	5	7	0	E D
7	Машинописный	6	4	0	E D
8	Признак1	7	1	0	E D
9	Признак2	7	1	0	E D
10	Признак3	7	1	0	E D
11	Печатный	6	1	0	E D
12	Кодированный	6	1	0	E D
13	Рукописный	6	1	0	E D
14	Требования ОСТ	5	1	0	E D
			Итого времени: ( ) часов		

Добавить

Рисунок 20 — Интерфейс редактора БД

Разрабатываемый программный модуль полностью отвечает всем поставленным требованиям. Редактор БД выполнен в едином стиле и готов к внедрению в платформу учебно-методического обеспечения с помощью фреймов сайта. Модуль готов к эксплуатации и последующим доработкам с расширением функциональных возможностей.

### 3.3 Выводы по главе 3

В ходе выполнения поставленной задачи разработана платформа учебно-методического обеспечения. Она удовлетворяет всем заявленным требованиям, согласно техническому заданию. Платформа реализована с помощью средств CMS Wordpress и дополнительных модулей. База данных осуществлена с помощью СУБД Phpmyadmin. Редактор базы данных реализован на таких языках программирования как PHP и JavaScript.



## ЗАКЛЮЧЕНИЕ

В ходе выполнения бакалаврской работы проведен обзор теоретических данных для понимания контекста разработки веб-платформы. Были выделены следующие проблемы в сфере электронного обучения студентов:

– В настоящий момент, рассмотрев существующие электронные обучающие системы, можно сделать вывод, что большинство систем образовательный процесс направлен на общие массы обучающихся и существует проблема индивидуального подхода обучения, учитывая индивидуальные особенности обучающихся.

Для того, чтобы решить выявленную проблему, необходимо разработать веб-платформу учебно-методического обеспечения, а также редактор базы данных для работы с понятийным аппаратом изучаемой учебной дисциплины.

Проанализировав сформулированные варианты использования, был сделан вывод, что с точки зрения потенциальных рисков и архитектурной значимости наиболее существенным прецедентом является «Работа с БД» принадлежащий актору «Эксперт».

Для проектирования модуля редактора БД предметной области учебной дисциплины использовались UML-диаграммы.

Диаграммы отображают, каким образом будет работать и разрабатываемый модуль.

Также, была проведена классификация пользователей платформы, рассмотрено взаимодействие ее компонент и схема работы, описана концептуальная схема разрабатываемого сайта, были выбраны средства для разработки платформы.

В третьей главе представлена реализация платформы учебно-методического обеспечения и редактора БД для работы с предметной областью.

Платформа реализована с помощью средств CMS Wordpress и дополнительных модулей. База данных осуществлена с помощью СУБД

Phpmyadmin. Редактор базы данных реализован на таких языках программирования как PHP и JavaScript.

Разрабатываемая платформа полностью отвечает всем поставленным требованиям, готова к эксплуатации и последующим доработкам с расширением функциональных возможностей.

## **СПИСОК СОКРАЩЕНИЙ**

ЭОР – электронные образовательные ресурсы.

ИКТ – информационно-коммуникационных технологий.

КСО – компьютерное средство обучения.

ООП – основные образовательные программы.

БД – база данных.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Перфильев, Д. А. Междисциплинарный курсовой проект [Электронный ресурс] / Д. А. Перфильев, К. В. Раевич. - Красноярск : СФУ, 2016. - 207 с. – Режим доступа [https://e.sfu-kras.ru/pluginfile.php/883706/mod\\_resource](https://e.sfu-kras.ru/pluginfile.php/883706/mod_resource)
2. ГОСТ 34.602-89. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. – 6 с.
3. СТО 4.2–22–2009 Система менеджмента качества. Организация учета и хранения документов. – Введ. 22.12.2009. – Красноярск : ИПК СФУ, 2009. – 41 с.
4. Agile СОСОМО II [Электронный ресурс]: Техническая документация модели СОСОМО II – Режим доступа: <http://csse.usc.edu/AgileСОСОМО>.
5. Бюллетень лаборатории математического, естественнонаучного образования и информатизации: рецензируемый сборник научных трудов. Москва: МГПУ, 2012. – 391 с.
6. Роберт И. В. Теория и методика информатизации образования (психолого-педагогический и технологический аспекты). 2-е изд., доп. М.: ИИО РАО, 2008. 274 с.
7. Сатунина, А. Е. Электронное обучение: плюсы и минусы / А.Е. Ступина // Современные проблемы науки и образования. 2006. – № 1 – С. 89-90.
8. Башмаков, И. А. Разработка компьютерных учебников и обучающих систем. / И.А. Башмаков // Москва.: Информационно-издательский дом «Филинь», 2003. – 616 с.
9. Токтарова, В. И. Модель учебно-методического обеспечения образовательного процесса вуза в контексте электронного обучения / В.И. Токарева / Вестник Марийского государственного университета. – 2013. – С. 87-90.
10. Фролов, Ю. В., Махотин, Д. А. Компетентностная модель как основа оценки качества подготовки специалистов // Высшее образование сегодня. 2004. № 8. – С. 34–41.

11. Токтарова, В.И. Технология создания и применения электронных учебно-методических комплексов в образовательном процессе высшей школы // Вестник Московского городского педагогического университета. Сер.: Информати-ка и информатизация образования. 2008. № 11. – С. 355-357.

12. О внесении изменений в Закон Российской Федерации «Об образовании» в части применения электронного обучения, дистанционных образовательных технологий»: ферр.закон Российской Федерации от 28.02.2012 г № 11 – ФЗ // Российская газета – 2012. – 2 мар.

13. Диаграмма прецедентов [Электронный ресурс] – Режим доступа: [https://ru.wikipedia.org/wiki/Диаграмма\\_прецедентов](https://ru.wikipedia.org/wiki/Диаграмма_прецедентов)

14. Диаграмма деятельности [Электронный ресурс] – Режим доступа: [https://ru.wikipedia.org/wiki/Диаграмма\\_деятельности](https://ru.wikipedia.org/wiki/Диаграмма_деятельности).

15. Диаграмма классов [Электронный ресурс] – Режим доступа: [https://ru.wikipedia.org/wiki/Диаграмма\\_классов](https://ru.wikipedia.org/wiki/Диаграмма_классов).

16. Диаграмма последовательности [Электронный ресурс] - Режим доступа: [https://ru.wikipedia.org/wiki/Диаграмма\\_последовательности](https://ru.wikipedia.org/wiki/Диаграмма_последовательности)

17. Диаграмма компонентов [Электронный ресурс] – Режим доступа: [https://ru.wikipedia.org/wiki/Диаграмма\\_компонентов](https://ru.wikipedia.org/wiki/Диаграмма_компонентов).

18. Наумов, И. С., Выхованец, В. С. Оценка трудности и сложности учебных задач на основе синтаксического анализа текстов // Информационные технологии в управлении. Управление большими системами. 2014. № 48. С. 97–131.

19. Статья «Диаграмма развертывания» [Электронный ресурс] – Режим доступа: [https://ru.wikipedia.org/wiki/Диаграмма\\_развертывания](https://ru.wikipedia.org/wiki/Диаграмма_развертывания).

## ПРИЛОЖЕНИЕ А

### Техническое задание

Настоящее техническое задание, оформленное в соответствии с ГОСТ //19.201- 78//сто содержит назначения и цели, требования к системе, этапы её разработки.

Общее описание платформы

Наименование: Веб-платформа учебно-методического обеспечения

Разработчик: Приловский С.А.

Заказчик: Кафедра систем искусственного интеллекта.

Начало: 30 апреля 2018

Окончание: 27 мая 2018

Назначение и цели системы

Система направлена на автоматизацию процесса обучения посредством составления тематических разделов, планового графика деятельности обучаемого в освоении учебной дисциплины, адаптивного управления деятельностью обучаемого при реализации учебного процесса.

Целью данного проекта является разработка автоматизированной информационной системы для описания предметной области учебной дисциплины и формирования на ее основе тематических разделов учебного плана, индивидуального графика обучения, анализа успеваемости студентов.

Так же для отражения содержания каждого модуля в учебной программе, его объем и логическую сложность необходимо разработать модуль визуализации понятий предметной области с помощью графовой модели представления.

Требования к системе

Пользователям доступны все функций платформы только после регистрации и подтверждения администратором учетной записи пользователя. Сайт должен иметь простой и понятный интерфейс для пользователей. В систему требуется внедрить модули для работы с учебным курсом. Необхо-

димо предоставить возможность доработки или изменения системы в соответствии с требованиями заказчика.

*Требования к функциям (задачам), выполняемым системой.* Сайт должен состоять из следующих основных элементов:

- редактор базы данных для работы с предметной областью;
- модуль визуализации понятий предметной области;
- модуль формирования тематических разделов;
- модуль индивидуального графика обучения.

Пользовательский интерфейс сайта должен обеспечивать наглядное, легко понятное отображение структуры информации размещенной на нем, быстрый и логичный переход к разделам и страницам. Навигационные элементы должны обеспечивать однозначное понимание пользователем их смысла: ссылки на страницы должны быть снабжены заголовками, условные обозначения соответствовать общепринятым. Графические элементы навигации должны быть с дополнительными подписями.

Система должна обеспечивать навигацию по всем доступным пользователю ресурсам и отображать соответствующую информацию.

При выборе какого-либо из пунктов меню пользователем должна загружаться соответствующая ему информационная страница.

*Требования к персоналу.* Для эксплуатации сайта определены следующие роли:

- Администратор: выполняет функции по поддержанию работоспособности сайта;
- Эксперт: работа с основными функциями и модулями сайта.

*Показатели назначения.* Сайт должен обеспечивать бесперебойную одновременную работу нескольких экспертов в системе. Своевременный отклик о выполнении задач системы, не более 1,5 секунд.

*Требования к надежности.* Доступность – время, затрачиваемое на восстановления системы после отказа не должно превышать 1 часа.

Максимально допустимое количество ошибок или дефектов – 1 ошибка на 5000 строк кода.

*Требования к безопасности.* Безопасность информации сайта поддерживается администратором. Исходный код сайта доступен только разработчикам и программистам, обслуживающим сайт.

Для пользователей сайта исходный код является закрытым.

*Требования к эргономике и технической эстетике.* Взаимодействие пользователя и сайта осуществляется по средствам графического интерфейса. Интерфейс должен соответствовать требованиям выполняемых функций. Навигация по сайту должна быть представлена в удобном виде. Ввод и вывод данных сайта, выполнение команд и отображение результатов исполнения должны выполняться в интерактивном виде.

*Требования к эксплуатации и техническому обслуживанию ремонту и хранению.* Для эксплуатации сайта необходимо обеспечить безотказную работу сервера. Периодическое ТО должно проводиться в соответствии с прилагаемой документацией. При вводе системы в опытную эксплуатацию должна быть создана резервная копия программного обеспечения и обрабатываемой информации.

Размещение помещений и их оборудование должны исключать возможность бесконтрольного проникновения в них посторонних лиц и обеспечивать сохранность находящихся в этих помещениях конфиденциальных документов и технических средств.

Размещение оборудования, технических средств должно соответствовать требованиям техники безопасности, санитарным нормам и требованиям пожарной безопасности.

Квалификация персонала и его подготовка должны соответствовать технической документации.[2]

*Требования к лингвистическому обеспечению системы.* Все прикладное программное обеспечение системы для организации взаимодействия с пользователем должно использовать русский или английский языки.



*Требования к программному обеспечению системы.* Используемое при разработке программное обеспечение и библиотеки программных кодов должны иметь широкое распространение, быть общедоступными и использоваться в промышленных масштабах.

*Требования к информационной и программной совместимости.* Система должна функционировать на ПЭВМ с одной из операционных систем.

Для реализации интерфейса между системой и другими программами из состава ПО ПЭВМ должны использоваться средства буфера обмена операционной системы.

*Требования к техническому обеспечению.* В состав комплекса должны следующие технические средства:

- серверы БД;
- ПК администраторов;
- ПК пользователей.

Требования к техническим характеристикам серверов БД:

- память: 1 Гб;
- процессор: 2,0 ГГц и выше;
- тип процессора процессор x64: AMD Opteron, AMD Athlon 64, IntelXeon с поддержкой Intel EM64T, IntelPentium IV.

Требования к техническим характеристикам ПК администратора, диспетчера:

- процессор – Intel I3;
- объем оперативной памяти – 4 ГБ;
- дисковая подсистема – 100 Гб;
- сетевой адаптер – 100 Мбит;
- Windows 7 и выше.

Требования к техническим характеристикам ПК пользователей:

- процессор – Intel Celeron;
- объем оперативной памяти – 1 ГБ;

- сетевой адаптер – 1 Мбит;
- Windows 7 и выше.

*Требования к организационному обеспечению.* Организационное обеспечение системы должно быть достаточным для эффективного выполнения персоналом возложенных на него обязанностей при осуществлении автоматизированных и связанных с ними неавтоматизированных функций системы.

К работе с системой должны допускаться сотрудники, имеющие навыки работы на персональном компьютере, ознакомленные с правилами эксплуатации и прошедшие обучение работе с системой.

*Требования к методическому обеспечению.* В состав методического обеспечения системы входят следующие документы:

- пояснительная записка, оформленная в соответствии с СТО 4.2–07–2014 [3];
- справка для пользователей сайта.

Состав и описание работ по созданию системы

1. Обоснование необходимости разработки программы:

- постановка задачи;
- сбор исходных материалов;
- обоснование необходимости проводимости работ по разработке программного продукта.

2. Научно-исследовательские работы:

- предварительный выбор методов решения задач;
- определение требований к техническим средствам;
- определение принципиальной возможности решения поставленной задачи.

3. Разработка и утверждение технического задания:

- определение требований к системе;
- определение стадий, этапов и сроков разработки платформы и документации для нее;

- выбор языков программирования;
- согласование и утверждение технического задания.

#### 4. Разработка технического проекта:

- разработка алгоритма решения задачи;
- разработка структуры программы;
- окончательное определение конфигураций технических средств.

#### 5. Разработка веб-платформы:

- программирование и отладка платформы.

#### Порядок внесения изменений в техническое задание

Настоящее техническое задание может уточняться и дополняться в процессе создания технического регламента совместными решениями сторон, оформленными в виде дополнения к ТЗ в соответствии с требованиями ГОСТ 19.201-78.

## ПРИЛОЖЕНИЕ Б

### Исходный код редактора БД

#### Index.php

```
<!DOCTYPE html>
<html>
<head>
  <title>Test DB</title>
  <link rel='stylesheet' type='text/css' href='css/style.css' />
  <script type='text/javascript' src='js/jquery-3.3.1.min.js'></script>
  <script type='text/javascript' src='js/Script.js'></script>
  <meta charset="utf-8">
</head>
<body>
  <div class='modal'>
    <div class='window'>
      <h2>Окно редактора</h2>
      <input name='Name' class='Name' placeholder='Имя
понятия'/>
      <input name='Inheritance' class='Inheritance' placehold-
er='Наследование'/>
      <input name='Time' class='Time' placeholder='Время'/>
      <input name='Coefficent' class='Coefficent' placehold-
er='Уровень освоения'/>
      <button class='save'
onclick='Modal.Save();'>Сохранить</button>
      <button onclick='Modal.Cancel();'>Отменить</button>
    </div>
  </div>
```

```

        <input class='search' placeholder='Поиск' type='search'
oninput='Page.Search();'/>
    <table>
        <tr class='bold'>
            <td><span>№</span></td>
            <td
                onclick='Page.Sort(1);'><span>Имя
понятия</span></td>
            <td><span>Наследование</span></td>
            <td><span>Время</span></td>
            <td><span>Уровень освоения</span></td>
            <td><span>Действия</span></td>
        </tr>
        <?php
include("php/DB.php");

$SQL = $pdo->prepare("SELECT * FROM Notion");
$SQL -> execute();
$DBS = $SQL -> fetchAll();
$AllTime = 0;
foreach ($DBS as $DB){
    echo "
    <tr>
        <td><span>".$DB['ID']."</span></td>
        <td><span>".$DB['Name']."</span></td>
        <td><span>".$DB['Inheritance']."</span></td>
        <td><span>".$DB['Time']."</span></td>
        <td><span>".$DB['Coefficent']."</span></td>
        <td>
            <button
onclick='Table.Edit("".$DB['ID']."');>E</button>

```

```

        <button
onclick="Table.Drop(".$DB['ID'].");">D</button>
    </td>
</tr>
";
$AllTime = $AllTime + $DB['Time'];
}
?>
<tr>
    <td></td>
    <td></td>
    <td></td>
    <td><span>Итого времени: (<?php echo $AllTime; ?>)
часов</span></td>
    <td></td>
    <td></td>
</tr>
</table>
<button onclick="Table.Add();">Добавить</button>
</body>
</html>

```

### Stule.css

```

html{font-family:-Apple-font, sans-serif;}
table {margin: 10px; width:100%;}
table .bold span{font-weight:bold;}
table .bold td{cursor:pointer;}
table tr td{padding:5px 10px; border-bottom:1px solid #ccc;}
button{outline:none; cursor:pointer; padding:5px 10px; border-
radius:3px; background:#dc5d3c; color:#fff; border:0; font-size:100%;}

```

```

        .modal{display:none; background:rgba(0,0,0,0.7); position:fixed;
left:0; top:0; width:100%; height:100%;}

        .modal .window{background:#fff; padding:10px; border-radius:3px;
box-shadow:0 5px 15px #171719; width:300px; margin:5vh auto;}

        .modal .window h2{margin:0; padding:0; padding-bottom:10px;}

        .modal .window input{width:calc(100% - 22px); padding:5px 10px;
border:1px solid #ccc; font-size:100%; border-radius:3px; margin-bottom:10px;}

```

### Script.js

```

var Sorter = true;

var Page = {
    Get: function(){
        $.get("php/PageGet.php", function(data){
            $("table").html(data);
        });
    },
    Sort: function(ID){
        if(Sorter){
            ID = ID+"a";
            Sorter = !Sorter;
        }
        else{
            ID = ID+"d";
            Sorter = !Sorter;
        }
        $.get("php/PageGet.php?sort="+ID, function(data){
            $("table").html(data);
        });
    },
    Search: function(){
        var Searcher = $(".search").val();

```

```

        if(Searcher != ""){
            $.post("php/Search.php",    {Searcher:Searcher},
function(data){
                $("table").html(data);
            });
        }
        else{
            Page.Get();
        }
    }
};
var Table = {
    Add: function(){
        $(".modal").fadeIn(500);
        $(".modal .window .save").attr("onclick", "Mod-
al.Add()");
    },
    Edit: function(id){
        $(".modal").fadeIn(500);
        $.getJSON("php/Get.php?id="+id, function(data){
            $(".modal .window .Name").val(data.Name);
            $(".modal .window
.Inheritance").val(data.Inheritance);
            $(".modal .window .Time").val(data.Time);
            $(".modal .window
.Coefficient").val(data.Coefficient);
            $(".modal .window .save").attr("onclick", "Mod-
al.Save("+id+"");
        });
    },

```



```

Drop: function(ID){
    if(confirm("Действительно удалить?")){
        $.post("php/Drop.php", {ID:ID}, function(data){
            Page.Get();
        });
    }
}

};

var Modal = {
    Add: function(){
        var Name = $(".modal .window .Name").val();
        var Inheritance = $(".modal .window .Inheritance").val();
        var Time = $(".modal .window .Time").val();
        var Coefficient = $(".modal .window .Coefficient").val();
        $.post("php/Add.php", {Name:Name, Inher-
itance:Inheritance, Time:Time, Coefficient:Coefficient}, function(data){
            Modal.Cancel();
            Page.Get();
        });
    },
    Save: function(ID){
        if(ID != ""){
            var Name = $(".modal .window .Name").val();
            var Inheritance = $(".modal .window
.Inheritance").val();

            var Time = $(".modal .window .Time").val();
            var Coefficient = $(".modal .window
.Coefficient").val();

            $.post("php/Save.php", {ID:ID, Name:Name, In-
heritance:Inheritance, Time:Time, Coefficient:Coefficient}, function(data){

```

```

        Modal.Cancel();
        Page.Get();
    });
}
},
Cancel: function(){
    $(".modal").fadeOut(500);
    $(".modal .window input").val("");
}
};

```

### **DB.php**

```

<?php
    $pdo = new PDO('mysql:host=localhost;dbname=forpercc_test',
"forpercc_test", "123456", array(
        PDO::ATTR_PERSISTENT => true
    ));
?>

```

### **Add.php**

```

<?php
    include("DB.php");
    $Name = htmlspecialchars(trim($_POST['Name']));
    $Inheritance = htmlspecialchars(trim($_POST['Inheritance']));
    $Time = htmlspecialchars(trim($_POST['Time']));
    $Coefficient = htmlspecialchars(trim($_POST['Coefficient']));
    $SQL = "INSERT INTO Notion (Name, Inheritance, Time,
Coefficient) VALUES (:Name, :Inheritance, :Time, :Coefficient)";
    $DB = $pdo->prepare($SQL);
    $DB->bindValue(':Name', $Name);
    $DB->bindValue(':Inheritance', $Inheritance);
    $DB->bindValue(':Time', $Time);

```

```
$DB->bindValue(':Coefficient', $Coefficient);  
$DB->execute();
```

```
?>
```

### **Drop.php**

```
<?php
```

```
include("DB.php");  
$ID = htmlspecialchars(trim($_POST['ID']));  
$SQL = "DELETE FROM Notion WHERE ID = :ID";  
$DB = $pdo->prepare($SQL);  
$DB->bindValue(':ID', $ID);  
$DB->execute();
```

```
?>
```

### **Get.php**

```
<?php
```

```
include("DB.php");  
if($_GET['id'] != null){  
    $SQL = $pdo->prepare("SELECT * FROM Notion WHERE ID  
= :id");  
    $SQL->bindvalue(":id", $_GET['id']);  
    $SQL -> execute();  
    $DB = $SQL -> fetch(PDO::FETCH_LAZY);  
    if($DB['ID'] == $_GET['id']){  
        $ARR = array(  
            "Name" => $DB['Name'],  
            "Inheritance" => $DB['Inheritance'],  
            "Time" => $DB['Time'],  
            "Coefficient" => $DB['Coefficient'],  
        );  
        echo json_encode($ARR);  
    }  
}
```

```

        else{
            echo "error1";
        }
    }
    else{
        echo "error2";
    }
?>

```

### PageGet.php

```

<?php
include("DB.php");
echo "<tr class='bold'>
        <td><span>ID</span></td>
        <td
onclick='Page.Sort(1);'><span>Name</span></td>
        <td><span>Inheritance</span></td>
        <td><span>Time</span></td>
        <td><span>Coefficent</span></td>
        <td><span>Действия</span></td>
    </tr>";
$ID = $_GET['sort'];
if($ID == null){
    $SQL = $pdo->prepare("SELECT * FROM Notion");
    $SQL -> execute();
    $DBS = $SQL -> fetchAll();
    $AllTime = 0;
    foreach ($DBS as $DB){
        echo "
        <tr>
            <td><span>".$DB['ID']."</span></td>

```

```

        <td><span>".$DB['Name']."</span></td>
        <td><span>".$DB['Inheritance']."</span></td>
        <td><span>".$DB['Time']."</span></td>
        <td><span>".$DB['Coefficient']."</span></td>
        <td>
                <button
onclick='Table.Edit(".$DB['ID'].");'>E</button>
                <button
onclick='Table.Drop(".$DB['ID'].");'>D</button>
        </td>
</tr>
";
$AllTime = $AllTime + $DB['Time'];
}
echo "
        <tr>
                <td></td>
                <td></td>
                <td></td>
                <td><span>Итого времени: ($AllTime)
часов</span></td>
                <td></td>
                <td></td>
        </tr>";
}
else{
        switch($ID){
                case "1a":
                        $SQL = $pdo->prepare("SELECT * FROM Notion
ORDER BY Name ASC");

```

```

$SQL -> execute();
$DBS = $SQL -> fetchAll();
$AllTime = 0;
foreach ($DBS as $DB){
    echo "
    <tr>
        <td><span>".$DB['ID']."</span></td>

<td><span>".$DB['Name']."</span></td>

<td><span>".$DB['Inheritance']."</span></td>

<td><span>".$DB['Time']."</span></td>

<td><span>".$DB['Coefficient']."</span></td>
        <td>
            <button
onclick='Table.Edit(".$DB['ID'].");'>E</button>
            <button
onclick='Table.Drop(".$DB['ID'].");'>D</button>
        </td>
    </tr>
    ";
    $AllTime = $AllTime + $DB['Time'];
}
echo "
    <tr>
        <td></td>
        <td></td>
        <td></td>

```

```

                                <td><span>Итого
                                времени:
($AllTime) часов</span></td>

                                <td></td>
                                <td></td>
                                </tr>";
break;
case "1d":
    $SQL = $pdo->prepare("SELECT * FROM Notion
ORDER BY Name DESC");
    $SQL -> execute();
    $DBS = $SQL -> fetchAll();
    $AllTime = 0;
    foreach ($DBS as $DB){
        echo "
        <tr>
                                <td><span>".$DB['ID']."</span></td>

                                <td><span>".$DB['Name']."</span></td>

                                <td><span>".$DB['Inheritance']."</span></td>

                                <td><span>".$DB['Time']."</span></td>

                                <td><span>".$DB['Coefficient']."</span></td>
                                <td>
                                <button
onclick='Table.Edit(".$DB['ID'].");'>E</button>
                                <button
onclick='Table.Drop(".$DB['ID'].");'>D</button>
                                </td>

```

```

        </tr>
        ";
        $AllTime = $AllTime + $DB['Time'];
    }
    echo "
        <tr>
            <td></td>
            <td></td>
            <td></td>
            <td><span>Итого          времени:
($AllTime) часов</span></td>
            <td></td>
            <td></td>
        </tr>";
        break;
    }
}
?>

```

### Save.php

```

<?php
    include("DB.php");
    $ID = htmlspecialchars(trim($_POST['ID']));
    $Name = htmlspecialchars(trim($_POST['Name']));
    $Inheritance = htmlspecialchars(trim($_POST['Inheritance']));
    $Time = htmlspecialchars(trim($_POST['Time']));
    $Coefficient = htmlspecialchars(trim($_POST['Coefficient']));
    $SQL = "UPDATE Notion SET Name = :Name, Inheritance =
:Inheritance, Time = :Time, Coefficient = :Coefficient WHERE ID = :id";
    $DB = $pdo->prepare($SQL);
    $DB->bindValue(':id', $ID);

```



```

$DB->bindValue(':Name', $Name);
$DB->bindValue(':Inheritance', $Inheritance);
$DB->bindValue(':Time', $Time);
$DB->bindValue(':Coefficient', $Coefficient);
$DB->execute();

```

?>

## Search.php

```

<?php
include("DB.php");

echo "<tr class='bold'>
        <td><span>ID</span></td>
        <td
onclick='Page.Sort(1);'><span>Name</span></td>
        <td><span>Inheritance</span></td>
        <td><span>Time</span></td>
        <td><span>Coefficient</span></td>
        <td><span>Действия</span></td>
</tr>";

$Searcher = $_POST['Searcher'];
if($Searcher != null){
    $SQL = $pdo->prepare("SELECT * FROM Notion WHERE
Name LIKE :search");
    $SQL->bindValue(':search', "%".$Searcher."%");
    $SQL -> execute();
    $DBS = $SQL -> fetchAll();
    $Count = 0;
    $AllTime = 0;
    foreach ($DBS as $DB){
        echo "

```

```

        <tr>
            <td><span>".$DB['ID']."</span></td>
            <td><span>".$DB['Name']."</span></td>
            <td><span>".$DB['Inheritance']."</span></td>
            <td><span>".$DB['Time']."</span></td>
            <td><span>".$DB['Coefficient']."</span></td>
            <td>
                <button
onclick="Table.Edit('".$DB['ID']."');">E</button>
                <button
onclick="Table.Drop('".$DB['ID']."');">D</button>
            </td>
        </tr>
";
    $Count++;
    $AllTime = $AllTime + $DB['Time'];
}
if($Count == 0){
    echo "<tr><td colspan='8'><span>Ничего не найдено
:(</span></td></tr>";
}
else{
    echo "
    <tr>
        <td></td>
        <td></td>
        <td></td>
        <td><span>Итого времени: ($AllTime)
часов</span></td>
        <td></td>

```

```

                <td></td>
            </tr>";
        }
    }
?>

```

### **CSV\_export.php**

```

<?php
    $mysqli = new mysqli("localhost", "forpercc_test", "123456",
"forpercc_test");

    if (mysqli_connect_errno()) {
        printf("Ошибка подключения: %s\n", mysqli_connect_error());
        exit();
    }

    $chek = 0;
    $invert = 0;
    $temp = 1;
    $poisk = 1;
    $arr = array();
    $ID = array();
    $Name = array();
    $Time = array();
    $Inheritance = array();
    $stroka = "";
    $csv_file = "";
    $csv_file .= 'id,value'. "\r\n";
    mysqli_query($mysqli, "SET NAMES CP1251");
    $query = "SELECT ID, Name, Time, Inheritance FROM Notion";
    $result = mysqli_query($mysqli,$query);
    while ($row = mysqli_fetch_assoc($result))

```

```

{ $ID[$temp]=$row["ID"];
  $Name[$temp]=$row["Name"];
  $Time[$temp]=$row["Time"];
  $Inheritance[$temp]=$row["Inheritance"];
  $temp++;
}
for($i=1;$i<count($Time);$i++)
{
  $Time[$i] = str_replace(".", "", $Time[$i]);
}
for($i=1;$i<$temp;$i++)
{
  $schek = 0;
  $invert = $Inheritance[$i];
  $arr[$i] = ".$Name[$i].('$Time[$i].)';
  while($invert>0)
  {
    while($invert!=$ID[$poisk])
    {
      $poisk++;
    }
    $arr[$invert]=".$Name[$poisk].('$Time[$poisk].)';
    $invert = $Inheritance[$poisk];
    $poisk = 1;
  }
  for ($j = 0 ; $j <= 100 ; $j++)
  {
    if (isset($arr[$j]))
    {
      if($schek==1)

```

```
        {
            $csv_file .= '!.$arr[$j].";
        }
    else
    {
        $csv_file .= ".$arr[$j].";
    }
    $chek = 1;
}
}
$arr=array();
$csv_file .= "\r\n";
}
$file_name = "subjectarea.csv";
$file = fopen($file_name,"w");
$csv_file = iconv('windows-1251',"UTF-8",$csv_file);
fwrite($file,trim($csv_file));
fclose($file);
?>
```

## ПРИЛОЖЕНИЕ В

### Презентационный материал

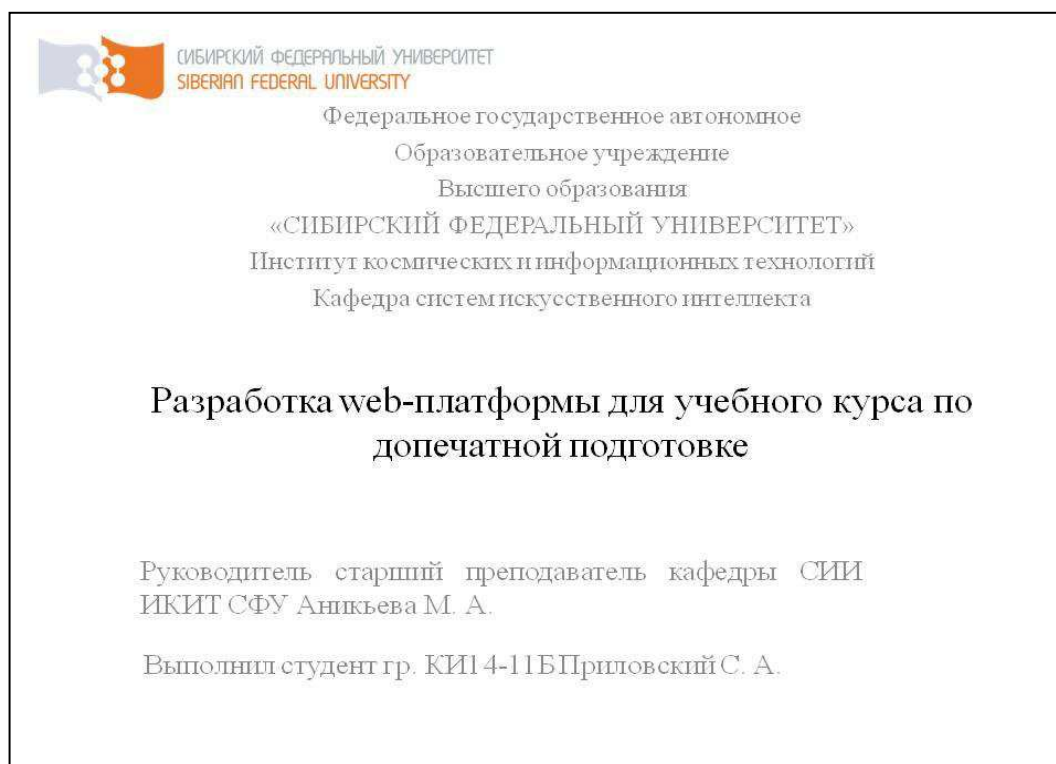


Рисунок В.1 — Слайд презентации №1

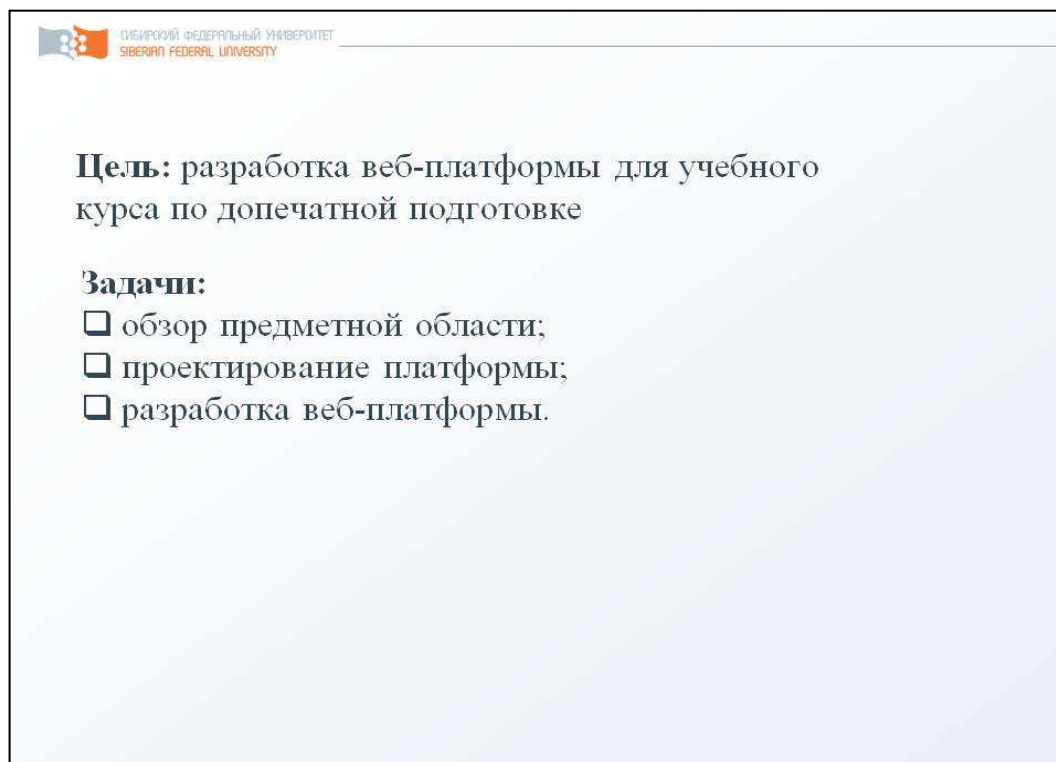


Рисунок В.2 — Слайд презентации №2

## Функциональные требования

**Платформа должна состоять из следующих основных элементов:**

- редактор базы данных для работы с предметной областью;
- модуль визуализации понятий предметной области;
- модуль формирования тематических разделов;
- модуль формирования индивидуального графика обучения.

**Дополнительные требования:**

- администрирование платформы;
- безопасность платформы;
- возможность доработки платформы.

Рисунок В.3 — Слайд презентации №3

## Средства разработки платформы



Javascript



PHP



phpMyAdmin



HTML



CMS Wordpress



CSS

Рисунок В.4 — Слайд презентации №4

### Схема работы и взаимодействие компонент разрабатываемой системы



Рисунок В.5 — Слайд презентации №5

### ER-диаграмма

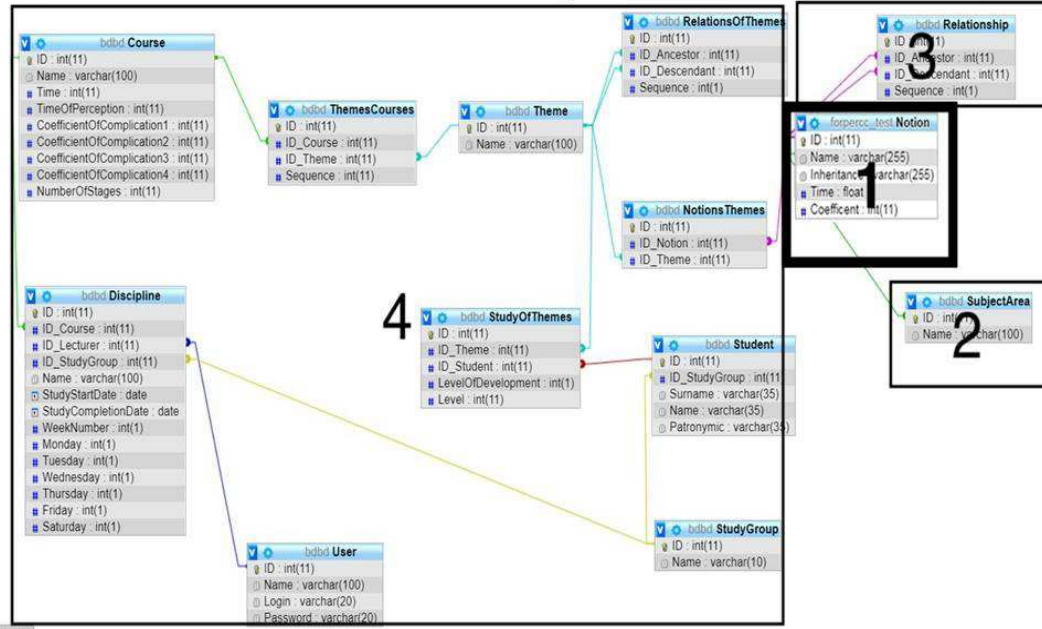


Рисунок В.6 — Слайд презентации №6



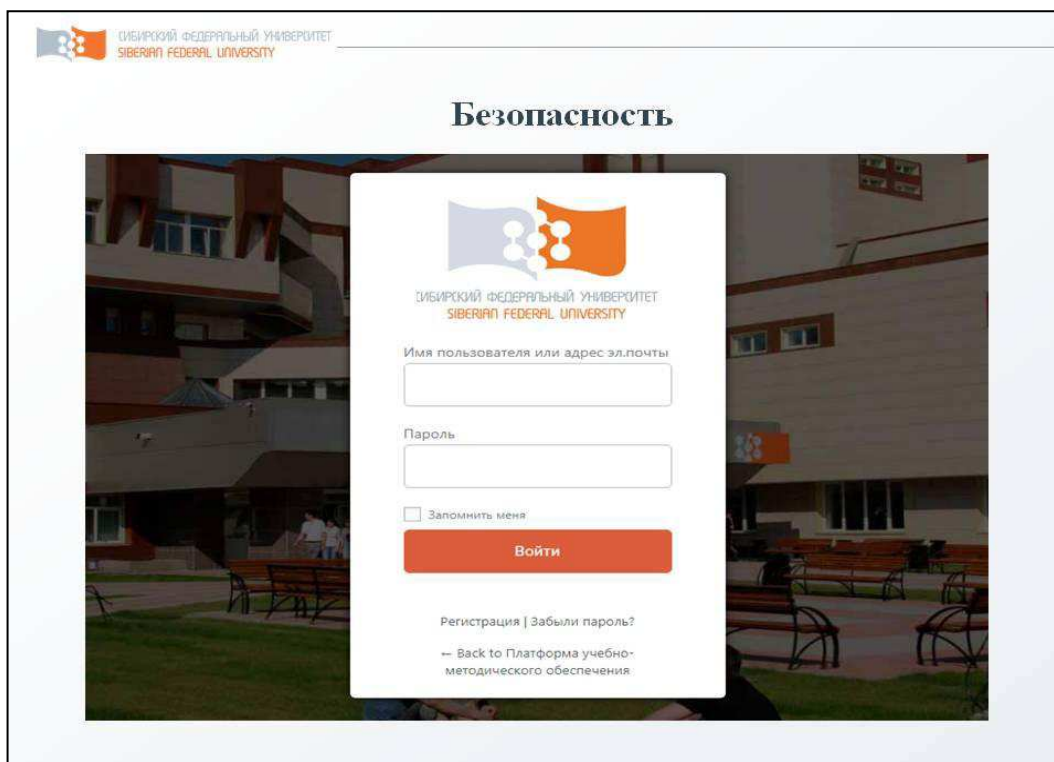


Рисунок В.7— Слайд презентации №7

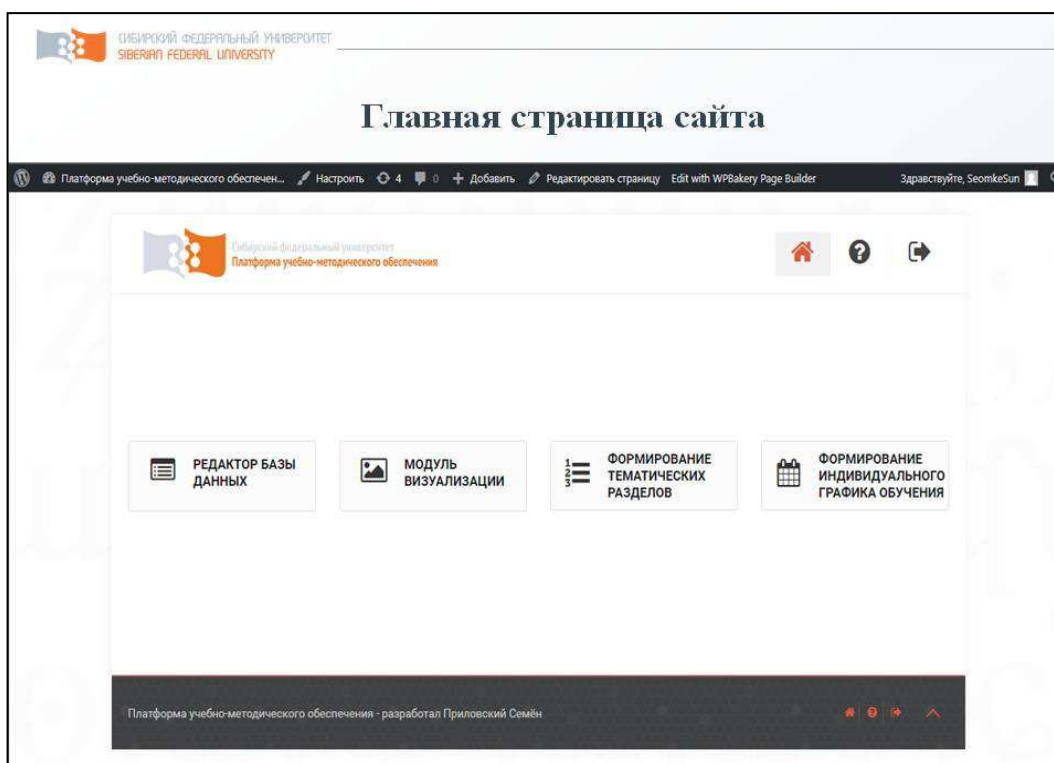


Рисунок В.8 — Слайд презентации №8

СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ  
SIBERIAN FEDERAL UNIVERSITY

## Редактор БД

Поиск

№	Имя понятия	Наследование	Время	Уровень освоения	Действия
1	Обработка текстов	0	36	0	E D
2	Набор текста	1	30	0	E D
3	НИС	2	24	0	E D
4	Анализ оригинала	3	10	0	E D
5	Характеристики авторских оригиналов	4	9	0	E D
6	Способ исполнения	5	7	0	E D
7	Машинописный	6	4	0	E D
8	Признак1	7	1	0	E D
9	Признак2	7	1	0	E D
10	Признак3	7	1	0	E D
11	Печатный	6	1	0	E D
12	Кодированный	6	1	0	E D
13	Рукописный	6	1	0	E D
14	Требования ОСТ	5	1	0	E D

Итого времени ( ) часов

[Добавить](#)

Рисунок В.9 — Слайд презентации №9

СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ  
SIBERIAN FEDERAL UNIVERSITY

## Редактор БД

Поиск

№	Имя понятия	Наследование	Время	Уровень освоения	Действия
1	Обработка текстов	0	36	0	E D
2	Набор текста	1	30	0	E D
3	НИС	2	24	0	E D
4	Анализ оригинала	3	10	0	E D
5	Характеристики авторских оригиналов	4	9	0	E D
6	Способ исполнения	5	7	0	E D
7	Машинописный	6	4	0	E D
8	Признак1	7	1	0	E D
9	Признак2	7	1	0	E D
10	Признак3	7	1	0	E D
11	Печатный	6	1	0	E D
12	Кодированный	6	1	0	E D
13	Рукописный	6	1	0	E D

**Окно редактора**

Обработка текстов

0

36

0

[Сохранить](#) [Отменить](#)

Рисунок В.10 — Слайд презентации №10

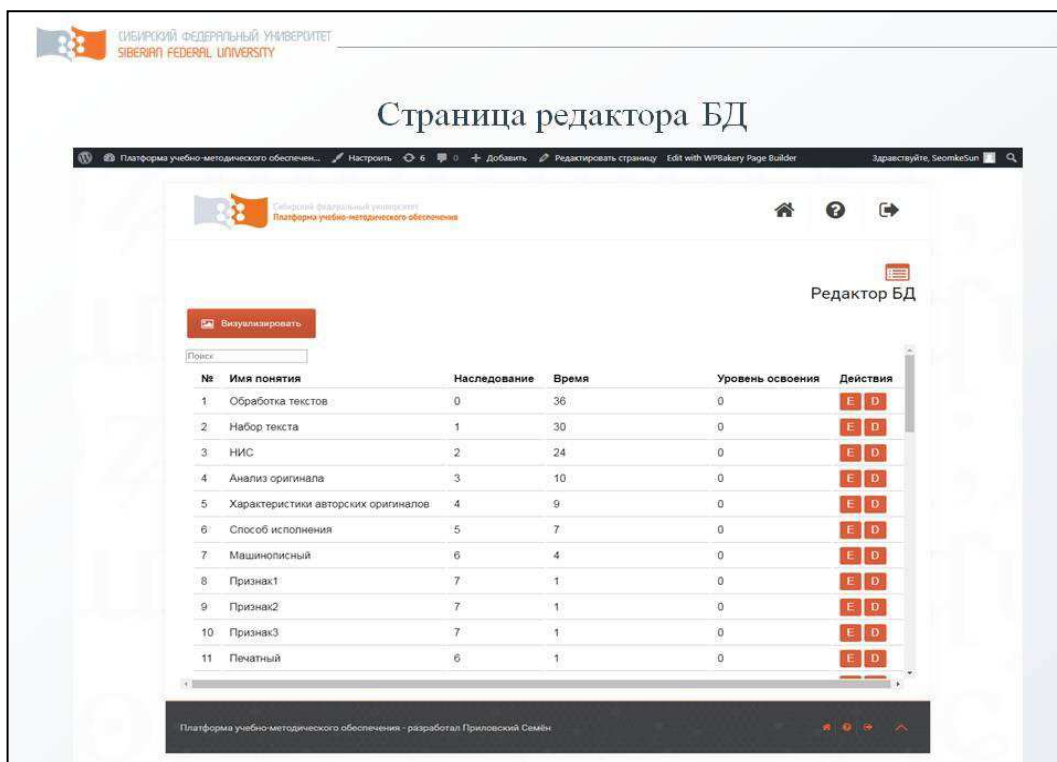


Рисунок В.11 — Слайд презентации №11

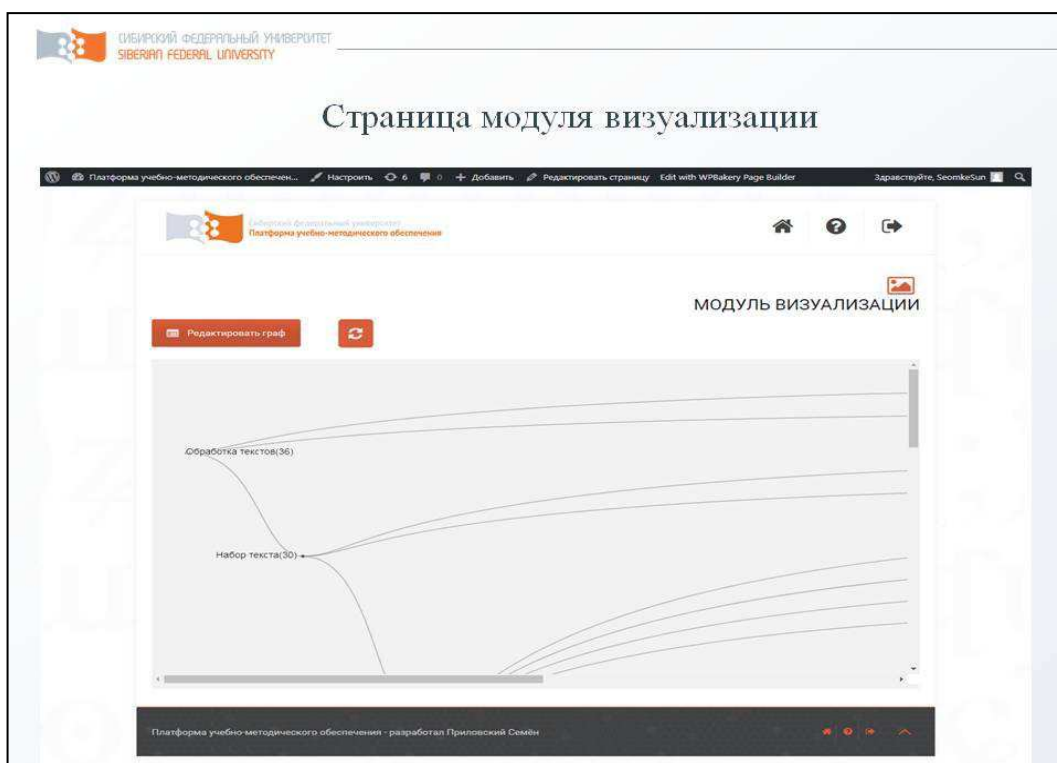


Рисунок В.12 — Слайд презентации №12

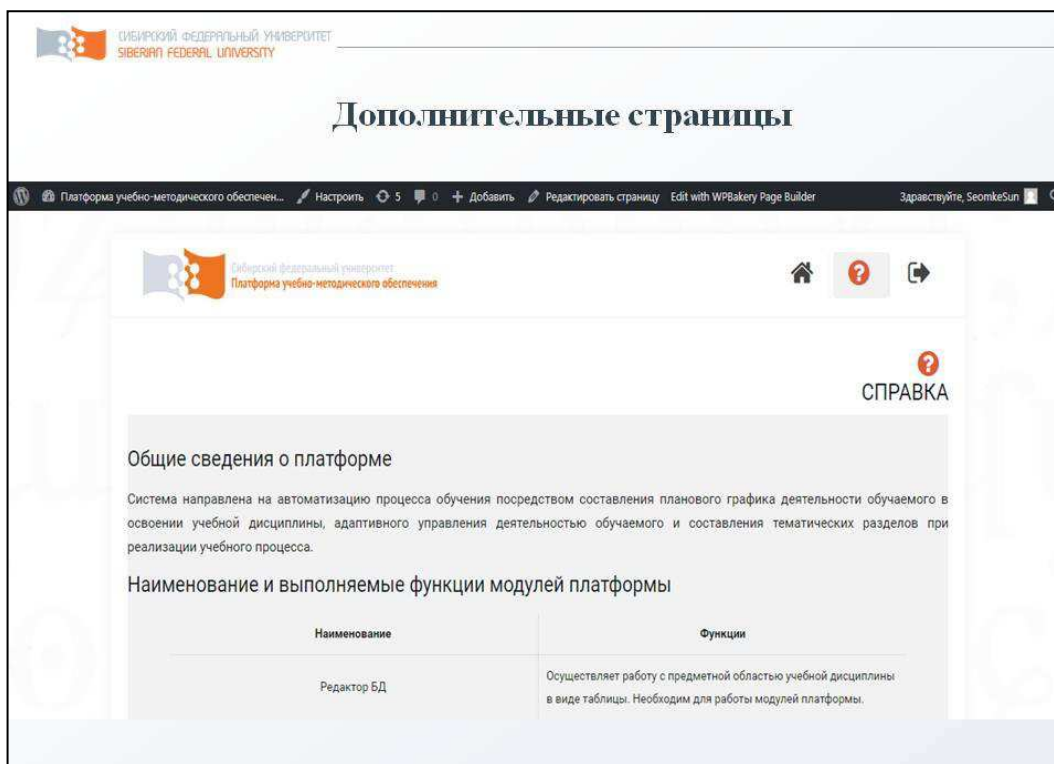


Рисунок В.13 — Слайд презентации №13

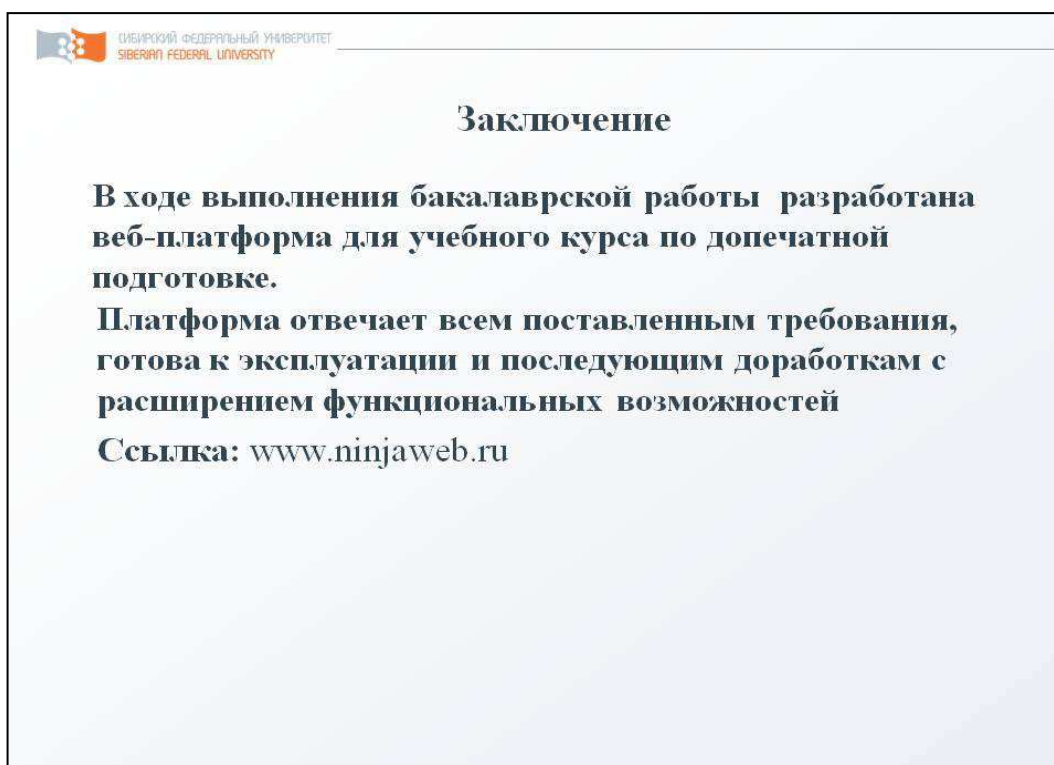


Рисунок В.14 — Слайд презентации №14



Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт космических и информационных технологий  
Кафедра систем искусственного интеллекта

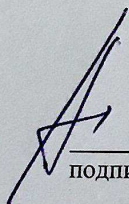
УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ Г. М. Цибульский  
подпись  
« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_ г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.02 «Информационные системы и технологии»

Разработка web-платформы для учебного курса по допечатной подготовке

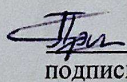
Руководитель

 19.06.18  
подпись, дата

ст. преп. каф. СИИ

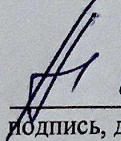
М. А. Аникьева

Выпускник

 19.06.18  
подпись, дата

С. А. Приловский

Нормоконтролер

 19.06.18  
подпись, дата

М. А. Аникьева

Красноярск 2018