

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ И ЦИФРОВЫХ ТЕХНОЛОГИЙ
КАФЕДРА ИНФОРМАЦИОННЫХ И РОБОТОТЕХНИЧЕСКИХ СИСТЕМ

**АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА
ПРИЁМА И УЧЁТА ДОКУМЕНТОВ В «ИНЖИНИРИНГОВОЙ
ШКОЛЕ НИУ БЕЛГУ»**

Выпускная квалификационная работа
обучающегося по направлению подготовки 09.03.02 Информационные
системы и технологии
очной формы обучения, группы 12001508
Нагайченко Владислава Эдуардовича

Научный руководитель
ст.пр. Удовенко И.В.

БЕЛГОРОД 2019

РЕФЕРАТ

Автоматизированная информационная система приёма и учёта документов в «инжиниринговой школе НИУ БелГУ» – Нагайченко Владислава Эдуардовича, выпускная квалификационная работа бакалавра Белгород, Белгородский государственный национальный исследовательский университет (НИУ «БелГУ»), количество страниц 66, включая приложения 82, количество рисунков 64, количество таблиц 6, количество использованных источников 43.

КЛЮЧЕВЫЕ СЛОВА: автоматизированная информационная система, база данных, веб-сайт, инжиниринговая школа, приём и учёт документов.

ОБЪЕКТ ИССЛЕДОВАНИЯ: открытая инжиниринговая школа НИУ «БелГУ».

ПРЕДМЕТ ИССЛЕДОВАНИЯ: автоматизированная информационная система приёма и учёта документов в открытой инжиниринговой школе НИУ «БелГУ».

ЦЕЛЬ РАБОТЫ: совершенствование процесса приёма и учёта документов в открытой инжиниринговой школе НИУ «БелГУ», путем разработки автоматизированной информационной системы.

ЗАДАЧИ ИССЛЕДОВАНИЯ: анализ предметной области и аналогов существующих программных средств; разработка функциональной и информационной модели; выбор программных средств для реализации; реализация программного обеспечения; тестирование и отладка программного продукта; разработка сопровождающей документации; обоснование эффективности проекта.

МЕТОДЫ ИССЛЕДОВАНИЯ: системный анализ и функциональное моделирование.

ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ: в результате работы была спроектирована и реализована автоматизированная информационная система для совершенствования процесса приёма и учёта документов в открытой инжиниринговой школе НИУ «БелГУ».

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 Аналитическая часть.....	7
1.1 Общая характеристика открытой инжиниринговой школы НИУ «БелГУ».....	7
1.2 Анализ существующих аналогов программных средств.....	9
1.3 Постановка задачи	11
2 Моделирование АИС	12
2.1 Разработка функциональной модели	12
2.2 Проектирование базы данных	17
2.3 Выбор программного обеспечения	19
3 Реализация программного обеспечения	22
3.1 Разработка базы данных.....	22
3.2 Разработка интерфейса АИС	26
3.3 Разработка серверной части АИС	34
3.4 Тестирование и отладка программного продукта	40
3.5 Сопровождающая документация	50
3.5.1 Руководство пользователя	50
3.5.2 Руководство администратора	53
3.6 Применение методики SWOT анализа для обоснования целесообразности разработки АИС в инжиниринговой школе	58
ЗАКЛЮЧЕНИЕ	62
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	64
ПРИЛОЖЕНИЕ А.....	67
ПРИЛОЖЕНИЕ В	71

ВВЕДЕНИЕ

Сегодня практически ни одно предприятие не обходится без автоматизации своей деятельности, так как высококачественная автоматизация может в значительной степени облегчить его работу, повысить производительность труда, сэкономить время на контроль и ведение документов, материальные и трудовые ресурсы. Также автоматизация позволяет сократить значительное количество ошибок, которые неизбежно возникают из-за человеческого фактора в процессе работы и всегда поддерживать данные в достоверном состоянии.

Автоматизированная информационная система (АИС) – это организационно-упорядоченная взаимосвязанная совокупность средств, и методов информационных технологий.

АИС обеспечивают сбор, хранение, обработку, поиск, выдачу информации, необходимой в процессе принятия решений задач. Помогают анализировать существующие проблемы и создавать новые продукты.

Объектом исследования является открытая инжиниринговая школа НИУ «БелГУ».

Предметом исследования является автоматизированная информационная система приёма и учёта документов в открытой инжиниринговой школе НИУ «БелГУ».

В инжиниринговой школе небольшой штат сотрудников, на которых возложено множество задач. Одной из этих задач является ведение документооборота, который сотрудники выполняют вручную и тратят много трудовых ресурсов.

Для оптимизации процесса было принято спроектировать и реализовать АИС приёма и учёта документов.

Актуальность создания АИС в инжиниринговой школе обусловлена сегодня необходимостью использования больших, и постоянно растущих,

объемов информации при решении статистических, управленческих и других задач.

А внедрение АИС позволит хранить, структурировать информацию и извлекать оптимальным для работников образом, то есть в считанные минуты поможет найти сведения о заявлениях, договорах, учениках, и т.д.

Таким образом, целью выпускной квалификационной работы является совершенствование процесса приёма и учёта документов в открытой инжиниринговой школе НИУ «БелГУ», путем разработки автоматизированной информационной системы.

Для достижения обозначенной цели необходимо решить перечень следующих задач:

- анализ предметной области и аналогов существующих программных средств;
- разработка функциональной и информационной модели;
- выбор программных средств для реализации;
- реализация программного обеспечения;
- тестирование и отладка программного продукта;
- разработка сопровождающей документации;
- обоснование эффективности проекта.

Выпускная квалификационная работа состоит из трех разделов: аналитическая часть, моделирование АИС, реализация программного обеспечения.

В первом разделе проводится аналитическое исследование открытой инжиниринговой школы НИУ «БелГУ».

Во втором разделе рассматриваются: разработка функциональной модели, проектирование базы данных, обоснование и выбор программного обеспечения.

В третьем разделе рассматриваются вопросы информационного и программного обеспечения разработки, описывается процесс разработки базы данных, визуального интерфейса и серверной части, технической

документации и приводятся итоги тестирования. Также обосновывается целесообразность разработки АИС и проводится SWOT анализ.

В заключении подведены основные итоги выпускной квалификационной работы, отмечены решенные задачи, и намечены возможные улучшения разработанной автоматизированной информационной системы в нескольких направлениях.

В приложениях представлены примеры работы АИС и листинги основных фрагментов кода.

Выпускная квалификационная работа выполнена на 66 страницах без приложения, и 82 страницах с приложениями, содержит 64 рисунка, 6 таблиц и 2 приложения.

1 Аналитическая часть

1.1 Общая характеристика открытой инженеринговой школы НИУ «БелГУ»

Объектом исследования является открытая инженеринговая школа НИУ «БелГУ».

Предметом исследования является автоматизированная информационная система приёма и учёта документов в открытой инженеринговой школе НИУ «БелГУ».

Научно-образовательный центр (НОЦ) «Инженеринговая школа НИУ «БелГУ» является структурным подразделением Инженерингового центра НИУ «БелГУ». Школа создана в рамках реализации проекта «Открытая инженеринговая школа НИУ «БелГУ» для детей и молодёжи» при поддержке Министерства образования и науки Российской Федерации, губернатора Белгородской области, Департамента образования Белгородской области, Департамента экономического развития Белгородской области.

Основная задача школы – развитие системного и проектного мышления на основе междисциплинарности.

Образовательный процесс Инженеринговой школы осуществляется в области инженеринга по естественно-научным, техническим, гуманитарным направлениям. Занятия имеют практическую направленность, а теоретическая информация сопровождается визуальными примерами, кейсами и деловыми играми. Из особенностей обучения следует отметить индивидуальный подход к каждому обучающемуся, командную работу школьников в доброжелательной атмосфере. В ходе обучения школьники выполняют научно-технические проекты совместно с партнерами и научными руководителями. Организованный таким образом, учебный процесс позволяет обучающимся адаптироваться к работе в команде, формирует высокую

мотивацию к научному творчеству и самостоятельной познавательной деятельности.

Кроме работы над общим проектом и текущего учебного процесса Инжиниринговая школа НИУ «БелГУ» обеспечивает множество дополнительных возможностей реализации научного и творческого потенциала детей: участие в сезонных школах (проведены 2 сезонные школы по 4 направлениям совместно с департаментом образования Белгородской области), предметных олимпиадах, научно-технических конкурсов инжиниринговых проектов.

На сегодняшний день модель «Открытая инжиниринговая школа НИУ «БелГУ» для детей и молодежи» включает в себя: сезонные проектно-образовательные сессии, online-обучение, конкурсы и олимпиады, обучение на постоянной основе по 18 направлениям инжиниринга, соответствующих вектору научно-технологического развития России, отраслевым рынкам матрицы НТИ и ведущим отраслям производства Белгородской области. Данная модель отображена на рисунке 1.

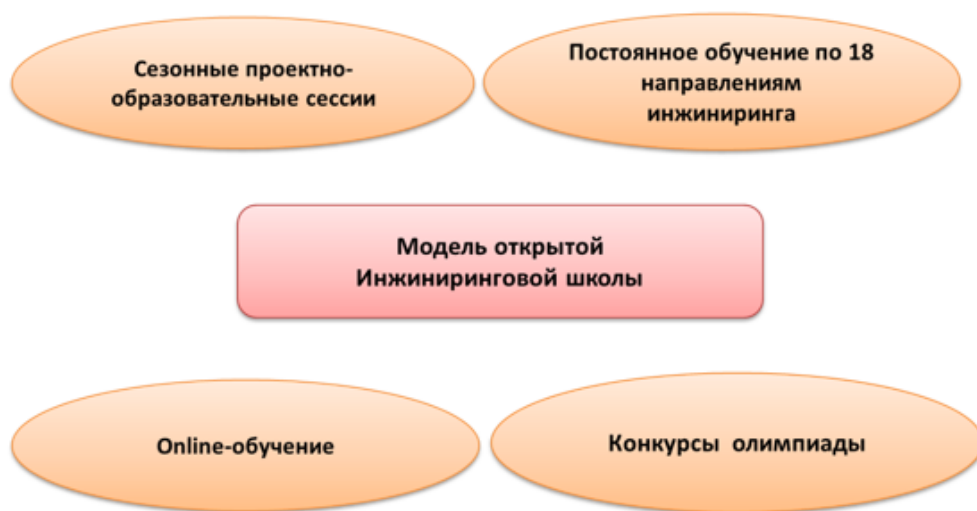


Рисунок 1 - Модель «Открытая инжиниринговая школа НИУ «БелГУ» для детей и молодежи»»

В Инжиниринговой школе прошли подготовку более 550 детей и подростков из Белгородской, Курской, Ленинградской, Московской областей, а также русскоязычные дети из Испании, Германии, Эквадора, Сингапура. Обучение в Инжиниринговой школе построено на постоянной основе.

Как уже отмечалось ранее одна из основных задач школы – выявление и подготовка талантливой молодежи на основе раннего развития творческих способностей, лидерских качеств, профессионального определения, получения комплексных знаний, навыков технопредпринимательства, проектного управления, а также практических навыков полного цикла инжиниринга (от идеи до стартапа) в соответствии с направлением национальной технологической инициативы (НТИ) – «Таланты».

В модели Национальной технологической инициативы задача обеспечения компаний кадрами нового типа основывается, с одной стороны, на проектировании технологий, формирующих перспективные рынки, и компетенций, необходимых для генерации прорывных решений, с другой стороны, на построении системы раннего выявления и развития талантов, создании среды, позволяющей этим талантам реализовать свой потенциал.

1.2 Анализ существующих аналогов программных средств

В настоящее время существует множество различных автоматизированных информационных систем приёма и учёта документов, которые используются на различных предприятиях.

В таблице 1 представлены наиболее популярные аналоги разрабатываемой автоматизированной информационной системы приёма и учёта документов, существующие на рынке сегодня.

Таблица 1 - Достоинства и недостатки рассматриваемых аналогов

Наименование	Ведение договоров	Наличие веб - клиента	Интерфейс	Возможность регистрации пользователей	Открытый исходный код	Быстрый Старт
DocsVision	Докупается отдельно	Приобретается отдельно	+	+	–	–
SharePoint	+	Функция не заявлена	+	+	–	–
Company Media	Докупается отдельно	+	+	+	–	–
Directum	Докупается отдельно	+	–	+	–	+
Разрабатываемая АИС	+	+	+	+	+	+

Из таблицы видно, что разрабатываемая АИС для приёма и учёта документов будет иметь преимущества перед существующими системами.

Одним из главных преимуществом будет реализованный узконаправленный полноценный функционал системы, который используется в инжиниринговой школе, благодаря этому возможно быстрое внедрение системы.

Также система будет обладать удобным и интуитивно понятным интерфейсом. Работу в АИС возможно будет осуществлять через любой браузер, а открытый исходный код позволит с легкостью дополнить функционал системы в случае необходимости.

В связи с вышеперечисленным, создание собственной АИС более чем целесообразно.

1.3 Постановка задачи

АИС приёма и учёта документов в открытой инжиниринговой школе НИУ «БелГУ» необходима для достижения следующих целей:

- повышение достоверности данных;
- облегчение сбора и хранения информации;
- ускорение обработки информации;
- облегчение работы персонала.

Разрабатываемая АИС должна обладать следующими возможностями:

- формирование учётной записи в системе;
- возможность редактирования личных данных;
- возможность родителю внести в систему данные о детях по отдельности;
- возможность формирования заявления;
- возможность выбора направления обучения;
- возможность сформировать договор на каждого ребёнка;
- возможность формирования квитанции;
- формирование справки об обучении ребёнка в учреждении;
- ведение факта оплаты;
- возможность формирования приказа на зачисление;
- формирование группы;
- экспорт документов в формат PDF.

Вывод по первому разделу

В данном разделе выпускной квалификационной работы были рассмотрены общая характеристика предметной области и аналоги АИС для приёма и учёта документов, существующие на рынке. А также поставлены задачи, изучены плюсы и минусы разрабатываемого продукта и принято решение для создания собственной АИС приёма и учёта документов в открытой инжиниринговой школе НИУ «БелГУ».

2 Моделирование АИС

2.1 Разработка функциональной модели

Для реализации структурно-функциональной модели автоматизированной информационной системы приёма и учёта документов в открытой инжиниринговой школе НИУ «БелГУ» была выбрана программа AllFusion Process Modeler [1].

AllFusion Process Modeler – это программный продукт для реализации средств CASE-технологий, позволяющий проводить анализ, моделирование и описание бизнес-процессов [4]. Содержит три стандартные методологии: функциональное моделирование (IDEF0), моделирование потоков данных (DFD) и моделирование потоков работ (IDEF3) [24]. Каждая методологии уникальна по-своему и может быть отдельно выполнена, однако их совокупность представляет аналитику полной картины предметной области клиента [36].

Была разработана контекстная диаграмма, которая описывает систему в целом и ее взаимодействие с внешним миром [21]. После чего контекстная диаграмма была декомпозирована для подробного представления проблемных областей процесса. Данный процесс продолжается до тех пор, пока не была достигнута желаемая степень детализации.

На рисунке 2 представлена контекстная диаграмма бизнес-процесса приёма и учёта документов в открытой инжиниринговой школе НИУ «БелГУ». На вход поступает информация о клиенте (данные родителя и ребенка), о преподавателях, сотрудниках и т.д. Управляющими, регламентирующими и нормативными данными, которыми руководствуется работа АИС, являются законы РФ, устав НИУ «БелГУ» и инжиниринговой школы. Механизмами являются сотрудники инжиниринговой школы и АИС.

На выход поступает заключенный договор, справка об обучении, квитанция, сформированные группы, приказ на зачисление и т.д.

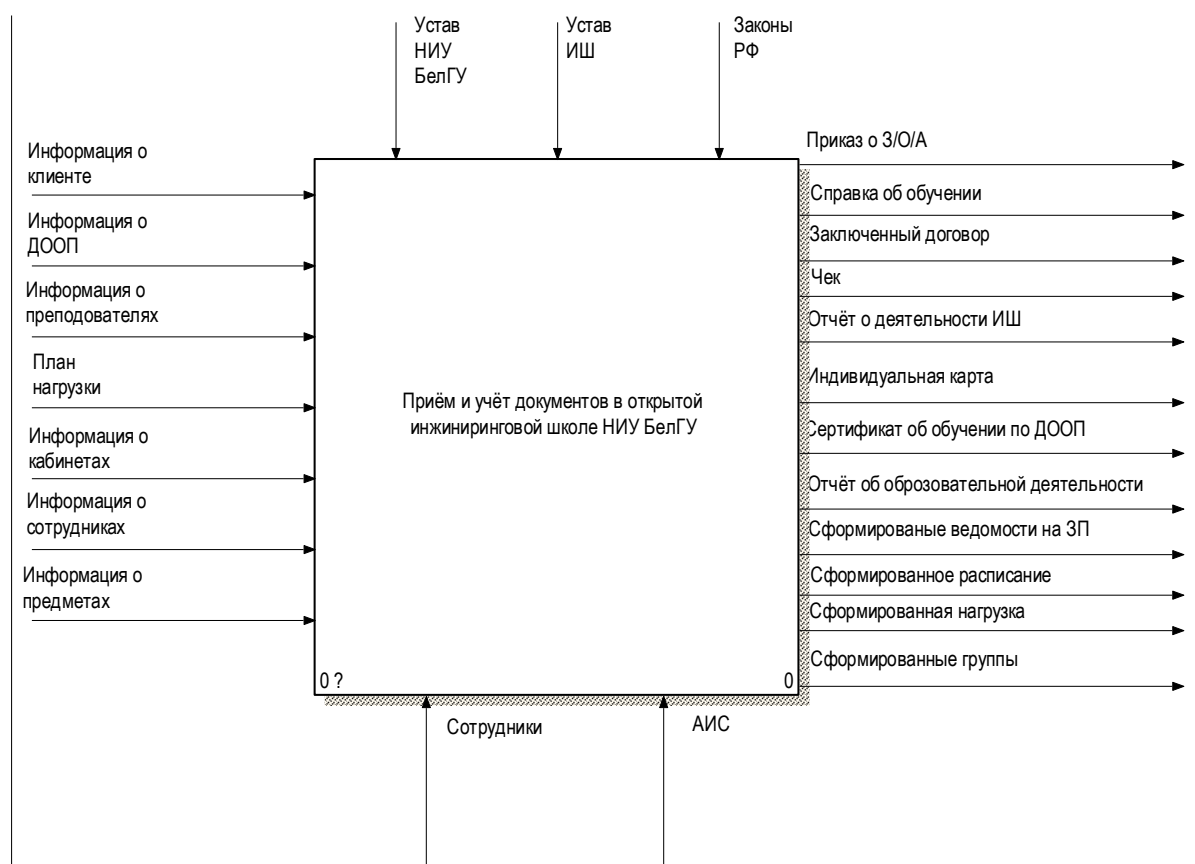


Рисунок 2 - Контекстная диаграмма

На рисунке 3 представлена декомпозиция контекстной диаграммы, которая содержит три основных процесса в методологии IDEF0: администрирование методической документации, администрирование ведения документов и администрирование учебного процесса.

В блоке администрирование методической документации входящими данными являются информация о клиенте и сотрудниках, а на выход поступает квитанция и договор, которые в свою очередь являются входящими данными в блок администрирование ведения документов.

В блоке администрирование ведения документов происходит обработка и заключение договора, ведение факта оплаты, формирование справки об обучении, формирование приказа на зачисление и др.

Формирование групп, нагрузки и расписания происходит в блоке в администрирование учебного процесса.

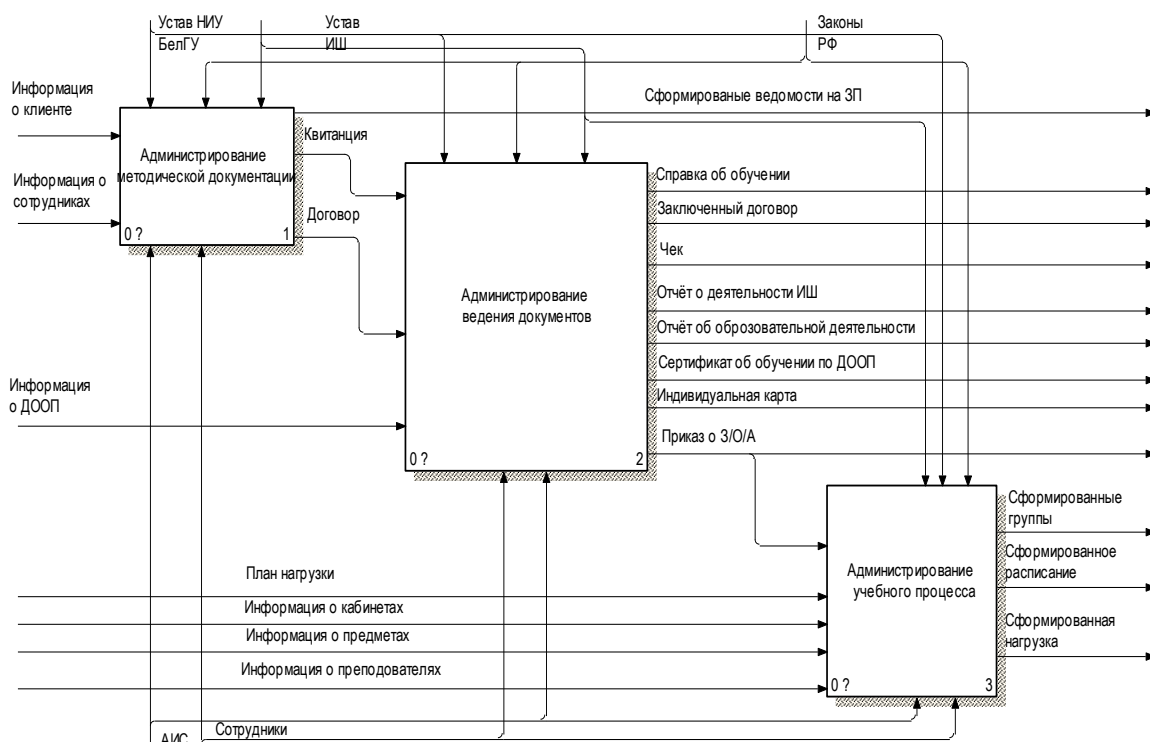


Рисунок 3 - Декомпозиция контекстной диаграммы

На рисунке 4 представлена декомпозиция блока администрирование методической документации, выполненная в методологии IDEF3.

Сначала происходит формирование учетной записи пользователя в системе, после чего начинается процесс формирования заявления на обучение, где пользователь предоставляет данные о ребенке и выбирает направление обучения.

После чего администратор, проверяет заявления и переводит его статус на «рассмотрено», после чего на основании поданного заявления формируется договор, который отобразится в личном кабинете пользователя.

В зависимости от статуса оплаты и статуса договора у пользователя в личном кабинете появится возможность получить квитанцию для оплаты обучения и справку об обучении.

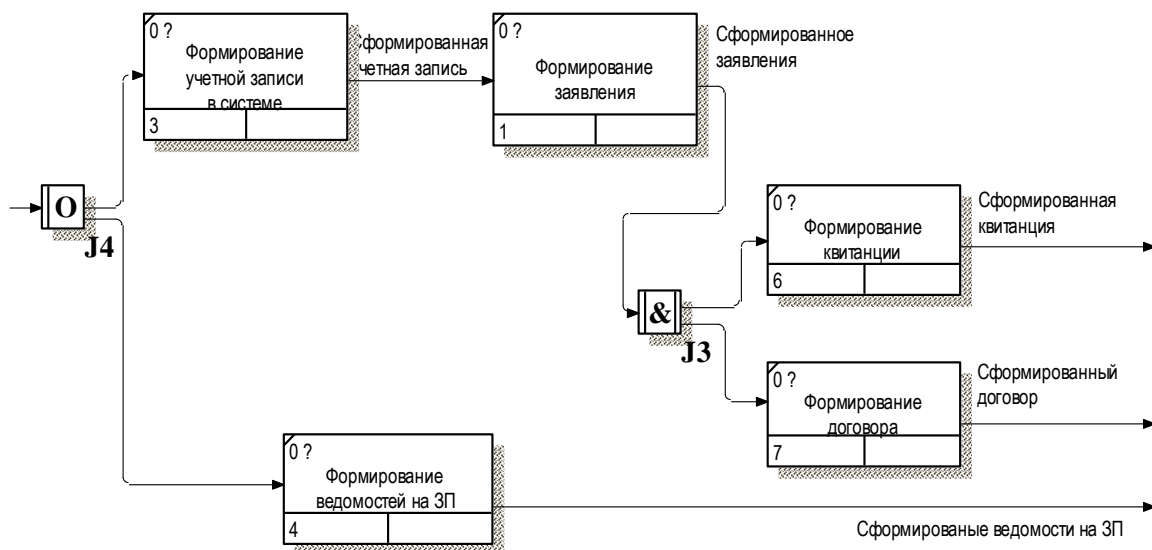


Рисунок 4 - Администрирование методической документации

На рисунке 5 представлена декомпозиция блока «Администрирование ведения документов», выполненная в методологии IDEF0.

Входящей информацией являются договор, квитанция и информация о ДООП.

Далее происходит процесс обработки и заключения договора, на основании которого формируется приказ на зачисление.

После формирования приказа, становится доступным получение справки об обучении. В свою очередь приказ поступает на вход блока учебный процесс, где формируется сертификат об обучении по ДООП, индивидуальная карта и отчет об образовательной деятельности.

Фиксация факта оплаты происходит в блоке «Оплата обучения», после чего выдается чек об оплаченных услугах, после чего выдается чек об облаченной услуге.

Управлением являются: законы РФ, устав НИУ «БелГУ» и устав инжиниринговой школы.

Механизмами являются: сотрудники и АИС.

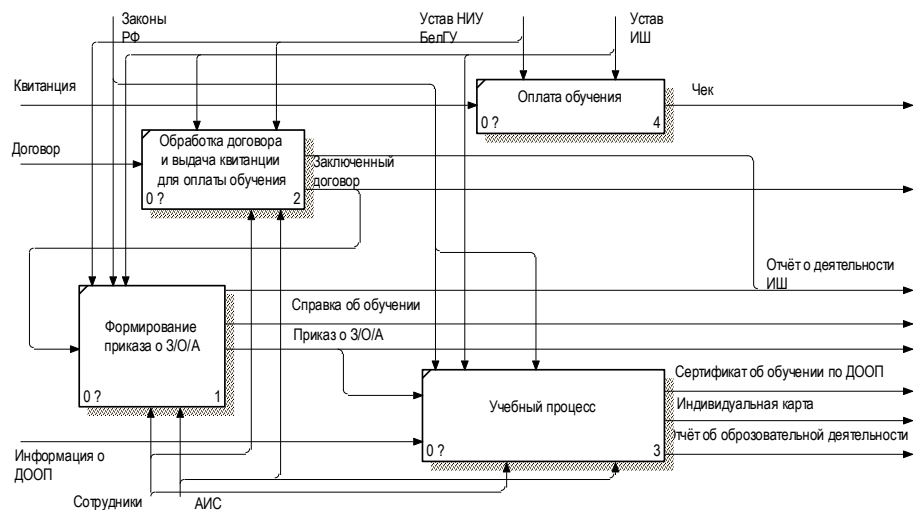


Рисунок 5 - Администрирование ведения документов

На рисунке 6 представлена декомпозиция блока «Администрирование учебного процесса» в методологии DFD. На основании приказа на зачисление формируются группы, которые заносятся в БД. У каждой группы есть свое направление обучения.

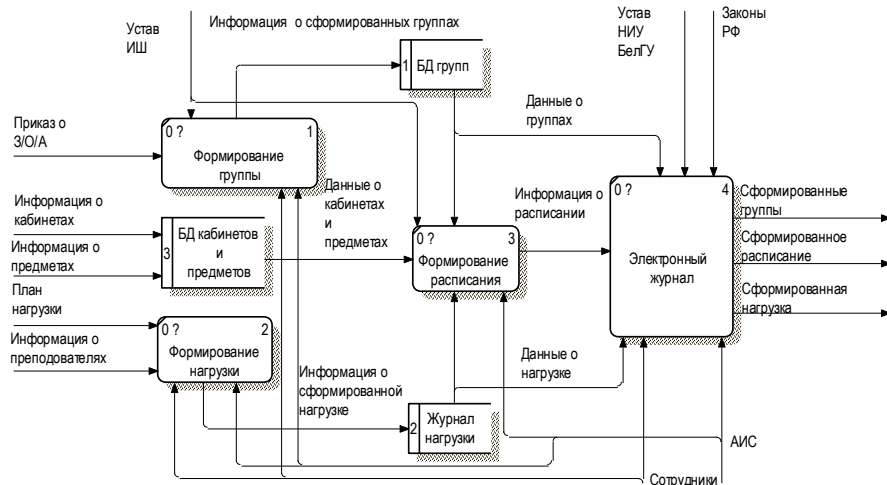


Рисунок 6 - Администрирование учебного процесса

Также в БД заносится информация о предметах и кабинетах. Происходит формирование нагрузки, которая записывается в журнал нагрузки. После чего из всего вышеперечисленного формируется расписание, которое поступает в электронный журнал вместе с информацией о группах и нагрузке.

2.2 Проектирование базы данных

Для построения информационной модели была выбрана программа MySQL Workbench.

MySQL Workbench - инструмент для визуального проектирования баз данных, интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных MySQL [28].

Была построена ER-диаграмма физического уровня, представленная на рисунке 7, на которой были выделены сущности, их атрибуты, первичные ключи, идентифицированы связи между сущностями.

Основной таблицей является таблица заявления (Statement). Из таблиц дети (kids) и пользователи (users) в заявление идет информация о персональных данных родителей и детей, такая как ФИО, дата рождения, адрес проживания, контактные данные и т.д. Из таблицы статус договора/заявления (Contract/Statement_status) поступает статус, который имеет значения «на рассмотрении» и «рассмотрено». Так же таблица заявление имеет связь с таблицей направления обучения (direction), которая содержит перечень дисциплин, преподаваемых в инженеринговой школе, и выбирается пользователем при подаче заявления.

Таблица договор связана с таблицей заявление, так как договор формируется на основании поданного заявления, которое имеет статус «рассмотрено». Все необходимые данные берутся через таблицу заявления (Statement).

Приказ на зачисление организуется на основе договоров и для того, чтобы осуществить формирование приказа на зачисление, необходимо связать его через промежуточную таблицу Student, которая содержит номер договора и номер группы. Номер группы таблица Student получает через связь с таблицей group.



Рисунок 7 - Физическая модель

На рисунке 8 представлена таблица, которая не имеет связей это новости (news).



Рисунок 8 - Таблица новости

В данную таблицу администратор сайта добавляет новую новость, которая будет видна пользователям на главной странице сайта.

2.3 Выбор программного обеспечения

Для разработки серверной части АИС приема и учёта документов в открытой инжиниринговой школе НИУ «БелГУ» был выбран язык программирования PHP и СУБД MySQL.

Язык PHP отлично подходит для написания web-приложений [26]. Скрипты выполняются на стороне сервера и это является огромным плюсом [37]. PHP способен генерировать и преобразовывать HTML документы и изображения разных форматов, файлы PDF, а также способен формировать данные в любом текстовом формате [5].

Самый оптимальный вариант для создания web-приложений различной сложности это использование PHP+MySQL [17].

MySQL - полноценная серверная СУБД. MySQL успешно работает с различными сайтами и веб приложениями, а также поддержка сервера MySQL автоматически включается в поставку PHP [10]. MySQL отлично сочетается с PHP и имеет множество преимуществ таких как: простота, широкий функционал, большое количество функций, обеспечивающих безопасность, которые поддерживаются по умолчанию, масштабируемость, скорость [39].

Исходя из вышесказанного целесообразность использования связки PHP+MySQL является оправданной.

Для разработки серверной части был выбран фреймворк, Yii 2 - это MVC-фреймворк.

Yii 2 - это фреймворк для веб-программирования общего назначения, который может быть использован для разработки практически любых веб-приложений. Особенно подходит для разработки приложений с большим потоком трафика, таких как (CMS) [35].

Yii 2 очень эффективен, имеет большой спектр возможностей и отличается качественной документацией и соответствует требованиям при разработке серьезных веб-приложений, чем и превосходит другие фреймворки PHP [40].

MVC - это паттерн проектирования веб - приложений, который включает в себя несколько более мелких шаблонов [7].

Этот шаблон разделяет работу веб-приложения на три отдельные функциональные роли: модель данных (model), пользовательский интерфейс (view) и управляющую логику (controller) [9]. Таким образом, изменения, вносимые в один из компонентов, оказывают минимально возможное воздействие на другие компоненты [22]. Структура MVC представлена на рисунке 9.

В данном паттерне модель не зависит от представления или управляющей логики, что делает возможным проектирование модели как независимого компонента и, например, создавать несколько представлений для одной модели [41].

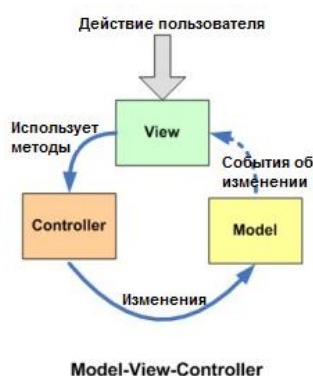


Рисунок 9 - Model-View-Controller

Для реализации визуальной части АИС были выбраны языки HTML, CSS, JavaScript.

HTML (HyperText Markup Language) - стандартный язык разметки гипертекстовых страниц в Интернете [32].

CSS (Cascading Style Sheets - каскадные таблицы стилей) - формальный язык описания внешнего вида документа, написанного с использованием языка разметки [27].

JavaScript – язык сценариев, исполняемых на стороне клиента, который позволяет создавать интерактивные HTML-страниц. JavaScript запускается в Web-браузере и не используется на стороне сервера [29]. Сценарии на стороне клиента позволяют пользователю интерактивно взаимодействовать с Web-страницей после того, как она была обработана сервером и загружена Web-браузером [38]. Связка данных языков позволят создать подходящий и отзывчивый интерфейс для нашей АИС.

Вывод по второму разделу

В данном разделе выпускной квалификационной работы были разработаны функциональная и информационная модель для АИС приёма и учета документов в инженеринговой школе НИУ «БелГУ». А также рассмотрены и выбраны средства реализации для визуального интерфейса и серверной части.

3 Реализация программного обеспечения

3.1 Разработка базы данных

База данных (БД) – совокупность взаимосвязанных, хранящихся вместе данных при наличии такой минимальной избыточности, которая допускает их использование оптимальным образом для одного или нескольких приложений.

Создание базы данных, ее поддержка и обеспечение доступа пользователей к ней осуществляется централизованно с помощью специального программного инструментария – системы управления базами данных [18].

Система управления базами данных (СУБД) – это комплекс программных и языковых средств, необходимых для создания баз данных, поддержания их в актуальном состоянии и организации поиска в них необходимой информации [8].

Для организации работы системы в СУБД MySQL была создана база данных, которая состоит из 23 таблиц, представленная в таблице 2.

Таблица 2 - Список таблиц базы данных

Название таблицы	Описание таблицы
Users	Содержит данные зарегистрированных пользователей
Kids	Содержит информацию о детях
Statement	Перечень заявлений
Contract	Перечень договоров
Student	Информация о учениках школы
Group	Содержит список учебных групп
Order	Приказ на зачисление

Продолжение таблицы 2

Название таблицы	Описание таблицы
Reference	Справка об обучении
Receipt	Чек
Payment	Статус оплаты обучения
Contract/Statement Status	Статус договора и заявления
Direction	Содержит перечень направлений подготовки
Dicspline	Содержит перечень предметов
auth_assignment	Назначения
auth_item	Разрешения
auth_rule	Правила
auth_item_child	Маршруты
news	Новости
teacher	Содержит информацию о преподавателях школы
rooms_has_direction	
rooms	Перечень кабинетов

Структура таблицы Statement (заявление) представлена в таблице 3.

Таблица 3 - Структура таблицы Statement

Имя поля	Ключ	Тип, дина	Обязательность значения
idStatement	Pk	Integer	NOT NULL
IdUsers	Fk1	Integer	NOT NULL
iddirection	Fk2	Integer	NOT NULL
idKids	Fk3	Integer	NOT NULL
statement_status_idstatement_status	Fk4	Integer	NOT NULL

Продолжение таблицы 3

Имя поля	Ключ	Тип, дина	Обязательность значения
date		Date	NOT NULL
image		text	NOT NULL
image2		text	NOT NULL
day1		int	NOT NULL
day2		int	NOT NULL
month1		int	NOT NULL
month2		int	NOT NULL

Создание структуры таблицы представлено в листинге 1.

Листинг 1 - Создание таблицы Statement

```
CREATE TABLE
`Statement` (
  `idStatement` int(11) NOT NULL,
  `IdUsers` int(11) NOT NULL,
  `iddirection` int(11) NOT NULL,
  `idKids` int(11) NOT NULL,
  `date` text,
  `statement_status_idstatement_status` int(11) NOT NULL,
  ` day1` int(11) NOT NULL,
  ` day2` int(11) NOT NULL,
  ` month1` int(11) NOT NULL,
  ` month2` int(11) NOT NULL,
  ` image1` text NOT NULL,
  ` image2` text NOT NULL, )
ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

После создания таблицы были созданы индексы для данной таблицы.

Первичный ключ (primary key) представляет собой один из примеров уникальных индексов и применяется для уникальной идентификации записей таблицы [34].

Внешний ключ (foreign key) SQL - это ключ, используемый для объединения двух таблиц [43].

Поле idStatement является первичным ключом, а поля idUsers, iddirection, idKids, statement_status_idstatement_status внешними ключами, которые необходимы для связи с таблицами: Users, Kids, Direction, Contract/Statement_status, что продемонстрировано в листинге 2.

Листинг 2 - Создание индексов таблицы Statement

```
ALTER TABLE `Statement`  
ADD PRIMARY KEY (`idStatement`),  
ADD KEY `idUsers_idx` (`IdUsers`),  
ADD KEY `fk_Statement_direction1_idx` (`iddirection`),  
ADD KEY `fk_Statement_Kids1_idx` (`idKids`),  
ADD KEY `fk_Statement_statement_status1_idx`  
(`statement_status_idstatement_status`),  
ADD KEY `fk_Statement_kids_status1_idx`  
(`idkids_status`);
```

Для удобства были созданы для каждой таблицы автоинкрементируемые поля по первичному ключу. Пример создания автоинкремента для таблицы Statement представлен в листинге 3.

Листинг 3 - Создание автоинкремента

```
AUTO_INCREMENT для таблицы `Statement`  
ALTER TABLE `Statement`  
MODIFY `idStatement`  
int(11) NOT NULL AUTO_INCREMENT,  
AUTO_INCREMENT=29;
```

Так же необходимо было для каждой таблицы, которая имеет связь с другими таблицами создать ограничения внешнего ключа таблицы `Statement`, программный код представлен в листинге 4.

Листинг 4 - Создание ограничения внешнего ключа

```
ALTER TABLE `Statement`  
  ADD CONSTRAINT `fk_Statement_Kids1`  
  FOREIGN KEY (`idKids`)  
  REFERENCES `Kids` (`idKids`)  
  ON DELETE NO ACTION ON UPDATE NO ACTION,  
ADD CONSTRAINT `fk_Statement_direction1` FOREIGN KEY (`iddirection`) REFERENCES  
`direction` (`iddirection`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  ADD CONSTRAINT `fk_Statement_kids_status1`  
  FOREIGN KEY (`idkids_status`)  
  REFERENCES `kids_status` (`idkids_status`)  
  ON DELETE NO ACTION ON UPDATE NO ACTION,  
ADD CONSTRAINT `fk_Statement_statement_status1`  
  FOREIGN KEY (`statement_status_idstatement_status`)  
  REFERENCES `statement_status` (`idstatement_status`)  
  ON DELETE NO ACTION ON UPDATE NO ACTION,  
ADD CONSTRAINT `statement_ibfk_1`  
  FOREIGN KEY (`IdUsers`) REFERENCES `user` (`id`);
```

Для всех остальных таблиц создание структуры, индексов и ограничений было выполнено по аналогии с таблицей `Statement`.

Наполнение БД тестовыми данными являлось завершающим этапом разработки базы данных.

3.2 Разработка интерфейса АИС

Визуальный интерфейс является одной из важнейших частей приложения, от качества интерфейса во многом зависит эффективность системы, поэтому очень важно обеспечить эргономичность, удобство и соответствие целям АИС [30].

На начальной стадии разработки была разработана главная страница сайта.

На рисунке 10 представлена шапка главной страницы сайта.



Рисунок 10 - Шапка на главной странице сайта

На рисунке 11 продемонстрирован слайдер на главной странице сайта.

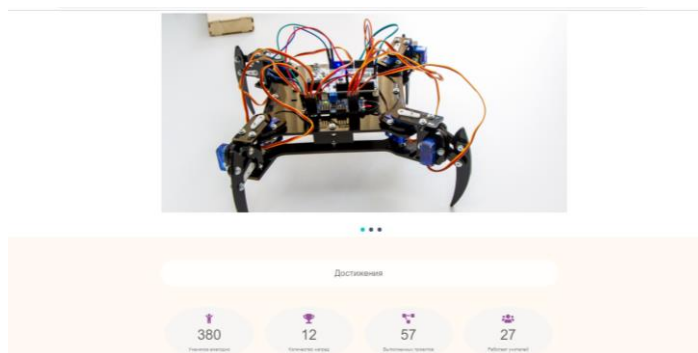


Рисунок 11 - Слайдер на главной странице сайта

На рисунке 12 изображены подвал и карточки на главной странице сайта.

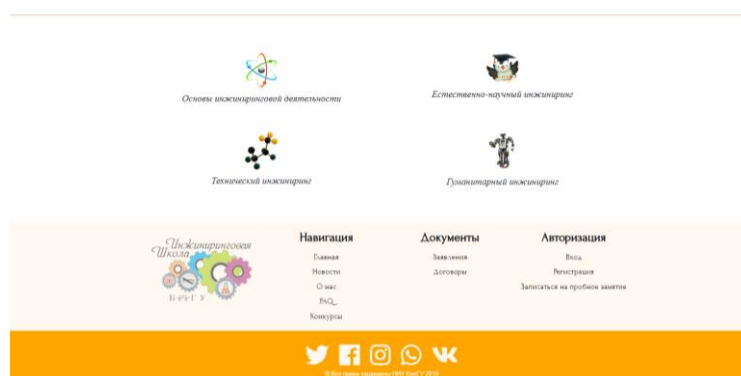


Рисунок 12 – Подвал на главной странице сайта

На главной странице в верхней её части было сделано навигационное меню с ссылками на другие страницы. Меню содержит 8 ссылок.

Под меню располагается «Шапка», которая содержит название открытой инжиниринговой школы и ссылку на официальный сайт НИУ БелГУ, представленная на рисунке 10.

На странице также помещен основной блок с контентом, который содержит слайдер, карточки и различную информацию, представленный на рисунках 11,12.

На рисунке 12 представлен «Подвал» сайта, который имеет ссылки на документы, официальные социальные сети открытой инжиниринговой школы, а также дублирует ссылки навигационного меню.

Следующим этапом разработки интерфейса было создание форм регистрации и авторизации пользователей. Формы включают поля «Логин» и «Пароль». Форма регистрация представлена на рисунке 13, форма авторизация представлена на рисунке 14.

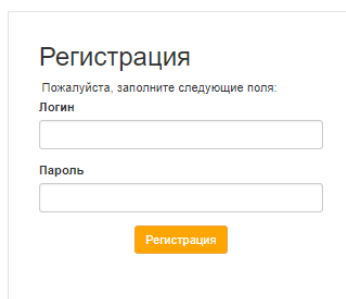


Рисунок 13 - Форма регистрации

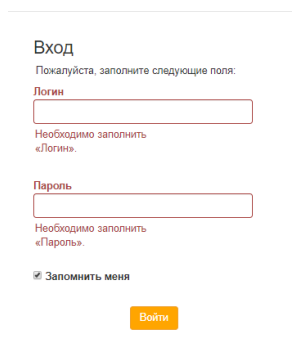


Рисунок 14 - Форма авторизации

После создания форм был создан личный кабинет, представленный на рисунке 15, который содержит меню для навигации и блок для работы с данными, представленный на рисунках А.1-А.3 в приложении А.

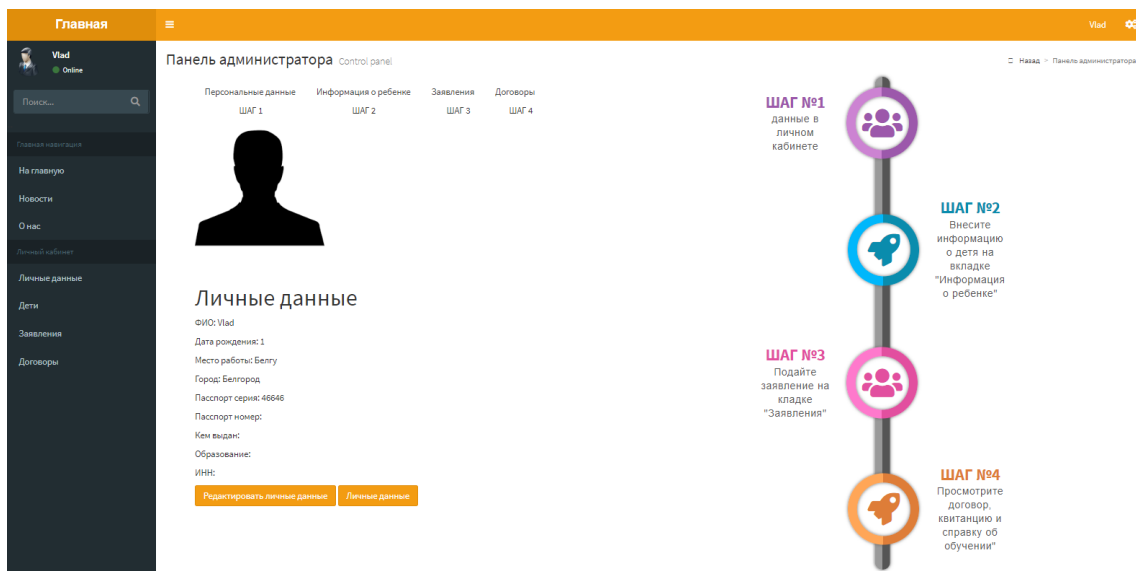


Рисунок 15 - Личный кабинет

Также была создана «Панель администратора», которая содержит меню выбора нужной таблицы и основную часть, где непосредственно находится таблица, представленная на рисунке 16. В основной части содержится информация и происходит редактирование данных администратором.

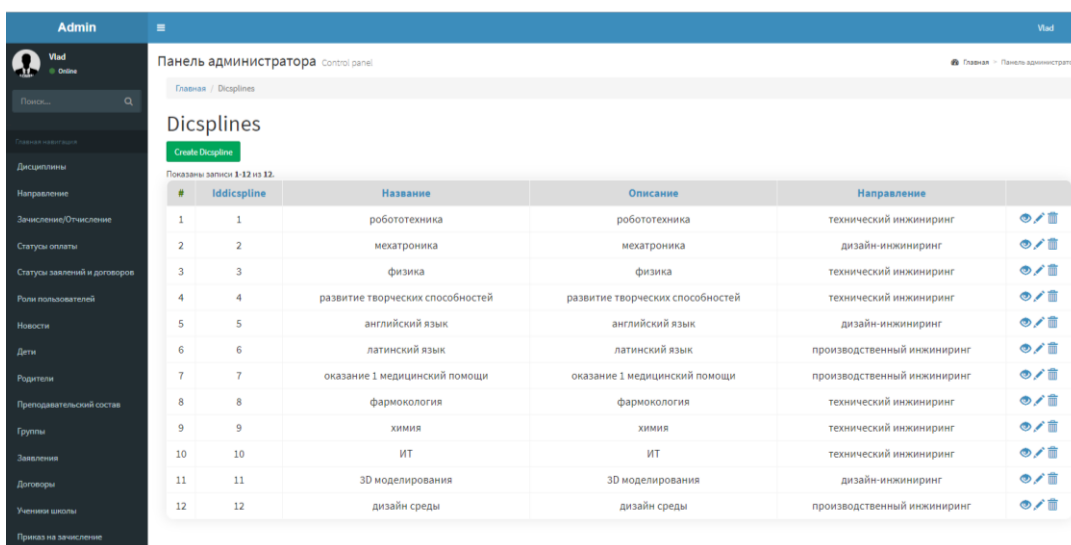


Рисунок 16 - Панель администратора

Также была создана страница «О нас», представленная на рисунке 17, которая содержит информацию об открытой инжиниринговой школе НИУ «БелГУ», преподавателях и контактные данные школы, это продемонстрировано в приложении А рисунки А.4-А.7.

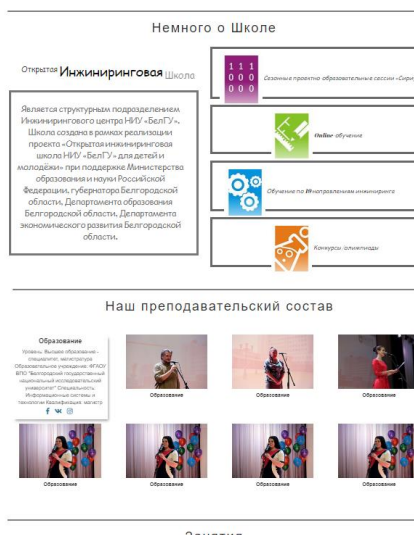


Рисунок 17 - Информация о школе и преподавателях

Пример созданной формы ввода данных для заявления представлен на рисунке 18. Формы для всех остальных таблиц были созданы по аналогии.

The image shows a web form for entering application data. The form is titled "Главная / Заявление / 88 / Изменить" and has a "Сохранить" (Save) button. The form contains the following fields:

- Статус заявления: Рассмотрено (dropdown menu)
- Направление: География (dropdown menu)
- Отчисление/Зачисление: На зачисление (dropdown menu)
- Дата: 2018-12-03 (text input)
- Пользователь: Владик нагайченко (dropdown menu)
- Данные родителя:
 - E-mail: nagaychenko@gmail.com2 (text input)
- Место работы: Белгу (text input)
- Должность: Студент (text input)
- Квартира: 43 (text input)
- Страна: Россия (text input)
- Регион: Южный (text input)
- Район: (text input)

Рисунок 18 - Форма ввода данных

Также были разработаны шаблоны справки об обучении и договора, представленные в приложении А рисунки А.8, А9.

На рисунке 19 представлен шаблон квитанции для оплаты обучения.

Извещение	ФГАОУ ВО «Белгородский государственный национальный исследовательский университет» (наименование получателя платежа) ИНН 3123035312; КПП 312301001; ОКАТО 14401365000; ОКТМО 14701000001 р/с № 40503810207004000002 в Белгородском отделении №8592 ОАО «Сбербанк России»; БИК 041403633; кор/счет 30101810100000000633; КОД 07430201010010000130 за обучение 18/19 уч.г. этап Ф.И.О. заказчика
Кассир	Адрес заказчика Город: , улица: , корпус: , дом: , квартира: Договор № 31 Сумма платежа <input type="text"/> руб. <input type="text"/> коп.

Рисунок 19 - Шаблон квитанции для оплаты обучения

На рисунке 20 представлен шаблон заявления на обучение.

	Ректору НИУ БелГУ О.Н. Полухину Владик нагайченко
От	
Заявление	
Прошу зачислить мою(моего) дочь(сына) Сына Ф.И.О. (полностью, печатными буквами) Владик нагайченко На обучение по дополнительной общеобразовательной общеразвивающей программе (наименование программы) География в ФГАОУ ВПО "Белгородский государственный национальный исследовательский университет" в срок с 1 января 2018 г. по 1 января 2019 г.	
Сообщаю о сыне (дочери) следующие сведения:	
1. Дата рождения 2010-10-20 полных лет 11 2. Адрес проживания страна Россия регион Северный район Воронежский населенный пункт Воронеж улица Юбилейная корпус (строение) 2 дом 12 квартира 4 7. Контактные телефоны 8-800-555-35-35 8. Электронная почта Vla11dik@mail.com 9. Школа №100	
О себе сообщаю	
1. Дата рождения полных лет 17.12.1997 2. Место работы Белгу 3. Должность Студент 4. Адрес проживания: Страна Россия регион Южный район Белгородский населенный пункт Белгород	

Рисунок 20 - Шаблон заявления

На рисунке 21 представлена страница «Услуги».



Рисунок 21 - Услуги

На рисунке 22 представлена страница «Оборудование в инженеринговой школе».

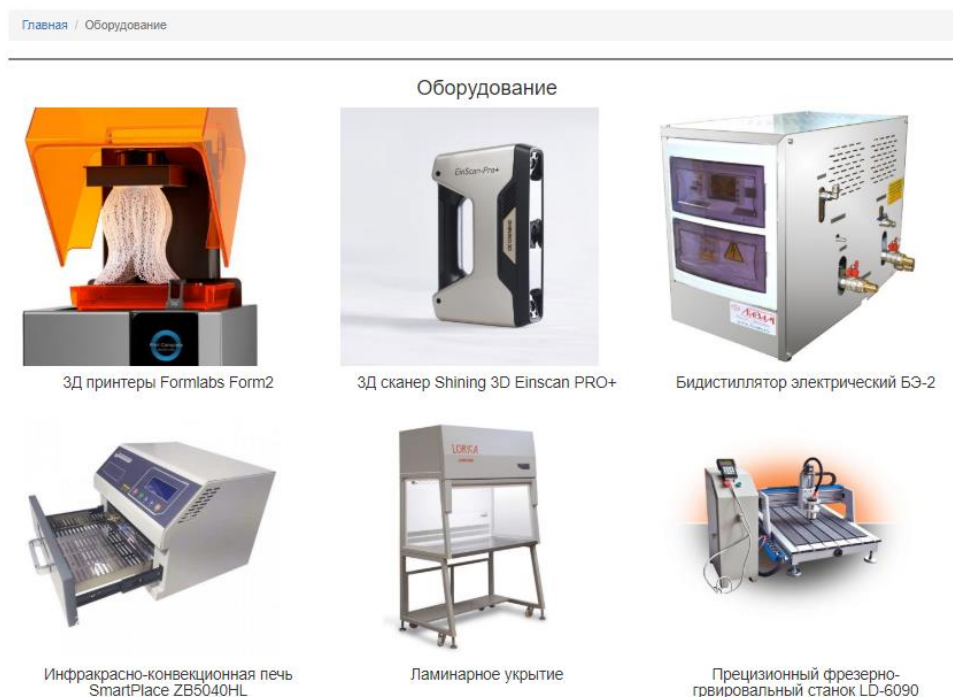


Рисунок 22 - Оборудование

На рисунках 23, 24 представлены страницы «Партнеры» и «Запись на пробное занятие».

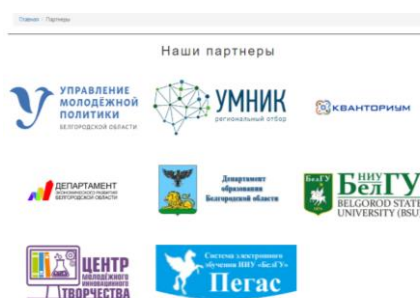


Рисунок 23 - Партнёры

Рисунок 24 - Форма записи на пробное занятие

Завершающим этапом разработки визуального интерфейса было создание страницы «Новости», продемонстрировано на рисунке 25.

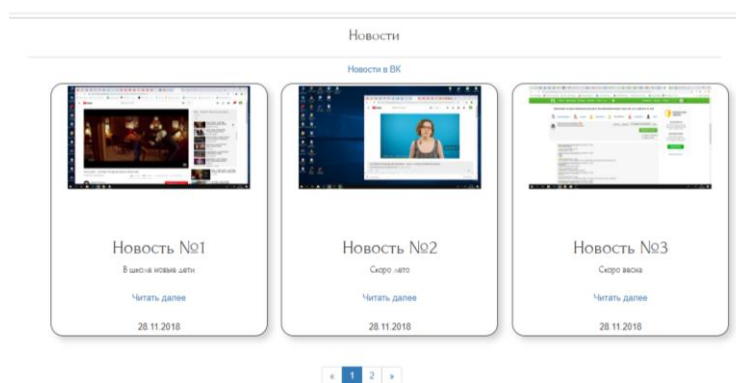


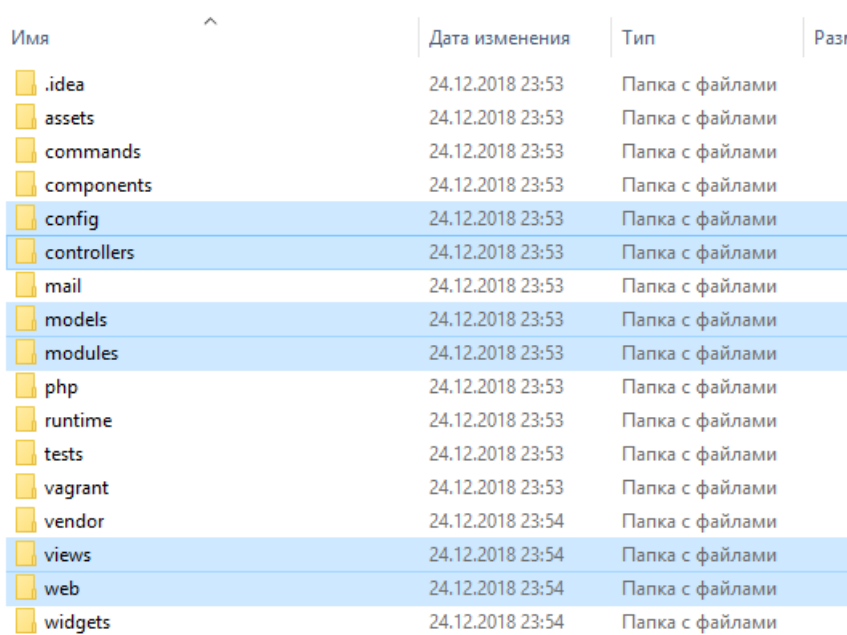
Рисунок 25 - Страница новости

Разработанные страницы содержат информацию для пользователя об открытой инженеринговой школе НИУ «БелГУ».

3.3 Разработка серверной части АИС

После создания базы данных и визуального интерфейса, была реализована функциональная часть программного продукта.

Структура проекта представлена на рисунке 26.



Имя	Дата изменения	Тип	Разм
.idea	24.12.2018 23:53	Папка с файлами	
assets	24.12.2018 23:53	Папка с файлами	
commands	24.12.2018 23:53	Папка с файлами	
components	24.12.2018 23:53	Папка с файлами	
config	24.12.2018 23:53	Папка с файлами	
controllers	24.12.2018 23:53	Папка с файлами	
mail	24.12.2018 23:53	Папка с файлами	
models	24.12.2018 23:53	Папка с файлами	
modules	24.12.2018 23:53	Папка с файлами	
php	24.12.2018 23:53	Папка с файлами	
runtime	24.12.2018 23:53	Папка с файлами	
tests	24.12.2018 23:53	Папка с файлами	
vagrant	24.12.2018 23:53	Папка с файлами	
vendor	24.12.2018 23:54	Папка с файлами	
views	24.12.2018 23:54	Папка с файлами	
web	24.12.2018 23:54	Папка с файлами	
widgets	24.12.2018 23:54	Папка с файлами	

Рисунок 26 - Структура проекта

В папке config содержатся файлы для подключения БД и различных модулей, что представлено в листинге приложения В.

В папках controllers, models и view соответственно располагаются контроллеры, модели и виды, а в папке modules находится созданный модуль администратора со своими контроллерами и видами и отдельным layout.

Для начала были созданы модели для всех таблиц, имеющих в БД. Пример создания модели для таблицы discipline представлен в листинге приложения В. Класс Discipline наследуется от ActiveRecord.

Active Record обеспечивает объектно-ориентированный интерфейс для доступа и манипулирования данными, хранящимися в базах данных [19]. Класс Active Record соответствует таблице в базе данных, объект Active Record соответствует строке этой таблицы, а атрибут объекта Active Record представляет собой значение отдельного столбца строки [6]. Вместо непосредственного написания SQL-выражений вы сможете получать доступ к атрибутам Active Record и вызывать методы Active Record для доступа и манипулирования данными, хранящимися в таблицах базы данных [13].

Функция `tableName` возвращает имя таблицы из БД.

Функция `rules` устанавливает правила валидации [31]. Так `'required'` делает поля обязательными для заполнения, тут же указывается тип данных для всех полей и их размер.

В функциях `attributeLabels` устанавливается имя атрибутов таблицы.

Функция `getDirectionIdirection` устанавливает связь между классами `Direction` и `Dicspline`.

В листинге приложения В представлено создание для данной таблицы модели поиска `DicsplineSearch`.

Далее были созданы нужные контроллеры и в них прописаны действия.

Контроллеры являются частью MVC архитектуры [20]. Это объекты классов, унаследованных от `yii\base\Controller`, отвечающие за обработку запроса и генерирование ответа [11]. В сущности, после обработки запроса приложениями, контроллеры проанализируют входные данные, передадут их в модели, вставят результаты модели в представления, и в конечном итоге сгенерируют исходящие ответы [14].

Контроллеры состоят из действий, которые являются основными блоками, к которым может обращаться конечный пользователь и запрашивать исполнение того или иного функционала [15]. В контроллере может быть одно или несколько действий [2].

Пример создания контроллера `SiteController` представлен в листинге приложения В. Действия созданного контроллера представлены в таблице 4.

Таблица 4 - Действия SiteController

Действие	Описание действия
actionIndex	Главная страница сайта
actionLogin	Переход на форму регистрации
actionLogout	Выходи из учетной записи
actionAbout	страница о нас
actionSignup	Авторизация
actionInfo	Новости

После создания контроллеров были созданы представления, представленные в таблице 5. Для каждой таблицы в большинстве случаев создаются несколько представлений.

Таблица 5 - Представления для таблицы Contract

Название	Описание
_form	Форма создания и редактирования договоров
_search	Поиск
index	Главное представление
update	Изменение
view	Представление
View-pdf	Представление печати

Далее был создан отдельный модуль для панели администратора, где он сможет редактировать все необходимые данные в таблицах. В данном модуле также созданы контроллеры и представления.

Для того, чтобы ограничить доступ обычного пользователя от администратора, были созданы роли пользователей, для этого был подключен модуль rbac, что представлено на рисунке 27.

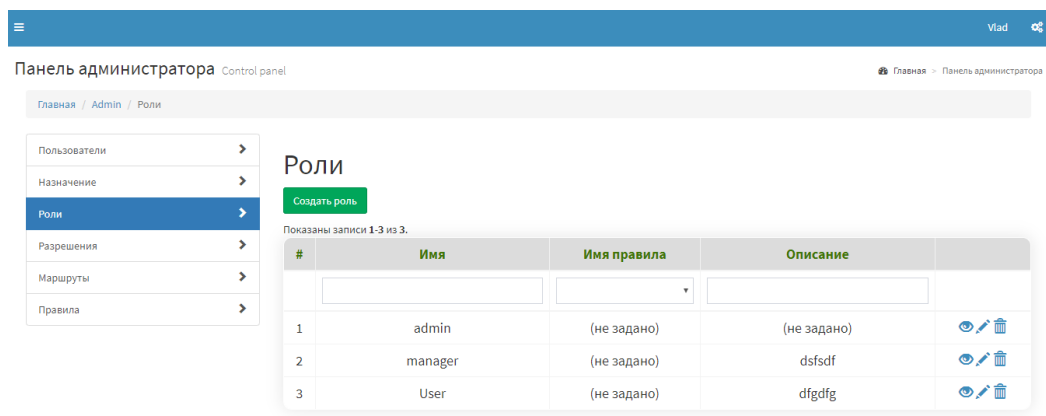


Рисунок 27 - Панель администратора rbac

Также к модулю администратора в config был подключен отдельный layout с навигацией и отдельным пунктом меню для входа в панель администратора, который будет виден только администратору.

Для удобства выбора и заполнения данных необходимо было выполнить реализацию выпадающих списков.

Пример реализации выпадающего списка с корректным отображением имен представлен в листинге 5. В переменную \$model_kids записывается результат запроса, в котором с помощью метода find() выполняется выборка всех данных из модели Kids. После с помощью цикла foreach заполняется массив значениями KidsName, который передается во view, где добавляется элемент dropDownList, куда передается данный массив.

Листинг 5 - Реализация выпадающего списка

```
$model_kids = Kids::find()->orderBy('KidsName')->all();
foreach($model_kids as $value){
    $arrKids[$value->idKids]= $value->KidsName;}
return $this->render('index', [
    'arrKids'=>$arrKids,
]);
<?= $form->field($model, 'iddirection')->dropDownList($arrDirection, ['prompt'=>""]) ?>
```

Для реализации автозаполнения полей по выбранному значению из выпадающего списка использовался формат JSON.

Первым делом было создано действие в контроллере листинге 6. В действие в переменную записывается результат запроса, где выбираются дети с выбранным id, который поступает из выпадающего списка в действие в контроллере с помощью параметра \$id. После необходимо вернуть переменную в формате JSON, который используется для представления объектов в виде строки. В представлении по выбранному значению в выпадающем списке заполняются необходимые поля, что представлено в листинге 6.

Листинг 6 - Реализация автозаполнения полей в контроллере

```
public function actionLists($id){
    \Yii::$app->response->format = \yii\web\Response::FORMAT_JSON;
    $info = Kids::find()
        ->where(['idKids'=>$id])
        ->all();
    return Json::encode($info);
}
```

Листинг 7 - Реализация автозаполнения полей во view

```
<?= $form->field($model, 'idKids')->dropDownList($arrKids, [
    'prompt' => 'Выбрать',
    'onchange'=>'$.get( "'.Url::toRoute(['statement/lists'])."', { id : $(this).val() })
.done(function(data) {
    var json = $.parseJSON(data);
    $("#statement-kemail").val(json[0].email);
    $("#statement-kkvartira").val(json[0].kvartira);
    $("#statement-kcountry").val(json[0].country);
    $("#statement-kregion").val(json[0].region);
    $("#statement-klocality").val(json[0].locality);
    $("#statement-kbody").val(json[0].body);
    $("#statement-kstreet").val(json[0].street);
    $("#statement-khouse").val(json[0].house);
    $("#statement-knumber").val(json[0].number);
    $("#statement-kdr").val(json[0].date_b);
    $("#statement-kschool").val(json[0].school);
```

```
$( "#statement-kcity").val(json[0].city);
});'
]) ?>
```

Для реализации печати документов был использован модуль MPDF, который в действии контроллера вызывает представление и преобразовывает его из формата .php в формат PDF, это продемонстрировано в листинге 8.

Листинг 8 - Функция GenPdf

```
public function actionGenPdf($id) {
    $pdf_content=$this->renderPartial('view-pdf', [
        'model' => $this->findModel($id), ]); $mpdf->Output();
    exit; }
```

Для создания заявления был выполнен SQL запрос, представленный в листинге приложения В.

Для реализации выбора договора с помощью check и сохранения его в промежуточную таблицу был прописан код, представленный в листинге приложения В.

Для получения данных модели User использовалась следующая команда: \$id = Yii::\$app->user->identity->id;

Для отображения данных с помощью таблицы использовался виджет GridView, а для создания форм использовался виджет Yii2 ActiveForm, что представлено в листинге приложения В.

В листинге приложения В, представлено получение данных конкретной модели с помощью функции findModel с принимаемым параметром \$id.

Для реализации возможности загрузки копии паспорта в модели Image1 был прописан код, представленный в листинге приложения В.

В правилах были прописаны разрешенные форматы для загружаемых файлов, в данном случае форматы изображений jpg и png.

Для предотвращения ситуации перезаписи изображения, когда разные пользователи загрузят на сервер изображения с одинаковым названием, необходимо с помощью md5 закодировать название изображения тем самым обеспечив ему уникальное название.

В листинге 9 представлен метод `setImage`, созданный в контроллере заявления.

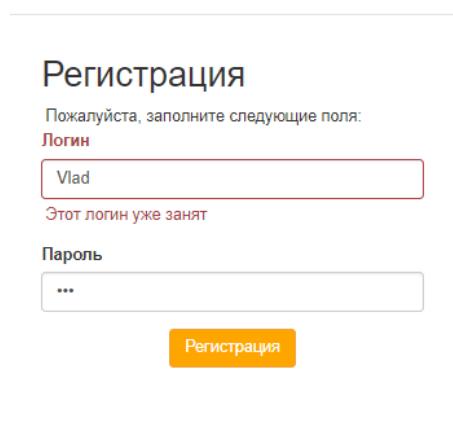
Листинг 9 - метод `setImage`

```
public function actionSetImage($id){
    $model = new Image; if(Yii::$app->request->isPost)
    { $photo = $this->findModel($id);
    $file = UploadedFile::getInstance($model,'image');
    if($photo->saveImage($model->uploadFile($file, $photo->image))){
        return $this->redirect(['view','id'=>$photo->idStatement]);
    }return $this-> render('image',['model'=>$model]);}
```

3.4 Тестирование и отладка программного продукта

Тестирование программного обеспечения - процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определенным образом [23].

Для начала была протестирована форма регистрации пользователей. При вводе имеющегося логина должно быть оповещение что данный логин уже занят, что продемонстрировано на рисунке 28.



Регистрация

Пожалуйста, заполните следующие поля:

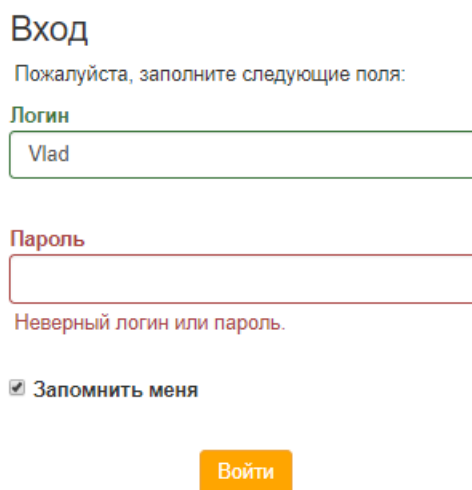
Логин

Этот логин уже занят

Пароль

Рисунок 28 - Ввод существующего логина

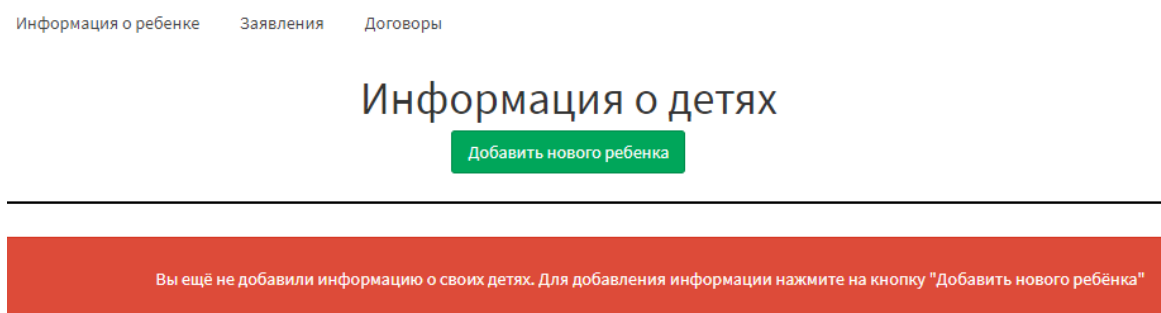
В форме входа ввели неверный логин или пароль и увидели соответствующее сообщение о неверно введенных данных, представленное на рисунке 29.



The screenshot shows a login form titled "Вход". Below the title is the instruction "Пожалуйста, заполните следующие поля:". There are two input fields: "Логин" (Login) containing the text "Vlad" and "Пароль" (Password), which is empty. Below the password field is a red error message: "Неверный логин или пароль.". At the bottom left, there is a checked checkbox labeled "Запомнить меня". At the bottom center, there is an orange button labeled "Войти".

Рисунок 29 - Неверный пароль или логин

Если пользователь не вносил данные о детях он увидит соответствующее сообщение (рисунок 30).



The screenshot shows a user profile page with a navigation bar containing "Информация о ребенке", "Заявления", and "Договоры". The main heading is "Информация о детях". Below the heading is a green button labeled "Добавить нового ребенка". A red banner at the bottom contains the text: "Вы ещё не добавили информацию о своих детях. Для добавления информации нажмите на кнопку \"Добавить нового ребёнка\"".

Рисунок 30 - Информация о незаполненных данных

После ввода данных должны появиться данные ребенка, представленные на рисунке 31 и отобразиться информация в личном кабинете, что представлено на рисунке 32.

Личный кабинет

Главная / Дети

Изменить Удалить

Имя	TEST1
Дата рождения	17.12.1997
Полных лет	21
Страна	Россия
Регион	Северный
Район	Белгородский
Населенный пункт	Белгород
Улица	test
Корпус	1
Дом	1
Квартира	1
Контактные данные	8-800-800-88-00
Школа	№1
Email	test@mail.com

Рисунок 31 - Заполненная информация о ребенке

Личные данные Информация о ребенке Заявления Договоры

Информация о детях

Добавить нового ребенка

Редактировать личные данные Личные данные

ФИО	TEST1
Дата рождения	Дата рождения 17.12.1997
Возраст	21
Номер	8-800-800-88-00
Email	test@mail.com
Школа	№1
Адрес	Страна Россия Регион Северный Город Белгород Район Белгородский Улица test Корпус 1 Дом 1 Квартира 1

Рисунок 32 - Информация о ребенке в личном кабинете

При подаче заявления есть обязательные поля при не заполнении которых, пользователь не сможет отправить заявление на проверку. Для проверки корректности работы было пропущено несколько полей и проверен результат, представленные на рисунке 33.

2

Дом

12

Квартира

Необходимо заполнить «Квартира».

Контактные данные

Необходимо заполнить «Контактные данные».

Школа

Необходимо заполнить «Школа».

Email

test@mail.com

Населенный пункт

Белгород

Сохранить

Рисунок 33 - Ошибка заполнения обязательных полей

При загрузке копии паспорта и свидетельства о рождении разрешены только форматы, применяемые для изображений. В поле для загрузки был загружен файл с расширением docx. На рисунке 34 представлено оповещение о разрешенных форматах.

Паспорт

Выберите файл Документ ...ord.docx

Разрешена загрузка файлов только со следующими расширениями: jpg, png.

Загрузить

Рисунок 34 - Оповещение о разрешенных форматах

Отображения информации о заявлении и его статусе в личном кабинете после подачи (рисунок 35).

Подробнее...

ФИО	TEST1
Заявление №	130
Направление	нейроинжиниринг
Статус заявления	На рассмотрении
Статус заявления	2019-04-10
Копия паспорта	
Копия свидетельства	

Рисунок 35 - Отображении информации о заявлении

На рисунке 36 представлена проверка корректности отображения копии паспорта и свидетельства о рождении в личном кабинете после загрузки.



ФИО	TEST1
Заявление №	136
Направление	аэрокосмический инжиниринг
Статус заявления	Рассмотрено
Статус заявления	2019-05-08
Копия паспорта	
Копия свидетельства	

Рисунок 36 - Копия паспорта и свидетельства о рождении

После подачи проверили отображение заявления у администратора, а также работу сортировки по полям, что представлено на рисунке 37.

Заявление

Создать заявление

Показаны записи 1-1 из 1.


#	Заявление №	Пользователь	Направление	Ребёнок	Дата	Статус заявления	Отчисление/ Зачисление	
1	136	Владик нагайченко	аэрокосмический инжиниринг	TEST1	2019-05-08	На рассмотрении	На зачисление	

Рисунок 37 - Отображения заявления у администратора и сортировка

На основании заявления создали договор и перешли в режим просмотра договора, для того чтобы убедиться в правильно отображении всех данных, это продемонстрировано на рисунке 38.

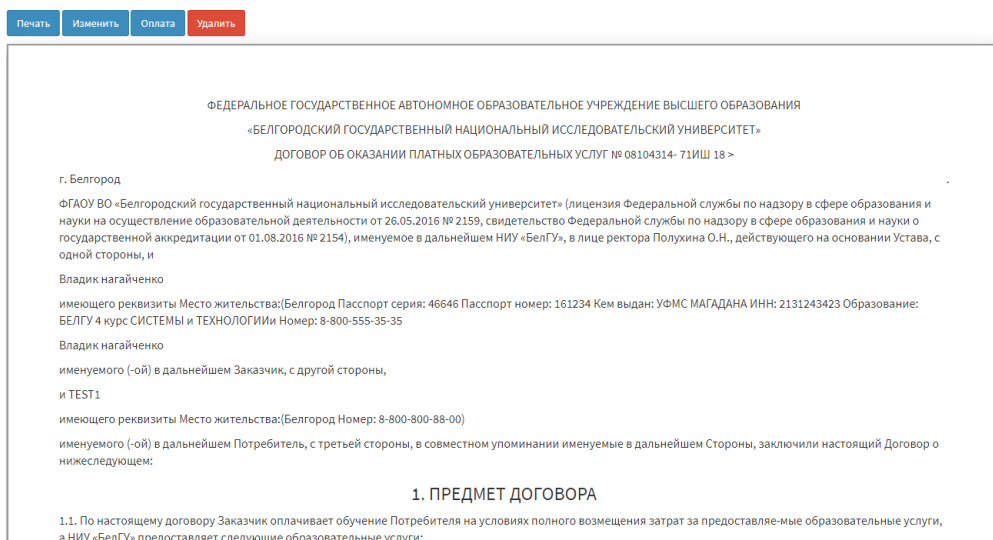


Рисунок 38 - Просмотр созданного договора

Отображение созданного договора в панели администратора представлено на рисунке 39.



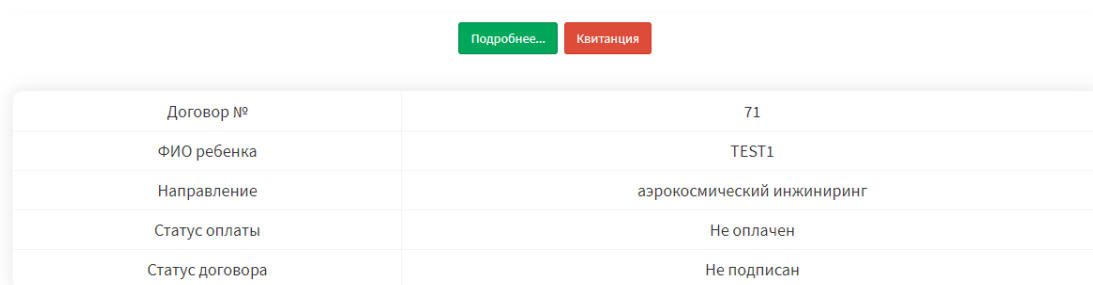
17	<input type="checkbox"/>	71	136	Подписан	Не оплачен	TEST1	Владик нагайченко	 
----	--------------------------	----	-----	----------	------------	-------	-------------------	--

Рисунок 39 - Договор в панели администратора

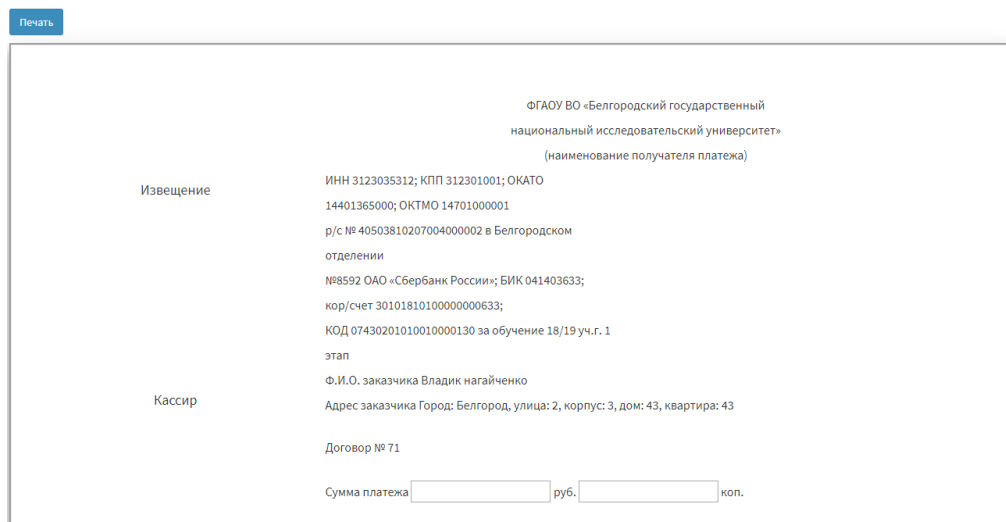
После формирования договора администратором у пользователя в личном кабинете на вкладке «Мои договоры» должны отобразиться договор и квитанция для оплаты обучения. Для проверки этого необходимо было зайти в личный кабинет и убедиться, что всё отобразилось верно, что представлено на рисунке 40.



Договор №	71
ФИО ребенка	TEST1
Направление	аэрокосмический инжиниринг
Статус оплаты	Не оплачен
Статус договора	Не подписан

Рисунок 40 - Договор и кнопка квитанции для оплаты

На рисунке 41 представлена квитанция для оплаты обучения в личном кабинете пользователя.



Печать

ФГАОУ ВО «Белгородский государственный национальный исследовательский университет»
(наименование получателя платежа)

Извещение

ИНН 3123035312; КПП 312301001; ОКАТО 14401365000; ОКТМО 14701000001
р/с № 40503810207004000002 в Белгородском отделении
№8592 ОАО «Сбербанк России»; БИК 041403633;
кор/счет 30101810100000000633;
КОД 07430201010010000130 за обучение 18/19 уч.г. 1 этап

Кассир

Ф.И.О. заказчика Владик нагайченко
Адрес заказчика Город: Белгород, улица: 2, корпус: 3, дом: 43, квартира: 43

Договор № 71

Сумма платежа руб. коп.

Рисунок 41 - Квитанция для оплаты обучения

Далее занесли ученика в группу, что представлено на рисунке 42, изменили статус оплаты и сформировали приказ на зачисление, что представлено на рисунке 43.

Главная / Ученики

Ученики

Добавить ученика

Показаны записи 1-1 из 1.




#	ID	Группа	Номер договора	Ученик	
1	180	07001508	71	TEST1	  

Рисунок 42 - Занесение ученика в группу

Главная / Приказ / 22 / Update

Изменить: 22

Группа №

07001508

Сохранить

Рисунок 43 - Формирование приказа на зачисление

После формирования приказа на зачисления у пользователя появилась кнопка справка об обучении в личном кабинете и пропала кнопка квитанции, представленная на рисунке 44.

Подробнее...

Справка

Договор №	71
ФИО ребенка	TEST1
Направление	аэрокосмический инжиниринг
Статус оплаты	Оплачен
Статус договора	Подписан

Рисунок 44 - Кнопка справка об обучении

На рисунке 45 представлена печатная форма договора об обучении.

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ»
ДОГОВОР ОБ ОКАЗАНИИ ПЛАТНЫХ ОБРАЗОВАТЕЛЬНЫХ УСЛУГ № 08104314-
71ИШ 18 >
г. Белгород
ФГАОУ ВО «Белгородский государственный национальный исследовательский
университет» (лицензия Федеральной службы по надзору в сфере образования
и науки на осуществление образовательной деятельности от 26.05.2016 № 2159,
свидетельство Федеральной службы по надзору в сфере образования и науки о
государственной аккредитации от 01.08.2016 № 2154), именуемое в
дальнейшем НИУ «БелГУ», в лице ректора Полухина О.Н., действующего на
основании Устава, с одной стороны, и
Владик нагайченко
имеющего реквизиты Место жительства:(Белгород Паспорт серия: 46646
Паспорт номер: 161234 Кем выдан: УФМС МАГАДАНА ИНН: 2131243423
Образование: БЕЛГУ 4 курс СИСТЕМЫ и ТЕХНОЛОГИИи Номер: 8-800-555-35-35
Владик нагайченко
именуемого (-ой) в дальнейшем Заказчик, с другой стороны,
и TEST1
имеющего реквизиты Место жительства:(Белгород Номер: 8-800-800-88-00)
именуемого (-ой) в дальнейшем Потребитель, с третьей стороны, в совместном
упоминании именуемые в дальнейшем Стороны, заключили настоящий
Договор о нижеследующем:

1. ПРЕДМЕТ ДОГОВОРА

1.1. По настоящему договору Заказчик оплачивает обучение Потребителя на
условиях полного возмещения затрат за предоставляе-мые образовательные
услуги, а НИУ «БелГУ» предоставляет следующие образовательные услуги:

Рисунок 45 - Печатная форма договора

Также было протестировано добавление новости на сайт, для этого в таблицу новости добавили тестовую запись, что продемонстрировано на рисунке 46.

Добавить новость

ID

Заголовок

Текст

Рисунок 46 - Добавление тестовой записи

На рисунке 47 представлен результат отображения добавления новости на сайт в раздел «Новости».

«Космический инжиниринг»



В инжиниринговой школе открывается каникулярная смена «Космический инжиниринг». С 25 по 30 марта школьники будут изучать химию, фармацевтику, биологию, математическое моделирование, программирование, робототехнику, английский язык и т.д.

Рисунок 47 - Добавленная запись

Последним этапом тестирования была проверка назначения ролей пользователям.

Зарегистрировали нового пользователя с логином Vlad2 и убедились, что ему доступен личный кабинет и не доступна панель администратора, что продемонстрировано на рисунке 48.

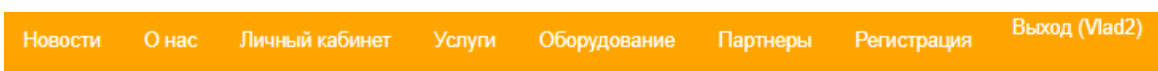


Рисунок 48 - Доступный личный кабинет

На рисунке 49 представлен вход в систему под ролью администратор и назначены пользователю Vlad2 права администратора.

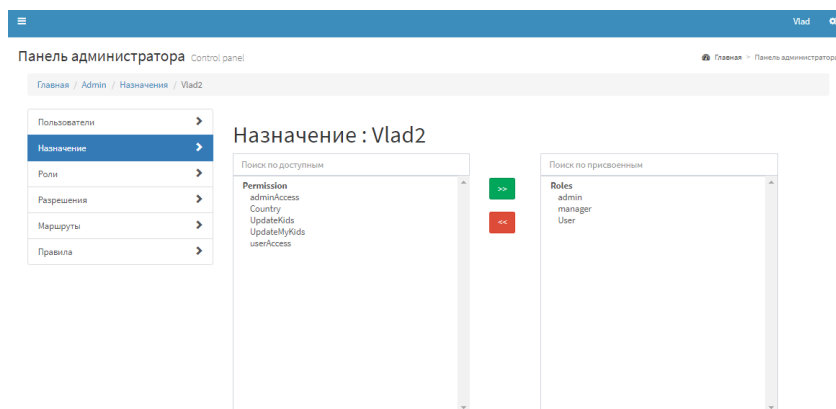


Рисунок 49 - Назначение роли

После снова выполнили вход под пользователем Vlad2 и убедились, что появились возможности администратора, что представлено на рисунке 50.

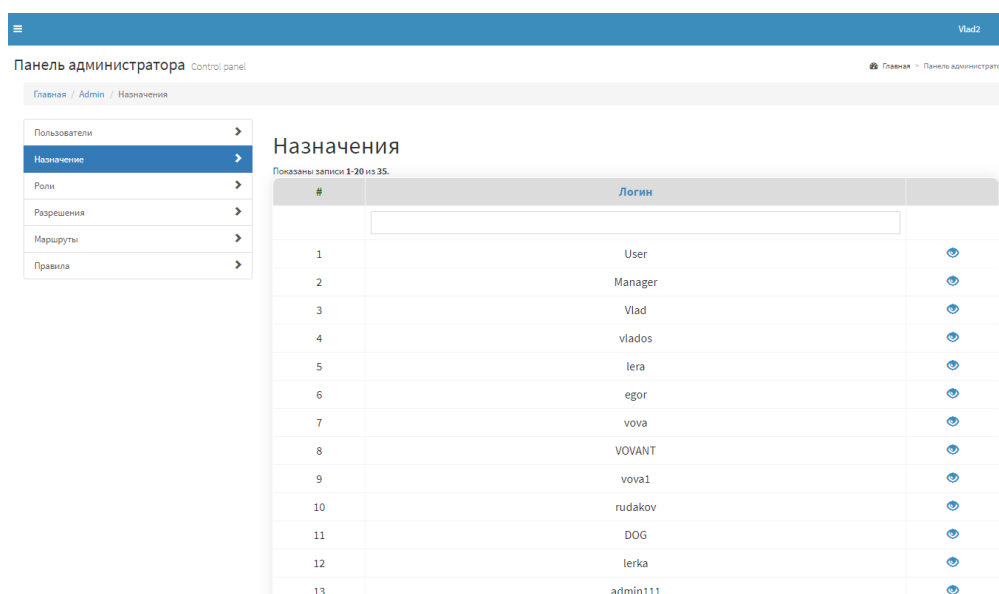


Рисунок 50 - Панель администратора у пользователя Vlad2

Теперь пользователь может назначать правила, маршруты, роли и разрешения другим юзерам, а также выполнять добавление, изменения и поиск во всех таблицах в панели администратора.

3.5 Сопровождающая документация

3.5.1 Руководство пользователя

Документ «Руководство пользователя» относится к пакету эксплуатационной документации. Основная цель «Руководства пользователя» заключается в обеспечении пользователя необходимой информацией для самостоятельной работы с программой или автоматизированной системой.

Регистрация и авторизация

Для регистрации на сайте необходимо на панели навигации выбрать пункт регистрация, что представлено на рисунке 51.

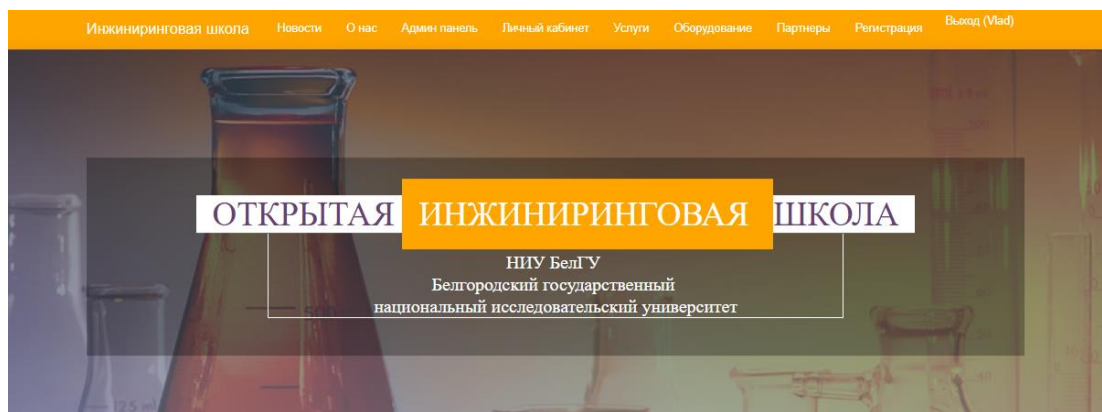


Рисунок 51 - Панель навигации

После заполнить следующие поля: логин и пароль, и нажать на кнопку регистрация, представленную на рисунке 52.

Рисунок 52 - Форма регистрации

После регистрации необходимо пройти авторизацию, для этого на панели навигации выбрать пункт вход и заполнять необходимые поля.

Управление личным кабинетом

Навигация по личному кабинету осуществляется с помощью бокового меню, которое имеет следующие пункты: «Личные данные», «Добавить информацию о ребенке», «Подать заявление», «Мои договоры».

На вкладке «Личные данные» выбрать кнопку «Редактировать личные данные», где необходимо заполнить все поля формы и нажать кнопку «Сохранить». Для загрузки фотографии для профиля в конце формы выбрать файл для загрузки и нажать кнопку «Загрузить».

Для добавления информации о ребенке необходимо на вкладке «Добавить информацию о ребенке» нажать на кнопку «Добавить нового ребенка», где необходимо заполнить все поля и нажать кнопку «Сохранить».

Для подачи заявления на обучения на вкладке «Подать заявление» нажать кнопку «Подать новое заявление». Выбрать из выпадающего списка ребёнка. Данные о родителе заполняются автоматически, данные о ребенке также заполняются автоматически при выборе ребенка из выпадающего списка. Далее выбрать направление обучения и сроки обучения и нажать кнопку «Сохранить».

Посмотреть печатную форму документа можно нажав на кнопку «Подробнее», там же необходимо загрузить копию паспорта и свидетельства о рождении.

Для печати заявления нажать кнопку «Экспорт в PDF», представленную на рисунке 53.

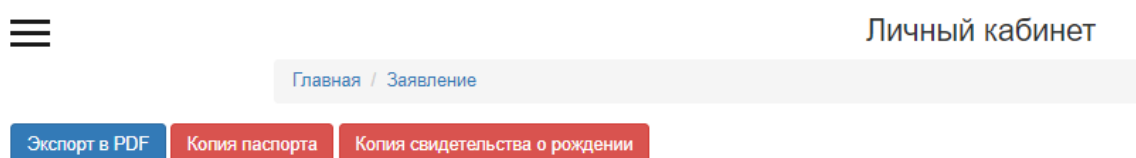


Рисунок 53 - Печать заявления

После подачи заявления администратор должен рассмотреть заявления и установить статус рассмотрено.

После составления администратором договора он отобразится на вкладке «Мои договоры».

Можно отследить статус оплаты и статус договора:

- Если договор имеет статус «Не оплачен» появится кнопка квитанция.
- Если договор имеет статус «Подписан», то появляется кнопка «Справка об обучении».

При нажатии на кнопку «Подробнее» можно увидеть печатную форму договора и распечатать его, нажав кнопку «Экспорт в PDF».

3.5.2 Руководство администратора

Одним из необходимых в процессе эксплуатации системы документов является «Руководство администратора» [16]. Роль администратора системы заключается в том, чтобы организовывать и поддерживать работу конечных пользователей, тогда как обеспечение технической работоспособности системы обычно обеспечивается другими людьми [12]. Таким образом, администратор лишь управляет работой систем [3].

Для входа в панель администратора необходимо на панели навигации выбрать пункт «Админ панель».

Добавление, удаление, редактирование

На рисунке 54 представлена панель администратора, которая состоит из бокового меню и рабочей области.

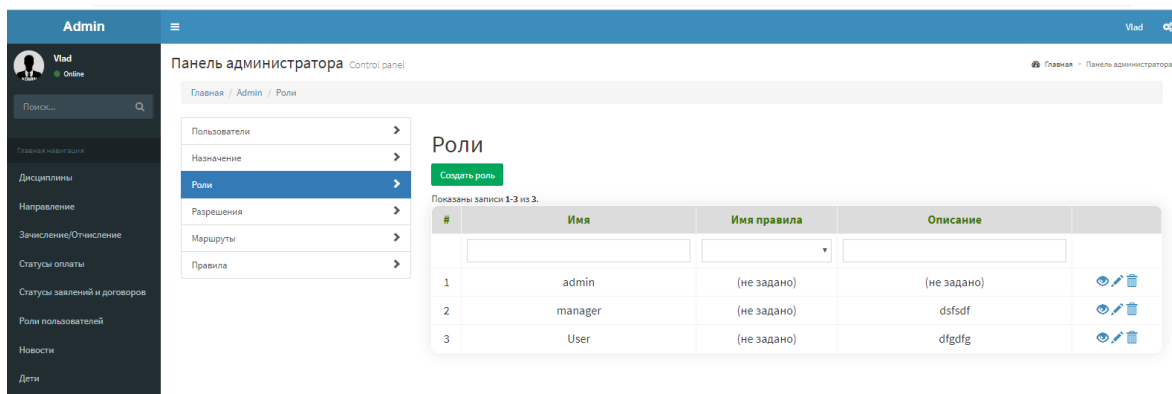


Рисунок 54 - Панель администратора

Для просмотра, редактирования и удаления записи таблицы необходимо выбрать напротив каждой записи таблицы иконку с действием, представленную на рисунке 55.



Рисунок 55 - Просмотр, редактирование, удаление записи

Для создания новой записи необходимо нажать кнопку «Создать», представленную на рисунке 56.

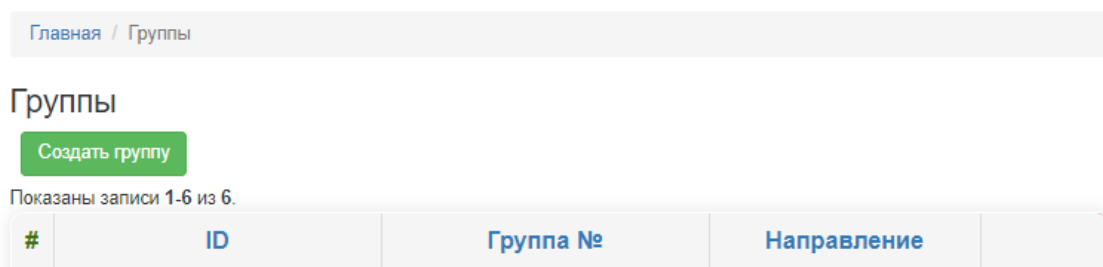


Рисунок 56 - Кнопка создать

Назначение ролей пользователям

Для назначения ролей в боковом навигационном меню необходимо выбрать пункт «Роли пользователей», показанный на рисунке 57. После открытия подменю со следующими пунктами: «Пользователи», «Назначение», «Разрешения», «Маршруты».

В пункте «Пользователи» содержится информация о пользователе, такая как дата создания и статус учетной записи.

Пункт «Назначение» нужен для назначения роли пользователю, например, для создания нового администратора.

В пункте «Разрешения» можно создавать разрешения с выбранными маршрутами.

В пункте «Маршруты» происходит выбор разрешенных действий для присвоения разрешениям.

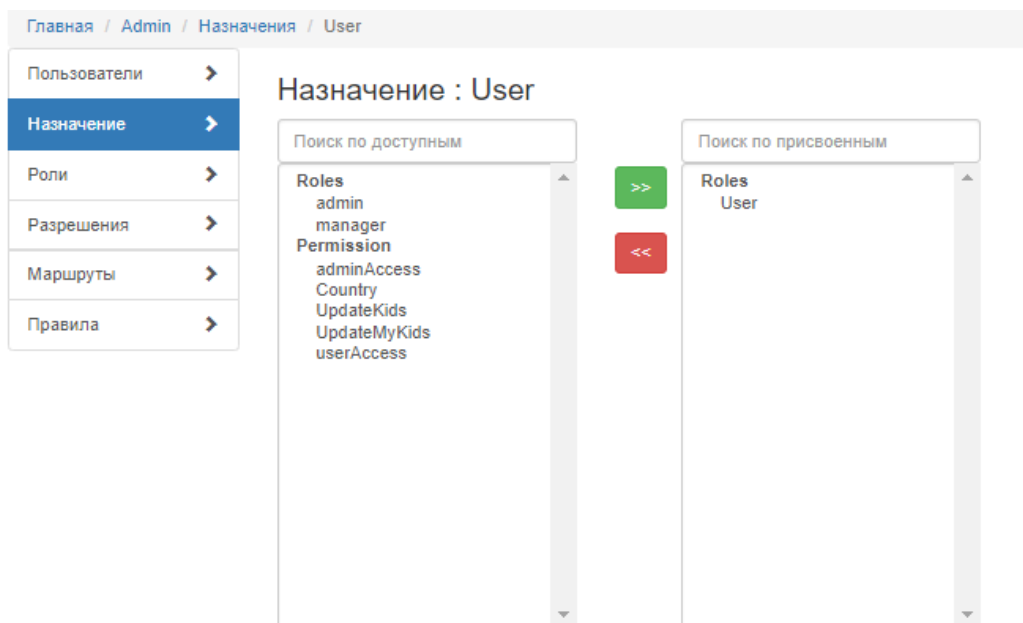


Рисунок 57 - Разрешенные маршруты

Для назначения прав доступа для пользователей, сначала необходимо выбрать разрешенные маршруты, представленные на рисунке 58.

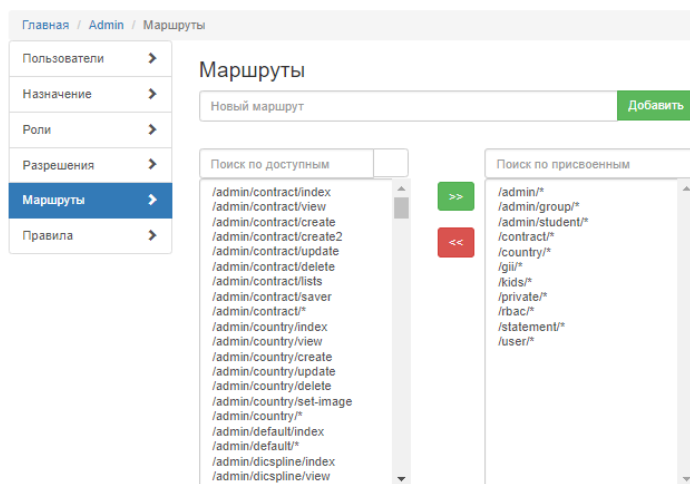


Рисунок 58 - Разрешенные маршруты

Затем ролям присвоить маршруты и разрешения, а в назначениях присвоить роли пользователям. При регистрации пользователю по умолчанию назначается роль User, которая даёт права на доступ в личный кабинет и подачу документов на поступление в школу.

Работа с заявлением, договором и приказом

Для просмотра выбранных заявлений необходимо выбрать пункт «Заявления», откроется таблица с поданными заявлениями, представленная на рисунке 59.













#	Заявление №	Пользователь	Направление	Ребёнок	Дата	Статус заявления	Отчисление/ Зачисление	
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
1	98	Владик нагайченко	География	ГОША	2018-12-03	Рассмотрено	На зачисление	  
2	99	Владик нагайченко	Информатика	ГОША	2018-12-03	Рассмотрено	На зачисление	  
3	100	Владик нагайченко	Химия	Владос	2018-12-05	Рассмотрено	На зачисление	  
4	101	ТЕСТИК	География	РЕБЕНОК	12.12.2018	Рассмотрено	На зачисление	  

Рисунок 59 - Поданные заявления

Для сортировки заявлений по направлению, дате, статусу и т.д., необходимо выбрать из выпадающего списка нужное значение или ввести данные в поля для поиска, представленные на рисунке 60.

Показаны записи 1-8 из 8.




#	Заявление №	Пользователь	Направление	Ребёнок	Дата	Статус заявления	Отчисление/ Зачисление	
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
1	101	ТЕСТИК	География	РЕБЕНОК	12.12.2018	<ul style="list-style-type: none">На рассмотренииНе подписанПодписанРассмотрено	На зачисление	  

Рисунок 60 - Сортировка

Для просмотра заявления в печатном виде необходимо нажать на «Просмотр». После проверки заявления нажать кнопку «Изменить» и изменить статус заявления на «Рассмотрено, что представлено на рисунке 61. После нажать конку «Сохранить».

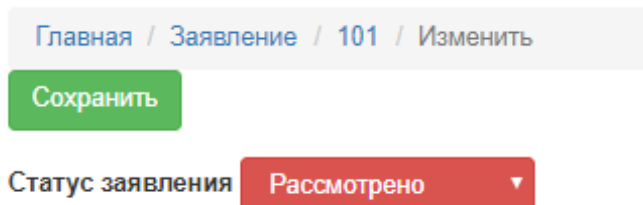


Рисунок 61 - Установка статуса заявления

Для создания договора по выбранному заявлению нажать на кнопку «Договор», представленную на рисунке 62.

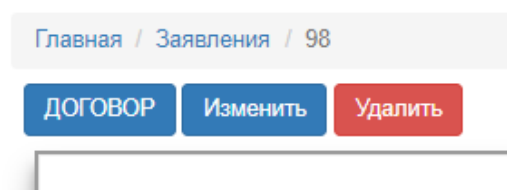






Рисунок 62 - Создание договора

Далее необходимо выбрать из выпадающих списков заявление, статус договора и статус оплаты, которые изображены на рисунке 63. Остальные данные заполняются автоматически.

The image shows a web form titled 'Формирование договора'. It contains three dropdown menus. The first is labeled 'Заявление' and has the text 'РОДИТЕЛЬ: Владик нагайченко РЕБЕНОК ГОША НАПРАВЛЕНИЕ География ЗАЯВЛЕНИЕ № 98'. The second is labeled 'Статус' and has the text 'Не подписан'. The third is labeled 'Статус оплаты' and has the text 'Не оплачен'.

Рисунок 63 - Формирование договора

Формирование приказа на зачисление осуществляется на вкладке «Договоры», где с помощью галочек выбираются необходимые договоры. Также выбирается из выпадающего списка группа для зачисления, после чего необходимо нажать кнопку «Занести», что представлено на рисунке 64.

16	<input checked="" type="checkbox"/>	70	135	Подписан	Оплачен	ГОША	Владик нагайченко	 
17	<input checked="" type="checkbox"/>	71	136	Подписан	Оплачен	TEST1	Владик нагайченко	 

ФИО ребёнка
07001508

Занести в группу

Рисунок 64 - Формирование приказа на зачисление.

Сформированный приказ можно посмотреть на вкладке «приказ на зачисление».

3.6 Применение методики SWOT анализа для обоснования целесообразности разработки АИС в инжиниринговой школе

Эффективность – выполнение требуемых функций при минимальных затратах ресурсов [44]. Эффективность – одно из наиболее общих экономических понятий, не имеющих пока, по-видимому, единого общепризнанного определения [42]. Это одна из возможных характеристик качества системы, а именно ее характеристика с точки зрения соотношения затрат и результатов функционирования системы [41].

Еще на этапе изучения предметной области необходимо было задуматься о целесообразности и эффективности разработки и внедрения АИС приёма и учёта документов в открытой инжиниринговой школе НИУ «БелГУ». При анализе деятельности инжиниринговой школы, было выяснено, что разработка и внедрение АИС является целесообразной по нескольким причинам.

Во-первых, для данного предприятия характерен довольно большой объем документов для приёма и учета, а автоматизация работы позволяет увеличить производительность труда благодаря:

- ускорению обработки информации;
- сокращению количества ручных операций;

– облегчение сбора, поиска и хранения информации.

Во-вторых, использование АИС позволит сократить затраты предприятия, отказавшись от покупки дорогостоящего ПО и выполнения дополнительных операций в сторонних программных продуктах, не предназначенных напрямую для приёма и учёта.

Используемый в целях анализа среды подход SWOT (название является аббревиатурой первых букв слов в английском переводе: сила (Strength), слабость (Weakness), возможности (Opportunities) и угрозы (Threats)) представляет собой часто применяемый метод, который дает возможность изучить влияние внешней и внутренней среды.

С помощью SWOT-анализа можно определить взаимодействия между сильными и слабыми сторонами, свойственными разработке, а также между возможностями и угрозами. С целью выявления всех перечисленных связей формируется SWOT-матрица [33].

Для реализованного проекта разработки АИС приёма и учёта документов в открытой инжиниринговой школе НИУ «БелГУ», SWOT-матрица представлена в таблице 6.

Проанализировав таблицу 6, представленную ниже, можно сделать вывод, что проект АИС приёма и учёта документов в инжиниринговой школе НИУ «БелГУ» является весьма перспективным и его необходимо реализовать.

Из таблицы 6 видно, что проект не имеет существенных угроз, но есть одна существенная слабая сторона – это необходимость подготовки сотрудников к работе с АИС. В случае реализации проекта, положительные стороны очевидны, так как повышается общая трудоспособность сотрудников, а в связи с этим сотрудник сможет обработать гораздо больший объем данных за определенный период.

На данный момент реализации разработки не мешают серьезные проблемы. Однако, необходимо повысить квалификацию персонала.

Таблица 6 - SWOT-матрица

Сильные стороны	Возможности		Угрозы		Итого
	1. Совершенствование разработки	2. Предпочтение более дорогому ПО	1. Появление новых конкурентов	2. Быстрое устаревание дизайна	
Низкая стоимость разработки.	0	++	++	0	+4
Многофункциональность	+	++	+	++	+6
Простота реализации	+	+	+	+	+4
Высокая скорость работы	+	+	+	+	+4
Низкая вероятность ошибки	+	+	+	+	+4
Итого	+4	+7	+6	+5	+22
Слабые стороны					
1. Необходимость ознакомления с программой	-	--	-	--	-6
2. Необходимость сопровождения системы	-	-	0	--	-4
Итого	-2	-3	-1	-4	-10
Общий итог	+2	+4	+5	+1	+12

Вывод по третьему разделу

В данном разделе выпускной квалификационной работы для АИС приёма и учета документов в открытой инжиниринговой школе НИУ «БелГУ» были разработаны база данных, визуальный интерфейс, серверная часть, шаблоны заявления, договора, квитанции и других документов.

Было выполнено тестирование и отладка программного продукта. Найдены и исправлены незначительные ошибки в поведении программы.

Написано руководство пользователю и администратору, где описан процесс работы с программным продуктом со стороны пользователя и администратора.

Также была обоснована целесообразность разработки автоматизированной информационной системы с экономической точки зрения, проведен SWOT анализ.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была разработана АИС приёма и учёта документов в открытой инжиниринговой школе НИУ «БелГУ». АИС имеет удобный и интуитивно понятный интерфейс.

В процессе разработки и проектирования решены следующие задачи:

– произведен анализ предметной области и аналогов существующих программных средств;

– разработана функциональная и информационная модель;

– выбраны программные средства для реализации;

– разработано ПО;

– произведено тестирование и отладка программного продукта;

– разработана сопровождающая документация;

– обоснована эффективность проекта.

Разработанная система обладает следующими возможностями:

– формирование учётной записи в системе;

– возможность редактирования личных данных;

– возможность родителю внести в систему данные о детях по отдельности;

– возможность формирования заявления;

– возможность выбора направления обучения;

– возможность сформировать договор на каждого ребёнка по отдельности;

– возможность формирования квитанции;

– формирование справки об обучении ребёнка в учреждении;

– ведение факта оплаты;

– ведение факта заключения договора.

– возможность формирования приказа на зачисление;

- формирование группы;
- экспорт документов в формат pdf.

Тестирование на начальных этапах выявило ряд незначительных ошибок в проектировании и реализации, которые в последствие были исправлены.

В результате была достигнута основная цель работы: совершенствование процесса приёма и учёта документов в открытой инженеринговой школе НИУ «БелГУ», путем разработки автоматизированной информационной системы.

Программа успешно работает в стандартном режиме эксплуатации, а также при различных несанкционированных действиях пользователя.

В будущем, возможны улучшения разработанной системы в следующих направлениях:

- добавление электронного журнала;
- формирование расписания;
- формирование отчёта об образовательной деятельности;
- формирование ведомостей на заработную плату сотрудников;
- выдача сертификата об обучении по ДООП.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 7.32 - 2001 Система стандартов по информации, библиотечному и издательскому делу. Отчёт о научно-исследовательской работе. Структура и правила оформления.
2. ГОСТ 19.101 - 77 Единая система программной документации. Виды программ и программных документов.
3. Маккоу, А. Веб-приложения на JavaScript. - М.:Питер, 2012. - 285с.
4. Бенедетти, Р. Изучаем работу с jQuery. - М.:Питер, 2012. - 512с.
5. Бенкен, Е. PHP, MySQL, XML. Программирование для Интернета; БХВ-Петербург - М. 2017. - 336 с.
6. Бибо, Б. jQuery. Подробное руководство по продвинутому JavaScript. - М.: Символ, 2012. - 384с.
7. Веллинг, Л. Разработка веб-приложений с помощью PHP и MySQL / Л. Веллинг, Л. Томсон. – М:Вильямс, 2010. 848с.
8. Дамашке, Г. PHP и MySQL; НТ Пресс - М., 2016. - 569 с.
9. Шлоснейгл, Д. Профессиональное программирование на PHP – М.: Вильямс, 2006. – 345 с.
10. Дронов, В. - HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов. 2011 - 414с.
11. Дронов, В. PHP, MySQL и Dreamweaver. Разработка интерактивных Web-сайтов; БХВ-Петербург - М., 2016. - 480 с.
12. Флэнаган, Д. JavaScript. Подробное руководство, 6-е издание. - Символ-Плюс, 2012. - 1080с.
13. Закас, Н. JavaScript. Оптимизация производительности. - М.:Символ, 2012. - 256с.
14. Зандстра, М. PHP. Объекты, шаблоны и методики программирования; Вильямс - М., 2016. - 560 с.

15. Колисниченко, Д. PHP и MySQL. Разработка Web-приложений; БХВ-Петербург - М., 2017. - 560 с.
16. Крокфорд, Д. JavaScript: сильные стороны. - М.:Питер, 2012. - 176с.
17. Кузнецов, М. PHP 5. Практика создания Web-сайтов; БХВ-Петербург - М., 2017. - 960 с.
18. Кузнецов, М. Объектно-ориентированное программирование на PHP /М. Кузнецов, И. Симдянов. – СПб.: БХВ-Петербург, 2007. 608с.
19. Кузнецов, М. Самоучитель PHP 5/6; БХВ-Петербург - М., 2017. - 972 с.
20. Лабберс, П. HTML5 для профессионалов: мощные инструменты для разработки современных веб-приложений. - М.: «Вильямс», 2011. - 453с.
21. Лагутин, К. Пояснительная записка к курсовому проекту/ К.В. Лагутин. – М.: МГПУ, 2012. 35 с.
22. Ловэйн, П. PHP 5. Объектно-ориентированное программирование. / П.Ловэйн. – М.:ИТ Пресс, 2007. 224с.
23. Локхарт, Д. Современный PHP. Новые возможности и передовой опыт; ДМК Пресс - М., 2016. - 304 с.
24. Ляпин, А. PHP — это просто. Начинаем с видеоуроков (+ CD-ROM); БХВ-Петербург - М., 2017. - 881 с.
25. Ляпин, Д. PHP - это просто. Начинаем с видеоуроков; БХВ-Петербург - М., 2017. - 642 с.
26. Макфарланд, Д. Большая книга CSS3 - М.:Питер, 2015. - 608с.
27. Мейер, Э. - CSS - каскадные таблицы стилей. Подробное руководство. 2008. - 575с.
28. Мишель Е. Д. Изучаем PHP и MySQL / Е. Д. Мишель, А. Ф. Джон. – СПб.: Символ-Плюс, 2008. 442с.
29. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript и CSS; Питер - М., 2017. - 204 с.

30. Ноубл, Д. HTML, XHTML и CSS для чайников, 7-е издание.- М.: «Диалектика», 2011. - 313с.
31. Пакульских, Е Пояснительная записка к курсовому проекту / Е.И.Пакульских. – М.: МГПУ, 2012. 35 с.
32. Прохоренок, Н. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / Н. Прохоренок. – СПб.: БХВ-Петербург, 2010. 900с.
33. Рассел, Д. SWOT-анализ / Джесси Рассел. - М.: Книга по Требованию, 2012. - 448 с.
34. Никсон, Р. Создаем динамические веб–сайты с помощью PHP, MySQL и JavaScript – СПб.: Питер, 2011.– 356 с.
35. Сафронов, М. Разработка веб-приложений в Yii 2; ДМК Пресс - М., 2016. - 392 с.
36. Скляр, Д. Трахтенберг Адам PHP. Рецепты программирования; Питер - М., 2017. - 784 с.
37. Советов, Б. Архитектура информационных систем / Б. Я. Советов, А. И. Водяхо, В. А. Дубенецкий, В. В. Цехановский. – М.: Академия, 2012. 228с.
38. Стефанов, С. JavaScript. Шаблоны, Пер. с англ. – СПб.: Символ–Плюс, 2011. – 272 с., ил.
39. Конверс, Т. PHP 5 и MySQL / Конверс, Т., Джойс П. и Кларк М. Библия пользователя – М.: Вильямс, 2006 – 716 с.
40. Фланаган, Д. Javascript: Подробное руководство. - М.:-O'RELLY, 2008. - 435с.
41. Фленов, М. PHP глазами хакера; БХВ-Петербург - М., 2016. - 991 с.
42. Хадсон, П. PHP. Справочник; КУДИЦ-Пресс - М., 2016. - 448 с.
43. Энди, Х. PHP/MySQL для начинающих; КУДИЦ-Образ - М., 2016. - 384 с.

ПРИЛОЖЕНИЕ А

Визуальный интерфейс АИС

[Редактировать личные данные](#) [Личные данные](#)

ФИО	Нагайченко Владислав
Дата рождения	Дата рождения 11.12.1997
Возраст	32
Номер	89803764346
Email	1147087@bsu.edu.ru
Школа	№1
Адрес	Страна Россия Регион Южный Город Белгород Район Белгородский Улица 12 Корпус 1 Дом 23 Квартира 43

Рисунок А.1 - Информация о детях

[Подробнее...](#)

ФИО	Нагайченко Владислав
Заявление №	123
Направление	технический инжиниринг
Статус заявления	Рассмотрено
Статус заявления	2019-04-03
Копия паспорта	
Копия свидетельства	

Рисунок А.2 - Работа с заявлениями

[Подробнее...](#) [Справка](#) [Квитанция](#)

Договор №	48
ФИО ребенка	ПРОВЕРКА
Направление	География
Статус оплаты	Не оплачен
Статус договора	Подписан

Рисунок А.3 - Работа с договорами

Программы Инжиниринговой Школы

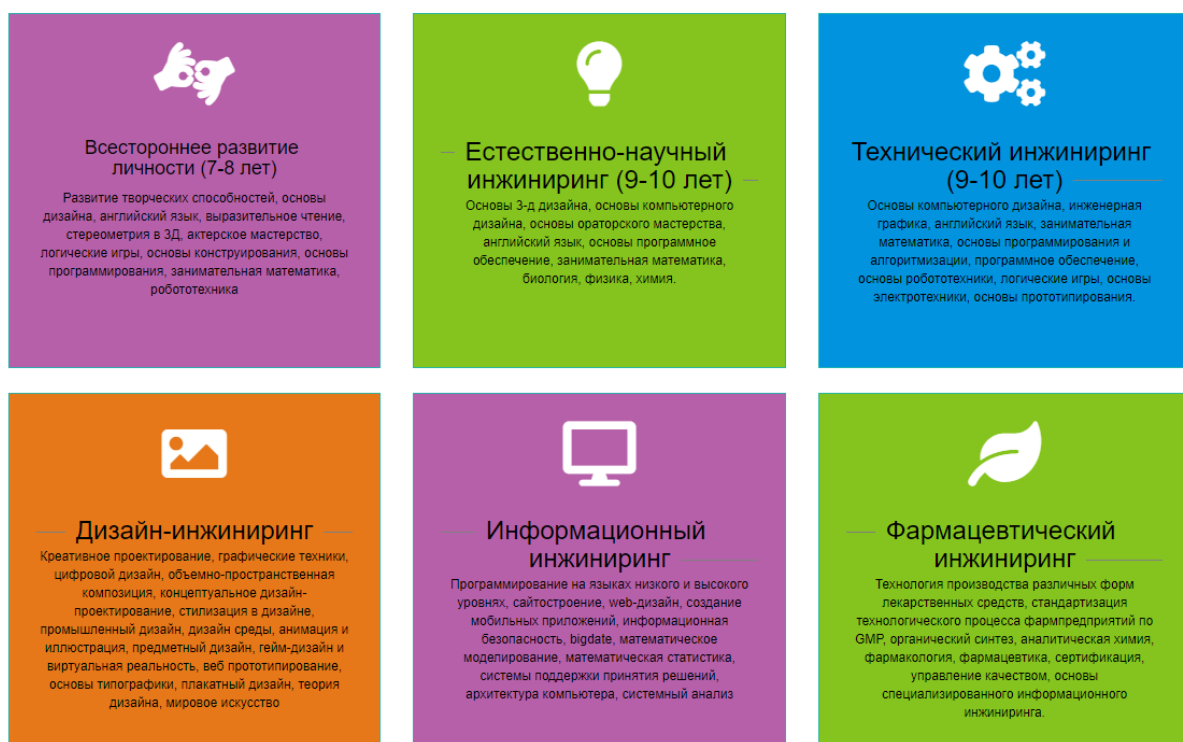


Рисунок А.4 - Программы школы

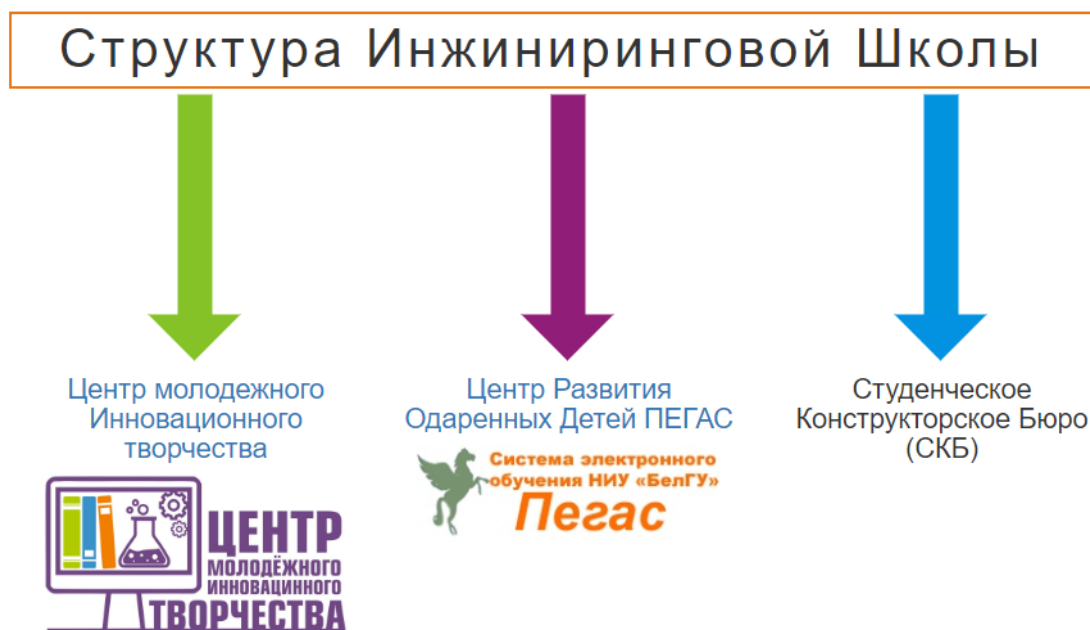


Рисунок А.5 - Структура школы

Занятия

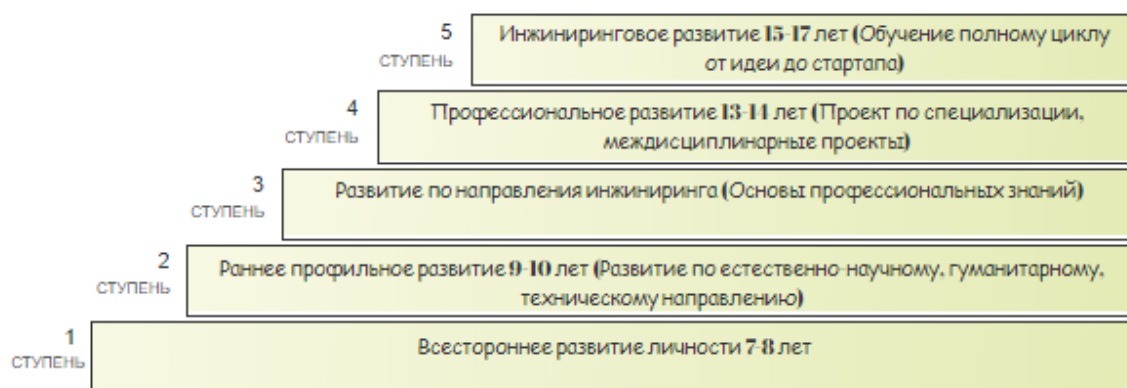


Рисунок А.6 - Ступени обучения

Наши Контакты

Адрес
ул. Студенческая, 11, корп.2, к.2

Директор
Худасова Ольга Геннадьевна

Почтовый ящик
ES@bsu.edu.ru

Заместитель директора
Варфоломеев Александр Валерьевич

Почтовый ящик
Varfolomeev@bsu.edu.ru

Телефон
8-910-711-18-88
(1722) 30-18-01
8-905-173-11-00

График работы
Пн 11:00 - 18:35
Вт 11:00 - 18:35
Ср 11:00 - 18:35
Чт 11:00 - 18:35
Пт 11:00 - 18:35
Сб 10:00 - 16:00
Вс Закрыто

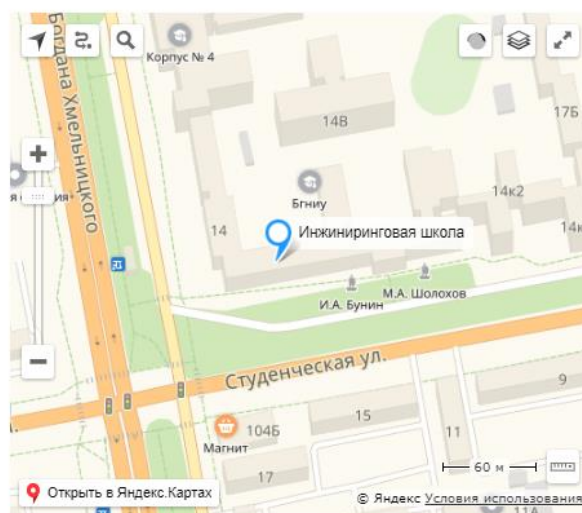


Рисунок А.7 - Контактные данные и карта проезда

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»
ДОГОВОР ОБ ОКАЗАНИИ ПЛАТНЫХ ОБРАЗОВАТЕЛЬНЫХ УСЛУГ № 08104314- 65ИШ 18 >

г. Белгород

ФГАОУ ВО «Белгородский государственный национальный исследовательский университет» (лицензия Федеральной службы по надзору в сфере образования и науки на осуществление образовательной деятельности от 26.05.2016 № 2159, свидетельство Федеральной службы по надзору в сфере образования и науки о государственной аккредитации от 01.08.2016 № 2154), именуемое в дальнейшем НИУ «БелГУ», в лице ректора Полухина О.Н., действующего на основании Устава, с одной стороны, и

Владик нагайченко

имеющего реквизиты Место жительства:(Белгород Паспорт серия: 46646 Паспорт номер: Кем выдан: ИНН: Образование: Номер: 8-800-555-35-35

Владик нагайченко

именуемого (-ой) в дальнейшем Заказчик, с другой стороны,

и MOLOKO MISHA

имеющего реквизиты Место жительства:(МОСКВА Номер: 89803764346)

именуемого (-ой) в дальнейшем Потребитель, с третьей стороны, в совместном упоминании именуемые в дальнейшем Стороны, заключили настоящий Договор о нижеследующем:

1. ПРЕДМЕТ ДОГОВОРА

1.1. По настоящему договору Заказчик оплачивает обучение Потребителя на условиях полного возмещения затрат за предоставляемые образовательные услуги, а НИУ «БелГУ» предоставляет следующие образовательные услуги:

	Описание платной образовательной услуги	Заполняется при оформлении:
1	Вид дополнительной общеобразовательной программы (дополнительная общеразвивающая программа, дополнительная общеобразовательная предпрофессиональная программа)	Дополнительная общеобразовательная (общеразвивающая) программа

Рисунок А.8 - Договор на оказание платных услуг

справка	
2019-05-08	№ 388
<p align="center">ПРОВЕРКА, 10.10.2018 года рождения, является обучающимся по дополнительной общеобразовательной общеразвивающей программе "География " (144 часов), договор об оказании платных образовательных услуг №08104314 - 48 ИШ от 2019 года.</p>	
Начало обучения 1 января	Окончание обучения 1 января
Приказ о зачислении	от _____ № 48
Справка выдана для предъявления по месту пребывания	
Проректор по качеству и дополнительному образованию	В.А. Шаповалов
Исп. Худасова О.Г., тел. +7 (4722)30-18-04	

Рисунок А.9 - Справка об обучении

ПРИЛОЖЕНИЕ В

Программный код

```
<?php
return [
    'class' => 'yii\db\Connection',
    'dsn' => 'mysql:host=localhost;dbname=mydb',
    'username' => 'root',
    'password' => '',
    'charset' => 'utf8'];
```

```
?php
$params = require __DIR__ . '/params.php';
$db = require __DIR__ . '/db.php';
$config = [
    'id' => 'basic',
    'basePath' => dirname(__DIR__),
    'bootstrap' => ['log'],
    'language' => 'ru',
    'aliases' => [
        '@bower' => '@vendor/bower-asset',
        '@npm' => '@vendor/npm-asset',
    ],
    'components' => [
        'assetManager' => [
            'class' => 'yii\web\AssetManager',
            'forceCopy' => true,
            'appendTimestamp' => true,
        ],
        'authManager' => [
            'class' => 'yii\rbac\DbManager',
        ],
    ],
    'request' => [
```

```

        'cookieValidationKey' => 'Zw2T5vkIze_KFhCup1LcJB2KSdJedzev',
        'enableCsrfValidation' => false,
    ],
    ],
    'db' => $db,
],
        'modules' => [
    'admin' => [
        'class' => 'app\modules\admin\Module',
    ],
        'rbac' => [
    'class' => 'mdm\admin\Module',
        'controllerMap'=>[
            'assignment'=>[
                'class' => 'mdm\admin\controllers\AssignmentController',
                'idField' => 'id',
                'usernameField' => 'username',
            ],
        ],
    ],
        'layout' =>'left-menu',
        'mainLayout'=> '@app/views/layouts/admin.php',
    ]
],
        'as access' => [
    'class' => 'mdm\admin\components\AccessControl',
    'allowActions' => [
        'site/*'
    ]
],
    'params' => $params,
];

```

```

class Dicspline extends \yii\db\ActiveRecord
{

```



```

public static function tableName()
{
    return 'dicspline';
}
public function rules()
{
    return [
        [['name', 'direction_iddirection'], 'required'],
        [['direction_iddirection'], 'integer'],
        [['name', 'time'], 'string', 'max' => 45],
        [['direction_iddirection'], 'exist', 'skipOnError' => true, 'targetClass' =>
Direction::className(), 'targetAttribute' => ['direction_iddirection' => 'iddirection']],
    ];
}
public function attributeLabels()
{
    return [
        'iddicspline' => 'Id',
        'name' => 'ДИСЦИПЛИНА',
        'time' => 'К-во часов',
        'direction_iddirection' => 'Направление',
    ];
}
public function getDirectionIdirection()
{
    return $this->hasOne(Direction::className(), ['iddirection' =>
'direction_iddirection']);
}
}

class DicsplineSearch extends Dicspline
{
    public function rules()
    {
        return [

```

```

        [['iddicspline', 'direction_iddirection'], 'integer'],
        [['name', 'time'], 'safe'],
    ];
}
public function scenarios()
{
    return Model::scenarios();
}
public function search($params)
{
    $query = Dicspline::find();
    $dataProvider = new ActiveDataProvider([
        'query' => $query,
    ]);
    $this->load($params);
    if (!$this->validate()) {
        return $dataProvider;
    }
    $query->andFilterWhere([
        'iddicspline' => $this->iddicspline,
        'direction_iddirection' => $this->direction_iddirection,
    ]);
    $query->andFilterWhere(['like', 'name', $this->name])
        ->andFilterWhere(['like', 'time', $this->time]);
    return $dataProvider;
}
}

```

```

<?php
namespace app\controllers;
use Yii;
use yii\filters\AccessControl;
use yii\web\Controller;
use yii\web\Response;

```

```

use yii\filters\VerbFilter;
use app\models\LoginForm;
use app\models\ContactForm;
use app\models\EntryForm;
use yii\data\Pagination;
use app\models\Country;
use app\models\SignupForm;
use app\models\User;
class SiteController extends Controller
{
    public function behaviors()
    {
        return [
            'access' => [
                'class' => AccessControl::className(),
                'only' => ['logout'],
                'rules' => [
                    [
                        'actions' => ['logout'],
                        'allow' => true,
                        'roles' => ['@'],
                    ],
                ],
            ],
            'verbs' => [
                'class' => VerbFilter::className(),
                'actions' => [
                    'logout' => ['post'],
                ],
            ],
        ];
    }

    public function actions()
    {

```

```

return [
    'error' => [
        'class' => 'yii\web\ErrorAction',
    ],
    'captcha' => [
        'class' => 'yii\captcha\CaptchaAction',
        'fixedVerifyCode' => YII_ENV_TEST ? 'testme' : null,
    ],
];
}

```

```

public function actionIndex()
{
    $query = Country::find();

    $pagination = new Pagination([
        'defaultPageSize' => 5,
        'totalCount' => $query->count(),
    ]);
    $countries = $query->orderBy('id')
        ->offset($pagination->offset)
        ->limit($pagination->limit)
        ->all();

    return $this->render('index2', [
        'countries' => $countries,
        'pagination' => $pagination,
    ]);
}

```

```

public function actionLogin()
{
    if (!Yii::$app->user->isGuest) {
        return $this->goHome();
    }
}

```

```

$model = new LoginForm();
if ($model->load(Yii::$app->request->post()) && $model->login() {
    // return $this->goBack();
        $id = Yii::$app->user->identity->id;
            //return $this->redirect(['user/update', 'id' => $id]);
            return $this->redirect(['user/index']);
    }
    $model->password = "";
    return $this->render('login', [
        'model' => $model,
    ]);
}
public function actionLogout()
{
    Yii::$app->user->logout();

    return $this->goHome();
}
public function actionContact()
{
    $model = new ContactForm();
    if ($model->load(Yii::$app->request->post()) && $model->contact(Yii::$app-
>params['adminEmail'])) {
        Yii::$app->session->setFlash('contactFormSubmitted');
        return $this->refresh();
    }
    return $this->render('contact', [
        'model' => $model,
    ]);
}
public function actionAbout()
{
    return $this->render('about');
}
    public function actionEntry()

```

```

    {
        $model = new EntryForm();
        if ($model->load(Yii::$app->request->post()) && $model->validate()) {
            return $this->render('entry-confirm', ['model' => $model]);
        } else {

            return $this->render('entry', ['model' => $model]);
        }
    }
}

public function actionSignup(){
    if (!Yii::$app->user->isGuest) {
        return $this->goHome();
    }
    $model = new SignupForm();
    if($model->load(\Yii::$app->request->post()) && $model->validate()){
        $user = new User();
        $user->username = $model->username;
        $user->password_hash = \Yii::$app->security->generatePasswordHash($model->password_hash);
        if($user->save()){
            $auth = Yii::$app->authManager;
            $role = $auth->getRole('User');
            $auth->assign($role, $user->getId());
            return $this->goHome();
        }
    }
    return $this->render('signup', compact('model'));
}

public function actionInfo($population)
{
    $country = Country::findOne(['population' => $population]);

    return $this->render('info',['country'=>$country, ]);
}
}

```

```

public function actionCreate()
{
    $model = new Statement();
    $parents=Yii::$app->user->identity->id;
    $query=('select
        contract.id,
        contract.idStatement,
        statement_status.name,
        user.username,
        user.name,
        kids.KidsName,
        statement.idUsers ,
        direction.direction_name
    from contract,user,statement,Kids ,direction, statement_status
    where
        contract.idStatement = statement.idStatement
        and statement.idUsers = user.id
        and contract.Status = statement_status.idstatement_status
        and statement.idKids = kids.idKids
        and statement.iddirection = direction.iddirection
        and statement.idUsers=
        '.$parents);
    $post=Yii::$app->db->createCommand($query)->queryAll();
}

```

```

public function actionSaver( $num ) {
    $check=Yii::$app->request->post('selection');
    $request = Yii::$app->request->post('Contract');
        foreach($request as $value){
            $Group = Group::find()
            ->where(['idgroup' => $value])
            ->all();
            foreach($Group as $values){
                $idGroup = $values->idgroup;}
        }
}

```

```

        }
        var_dump($idGroup);
        foreach($check as $value){
$KidsName = Contract::find()
->where(['id' => $value])
->all();
foreach($KidsName as $name){
$model = new Student;
$model->IdContracts = $name->id;
$model->IdGroup = $idGroup ;
$model->KidsName = $name->KidsName;
$model->save();    } } }

```

```

<?= GridView::widget([
    'dataProvider' => $dataProvider,
    'filterModel' => $searchModel,
    'columns' => [
        ['class' => 'yii\grid\SerialColumn'],
        'direction_name',
        'price',
        ['class' => 'yii\grid\ActionColumn'],
    ],
]); ?>

```

```

<?php $form = ActiveForm::begin([
    'action' => ['index'],
    'method' => 'get',
]); ?>
<?= $form->field($model, 'iddirection') ?>
<?= $form->field($model, 'direction_name') ?>
<?= $form->field($model, 'price') ?>
<div class="form-group">
    <?= Html::submitButton('Search', ['class' => 'btn btn-primary']) ?>
    <?= Html::resetButton('Reset', ['class' => 'btn btn-default']) ?>

```



```
</div>
```

```
<?php ActiveForm::end(); ?>
```

```
protected function findModel($id)
{
    if (($model = Statement::findOne($id)) !== null) {
        return $model;
    }
    throw new NotFoundHttpException('The requested page does not exist.');
```

```
<?php
namespace app\models;
use Yii;
use yii\base\Model;
use yii\web\UploadedFile;
class Image extends Model{
public $image;
public function rules(){
return [
[['image'],'file','extensions'=>'jpg,png']];
} public function attributeLabels(){ return [
'image'=>'Паспорт', ] }
public function uploadFile($file,$currentImage){
$this->image =$file;
if($this->validate()){
if(file_exists((Yii::getAlias('@webroot').'/uploads/'.$currentImage))){
}
$filename = strtolower(md5(uniqid($file->basename))).'.'.$file->extension;
$file->saveAs(Yii::getAlias('@webroot').'/uploads/'.$filename);
return $filename;
} }
private function getFolder(){
return Yii::getAlias('@webroot').'/uploads/'; }
```

Выпускная квалификационная работа выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

« ___ » _____ Г.

(подпись)

(Ф.И.О.)