

Министерство науки и высшего образования Российской Федерации

Федеральное бюджетное образовательное учреждение высшего образования
«Магнитогорский государственный технический
университет им. Г.И. Носова»

Многопрофильный колледж

Отделение №1 «Информационные
технологии»
ПЦК «Информатика и вычислительная
техника»

Допустить к защите
Заведующий отделением
_____/ Н.В. Сидорова
« ____ » _____ 2019г.

ДИПЛОМНЫЙ ПРОЕКТ
Д.09.02.03.19.ДП.ПЗ

Студента _____ Лысикова Ивана Евгеньевича

На тему _____ Разработка модуля «Паспортизация помещений» в рамках информатизации
деятельности Многопрофильного колледжа

Состав дипломного проекта:

1. Пояснительная записка на _____ страницах
2. Графическая часть на _____ листах

Руководитель _____ преподаватель МпК Зорина Ирина Геннадьевна
(подпись, дата, должность, ученая степень, звание, Ф.И.О.)

Консультанты _____ преподаватель МпК Котельникова Юлия Михайловна
_____ к.п.н., преподаватель Щеблева Светлана Викторовна
(подпись, дата, должность, ученая степень, звание, Ф.И.О.)

Рецензент _____
(подпись, дата, должность, ученая степень, звание, Ф.И.О.)

Председатель ПК/ПЦК _____ преподаватель МпК Зорина Ирина Геннадьевна
(подпись, дата, должность, ученая степень, звание, Ф.И.О.)

Отметка нормоконтролера
_____/С.В. Щеблева/
« ____ » _____ 2019

Студент _____
(подпись)
« ____ » _____ 2019г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
1 АНАЛИТИЧЕСКАЯ ЧАСТЬ	8
1.1 Описание и анализ предметной области.....	8
1.2 Анализ различных методик формирования паспортов учебных помещений.....	9
1.3 Обоснование актуальности разработки приложения	13
1.4 Обоснование выбора средств автоматизации для создания программного продукта	14
1.5 Цель и задачи проекта	16
2 ПРОЕКТНАЯ ЧАСТЬ.....	19
2.1 Моделирование программного обеспечения.....	19
2.1.1 Построение информационной модели	19
2.1.2 Проектирование диаграммы прецедентов	19
2.1.3 Проектирование схемы бизнес-процессов и структуры паспорта помещения	20
2.2 Логическое и физическое проектирование базы данных.....	23
2.3 Разработка базы данных	33
2.4 Разработка физической и логической структуры сайта	46
2.5 Создание запросов	50
2.6 Проектирование пользовательского интерфейса.....	52
2.7 Руководство пользователя.....	54
2.7.1 Требования к уровню подготовки пользователя.....	55
2.7.2 Требования к программной среде	55
2.7.3 Описание операций, выполняемых пользователями модуля «Паспортизация помещений»	55
2.8 Руководство администратора.....	62
3 ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ.....	67
3.1 Законодательство в области защиты информации	67
3.2 Общие положения и уровни информационной безопасности	67
3.3 Виды угроз информационной безопасности	69
3.4 Организация доступа пользователей к информационным системам.....	70
3.5 Обеспечение процедуры резервирования и копирования информации, применение сканеров уязвимости, обеспечение защиты от сбоев в электропитании	72
3.6 Информационная безопасность модуля «Паспортизация помещений»	74
4 ЭКОНОМИЧЕСКАЯ ЧАСТЬ	76
ЗАКЛЮЧЕНИЕ.....	82

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	83
ПРИЛОЖЕНИЕ А Листинг 1	85
ПРИЛОЖЕНИЕ Б Листинг 2.....	91

ВВЕДЕНИЕ

Информационные технологии в образовании в настоящее время являются необходимым условием в информационной цивилизации. Современные технологии и телекоммуникации позволяют изменить характер организации учебно-воспитательного процесса, полностью погрузить обучаемого в информационно-образовательную среду, повысить качество образования, мотивировать процессы восприятия информации и получения знаний.

Процессы взаимодействия с информацией базируются на информационных и коммуникационных технологиях. Информационные и коммуникационные технологии – это процессы поиска, сбора, хранения, распространения и обработки информации.

Информационные технологии действуют в образовании, культуре, бизнесе, государстве и во многих других сферах жизнедеятельности общества. Они оказывают сильное влияние на информатизацию образования.

Информатизация образования – это внедрение информационных технологий в сферу образования. Основная цель с точки зрения информационных технологий повышение качества образования.

Новые информационные технологии создают среду компьютерной и телекоммуникационной поддержки организации и управления в различных сферах деятельности, в том числе в образовании.

Интеграция информационных технологий в образовательные программы осуществляется на всех уровнях: школьном, вузовском и послевузовском обучении.

Проникновение современных информационных технологий в сферу образования позволяет педагогам качественно изменить содержание, методы и организационные формы обучения.

Отечественная система образования стремительными шагами вливается в процесс создания современной национальной информационной среды, совершенствуя методы и модели педагогической практики, внедряя дистанционное обучение и повышая квалификацию преподавателей в области современных информационных технологий. Кроме того, повсеместное распространение средств современной вычислительной техники и развитие информационных технологий стимулируют разработку автоматизированных информационных систем для решения задач управления образованием. Умение воспользоваться всем вышеперечисленным определяет информационную культуру педагога.

Автоматизированные информационные системы – системы сбора, хранения, переработки, передачи и представления информации, базирующиеся на электронной технике

и системах телекоммуникации. Их совершенствование и развитие оказывает существенное влияние на характер производства, научных исследований, образование, культуру, быт, социальные отношения и структуры. Это оказывает как прямое влияние на содержание образования, связанное с уровнем научно-технического прогресса, так и косвенное, связанное с появлением новых информационных и производственных технологий, с появлением новых профессиональных умений и навыков, потребность в которых резко возрастает.

Обучение студентов по программам подготовки специалистов должно осуществляться в помещениях (кабинеты, лаборатории, мастерские и др.), соответствующих требованиям ФГОС. Для проведения всех видов лабораторных работ и практических занятий, дисциплинарной, междисциплинарной и модульной подготовки, учебной практики, предусмотренных учебным планом, образовательная организация должна располагать материально-технической базой, отвечающей действующим санитарным и противопожарным нормам.

Задачи паспортизации помещений учебного заведения:

- анализировать состояние помещения;
- обеспечивать соответствие требованиям стандартов образования;
- обеспечивать соответствия санитарно-эпидемиологическим требованиям;
- требованиям пожарной безопасности;
- определить основные направления работы по приведению помещения в

соответствие требованиям учебно-методического, материально-технического обеспечений образовательного процесса по реализуемым программам.

1 АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1 Описание и анализ предметной области

В соответствии с ФГОС образовательная организация, реализующая программы подготовки специалистов, должна располагать материально-технической базой, обеспечивающей проведение всех видов лабораторных работ и практических занятий, дисциплинарной, междисциплинарной и модульной подготовки, учебной практики, предусмотренных учебным планом образовательной организации. Материально-техническая база должна соответствовать действующим санитарным и противопожарным нормам. При этом в каждом ФГОС оговорен перечень кабинетов, лабораторий, мастерских и других помещений для организации учебного процесса.

Учебное помещение – это помещение, оснащенное наглядными пособиями, учебным оборудованием, мебелью и техническими средствами обучения, в котором проводится учебная, факультативная и внеклассная работа со студентами в полном соответствии с действующими ФГОС СПО, учебными планами, основными профессиональными образовательными программами и программами подготовки квалифицированных рабочих, служащих.

Обязательная паспортизация помещений аудиторного фонда учебного заведения разрабатывается в соответствии с действующим законодательством РФ:

- типовые положения об образовательном учреждении высшего профессионального образования (вышем учебном заведении), утвержденное постановлением Правительства Российской Федерации от 14 февраля 2008 г. № 71 и среднего профессионального образования (среднем специальном учебном заведении), утвержденное постановлением Правительства РФ от 18 июля 2008 г. № 543 [4,5];

- Закон РФ 29 декабря 2012 года N 273-ФЗ "Об образовании в Российской Федерации" (с изменениями и дополнениями) [6];

- Федеральные государственные образовательные стандарты высшего и среднего профессионального образования;

- санитарно-эпидемиологические правила и нормативы СанПиН 2.4.2.2821-10 «Санитарно-эпидемиологические требования к условиям и организации обучения в общеобразовательных учреждениях» (с изменениями и дополнениями) [7];

- Устав образовательной организации.

Паспортизация помещений учебного заведения проводится ежегодно в начале учебного года, в настоящий момент результаты данной процедуры учитываются вручную в паспортах в бумажном виде.

Цель паспортизации помещений учебного заведения – проанализировать состояние помещения, его готовность к обеспечению требований стандартов образования, соответствия санитарно-эпидемиологическим требованиям, требованиям пожарной безопасности, определить основные направления работы по приведению помещения в соответствие требованиям учебно-методического, материально-технического обеспечений образовательного процесса по реализуемым программам.

При лицензировании и аккредитации учебного заведения особое внимание уделяется отчетам по использованию аудиторного фонда.

В каждой образовательной организации создается своя форма паспорта аудитории, где, как правило, фиксируются такие характеристики, как информация о номере кабинета и его функциональном назначении; фамилия ответственного за кабинет; площадь кабинета; перечень имеющейся в нем мебели, оборудования, в том числе, компьютерного, приборов, технических средств, наглядных пособий, таблиц, карт, учебников, методических пособий и т. д. с указанием количества и инвентарных номеров. Не всегда данное представление отражает в полной мере характеристику помещения и соответствие его назначению.

1.2 Анализ различных методик формирования паспортов учебных помещений

При оценке существующих вариантов паспортизации учебных помещений были проанализированы различные методики формирования паспортов. В основном, это паспорта, созданные в формате .doc по существующим правилам для данного учебного заведения. Для отдельных заведений эти правила оформляются в виде положений об учебных кабинетах и лабораториях. Другие, более близкие к званию «системы», представляют собой базу данных без какого-либо интерфейса. Навигация таких систем максимально проста, в связях таблиц множество ошибок, материал достаточно скудный, в связи с чем, мотивация заполнять качественно и ответственно у сотрудников пропадает. Примеры паспортов представлены на рисунках 1.1-1.4.

ПАСПОРТ КАБИНЕТА БИОЛОГИИ
НА 2013/2014 УЧЕБНЫЙ ГОД

Фамилия, имя, отчество заведующей кабинетом: Хабенко Надежда Николаевна
Класс, ответственный за кабинет: 9 «б»

Классы, для которых оборудован кабинет: 6 – 11 кл.

Площадь кабинета 53 кв.м. + лаборантская площадью 16,7 кв.м
Число посадочных мест: 30

ИНСТРУКЦИЯ
Правила поведения в кабинете
биологии

1. В кабинет входить только после разрешения учителя.
2. Во время перемены приготовить ученические принадлежности к уроку.
3. В кабинете не разрешается во время перемены играть в подвижные игры.
4. Категорически запрещается открывать оконные рамы, портить мебель.
5. Нельзя без разрешения учителя пользоваться электрической розеткой.
6. По окончании урока оставить своё рабочее место в чистоте, убрать бумажки, опилки от карандаша в корзину для мусора, подвинуть стул к парте.

Инструкцию составила
(Н.Н.Хабенко)

Рисунок 1.1 – Фрагмент паспорта кабинета школы №63, г. Красноярск

Документация по организации работы учебного кабинета

Требования к учебным кабинетам

1. Наличие паспорта и плана работы учебного кабинета на учебный год.
2. Соблюдение правил техники безопасности, санитарно - гигиенических норм в учебном кабинете.
3. Укомплектованность кабинета оборудованием и учебно - методическим комплексом средств обучения.
4. Соответствие оборудования и учебно - методического комплекса средств обучения по профилю кабинета.
5. Наличие в кабинете комплекса материалов для диагностики качества обучения по профилю кабинета.
6. Обеспеченность кабинета учебниками, дидактическими и раздаточными материалами по его профилю.
7. Наличие в кабинете стендового материала, который носит обучающий характер:
 - рекомендаций по выполнению домашних работ;
 - рекомендаций по подготовке к различным формам учебно - познавательной деятельности (практикум, семинар, лабораторная работа, тестирование, зачет, коллоквиум, собеседование, экзамен).
8. Расписание работы учебного кабинета.

Документация учебного кабинета

1. Паспорт учебного кабинета.
2. Инвентарная ведомость на имеющееся оборудование.
3. Инструкция по правилам техники безопасности при работе в учебном кабинете (в кабинетах физики, химии, биологии, учебных мастерских, спортивном зале).
4. График работы учебного кабинета.
5. Акт приёмки учебного кабинета администрацией школы.
6. План работы учебного кабинета на учебный год.

Структура паспорта учебного кабинета

МУНИЦИПАЛЬНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ – СРЕДНЯЯ
ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА № 17

ПАСПОРТ КАБИНЕТА № _____

Учебный год:

Ф.И.О. учителей, ответственных за кабинет

Ответственный класс:

Оценка состояния кабинета

Сентябрь	Январь	Июнь

График занятости кабинета № ... на первое (второе)
полугодие 200 /0 учебного года

Урок	Поне- дельник	Вторник	Среда	Четверг	Пятница	Суббота
	Класс	Класс	Класс	Класс	Класс	Класс
1						
2						
3						
4						
5						
6						
7						
8						
9						

Рисунок 1.2 – Фрагмент паспорта кабинета Хилинской средней школы

муниципальное автономное общеобразовательное учреждение
«Средняя общеобразовательная школа № 56 с углубленным изучением математики»
города Магнитогорска

Паспорт учебного кабинета английского языка

кабинет № 11
Ответственный за кабинет:
Сафонова Ольга Ивановна

1. Общая характеристика кабинета

Площадь	33,4 м²
Освещение кабинета	Естественное, 2 окна Искусственное, 16 источников освещения Дополнительное освещение перед доской
Воздушно-вентиляционный режим кабинета	стационарный
Покрывание стен кабинета	обои
Покрывание пола кабинета	линолеум
Раковина с подведенным водоснабжением	нет
Обеспечение питьевого режима	нет

2. Материально-техническое обеспечение учебной деятельности

<i>Рабочее место педагога</i>	
Стол письменный (учительский)	1
Стул (учительский)	1
Классная доска	1
<i>Рабочее место обучающихся</i>	
Стул ученический	18
Стол ученический	9

3. Программно-методическое обеспечение

Федеральный компонент государственного образовательного стандарта начального общего, основного общего и среднего (полного) общего образования	1
Примерная программа основного общего образования. Английский язык	1
Примерная программа основного общего образования. Английский язык. Базовый уровень	1
Примерная программа среднего (полного) общего образования по английскому языку. Базовый уровень.	1
Программа курса «Английский язык» 5-9 классы / авт.-сост. Ю.А.Комарова, И.В.Ларионова. — М.: ООО «Русское слово-учебник», 2014. — 104 с. (Инновационная школа).	
Программа курса «Английский язык» 10-11 классы. Базовый уровень / авт.-сост. Ю.А.Комарова, И.В.Ларионова. — М.: ООО «Русское слово-учебник», 2014. — 56 с. (Инновационная школа).	
Рабочая программа по английскому языку для 6-9 класса на 2015-2016 учебный год	1
Рабочая программа по английскому языку для 10-11 класса на 2015-2016 учебный год	1

4. Специализированный программно-аппаратный комплекс педагога

Персональный компьютер с предустановленным программным обеспечением и выходом в интернет FLATRON L1922S	1
Медиа проектор BANO	1
Экран BRAUN Photo technik	1
Колонки	2

Рисунок 1.3 – Фрагмент паспорта кабинета школы №56, г. Магнитогорск

Образец паспорта кабинета

Кабинет/ Лаборатория/ Полигон/ Студия «_____»
Ответственный за кабинет: _____

1. Материально-техническое состояние кабинета/ лаборатории/ полигона/ студии:

Адрес: _____

Номер кабинета: _____

Объемные показатели кабинета: площадь _____ м²
количество рабочих мест — _____

Освещение кабинета: естественное — _____ окна искусственное — _____ ламп

Воздушно-вентиляционный режим кабинета: _____

Окраска стен: _____

Покраска пола: _____

Стационарное оборудование: _____

Наличие огнетушителей: _____

Наличие аптечки: _____

Наличие подсобного помещения: _____

Сроки проведения ремонта кабинета: _____ год

2. Дидактическая система кабинета/ лаборатории/ полигона/ студии:

2.1. Оборудование и оснащение

2.2. Учебно-методические материалы

ФГОС СПО _____

Выписка из квалификационной характеристики специальности _____

Учебные программы, программы модулей (разделы модулей) _____

2.3. Наглядные и технические средства обучения

Изобразительные (образы и условно-схематические) средства обучения (фотографии, картины, плакаты, рисунки, карты, таблицы, диаграммы, схемы, графики, компьютерные презентации и др.) _____

Натуральные средства обучения (природные объекты, материалы, приборы и др.) _____

Локальные ТСО (экранные (диапозитивы, диафильмы и пр.), звуковые (грампластинки, магнитофонные записи, лазерные диски и пр.), аудиовизуальные (кинофильмы, видеофильмы)) _____

Вычислительная техника и персональные _____

Рисунок 1.4 – Фрагмент образца паспорта кабинета факультета СПО ФГБОУ ВО «СГЭУ»

В 2017 году была разработана автоматизированная информационная система

«Паспортизация учебных помещений», но она содержала ряд проблем, которые мешали правильной работе системы. Фрагменты системы приведены на рисунках 1.5 – 1.7

Основные проблемы:

- аудитории фиксировались за ответственными;
- отсутствовала привязка аудиторного фонда к типам помещений, определенных ФГОС;
- отсутствовала привязка инструкций к типам помещений, определенных ФГОС;
- не полностью реализована обработка справочной информации;
- перегружена процедура заполнения паспортов преподавателями;
- код программы избыточен и содержал ошибки.

Помещение	A207		
Тип	Лаборатория		
Специальность	09.02.03 Программирование в компьютерных системах		
Лаборатория	системного и прикладного программирования		

2. Программно-методическое обеспечение

2.2 Программы профессиональных модулей, дисциплин и практик

№ п/п	Дисциплина	№ протокола и дата утверждения	
1	МДК.01.01 Системное программирование	№1 от 7.9.2016	Удалить

Выберите ПМ, дисциплину или практику

№1 от 7.9.2016
 № протокола ПЦК: Дата утверждения:

2.3 Перечень таблиц, плакатов и других дидактических материалов

Наименование	Количество
--------------	------------

3. Организационная оснащенность

3.1 Инструкции по ТБ, ОТ, ПБ и ОТ при проведении лабораторно-практических работ

СМК-ИОТ-01-12 Инструкция по оказанию доврачебной помощи при несчастных случаях

СМК-ИОТ-ПБЭМ-01-12 По охране труда для пользователей персональных электронно-вычислительных машин (ПЭВМ)

СМК-МПБ-01-12 Инструкция о мерах пожарной безопасности

Инструкции по ТБ, ОТ, ПБ и ОТ


3.2 Журнал регистрации инструктажей на рабочем месте

№ ведется с

3.3 Иллюстративная часть

Наименование	Инвентаризационный номер	Количество
Доска магнитная	E01360749	<input type="text"/>
Карниз	B000468.1	<input type="text"/>
Стул офисный	B010000000001516	<input type="text"/>
Стол ученический	B010000000001513	<input type="text"/>
Кресло офисное	B2101060233	<input type="text"/>
Стол компьютерный угловой	B1101060299	<input type="text"/>
Штора	B000046.2	<input type="text"/>

Рисунок 1.5 – Фрагмент формы редактирования паспорта кабинета



Федеральное государственное бюджетное образовательное учреждение высшего образования

МАГНИТОГОРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. Г.И. НОСОВА

Многопрофильный колледж

Паспорт кабинета/лаборатории/мастерской

Редактирование справочников

Добро пожаловать,

Администратор

Выйти

Формирование

Редактирование и печать

МТО

Схема помещения

Отчеты

Справочники

№ п/п	Идентификатор	Аудитория	Тип	Специальность					
1	4	A207	Лаборатория системного и прикладного программирования	09.02.03	Создать на основе	Изменить	Просмотр	Удалить	
2	5	A307	Лаборатория автоматизированных информационных систем [Специальность]	09.02.01	Создать на основе	Изменить	Просмотр	Удалить	
3	6	A307	Лаборатория информационно-коммуникационных систем	09.02.03	Создать на основе	Изменить	Просмотр	Удалить	
4	8	A309	Лаборатория технологии разработки базы данных	09.02.03	Создать на основе	Изменить	Просмотр	Удалить	
5	9	ИН-1	Лаборатория системного и прикладного программирования	09.02.03	Создать на основе	Изменить	Просмотр	Удалить	
6	10	ИН-1	Лаборатория интернет-технологий	09.02.01	Создать на основе	Изменить	Просмотр	Удалить	
7	11	П2	Мастерская салон парикмахерская	44.02.06	Создать на основе	Изменить	Просмотр	Удалить	
8	12	A306	Кабинет вычислительной техники	15.02.07	Создать на основе	Изменить	Просмотр	Удалить	
9	14	A407	Лаборатория системного и прикладного программирования	09.02.03	Создать на основе	Изменить	Просмотр	Удалить	
10	17	A112	Кабинет основ металлургического производства	22.02.01	Создать на основе	Изменить	Просмотр	Удалить	
11	18	У401	Кабинет гуманитарных и социально-экономических дисциплин	21.02.05	Создать на основе	Изменить	Просмотр	Удалить	
12	21	У401	Кабинет социально-экономических дисциплин	38.02.01	Создать на основе	Изменить	Просмотр	Удалить	
13	22	A405	Кабинет структуры транспортной системы	23.02.04	Создать на основе	Изменить	Просмотр	Удалить	
14	23	A405	Кабинет проектирования зданий и сооружений	08.02.01	Создать на основе	Изменить	Просмотр	Удалить	
15	24	A405	Кабинет инженерной графики	23.02.03	Создать на основе	Изменить	Просмотр	Удалить	
16	25	A405	Кабинет инженерной графики	23.02.04	Создать на основе	Изменить	Просмотр	Удалить	
17	26	A405	Кабинет инженерной графики	44.02.06	Создать на основе	Изменить	Просмотр	Удалить	
18	27	A405	Кабинет инженерной графики	09.02.01	Создать на основе	Изменить	Просмотр	Удалить	
			Кабинет информационных технологий в профессиональной	19.02.10	Создать на основе	Изменить	Просмотр	Удалить	

Рисунок 1.6 – Фрагмент «Список паспортов преподавателей»

Помещение	A-015
Тип:	Кабинет
Специальность:	08.02.01 Строительство и эксплуатация зданий и сооружений
Кабинет	социально-экономических дисциплин

Рисунок 1.7 – Фрагмент формы заполнения паспорта

1.3 Обоснование актуальности разработки приложения

Рассмотрев систему заполнения паспортов аудиторий в Многопрофильном колледже ФГБОУ ВО «Магнитогорский государственный технический университет им. Г.И. Носова» были выявлены следующие проблемы:

- отсутствие единой системы заполнения паспортов. При малейшей невнимательности преподавателя к изменениям во ФГОС заполненный им документ теряет свою актуальность. Материально-техническое оснащение хранится либо в разных архивах, либо в электронном виде на разных носителях, вследствие чего тяжело найти нужную информацию;

- хранение многочисленных архивов бумажных носителей;
- неудобство ручного заполнения огромных журналов инвентаризации;
- формирование сводных отчетов по использованию аудиторного фонда происходит вручную.

Все указанные недостатки приводят к:

- загромождению рабочего места сотрудников, в чьи обязанности входит эта рутинная работа;
- большим затратам на бумагу;
- большим затратам времени на поиск и проверку соответствия необходимой информации;
- отсутствию унификации внешнего представления паспортов.

Применение современных информационных технологий позволит решить эти проблемы, объединив весь архив инвентаризации в одной базе данных и предоставив сотрудникам в виде web-приложения с интуитивно понятным интерфейсом выполнить процесс паспортизации.

1.4 Обоснование выбора средств автоматизации для создания программного продукта

Почти любая система, представленная в виде сайта, разработана с применением базы данных. Для создания базы данных при разработке web-приложения, как правило, используется СУБД MySQL.

MySQL – одна из самых популярных и самых распространенных СУБД в интернете. Данная СУБД получила свою популярность за скорость работы, доступность функционала и весьма понятный интерфейс. Идеально подходит для web-сайтов, как небольших, так и достаточно крупных.

MySQL отличается высокой безопасностью. Работа в ней не должна вызывать трудностей даже у начинающих разработчиков. Поддержка сервера MySQL автоматически включается в поставку PHP. Весьма важным фактором является ее финансовая свобода. Программа распространяется на условиях общей лицензии GNU. Длительное хранение информации является очень часто поставляемой задачей и постоянно встречается в web - программировании: подсчет уникальных посетителей, хранение переписки и т. д. Код с использованием базы данных достаточно компактный, что позволяет легко совершать отладку. Также стоит упомянуть о скорости – выборка нужной информации из общего потока в базе данных происходит значительно быстрее, чем из файлов.

Для написания серверной части был выбран скриптовый язык общего назначения php.

PHP – скриптовый язык общего назначения, интенсивно применяемый для разработки web -приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических web -сайтов.

Для разметки страницы использовался стандартизированный язык html.

HTML – стандартный язык разметки документов во Всемирной паутине. Большинство web -страниц содержат описание разметки на языке HTML. Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.

Запросы к базе данных выполнялись на языке структурированных запросов SQL.

SQL – формальный непроцедурный язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных, управляемой соответствующей системой управления базами данных (СУБД). SQL основывается на исчислении кортежей.

PDO – PHP Data Objects – это прослойка, которая предлагает универсальный способ работы с несколькими базами данных.

PDO обеспечивает абстракцию доступа к данным, это значит, что вне зависимости от того, какая конкретная база данных используется, можно пользоваться одинаковыми функциями для выполнения запросов и выборки данных. По сравнению с расширениями mysql и mysqli, PDO использует объектно-ориентированную парадигму, также имеется поддержка СУБД SQLite, PostgreSQL.

Основным преимуществом PDO является prepared statement.

Prepared statement — это заранее скомпилированное SQL-выражение, которое может быть многократно выполнено путем отправки серверу лишь различных наборов данных. Дополнительным преимуществом является невозможность провести SQL-инъекцию через данные, используемые в placeholder'ax.

Для программирования клиентской части был выбран JavaScript с использованием библиотеки расширения JQuery.

JavaScript как и PHP является мультипарадигменным с динамической типизацией но главное отличие в том что код выполняется на стороне клиента. JavaScript представляет собой программный код — набор инструкций, который не требует предварительной обработки (например, компиляции) перед запуском.

Его код интерпретируется движком браузера во время загрузки веб-страницы. Интерпретатор браузера выполняет построчный анализ, обработку и выполнение исходной программы или запроса. Сценарии JavaScript бывают встроенные, т.е. их содержимое является частью документа, и внешние, хранящиеся в отдельном файле с расширением .js. Каждый html-элемент имеет JavaScript-события. Используя JavaScript, можно вносить изменения в страницу, изменять стили элементов, добавлять или удалять теги. Также есть возможность узнать о своих действиях пользователя на странице (уменьшение или

увеличение рабочей области экрана, клики с помощью мышки, нажатия любых клавиш, прокрутка страницы). Имеется возможность получения доступа к любому элементу Html-кода и манипулирование с этим элементом. Возможности загрузки данных без перезагрузки страницы, ввод сообщений, установка или считывание cookie и выполнение множества других действий.

Основная уникальность этого языка в том, что он имеет поддержку почти у всех браузеров и имеет полную интеграцию с ними. JavaScript имеет некоторые ограничения.

Отсутствует доступ к файлам, которые размещены на пользовательском компьютере. В целом нет доступа за пределы самой веб-страницы. Исключением могут быть только файлы cookie, которые могут быть считаны и записаны с помощью JavaScript. Объекты такого типа обычно нужны лишь для того, чтобы страница загружалась быстрее. Нет возможности выполнения кроссдоменных запросов, то есть получение доступа к веб-страницам, которые находятся на другом домене, даже если они открыты в соседних вкладках.

Отсутствует возможность защиты исходного текста на странице и запрета на копирование изображений или текста со страницы. При этом остается возможность добавлять некоторые ограничения, что позволяют функции JavaScript.

jQuery библиотека JavaScript, содержащая в себе готовые функции языка JavaScript, все операции jQuery выполняются из кода JavaScript. Для выбора элементов используются селекторы CSS. Выбор осуществляется с помощью функции jQuery() или \$(). При вызове функция \$() возвращает новый экземпляр объекта JQuery, который оборачивает элементы DOM и позволяет взаимодействовать с ними различными способами.

AJAX – это технология взаимодействия с сервером без перезагрузки страницы. Поскольку не требуется каждый раз обновлять страницу целиком, повышается скорость работы с сайтом и удобство его использования.

AJAX применяет асинхронную передачу данных. Такой подход позволяет пользователю совершать различные действия во время «фонового» обмена информации с сервером. Действует оповещение пользователя о протекающих процессах, чтобы он не подумал, что сайт «завис» либо на нем произошел какой-то сбой.

1.5 Цель и задачи проекта

Для рационального и эффективного использования рабочего времени ответственных за оформление документации по паспортизации учебных помещений, а также для унификации внешнего представления паспортов в Многопрофильном колледже ФГБОУ ВО

«Магнитогорский государственный технический университет им. Г.И. Носова» было принято решение разработать модуль «Паспортизация помещений», который позволит хранить и редактировать информацию об аудиторном фонде, его материальном, техническом и эргономическом обеспечении, выполнять поисковые запросы административным работникам, формировать паспорта помещений в едином стиле, создавать отчеты. Сформированный паспорт должен проходить процедуру согласования и утверждения.

Модуль «Паспортизация помещений» разрабатывается с целью автоматизации процесса формирования паспорта учебного помещения для учебных заведений среднего профессионального образования.

Разрабатываемая система станет помощником в контроле оснащения оборудованием всех учебных помещений. Областью применения проекта может быть любая организация, предоставляющая образовательные услуги.

Система будет выполнять функции создания, редактирования, печати, хранения паспортов учебных помещений, а также формирования отчетов и хранение всей информации об аудиторном фонде учебного заведения.

Модуль разрабатывается в виде web-приложения, что позволит воспользоваться ей без предварительной подготовки рабочего места. Система должна являться многопользовательской, при этом преподаватель может заполнить паспорт только принадлежащей ему аудитории.

Целью дипломного проекта является разработка модуля «Паспортизация помещений» в рамках информатизации деятельности Многопрофильного колледжа»

Для выполнения дипломного проекта поставлены следующие задачи:

- описание предметной области, выбранной для изучения;
- определение проблем исследования и возможные варианты решения этих проблем;
- оценка существующих решений, выявленных проблем;
- анализ и оценка существующих программных и технических средств, используемых для решения проблемы;
- обоснование и формулировка цели и задач исследования;
- разработка модели (концептуальная, информационная и др.);
- декомпозиция проблемы;
- построение структуры, схемы данных;
- построение схемы функционирования программного продукта;
- проектирование пользовательского интерфейса (выбор диалога, структура справочной системы);

- разработка модуля «Паспортизация помещений» в рамках информатизации деятельности Многопрофильного колледжа;
- организация информационной безопасности модуля «Паспортизация помещений»;
- расчет трудоемкости разработки программного продукта и определение его цены;
- оценка экономической эффективности.

2 ПРОЕКТНАЯ ЧАСТЬ

2.1 Моделирование программного обеспечения

2.1.1 Построение информационной модели

DFD – диаграммы потоков данных. Так называется методология графического структурного анализа, описывающая внешние по отношению к системе источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ. Фрагмент диаграммы потоков данных представлен на рисунке 2.1.

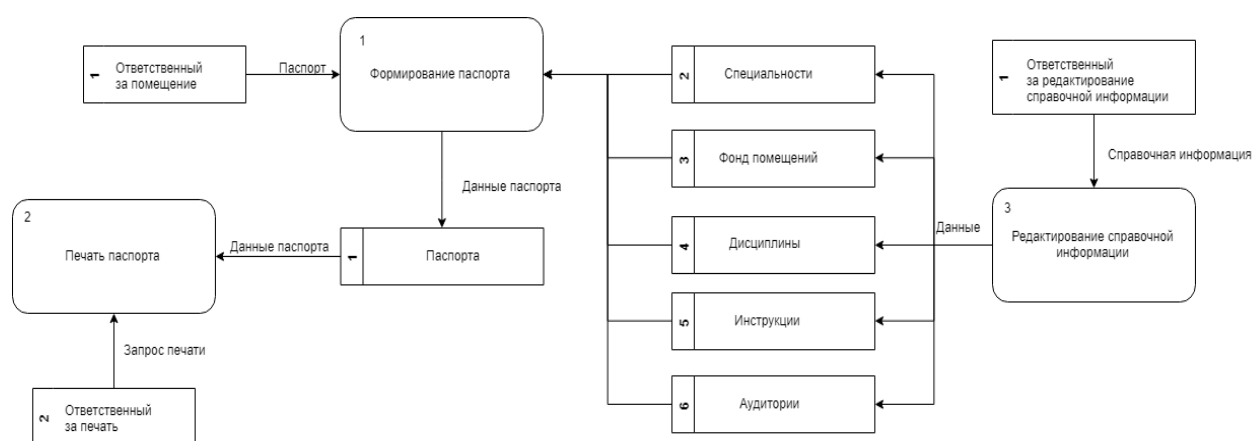


Рисунок 2.1 – Фрагмент диаграммы потоков данных

2.1.2 Проектирование диаграммы прецедентов

Диаграмма прецедентов (диаграмма вариантов использования) в UML – диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Прецедент – возможность моделируемой системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних требований к системе.

Основное назначение диаграммы – описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему.

При моделировании системы с помощью диаграммы прецедентов системный аналитик стремится:

- чётко отделить систему от её окружения;
- определить действующих лиц (актёров), их взаимодействие с системой и ожидаемую функциональность системы;
- определить в глоссарии предметной области понятия, относящиеся к детальному описанию функциональности системы (то есть, прецедентов).

Работа над диаграммой может начаться с текстового описания, полученного при работе с заказчиком. При этом нефункциональные требования (например, конкретный язык или система программирования) при составлении модели прецедентов опускаются (для них составляется другой документ).

Диаграмма прецедентов для модуля «Паспортизация помещений» представлена на рисунке 2.2.

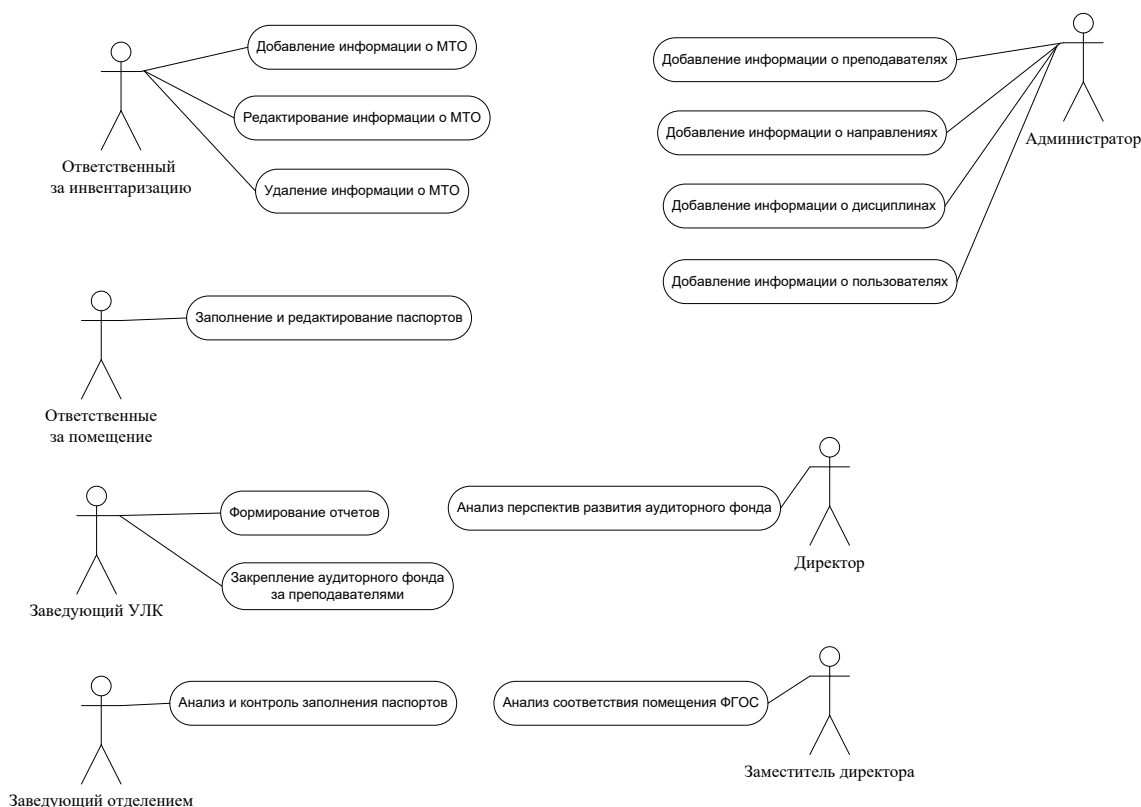


Рисунок 2.2 – Диаграмма прецедентов

2.1.3 Проектирование схемы бизнес-процессов и структуры паспорта помещения

Были проанализированы процедура паспортизации учебных помещений и учитываемые характеристики при составлении паспортов на многочисленных примерах различных образовательных организаций, представляющие приказы, рабочие инструкции, положения о проведении паспортизации учебных помещений, наполнение самих паспортов и

порядок ввода их в действие. Были выделены пользователи системы и определены их функции:

- ответственные за помещение имеют доступ к заполнению и редактированию паспорта только закрепленной за ними распоряжением директора лаборатории/мастерской/кабинета;
- ответственные по инвентаризации – права на редактирование, добавление и удаление информации о МТО;
- заведующий учебно-лабораторным комплексом (УЛК) – полный доступ к информации в системе, формирование поисковых запросов для составления отчетной документации;
- администратор – полный доступ к системе.

На основе анализа разработана схема функционирования бизнес-процессов (рисунок 2.3) после разработки и ввода модуля «Паспортизация помещений» в эксплуатацию и структура паспорта, учитывающая основные характеристики помещения для представления его соответствия назначению (рисунок 2.4).

Таким образом, ответственные по инвентаризации заполняют справочники о материально-техническом оборудовании, заведующий УЛК в соответствии с ежегодным распоряжением директора колледжа о закреплении аудиторного фонда за преподавателями вносит актуальную информацию о назначенных ответственных за помещение, при необходимости редактирует планы помещения со схемой размещения оборудования и схемы подключения электрооборудования. Ответственные за помещение заполняют и редактируют паспорта в системе. Далее паспорта в файловом виде согласовываются и визируются заведующим отделением, заведующим УЛК и заместителем директора по учебно-методической/учебно-производственной работе. В печатном виде паспорта передаются на утверждение директору колледжа.

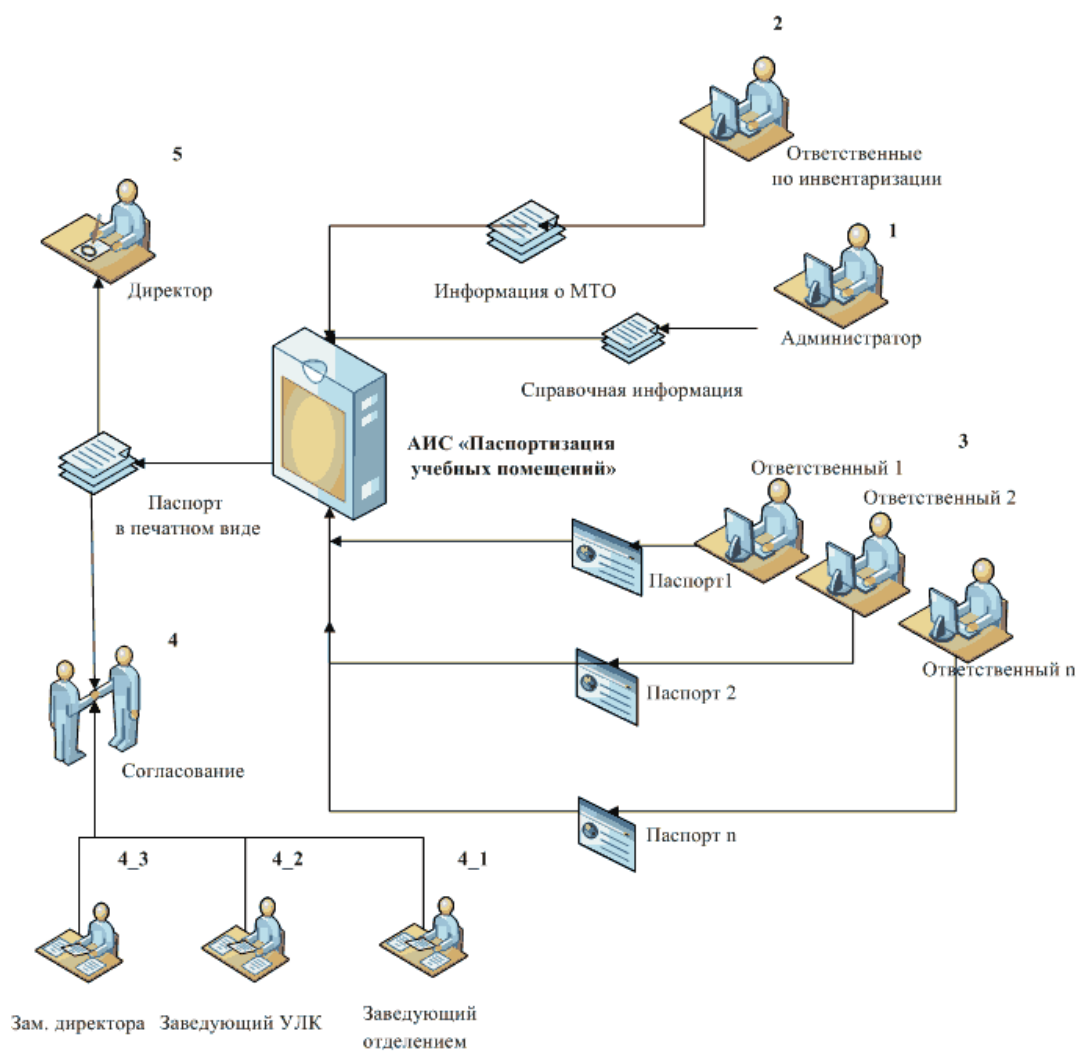


Рисунок 2.3 – Схема взаимодействия бизнес-процессов при паспортизации учебных помещений

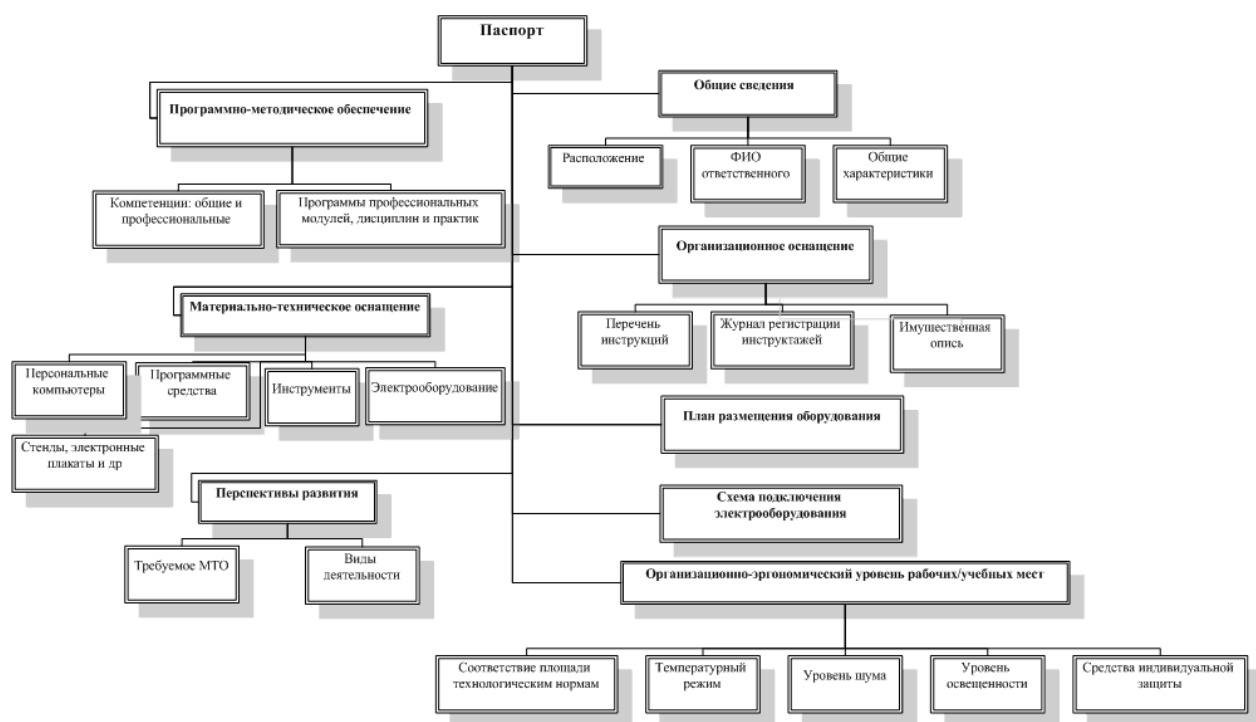


Рисунок 2.4 – Структура паспорта учебного помещения

2.2 Логическое и физическое проектирование базы данных

Логическое проектирование – создание схемы базы данных на основе конкретной модели данных, например, реляционной модели данных. Для реляционной модели данных даталогическая модель – набор схем отношений, обычно с указанием первичных ключей, а также «связей» между отношениями, представляющих собой внешние ключи.

Физическое проектирование – создание схемы базы данных для конкретной системы управления базой данных. Специфика конкретной СУБД может включать в себя ограничения на именованное объектов базы данных, ограничения на поддерживаемые типы данных и т. п. Кроме того, специфика конкретной СУБД при физическом проектировании включает выбор решений, связанных с физической средой хранения данных (выбор методов управления дисковой памятью, разделение БД по файлам и устройствам, методов доступа к данным), создание индексов и т. д.

Целью логического проектирования является преобразование концептуальной модели (рис. 2.5) на основе выбранной модели данных в логическую модель, независимую от особенностей используемой в дальнейшем СУБД для физической реализации базы данных.

Для ее достижения должны быть выполнены следующие процедуры:

- выбор модели данных. Как правило, выбирается реляционная модель данных, в связи с достаточной наглядностью представления таблиц с данными, а также для удобства работы с вышеупомянутыми таблицами;

– нормализация данных. На данном этапе работы выполняется проверка корректности структуры таблиц, посредством применения к ним процедуры нормализации. По итогу данного этапа получается весьма гибкий проект базы данных, позволяющий свободно редактировать необходимую информацию.

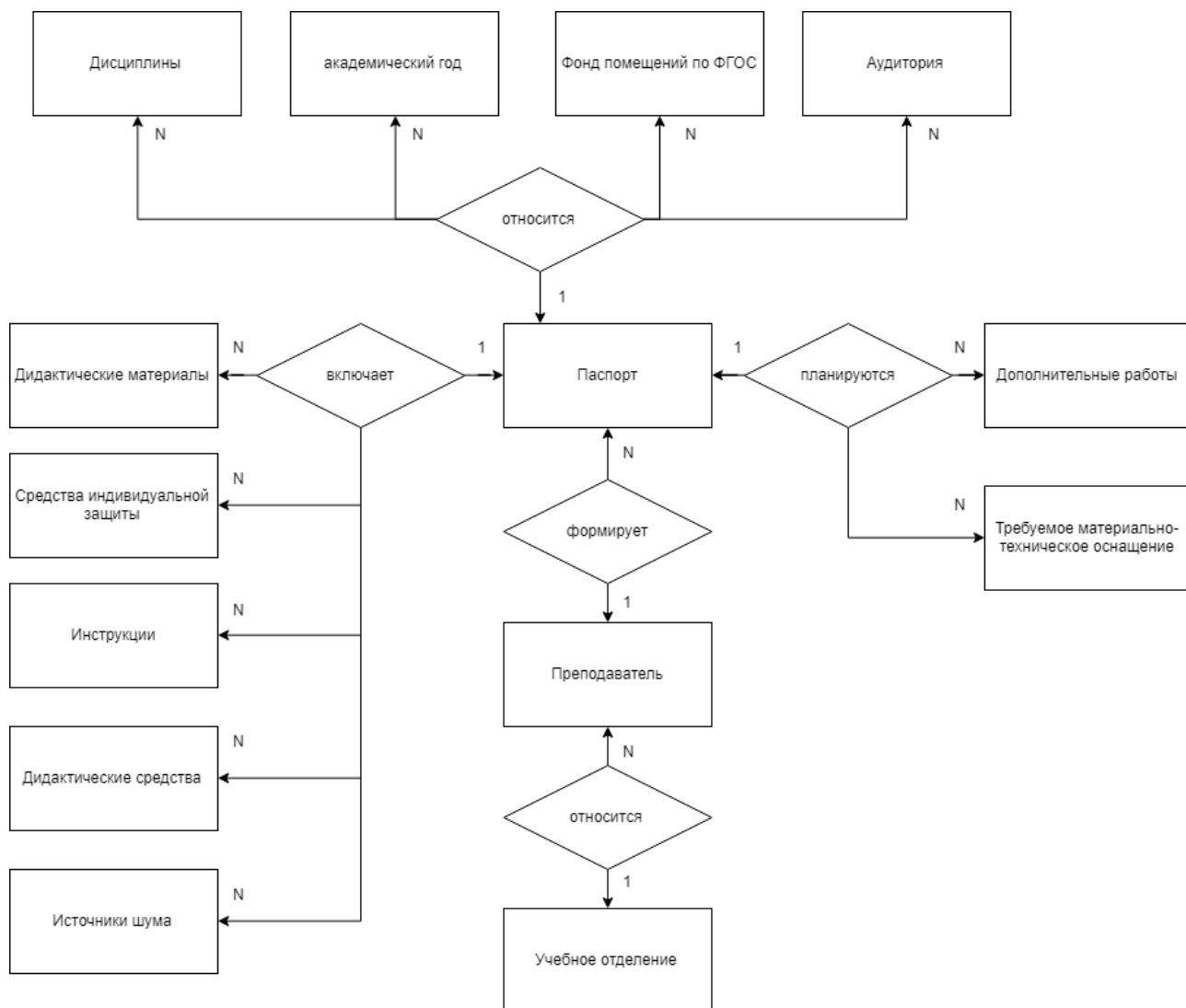


Рисунок 2.5 – Концептуальная модель базы данных

В результате анализа предметной области и логического проектирования сформирована физическая схема базы данных Passport (рис. 2.6).

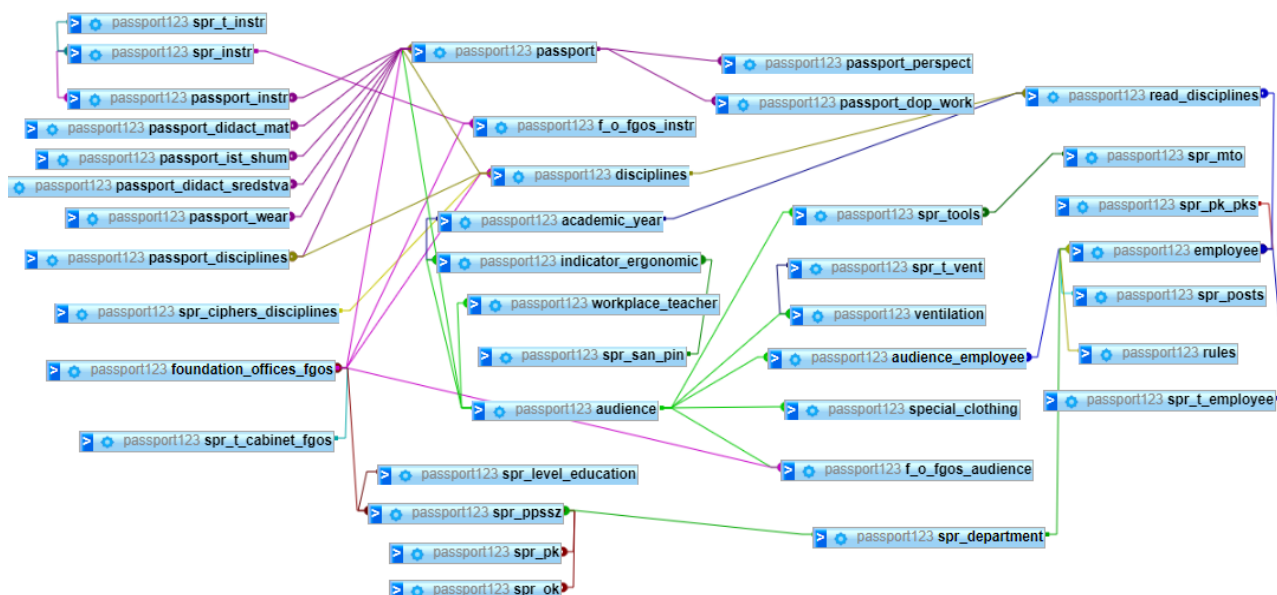


Рисунок 2.6 – Физическая схема базы данных

В процессе логического и физического проектирования в базе данных паспортизации помещений сформировались следующие таблицы:

Справочник инструкций – таблица данных, содержащая перечень инструкций по технике безопасности, охране труда, пожарной безопасности при проведении лабораторно/практических работ (SPR_INSTR).

Структура таблицы «SPR_INSTR»:

- ‘id’ – идентификатор инструкции;
- ‘types_instructions’ – тип инструкции;
- ‘name’ – наименование инструкции.

Справочник шифров дисциплин – таблица данных, содержащая перечень шифров дисциплин в соответствии с ФГОС (SPR_CIPHERS_DISCIPLINES).

Структура таблицы «SPR_CIPHERS_DISCIPLINES»:

- ‘id_ciphers_dis’ – идентификатор шифра дисциплин;
- ‘cipher_dis’ – название шифра;
- ‘description’ – расшифровка шифра.

Справочник типа помещения – таблица данных, содержащая перечень типов учебных помещений в соответствии с ФГОС (SPR_T_CABINET_FGOS).

Структура таблицы «SPR_T_CABINET_FGOS»:

- ‘id’ – код типа помещения;
- ‘name’ – название типа помещения.

Справочник программ подготовки специалистов среднего звена – таблица данных, содержащая перечень наименований программ подготовки специалистов среднего звена

(ППССЗ) (SPR_PPSSZ).

Структура таблицы «SPR_PPSSZ»:

‘id_specialty’ – идентификатор специальности;

‘specialty’ – код специальности;

‘name’ – название специальности;

‘id_level_education’ – идентификатор уровня обучения;

‘id_department’ – идентификатор учебного отделения.

Справочник общих компетенций – таблица данных, содержащая перечень общих компетенций (SPR_OK).

Структура таблицы «SPR_OK»:

‘id’ – идентификатор общей компетенции;

‘short_name’ – короткое название общей компетенции;

‘description’ – описание общей компетенции;

‘id_ppsz’ – идентификатор специальности.

Справочник профессиональных компетенций – таблица данных, содержащая перечень профессиональных компетенций по каждой ППССЗ (SPR_PK).

Структура таблицы «SPR_PK»:

‘id’ – идентификатор профессиональной компетенции;

‘short_name’ – короткое название профессиональной компетенции;

‘description’ – описание профессиональной компетенции;

‘id_ppsz’ – идентификатор специальности.

Справочник санитарных правил и норм – таблица данных, содержащая перечень нормативов санитарных норм для помещений разного типа (SPR_SAN_PIN).

Структура таблицы «SPR_SAN_PIN»:

‘id’ – идентификатор нормы СанПин;

‘index’ – название нормы СанПин;

‘cipher’ – шифр нормы СанПин;

‘unit_measure’ – единица измерения нормы СанПин;

‘norm_san_pin’ – значение нормы СанПин.

Справочник уровней обучения – таблица данных, содержащая перечень уровней обучения (SPR_LEVEL_EDUCATION).

Структура таблицы «SPR_LEVEL_EDUCATION»:

‘id’ – идентификатор уровня обучения;

‘level’ – название уровня обучения.

Справочник отделений – таблица данных, содержащая перечень номеров,

наименований отделений и ФИО заведующих отделений учебного заведения (SPR_DEPARTMENT).

Структура таблицы «SPR_DEPARTMENT»:

‘id’ – идентификатор учебного отделения;

‘name’ – название учебного отделения;

‘head_department’ – заведующий учебного отделения;

‘short_name’ – фамилия и инициалы заведующего отделения.

Справочник типов вентиляции – таблица данных, содержащая перечень типов вентиляции в помещениях (SPR_T_VENT).

Структура таблицы «SPR_T_VENT»:

‘id’ – идентификатор типа вентиляции;

‘type_ventilation’ – название типа вентиляции.

Справочник инструментов – таблица данных, содержащая перечень используемого инструментального оборудования и их инвентарные номера (SPR_TOOLS).

Структура таблицы «SPR_TOOLS»:

‘id’ – идентификатор оборудования;

‘name’ – название оборудования;

‘year_purchase’ – год приобретения оборудования;

‘invent’ – инвентарный номер оборудования;

‘Count’ – количество оборудования;

‘type_mto’ – тип оборудования;

‘audience’ – идентификатор аудитории;

‘Owner’ – ответственный за оборудование.

Справочник должностей – таблица данных, содержащая перечень должностей сотрудников (SPR_POSTS);

Структура таблицы «SPR_POSTS»:

‘id’ – идентификатор должности;

‘name’ – наименование должности.

Справочник типа сотрудника – таблица данных, содержащая перечень типов занятости (SPR_T_EMPLOYEE);

Структура таблицы «SPR_T_EMPLOYEE»:

‘id’ – идентификатор типа занятости;

‘name’ – наименование типа занятости.

Справочник предметных и предметно-цикловых комиссий (ПК/ПЦК) – таблица данных, содержащая перечень предметных и предметно-цикловых комиссий учебного

заведения (SPR_PK_PKS);

Структура таблицы «SPR_PK_PKS»:

‘id’ – идентификатор типа занятости;

‘name’ – наименование комиссии;

‘commission’ – тип комиссии;

‘cipher’ – шифр типа комиссии.

Справочник видов инструкций – таблица данных, содержащая перечень видов инструкций (SPR_T_INSTR);

Структура таблицы «SPR_T_INSTR»:

‘id’ – идентификатор типа инструкции;

‘types_instructions’ – шифр типа инструкции;

‘name’ – наименование типа инструкции.

Справочник прав – таблица данных, содержащая типы прав доступа (RULES);

Структура таблицы «RULES»:

‘id’ – идентификатор типа прав доступа;

‘name’ – наименование типа прав доступа.

Справочник типов материально-технического оснащения (MTO) – таблица данных, содержащая перечень справочной информации о типах МТО (SPR_MTO);

Структура таблицы «SPR_MTO»:

‘id’ – идентификатор типа материально-технического оснащения;

‘type_mto’ – наименование типа материально-технического оснащения.

Таблица «Читаемые дисциплины» – содержит информацию о том, какие дисциплины читаются какими преподавателями (READ_DISCIPLINES);

Структура таблицы «READ_DISCIPLINES»:

‘id’ – идентификатор читаемой дисциплины;

‘id_academic_year’ – академический год;

‘id_employee’ – идентификатор сотрудника;

‘id_disciplines’ – идентификатор дисциплины.

Таблица «Специальная одежда» – содержит информацию о находящейся в помещении специальной одежды (SPECIAL_CLOTHING);

Структура таблицы «SPECIAL_CLOTHING»:

‘id’ – идентификатор спецодежды;

‘name’ – название спецодежды;

‘actual_availability’ – оснащенность в %;

‘audience’ – идентификатор аудитории.

Таблица «Дисциплины» – содержит подробную информацию о преподаваемых дисциплинах (DISCIPLINES);

Структура таблицы «DISCIPLINES»:

- ‘id’ – идентификатор дисциплины;
- ‘id_ciphers’ – идентификатор типа дисциплины;
- ‘index_discipline’ – индекс дисциплины;
- ‘id_foundation’ – идентификатор помещения
- ‘name’ – название дисциплины;
- ‘year_fgos’ – год.

Таблица «Фонд кабинетов ФГОС» – содержит перечень кабинетов по ФГОС (FOUNDATION_OFFICES_FGOS);

Структура таблицы «FOUNDATION_OFFICES_FGOS»:

- ‘id_foundation’ – идентификатор помещения;
- ‘id_t_cabinet’ – идентификатор типа помещения;
- ‘name’ – название помещения;
- ‘id_ppssz’ – идентификатор специальности.

Таблица «Фонд кабинетов ФГОС. Аудитория» - содержит информацию о соответствии аудиторий фонду кабинетов ФГОС (F_O_FGOS_AUDIENCE);

Структура таблицы «F_O_FGOS_AUDIENCE»:

- ‘id’ – идентификатор соответствия помещений аудиториям;
- ‘id_audience’ – идентификатор аудитории;
- ‘id_foundation’ – идентификатор помещения.

Таблица «Фонд кабинетов ФГОС. Инструкции» - содержит информацию о соответствии инструкций фонду кабинетов ФГОС (F_O_FGOS_INSTR);

Структура таблицы «F_O_FGOS_INSTR»:

- ‘id’ – идентификатор соответствия помещений аудиториям;
- ‘id_instr’ – идентификатор инструкции;
- ‘id_foundation’ – идентификатор помещения.

Таблица «Рабочее место преподавателя» – содержит информацию о рабочем месте преподавателя (WORKPLACE_TEACHER);

Структура таблицы «WORKPLACE_TEACHER»:

- ‘id’ – идентификатор инвентаря;
- ‘name’ – название инвентаря;
- ‘audience’ – идентификатор аудитории;
- ‘count’ – количество.

Таблица «Показатели эргономики» – содержит информацию об организационно-эргономическом уровне рабочих и/или учебных мест (INDICATOR_ERGONOMIC);

Структура таблицы «INDICATOR_ERGONOMIC»:

‘id’ – идентификатор показателя эргономики;

‘index’ – идентификатор нормы СанПин;

‘indicator_values’ – показатели эргономики;

‘additional_indicators’ – дополнительные показатели эргономики;

‘academic_year’ – идентификатор академического года;

‘audience’ – идентификатор аудитории.

Таблица «Учебный год» – содержит информацию об учебных годах (ACADEMIC_YEAR);

Таблица «Вентиляция» – содержит информацию о наличии вентиляции в аудитории (VENTILATION);

Структура таблицы «VENTILATION»:

‘id’ – идентификатор помещения;

‘type_ventilation’ – тип вентиляции;

‘type_fan’ – тип вентилятора;

‘count’ – количество;

‘speed_exchange’ – скорость вращения;

‘room_volume’ – значение шума;

‘audience’ – идентификатор аудитории.

Таблица «Сотрудник» – содержит данные о сотрудниках (EMPLOYEE).

Структура таблицы «EMPLOYEE»:

‘id’ – идентификатор помещения;

‘surname’ – фамилия сотрудника;

‘name’ – имя сотрудника;

‘patronymic’ – отчество сотрудника;

‘id_pk_pkc’ – идентификатор ПК ПЦК;

‘id_type_employee’ – идентификатор типа деятельности

‘id_department’ – идентификатор отделения;

‘id_posts’ – идентификатор должности сотрудника;

‘login’ – логин;

‘password’ – пароль;

‘rules’ – идентификатор прав доступа;

Таблица «Сотрудник. Аудитория» – содержит данные о соответствии аудиторий

сотрудникам (AUDIENCE_EMPLOYEE).

Структура таблицы «AUDIENCE_EMPLOYEE»:

‘id’ – идентификатор помещения;

‘id_audience’ – идентификатор аудитории;

‘id_employee’ – идентификатор сотрудника;

Таблица «Паспорт. Инструкции» содержит информацию о внесенных в паспорт инструкциях для организации учебного процесса в данном помещении (PASSPORT_INSTR).

Структура таблицы «PASSPORT_INSTR»:

‘id’ – идентификатор соответствия помещений инструкциям;

‘id_instr’ – идентификатор инструкции;

‘id_passport’ – идентификатор паспорта.

Таблица «Паспорт. Дидактические материалы» содержит информацию о внесенных в паспорт дидактических материалах для организации учебного процесса в данном помещении (PASSPORT_DIDACT_MAT).

Структура таблицы «PASSPORT_DIDACT_MAT»:

‘id’ – идентификатор дидактического материала;

‘id_passport’ – идентификатор паспорта;

‘didact_mat’ – название дидактического материала;

‘didact_mat_gty’ – количество.

Таблица «Паспорт. Источники шума» содержит информацию о внесенных в паспорт источниках шума в данном помещении (PASSPORT_IST_SHUM).

Структура таблицы «PASSPORT_IST_SHUM»:

‘id’ – идентификатор источника шума

‘id_passport’ – идентификатор паспорта;

‘ist_shum’ – название источника шума;

Таблица «Паспорт. Дидактические средства» содержит информацию о внесенных в паспорт дидактических средствах для организации учебного процесса в данном помещении (PASSPORT_DIDACT_SREDSTVA).

Структура таблицы «PASSPORT_DIDACT_SREDSTVA»:

‘id’ – идентификатор дидактического средства;

‘id_passport’ – идентификатор паспорта;

‘name’ – название дидактического средства;

‘invent’ – инвентарный номер;

‘gty’ – количество;

‘year’ – год;

‘type_mto – тип дидактического средства.

Таблица «Паспорт. Спецодежда» содержит информацию о внесенной в паспорт спецодежды для организации учебного процесса в данном помещении (PASSPORT_WEAR).

Структура таблицы «PASSPORT_WEAR»:

‘id’ – идентификатор одежды;

‘id_passport’ – идентификатор паспорта;

‘name’ – название источника шума;

‘qty’ – количество.

Таблица «Паспорт. Перспективное планирование» содержит информацию о внесенном в паспорт перечне требуемого МТО помещения с целью повышения качества учебного процесса (PASSPORT_PERSPECT).

Структура таблицы «PASSPORT_PERSPECT»:

‘id’ – идентификатор требуемого материально-технического оснащения;

‘id_passport’ – идентификатор паспорта;

‘name’ – название требуемого материально-технического оснащения;

‘qty’ – количество;

‘description’ – описание.

Таблица «Паспорт. Виды деятельности» содержит информацию о внесенном в паспорт перечне планируемых работ с целью повышения качества учебного процесса в данном помещении (PASSPORT_DOP_WORK).

Структура таблицы «PASSPORT_DOP_WORK»:

‘id’ – идентификатор дополнительных работ;

‘id_passport’ – идентификатор паспорта;

‘name’ – название дополнительных работ;

‘qty’ – количество;

‘description’ – описание.

Таблица «Аудитория» содержит обобщенную информацию обо всех учебных помещениях (AUDIENCE).

Структура таблицы «AUDIENCE»:

‘id’ – идентификатор дополнительных работ;

‘number’ – номер кабинета;

‘number_jobs’ – количество мест;

‘number_special_jobs’ – количество рабочих мест;

‘responsible’ – ответственный за кабинет;

‘path_electric’ – путь к файлу электросхемы;

‘path_tables’ – путь к файлу схемы помещения;

‘address’ – адрес.

Таблица «Паспорт» содержит обобщенную информацию о составленных паспортах (PASSPORT).

Структура таблицы «PASSPORT»:

‘id’ – идентификатор паспорта;

‘id_audience’ – идентификатор аудитории;

‘id_foundation’ – идентификатор помещения;

‘discipline’ – идентификатор дисциплины;

‘number_jobs’ – количество мест;

‘number_special_jobs’ – количество рабочих мест;

‘number_protokol’ – номер протокола;

‘journal_date’ – дата утверждения;

‘temp_summer’ – температура летом;

‘temp_winter’ – температура зимой;

‘fact_shum’ – уровень шума;

‘fact_light_estestvennaya’ – уровень естественной освещенности;

‘fact_light_iskustvennaya’ – уровень искусственной освещенности;

‘date_priem’ – дата приемки;

‘date’ – дата формирования паспорта;

‘academic_year’ – академический год

На рисунке 2.4 приведена структура паспорта учебного помещения.

Таким образом, работа с БД сводится к хранению, систематизации, обработке и редактированию общих сведений о помещении, программно-методического обеспечения, материально-технического и организационного оснащения, плана размещения оборудования, схемы подключения электрооборудования, организационно-эргономического уровня рабочих/учебных мест и перспектив развития помещения для формирования паспорта учебного помещения.

2.3 Разработка базы данных

Создание таблицы «audience» (Аудитории):

```
CREATE TABLE `audience` (
```

```
  `id` SERIAL,
```

```
  `number` varchar(20) NOT NULL,
```

```

`space` double DEFAULT NULL,
`number_jobs` int(3) DEFAULT NULL,
`number_special_jobs` int(3) DEFAULT NULL,
`responsible` varchar(25) NOT NULL,
`path_electric` varchar(80) DEFAULT NULL,
`path_tables` varchar(80) DEFAULT NULL,
`address` text NOT NULL,
PRIMARY KEY(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «audience_employee» (Аудитории. Сотрудники):

```

CREATE TABLE `audience_employee` (
  `id` SERIAL,
  `id_audience` bigint(20) UNSIGNED NOT NULL,
  `id_employee` bigint(20) UNSIGNED NOT NULL,
  `level` tinyint(1) NOT NULL DEFAULT '0',
  PRIMARY KEY(id),
  FOREIGN KEY (`id_audience`) REFERENCES `audience` (`id`) ON UPDATE
  CASCADE,
  FOREIGN KEY (`id_employee`) REFERENCES `employee` (`id`) ON UPDATE
  CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «academic_year» (Академический год):

```

CREATE TABLE `academic_year` (
  `id` varchar(20) NOT NULL,
  PRIMARY KEY(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «disciplines» (Дисциплины):

```

CREATE TABLE `disciplines` (
  `id` SERIAL,
  `id_ciphers` bigint(20) UNSIGNED NOT NULL,
  `index_discipline` varchar(10) NOT NULL,
  `id_foundation` bigint(20) UNSIGNED DEFAULT NULL,

```



```

`name` varchar(100) NOT NULL,
`ppsz` int(11) NOT NULL,
`year_fgos` varchar(4) NOT NULL,
PRIMARY KEY(id),
FOREIGN KEY (`id_ciphers`) REFERENCES `spr_ciphers_disciplines` (`id_ciphers_dis`)
ON UPDATE CASCADE,
FOREIGN KEY (`id_foundation`) REFERENCES `foundation_offices_fgos`
(`id_foundation`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «employee» (Сотрудники):

```

CREATE TABLE `employee` (
`id` SERIAL,
`surname` varchar(30) NOT NULL,
`name` varchar(20) NOT NULL,
`patronymic` varchar(30) NOT NULL,
`id_pk_pkc` bigint(20) UNSIGNED NOT NULL,
`id_type_employee` bigint(20) UNSIGNED NOT NULL,
`id_department` bigint(20) UNSIGNED NOT NULL,
`id_posts` bigint(20) UNSIGNED NOT NULL,
`login` text NOT NULL,
`password` text NOT NULL,
`rules` bigint(20) UNSIGNED NOT NULL,
PRIMARY KEY(id)
FOREIGN KEY (`id_posts`) REFERENCES `spr_posts` (`id`) ON UPDATE CASCADE
FOREIGN KEY (`id_pk_pkc`) REFERENCES `spr_pk_pks` (`id`) ON UPDATE
CASCADE
FOREIGN KEY (`id_department`) REFERENCES `spr_department` (`id`) ON UPDATE
CASCADE
FOREIGN KEY (`id_type_employee`) REFERENCES `spr_t_employee` (`id`) ON
UPDATE CASCADE
FOREIGN KEY (`rules`) REFERENCES `rules` (`id`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «foundation_offices_fgos» (Фонд помещений по ФГОС):

```

CREATE TABLE `foundation_offices_fgos` (
  `id_foundation` SERIAL,
  `id_t_cabinet` bigint(20) UNSIGNED NOT NULL,
  `name` varchar(100) NOT NULL,
  `id_ppssz` varchar(10) NOT NULL,
  PRIMARY KEY(id_foundation),
  FOREIGN KEY (`id_t_cabinet`) REFERENCES `spr_t_cabinet_fgos` (`id`) ON UPDATE
CASCADE,
  FOREIGN KEY (`id_ppssz`) REFERENCES `spr_ppssz` (`id_specialty`) ON UPDATE
CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «f_o_fgos_instr» (Фонд помещений по ФГОС. Инструкции):

```

CREATE TABLE `f_o_fgos_instr` (
  `id` SERIAL,
  `id_instr` bigint(20) UNSIGNED NOT NULL,
  `id_foundation` bigint(20) UNSIGNED NOT NULL,
  PRIMARY KEY(id),
  FOREIGN KEY (`id_foundation`) REFERENCES `foundation_offices_fgos`
(`id_foundation`) ON UPDATE CASCADE,
  FOREIGN KEY (`id_instr`) REFERENCES `spr_instr` (`id`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «f_o_fgos_audience» (Фонд помещений по ФГОС. Аудитории):

```

CREATE TABLE `f_o_fgos_audience` (
  `id` SERIAL,
  `id_audience` bigint(20) UNSIGNED NOT NULL,
  `id_foundation` bigint(20) UNSIGNED NOT NULL,
  PRIMARY KEY(id),
  FOREIGN KEY (`id_foundation`) REFERENCES `foundation_offices_fgos`
(`id_foundation`) ON UPDATE CASCADE,
  FOREIGN KEY (`id_audience`) REFERENCES `audience` (`id`) ON UPDATE
CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «indicator_ergonomic» (Показатели эргономики):

```

CREATE TABLE `indicator_ergonomic` (
  `id` SERIAL,
  `index` varchar(80) NOT NULL,
  `indicator_values` decimal(5,2) NOT NULL,
  `additional_indicators` varchar(30) NOT NULL,
  `academic_year` varchar(20) NOT NULL,
  `audience` bigint(20) UNSIGNED NOT NULL,
  PRIMARY KEY(id),
  FOREIGN KEY (`index`) REFERENCES `spr_san_pin` (`index`) ON UPDATE
  CASCADE,
  FOREIGN KEY (`academic_year`) REFERENCES `academic_year` (`id`) ON UPDATE
  CASCADE,
  FOREIGN KEY (`audience`) REFERENCES `audience` (`id`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «passport» (Паспорта):

```

CREATE TABLE `passport` (
  `id` SERIAL,
  `id_audience` bigint(20) UNSIGNED NOT NULL,
  `id_foundation` bigint(20) UNSIGNED DEFAULT NULL,
  `discipline` bigint(20) UNSIGNED DEFAULT NULL,
  `number_jobs` int(11) NOT NULL,
  `number_special_jobs` int(11) NOT NULL,
  `number_protokol` varchar(20) NOT NULL,
  `journal_date` varchar(20) NOT NULL,
  `temp_summer` text NOT NULL,
  `temp_winter` text NOT NULL,
  `fact_shum` int(11) NOT NULL,
  `fact_light_estestvennaya` float NOT NULL,
  `fact_light_iskustvennaya` int(11) NOT NULL,
  `date_priem` varchar(20) NOT NULL,
  `date` text NOT NULL,
  `academic_year` varchar(20) NOT NULL,
  PRIMARY KEY(id),
  FOREIGN KEY (`id_audience`) REFERENCES `audience` (`id`) ON UPDATE

```

```

CASCADE,
    FOREIGN KEY (`id_foundation`) REFERENCES `foundation_offices_fgos`
(`id_foundation`) ON UPDATE CASCADE,
    FOREIGN KEY (`discipline`) REFERENCES `disciplines` (`id`) ON UPDATE
CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы « passport_didact_mat » (Паспорт. Дидактические материалы):

```

CREATE TABLE `passport_didact_mat` (
    `id` SERIAL,
    `id_passport` bigint(20) UNSIGNED NOT NULL,
    `didact_mat` text NOT NULL,
    `didact_mat_gty` int(11) NOT NULL,
    PRIMARY KEY(id),
    FOREIGN KEY (`id_passport`) REFERENCES `passport` (`id`) ON UPDATE
CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «passport_didact_sredstva» (Паспорт. Дидактические средства):

```

CREATE TABLE `passport_didact_sredstva` (
    `id` SERIAL,
    `id_passport` bigint(20) UNSIGNED NOT NULL,
    `name` text NOT NULL,
    `invent` text NOT NULL,
    `gty` int(11) NOT NULL,
    `year` text NOT NULL,
    `type_mto` text NOT NULL,
    PRIMARY KEY(id),
    FOREIGN KEY (`id_passport`) REFERENCES `passport` (`id`) ON UPDATE
CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «passport_disciplines» (Паспорт. Дисциплины):

```

CREATE TABLE `passport_disciplines` (
    `id` SERIAL,

```

```

        `id_discipline` bigint(20) UNSIGNED NOT NULL,
        `id_passport` bigint(20) UNSIGNED NOT NULL,
        PRIMARY KEY(id),
        FOREIGN KEY (`id_passport`) REFERENCES `passport` (`id`) ON UPDATE
        CASCADE,
        FOREIGN KEY (`id_discipline`) REFERENCES `disciplines` (`id`) ON UPDATE
        CASCADE,
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «passport_dop_work» (Паспорт. Дополнительные работы):

```

CREATE TABLE `passport_dop_work` (
    `id` SERIAL,
    `id_passport` bigint(20) UNSIGNED NOT NULL,
    `name` text NOT NULL,
    `description` text NOT NULL,
    PRIMARY KEY(id),
    FOREIGN KEY (`id_passport`) REFERENCES `passport` (`id`) ON UPDATE
    CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «passport_instr» (Паспорт. Инструкции):

```

CREATE TABLE `passport_instr` (
    `id` SERIAL,
    `id_instr` bigint(20) UNSIGNED NOT NULL,
    `id_passport` bigint(20) UNSIGNED NOT NULL,
    PRIMARY KEY(id),
    FOREIGN KEY (`id_passport`) REFERENCES `passport` (`id`) ON UPDATE
    CASCADE,
    FOREIGN KEY (`id_instr`) REFERENCES `spr_instr` (`id`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «passport_ist_shum» (Паспорт. Источники шума):

```

CREATE TABLE `passport_ist_shum` (
    `id` SERIAL,
    `id_passport` bigint(20) UNSIGNED NOT NULL,

```

```

        `ist_shum` text NOT NULL,
        PRIMARY KEY(id),
        FOREIGN KEY (`id_passport`) REFERENCES `passport` (`id`) ON UPDATE
CASCADE
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «passport_perspect» (Паспорт. Перспективное планирование):

```

CREATE TABLE `passport_perspect` (
    `id` SERIAL,
    `id_passport` bigint(20) UNSIGNED NOT NULL,
    `name` text NOT NULL,
    `qty` int(11) NOT NULL,
    `description` text NOT NULL,
    PRIMARY KEY(id),
    FOREIGN KEY (`id_passport`) REFERENCES `passport` (`id`) ON UPDATE
CASCADE
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «passport_wear» (Паспорт. Спецодежда):

```

CREATE TABLE `passport_wear` (
    `id` SERIAL,
    `id_passport` bigint(20) UNSIGNED NOT NULL,
    `name` text NOT NULL,
    `gty` text NOT NULL,
    PRIMARY KEY(id),
    FOREIGN KEY (`id_passport`) REFERENCES `passport` (`id`) ON UPDATE
CASCADE
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «read_disciplines» (Читаемые дисциплины):

```

CREATE TABLE `read_disciplines` (
    `id` SERIAL,
    `id_academic_year` varchar(20) NOT NULL,
    `id_employee` bigint(20) UNSIGNED NOT NULL,
    `id_disciplines` bigint(20) UNSIGNED NOT NULL,

```

```

PRIMARY KEY(id),
FOREIGN KEY (`id_academic_year`) REFERENCES `academic_year` (`id`) ON
UPDATE CASCADE,
FOREIGN KEY (`id_employee`) REFERENCES `employee` (`id`) ON UPDATE
CASCADE,
FOREIGN KEY (`id_disciplines`) REFERENCES `disciplines` (`id`) ON UPDATE
CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «rules» (Права пользователей):

```

CREATE TABLE `rules` (
  `id` SERIAL,
  `name` varchar(80) NOT NULL,
  PRIMARY KEY(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «rules» (Права пользователей):

```

CREATE TABLE `special_clothing` (
  `id` SERIAL,
  `name` varchar(100) NOT NULL,
  `actual_availability` double NOT NULL,
  `audience` bigint(20) UNSIGNED NOT NULL,
  PRIMARY KEY(id),
  FOREIGN KEY (`audience`) REFERENCES `audience` (`id`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «spr_ciphers_disciplines» (Шифры дисциплин):

```

CREATE TABLE `spr_ciphers_disciplines` (
  `id_ciphers_dis` SERIAL,
  `ciphers_dis` varchar(20) NOT NULL,
  `description` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «spr_department» (Отделения):

```

CREATE TABLE `spr_department` (

```

```

`id` SERIAL,
`name` varchar(100) NOT NULL,
`head_department` varchar(30) NOT NULL,
`short_name` text NOT NULL,
PRIMARY KEY(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «spr_instr» (Инструкции):

```

CREATE TABLE `spr_instr` (
  `id` SERIAL,
  `types_instructions` varchar(30) NOT NULL,
  `name` varchar(250) NOT NULL,
  PRIMARY KEY(id),
  FOREIGN KEY (`types_instructions`) REFERENCES `spr_t_instr` (`types_instructions`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «spr_level_education» (Уровни обучения):

```

CREATE TABLE `spr_level_education` (
  `id` SERIAL,
  `level` varchar(15) NOT NULL,
  PRIMARY KEY(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «spr_mto» (Типы материально-технического оснащения):

```

CREATE TABLE `spr_mto` (
  `id` SERIAL,
  `type_mto` varchar(50) NOT NULL,
  PRIMARY KEY(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «spr_ok» (Общие компетенции):

```

CREATE TABLE `spr_ok` (
  `id` SERIAL,
  `short_name` varchar(10) NOT NULL,
  `description` text NOT NULL,

```



```

`id_ppssz` varchar(10) NOT NULL,
PRIMARY KEY(id),
FOREIGN KEY (`id_ppssz`) REFERENCES `spr_ppssz` (`id_specialty`) ON UPDATE
CASCADE

```

```

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «spr_pk» (Профессиональные компетенции):

```

CREATE TABLE `spr_pk` (
  `id` SERIAL,
  `short_name` varchar(10) NOT NULL,
  `description` varchar(255) NOT NULL,
  `id_ppssz` varchar(10) NOT NULL,
  PRIMARY KEY(id),
  FOREIGN KEY (`id_ppssz`) REFERENCES `spr_ppssz` (`id_specialty`) ON UPDATE
  CASCADE

```

```

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «spr_pk_pks» (Профессиональные компетенции):

```

CREATE TABLE `spr_pk_pks` (
  `id` SERIAL,
  `name` varchar(100) NOT NULL,
  `commission` varchar(30) NOT NULL,
  `cipher` varchar(20) NOT NULL,
  PRIMARY KEY(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «spr_posts» (Должности):

```

CREATE TABLE `spr_posts` (
  `id` SERIAL,
  `name` varchar(50) NOT NULL,
  PRIMARY KEY(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «spr_ppssz» (Специальности):

```

CREATE TABLE `spr_ppssz` (
  `id_specialty` varchar(10) NOT NULL,

```

```

`specialty` varchar(20) NOT NULL,
`name` varchar(150) NOT NULL,
`id_level_education` bigint(20) UNSIGNED NOT NULL,
`id_department` bigint(20) UNSIGNED NOT NULL,
PRIMARY KEY(id_specialty),
FOREIGN KEY (`id_level_education`) REFERENCES `spr_level_education` (`id`) ON
UPDATE CASCADE,
FOREIGN KEY (`id_department`) REFERENCES `spr_department` (`id`) ON UPDATE
CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «spr_san_pin» (Нормы СанПин):

```

CREATE TABLE `spr_san_pin` (
  `id` SERIAL,
  `index` varchar(80) NOT NULL,
  `cipher` varchar(35) NOT NULL,
  `unit_measure` varchar(50) NOT NULL,
  `norm_san_pin` decimal(5,2) NOT NULL,
  PRIMARY KEY(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «spr_tools» (Материально-техническое оснащение):

```

CREATE TABLE `spr_tools` (
  `id` SERIAL,
  `name` varchar(150) NOT NULL,
  `year_purchase` int(4) UNSIGNED DEFAULT NULL,
  `invent` varchar(50) NOT NULL,
  `Count` int(4) NOT NULL,
  `type_mto` varchar(20) NOT NULL,
  `audience` bigint(20) UNSIGNED DEFAULT NULL,
  `Owner` varchar(25) NOT NULL,
  PRIMARY KEY(id),
  FOREIGN KEY (`type_mto`) REFERENCES `spr_mto` (`type_mto`) ON UPDATE
  CASCADE,
  FOREIGN KEY (`audience`) REFERENCES `audience` (`id`) ON UPDATE CASCADE

```

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

Создание таблицы «spr_t_cabinet_fgos» (Тип помещения по ФГОС):

```
CREATE TABLE `spr_t_cabinet_fgos` (
```

```
  `id` SERIAL,
```

```
  `name` varchar(30) NOT NULL,
```

```
  PRIMARY KEY(id)
```

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

Создание таблицы «spr_t_employee» (Типы занятости сотрудника):

```
CREATE TABLE `spr_t_employee` (
```

```
  `id` SERIAL,
```

```
  `name` varchar(15) NOT NULL,
```

```
  PRIMARY KEY(id)
```

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

Создание таблицы «spr_t_instr» (Типы инструкций):

```
CREATE TABLE `spr_t_instr` (
```

```
  `id` SERIAL,
```

```
  `types_instructions` varchar(30) NOT NULL,
```

```
  `name` varchar(250) NOT NULL,
```

```
  PRIMARY KEY(id)
```

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

Создание таблицы «spr_t_vent» (Типы вентиляции):

```
CREATE TABLE `spr_t_vent` (
```

```
  `id` SERIAL,
```

```
  `type_ventilation` varchar(30) NOT NULL,
```

```
  PRIMARY KEY(id)
```

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

Создание таблицы «ventilation» (Вентиляция):

```
CREATE TABLE `ventilation` (
```

```
  `id` SERIAL,
```

```
  `type_ventilation` varchar(30) NOT NULL,
```

```

`type_fan` varchar(15) NOT NULL,
`count` int(3) UNSIGNED NOT NULL,
`speed_exchange` double NOT NULL,
`room_volume` double DEFAULT NULL,
`audience` bigint(20) UNSIGNED NOT NULL,
PRIMARY KEY(id),
FOREIGN KEY (`type_ventilation`) REFERENCES `spr_t_vent` (`type_ventilation`) ON
UPDATE CASCADE,
FOREIGN KEY (`audience`) REFERENCES `audience` (`id`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Создание таблицы «workplace_teacher» (Рабочее место преподавателя):

```

CREATE TABLE `workplace_teacher` (
  `id` SERIAL,
  `name` varchar(50) NOT NULL,
  `count` int(8) UNSIGNED NOT NULL,
  `audience` bigint(20) UNSIGNED NOT NULL,
  PRIMARY KEY(id),
  FOREIGN KEY (`audience`) REFERENCES `audience` (`id`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

2.4 Разработка физической и логической структуры сайта

Логическая структура сайта - это подобный набор тематических рубрик с распределенными по соответствующим разделам документами и заранее спроектированными гиперсвязями между всеми страницами ресурса (рисунок 2.7).

Физическая структура подразумевает алгоритм размещения физических файлов по поддиректориям папки, в которой размещена автоматизированная информационная система (рисунок 2.8).



Рисунок 2.7 - Логическая структура сайта

Логическая структура сайта может изменяться в зависимости от прав доступа разных групп пользователей к тем или иным разделам.

Очевидно, что логическая и физическая структуры могут не совпадать, поскольку в общем случае физическая структура ресурса разрабатывается, исходя из удобства размещения файлов. Однако более или менее точное сохранение порядка следования логических разделов в физической структуре сайта позволяет избежать путаницы при последующем дополнении и обновлении материалов.

В данном дипломном проекте программный продукт реализован по шаблону MVC.

Шаблон MVC описывает простой способ построения структуры приложения, целью которого является отделение бизнес-логики от пользовательского интерфейса. В результате, приложение легче масштабируется, тестируется, сопровождается и, конечно же, реализуется.

В архитектуре MVC модель предоставляет данные и правила бизнес-логики, представление отвечает за пользовательский интерфейс, а контроллер обеспечивает взаимодействие между моделью и представлением.

Типичную последовательность работы MVC-приложения можно описать следующим образом:

При заходе пользователя на веб-ресурс, скрипт инициализации создает экземпляр приложения и запускает его на выполнение. При этом отображается вид, скажем главной страницы сайта.

Приложение получает запрос от пользователя и определяет запрошенные контроллер и действие. В случае главной страницы, выполняется действие по умолчанию (*index*).

Приложение создает экземпляр контроллера и запускает метод действия, в котором, к примеру, содержатся вызовы модели, считывающие информацию из базы данных.

После этого, действие формирует представление с данными, полученными из модели и выводит результат пользователю.

Модель — содержит бизнес-логику приложения и включает методы выборки, обработки и предоставления конкретных данных, что зачастую делает ее очень толстой, что вполне нормально. Модель не должна напрямую взаимодействовать с пользователем. Все переменные, относящиеся к запросу пользователя должны обрабатываться в контроллере. Модель не должна генерировать HTML или другой код отображения, который может изменяться в зависимости от нужд пользователя. Такой код должен обрабатываться в видах. Одна и та же модель, например, модель аутентификации пользователей может использоваться как в пользовательской, так и в административной части приложения. В таком случае можно вынести общий код в отдельный класс и наследоваться от него, определяя в наследниках специфичные для подприложений методы.

Вид — используется для задания внешнего отображения данных, полученных из контроллера и модели. Виды содержат HTML-разметку и небольшие вставки PHP-кода для обхода, форматирования и отображения данных. Не должны напрямую обращаться к базе данных. Этим должны заниматься модели. Не должны работать с данными, полученными из запроса пользователя. Эту задачу должен выполнять контроллер. Может напрямую обращаться к свойствам и методам контроллера или моделей, для получения готовых к выводу данных. Виды обычно разделяют на общий шаблон, содержащий разметку, общую для всех страниц (например, шапку и подвал) и части шаблона, которые используют для отображения данных выводимых из модели или отображения форм ввода данных.

Контроллер — связующее звено, соединяющее модели, виды и другие компоненты в рабочее приложение. Контроллер отвечает за обработку запросов пользователя. Контроллер не должен содержать SQL-запросов. Их лучше держать в моделях. Контроллер не должен содержать HTML и другой разметки. Её стоит выносить в виды. В хорошо спроектированном MVC-приложении контроллеры обычно очень тонкие и содержат только несколько десятков строк кода.










 components	04.03.2019 8:44	Папка с файлами	
 config	04.03.2019 8:44	Папка с файлами	
 controllers	04.03.2019 8:44	Папка с файлами	
 models	04.03.2019 8:44	Папка с файлами	
 uploads	04.03.2019 8:44	Папка с файлами	
 views	04.03.2019 8:44	Папка с файлами	
 web	04.03.2019 8:44	Папка с файлами	
 .htaccess	04.03.2019 8:44	Файл "HTACCESS"	1 КБ
 index.php	04.03.2019 8:44	JetBrains PhpStorm	1 КБ

Рисунок 2.8 - Физическая структура сайта

В папке components находятся все основные файлы (компоненты) такие как:

- Router.php – роутинг сайта;
- Autoload.php – автозагрузка классов;
- Controller.php –родительский класс контроллера;
- DataBase.php – подключение к базе;
- Helper.php – класс с функциями помощниками;
- Model.php – родительский класс модели;
- Pagination.php – класс пагинации;
- View.php – родительский класс представления;
- Widgets.php – класс виджетов;
- XLSXWriter.php – библиотека для экспорта в Excel.

В папке config находятся три файла с конфигурациями, а именно:

- db_params.php – содержит настройки для подключения к базе;
- routes.php – содержит шаблоны обрабатываемых URL;

Сайт использует схему разделения данных MVC которые разделяет web-приложение на три отдельных компонента models, views и controllers.

В папке models находятся файлы:

- AcademicYear.php – модель базы академического года;
- Ok.php – модель таблицы общие компетенции;
- Pk.php – модель таблиц профессиональные компетенции;
- Tools.php – модель таблицы инструменты;
- ReadDisciplines.php – модель таблицы читаемые дисциплины;
- Instr.php – модель таблицы инструкции;
- Disciplines.php – модель таблицы дисциплин;
- SanPin.php – модель таблицы нормы СанПин;

- Employee.php – модель таблицы сотрудников;
- SpecialClothing.php – модель таблицы спецодежда;
- FoundationOfficesFgos.php – модель таблицы фонд кабинетов по ФГОС;
- FoundationFgosAudience.php – модель таблицы соответствия фонда кабинетов ФГОС и аудиторий;
- FoundationFgosInstr.php – модель таблицы соответствия фонда кабинетов ФГОС и инструкций;
- TypeInstr.php – модель таблицы типы инструкций;
- LoginForm.php – модель формы входа;
- TypeMto.php – модель таблицы типы материально-технического оснащения;
- TypeVent.php – модель таблицы типы вентиляции;
- Passport.php – модель таблицы паспорта;
- Posts.php – модель таблицы должностей педагогического работника;
- Rpssz.php – модель таблицы специальностей;
- IndicatorErgonomic.php – модель таблицы научно-методической работы;
- Ventilation.php – модель таблицы вентиляция;
- WorkplaceTeacher.php – модель таблицы рабочее место преподавателя;
- DopWork.php – модель таблицы дополнительные работы;
- Perspect.php – модель таблицы перспективное планирование;
- User.php – модель таблицы пользователей.

Папка controllers содержит контроллеры ко всем моделям таблиц.

В папке views находятся папки представлений всех таблиц также содержится папка layouts которая хранит в себя шаблонные файлы, такие как header и footer разделенные на различные типы из которых формируются все страницы на сайте.

Папка web содержит папки с css – стилями, images – картинками и js – скриптами.

Файл .htaccess – содержит настройки Apache.

Файл index.php – содержит метод для вызова роутера и некоторые настройки, такие как отображение ошибок и т.п.

Следовательно, физическая структура сайта упорядоченная и аккуратная.

2.5 Создание запросов

Запрос – это средство выбора необходимой информации из базы данных. Вопрос, сформированный по отношению к базе данных, и есть запрос.

Запрос используется для получения информации, соответствующей определенным

требованиям, критериям. Помимо этого, с помощью них можно организовать добавление информации в базу данных, осуществлять удаление записей либо их изменить.

MySQL-запросы – это такие же запросы, но работа с которыми производится с помощью синтаксиса языка SQL.

Запрос на выборку данных общих записей из двух таблиц с условием, что значение поля `id_employee` соответствует полю `id` авторизованного пользователя и значение поля `academic_year` соответствует полю `academic_year`, которые хранятся в сессии, эти данные формируется для вывода списка паспортов, которые были созданы данным пользователем. Операторы `limit` и `offset` необходимы для работы пагинации (рисунок 2.9):

```
public function getPassportList($page = 1,$academic_year)
{
    $limit = Passport::SHOW_BY_DEFAULT;
    // Смещение (для запроса)
    $offset = ($page - 1) * self::SHOW_BY_DEFAULT;
    // Соединение с БД
    $db = DataBase::getInstance()->getDb();
    // Текст запроса к БД
    $sql = 'SELECT *,passport.id as passport_id FROM passport INNER JOIN
audience_employee ON passport.id_audience = audience_employee.id_audience WHERE
id_employee = :id_employee AND academic_year = :academic_year ORDER BY passport.id
DESC LIMIT :limit OFFSET :offset';
    // Используется подготовленный запрос
    $result = $db->prepare($sql);
    $result->bindParam(':limit', $limit, PDO::PARAM_INT);
    $result->bindParam(':offset', $offset, PDO::PARAM_INT);
    $result->bindValue(':id_employee',$_SESSION['Auth']['id']);
    $result->bindValue(':academic_year',$academic_year);
    // Выполнение коенды
    $result->execute();
    // Получение и возврат результатов
    $i = 0;
    $PassportList = array();
    while ($row = $result->fetch()) {
        $PassportList[$i]['passport_id'] = $row['passport_id'];
        $PassportList[$i]['id_audience'] = $row['id_audience'];
```

```

$PassportList[$i]['id_foundation'] = $row['id_foundation'];
$PassportList[$i]['discipline'] = $row['discipline'];
$i++;
}
return $PassportList;
}

```





ФГБОУ ВО "Магнитогорский государственный технический университет им. Г.И. Носова" Многопрофильный колледж Паспортизация учебных помещений						
ФОРМИРОВАНИЕ ПАСПОРТА ПАСПОРТА ПРЕПОДАВАТЕЛЯ						
Паспорта преподавателя						
№	Аудитория	Тип	Помещение	Специальность	Дисциплина	
265	A207	Кабинет	социально-экономических дисциплин	08.02.01	ПД.03 Физика 2015	 
264	A207	Кабинет	социально-экономических дисциплин	08.02.01	ПД.02 Информатика 2015	 
1						

Рисунок 2.9 - Результат запроса

2.6 Проектирование пользовательского интерфейса

Интерфейс сайта представляет собой визуальное отображение на экране монитора компьютера. Именно он отвечает за то, насколько удобно пользователю будет взаимодействовать с сайтом. Модуль «Паспортизация помещений» выполнен в соответствии с информационной политикой ФГБОУ ВО «Магнитогорский государственный технический университет им. Г.И.Носова», отвечает концепции фирменного стиля и содержит атрибуты бренда (рисунок 2.10).


 ФГБОУ ВО "Магнитогорский государственный технический университет им. Г.И. Носова" Многопрофильный колледж Паспортизация учебных помещений		Гость Вход
ФОРМИРОВАНИЕ ПАСПОРТА ПАСПОРТА ПРЕПОДАВАТЕЛЯ		
<p>В соответствии с ФГОС образовательная организация, реализующая программы подготовки специалистов, должна располагать материально-технической базой, обеспечивающей проведение всех видов лабораторных работ и практических занятий, дисциплинарной, междисциплинарной и модульной подготовки, учебной практики, предусмотренных учебным планом образовательной организации. Материально-техническая база должна соответствовать действующим санитарным и противопожарным нормам. При этом в каждом ФГОС оговорен перечень кабинетов, лабораторий, мастерских и других помещений для организации учебного процесса. Учебное помещение – это помещение, оснащенное наглядными пособиями, учебным оборудованием, мебелью и техническими средствами обучения, в котором проводится учебная, факультативная и внеклассная работа со студентами в полном соответствии с действующими ФГОС СПО, учебными планами, основными профессиональными образовательными программами и программами подготовки квалифицированных рабочих, служащих.</p> <p>Обязательная паспортизация помещений аудиторного фонда учебного заведения разрабатывается в соответствии с действующим законодательством РФ:</p> <ul style="list-style-type: none"> типовые положения об образовательном учреждении высшего профессионального образования (высшем учебном заведении), утвержденное постановлением Правительства Российской Федерации от 14 февраля 2008 г. № 71 и среднего профессионального образования (среднем специальном учебном заведении), утвержденное постановлением Правительства РФ от 18 июля 2008 г. № 543 [4,5]; Закон РФ 29 декабря 2012 года N 273-ФЗ "Об образовании в Российской Федерации" (с изменениями и дополнениями) [6]; Закон РФ 29 декабря 2012 года N 273-ФЗ "Об образовании в Российской Федерации" (с изменениями и дополнениями) [6]; Федеральные государственные образовательные стандарты высшего и среднего профессионального образования; санитарно-эпидемиологические правила и нормативы СанПиН 2.4.2.2821-10 «Санитарно-эпидемиологические требования к условиям и организации обучения в общеобразовательных учреждениях» (с изменениями и дополнениями) [7]; Устав образовательной организации. <p>Паспортизация помещений учебного заведения проводится ежегодно в начале учебного года, в настоящий момент результаты данной процедуры учитываются</p>		

Рисунок 2.10 - Главная страница сайта

Интерфейс можно разделить условно на 5 частей:

- фиксированная шапка сайта, на которой расположена кнопка «Вход», открывающая форму авторизации (рис. 2.11);
- статическая шапка сайта, на которой расположен логотип и наименование организации, и название информационной системы (рис. 2.12);
- навигация – меню, с помощью которого происходит переход по разделам сайта (рис. 2.13);
- область действий – панель, отображающая возможные действия, предусмотренные системой, такие как фильтрация по записям, добавление новой записи и др. (рис. 2.14);
- область контента – содержит основную информацию раздела, в большинстве случаев это таблицы данных, также в этой области можно удалить запись или перейти к редактированию информации (рис. 2.15).

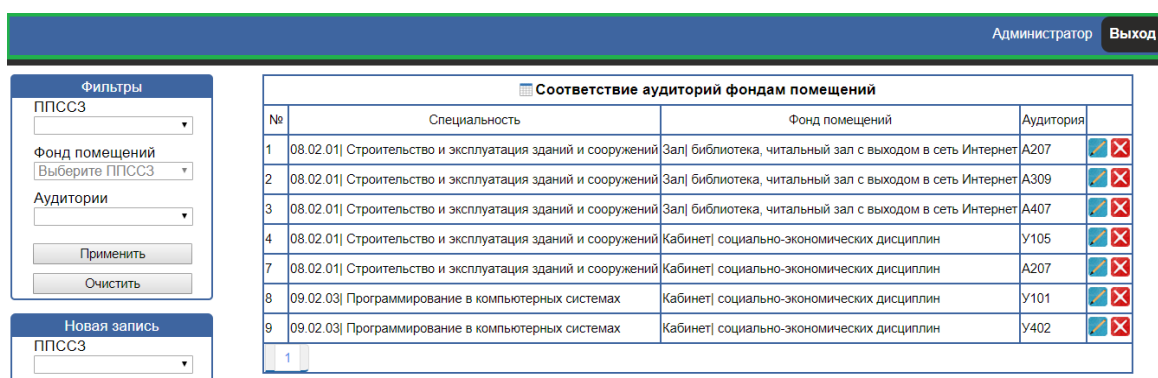


Рисунок 2.11 – Фиксированная шапка сайта



Рисунок 2.12 - Статическая шапка сайта

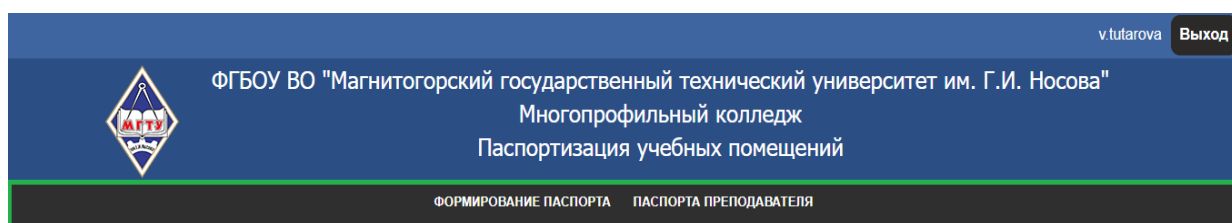


Рисунок 2.13 – Меню навигации

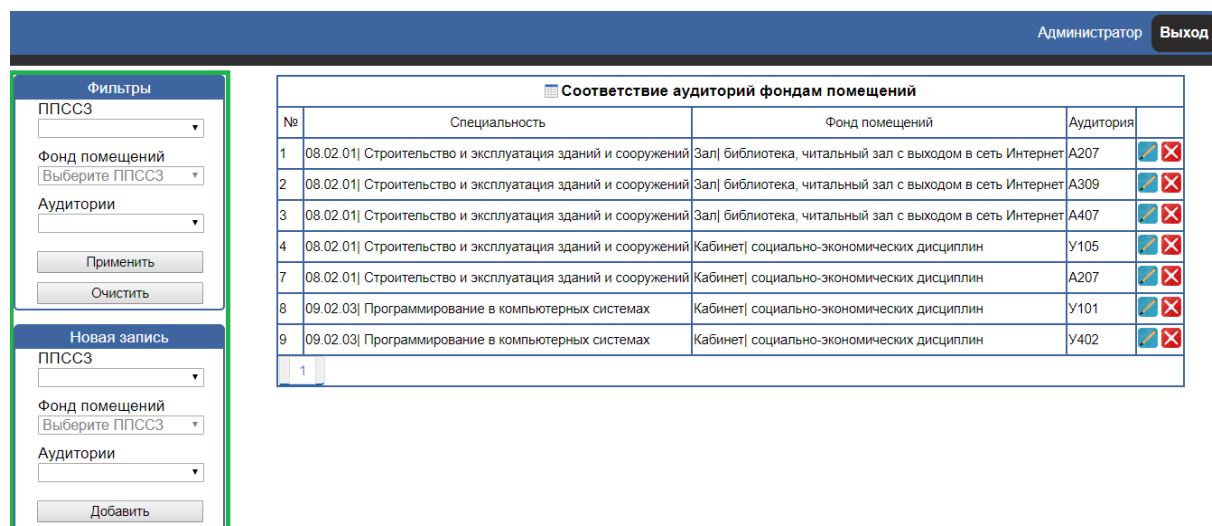


Рисунок 2.14 – Область действий

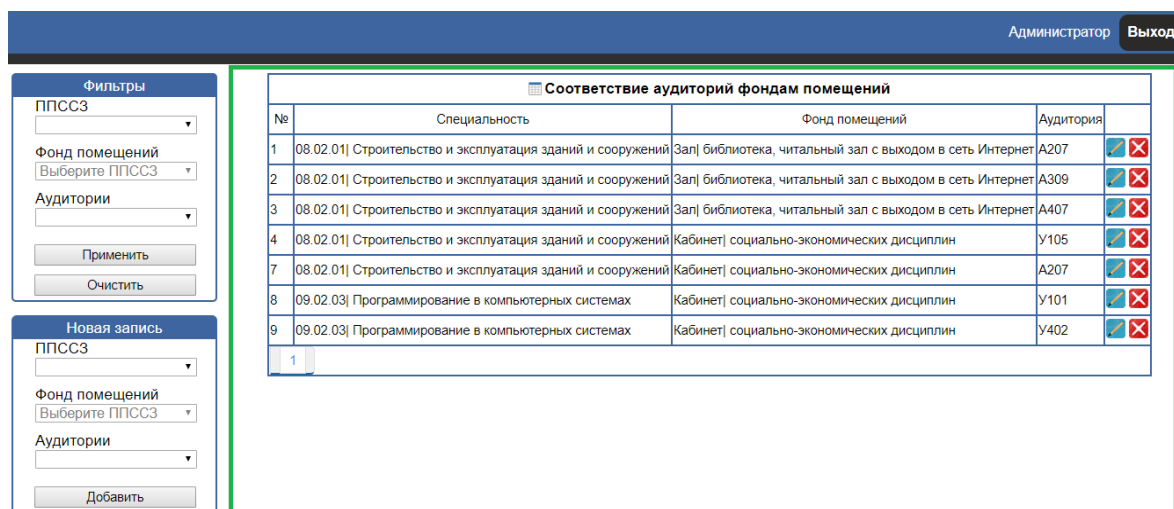


Рисунок 2.15 – Область контента

2.7 Руководство пользователя

Модуль «Паспортизация помещений» разработана с целью автоматизации процесса формирования паспорта помещений для учебных заведений среднего профессионального образования.

Созданное web-приложение позволяет решить следующие задачи:

- формирование паспорта учебного помещения и его проверке в соответствии с требованиями ФГОС;
- удобное редактирование справочной информации необходимой для формирования паспортов помещений;
- формирование и вывод на печать всех паспортов помещений в едином стиле.

2.7.1 Требования к уровню подготовки пользователя

Для выполнения функций пользователя «Ответственный за помещение» сотрудник должен обладать следующими знаниями и навыками:

- знание персонального компьютера;
- умение работать с файловой системой в операционной среде Microsoft Windows;
- иметь теоретические знания о заполнении паспорта помещения.

2.7.2 Требования к программной среде

Для подключения к модулю «Паспортизация помещений» компьютер должен удовлетворять следующим требованиям:

- ОС Windows XP/Windows Vista/Windows7/ Windows8(8.1)/Windows 10;
- доступ к локальной сети Многопрофильного колледжа;
- Google Chrome/Mozilla Firefox/Opera.

2.7.3 Описание операций, выполняемых пользователями модуля «Паспортизация помещений»

Электронная версия паспортизации учебных помещений предназначена для работы в локальной сети Многопрофильного колледжа ФГБОУ ВО «Магнитогорский государственный технический университет им. Г.И. Носова» (МГТУ). При загрузке сайта появится главная страница модуля «Паспортизация помещений» (рисунок 2.10).

Сотрудникам для работы с программой, необходимо авторизоваться в системе, для этого нужно в правой верхней части экрана нажать кнопку «Вход» после чего откроется форма авторизации (рисунок 2.16), на которой необходимо ввести логин и пароль, затем нажать на кнопку «Войти». При успешной авторизации внизу блока появится логин пользователя (рисунок 2.17). Этот шаг является обязательным, так как в связи с политикой

конфиденциальности преподаватель может составлять паспорт помещения и актуализировать его только в соответствии с распоряжением директора колледжа.

Рисунок 2.16 – Форма авторизации

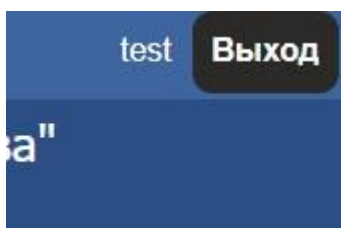


Рисунок 2.17 – Содержание фиксированной шапки при успешной авторизации

Если пользователь авторизовался как ответственный за помещение, то будет необходимо перед переходом по разделам выбрать нужный академический год из выпадающего списка (рис. 2.18).

Рисунок 2.18 – Академический год

Заполнение паспорта

Выберите раздел «Формирование паспорта» на панели главного меню (рис. 2.13) для формирования паспорта кабинета/лаборатории/мастерской.

Поля Помещение, Тип кабинета, Фонд помещений, Аудитория, Дисциплины и Инструкции заполняются выбором элементов из выпадающего списка:

- выбрать специальность (рис. 2.19);

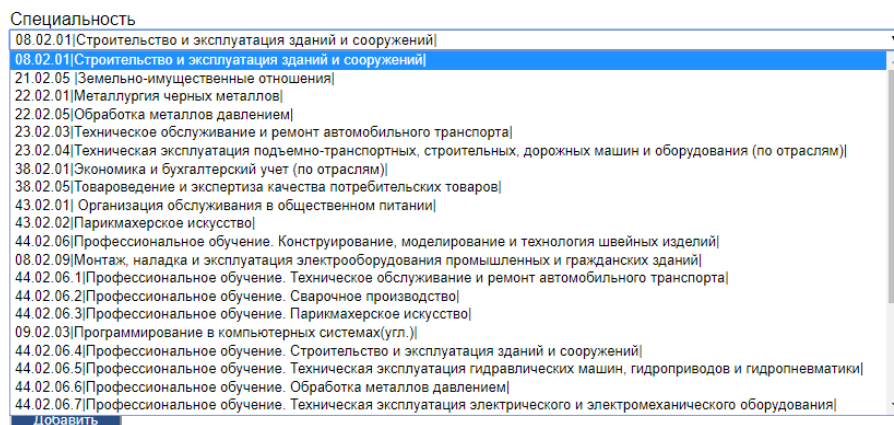


Рисунок 2.19 – Список специальностей

- выбрать тип помещения (рис. 2.20);

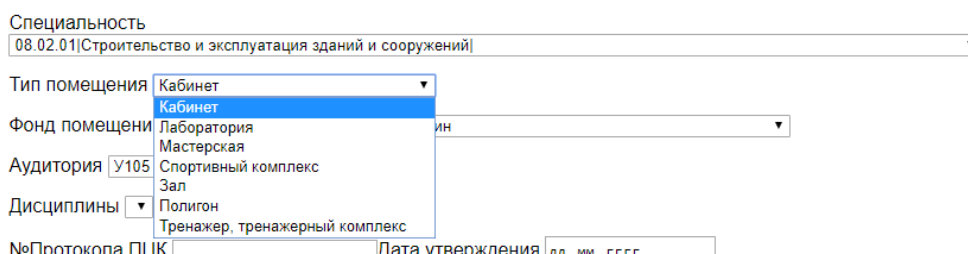


Рисунок 2.20 – Список типов помещений

- при выборе специальности и типа помещения будет сформирован соответствующий список фонда помещений по ФГОС (рис. 2.21);

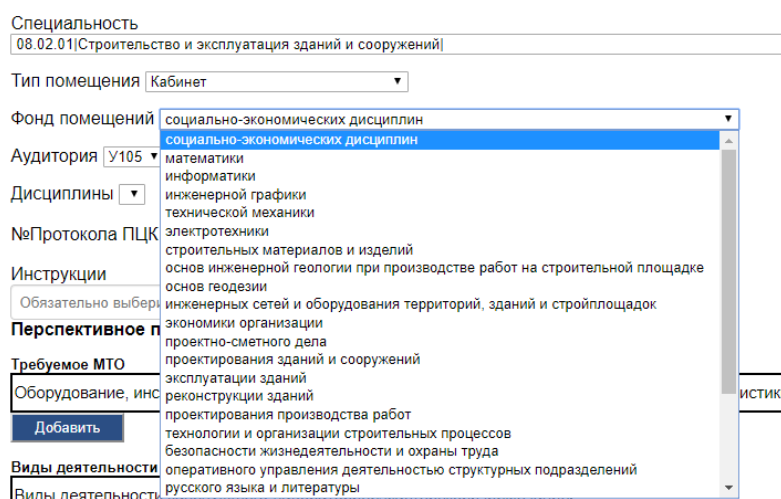


Рисунок 2.21 – Список типов помещений

– в соответствии с выбранным помещением будет сформирован соответствующий список аудиторий (рис. 2.22), дисциплин (рис. 2.23) и инструкций (рис. 2.24).

Фонд помещений социально-экономических дисциплин

Аудитория У105

Дисциплины А207

Рисунок 2.22 – Список аудиторий соответствующих фонду помещений по ФГОС

Фонд помещений социально-экономических дисциплин

Аудитория У105

Дисциплины ПД.02 Информатика 2015

№Протокола ПД.03 Физика 2015 Дата утверждения дд.мм.гггг

Рисунок 2.23 – Список дисциплин соответствующих фонду помещений по ФГОС

Инструкции

Обязательно выберите инструкции

Выбрать всё Убрать выделение

☐ П | СМК-К-ИОТ-02-11 Инструкция по охране труда мастерской малярных работ

☐ ТБ | СМК-ИОТ-01-12 Инструкция по оказанию доврачебной помощи при несчастных случаях

Рисунок 2.24 – Список инструкций соответствующих фонду помещений по ФГОС

Далее требуется заполнение текстового поля «№ Протокола» (рис. 2.25), полей даты «Дата утверждения» (рис. 2.26) и «Дата ежегодной приемки» (рис. 2.27).

№Протокола ПЦК Дата утверждения дд.мм.гггг

Рисунок 2.25 – Текстовое поле «№ Протокола» и поле даты «Дата утверждения»

Дата утверждения дд.мм.гггг

Май 2019

Пн	Вт	Ср	Чт	Пт	Сб	Вс
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

Рисунок 2.26 – Поле даты «Дата утверждения»

Рисунок 2.27 – Поле даты «Дата ежегодной приемки»

Перечисленные списки и поля являются обязательными для заполнения, если будут введены не все необходимые данные, то отобразится сообщение «Введены не все данные» (рис. 2.28).

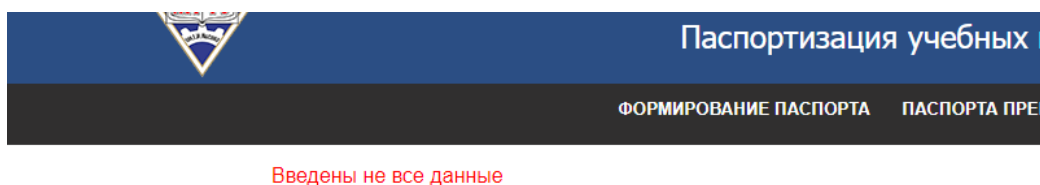


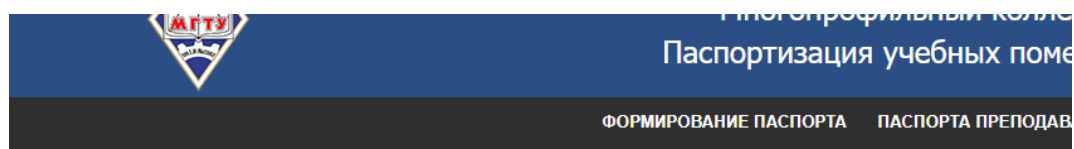
Рисунок 2.28 – Сообщение «Введены не все данные»

После заполнения нужных для добавления паспорта данных, пользователь может заполнить таблицы «Требуемое материально-техническое оснащение» (рис. 2.29) и «Виды деятельности» (рис. 2.30), если данная процедура необходима, для формирования паспорта эти данных необязательны.

Рисунок 2.29 – Требуемое материально-техническое оснащение

Рисунок 2.30 – Виды деятельности

При успешном формировании паспорта пользователь увидит сообщение «Паспорт добавлен» (рис. 2.31).



Паспорт добавлен

Рисунок 2.31 – Сообщение об успешном добавлении паспорта

Редактирование паспорта

Для перехода к редактированию нужно перейти в раздел «Паспорта преподавателя» и выбрать необходимый паспорт из списка (рис. 2.32).





ФОРМИРОВАНИЕ ПАСПОРТА ПАСПОРТА ПРЕПОДАВАТЕЛЯ						
Паспорта преподавателя						
№	Аудитория	Тип	Помещение	Специальность	Дисциплина	
265	A207	Кабинет	социально-экономических дисциплин	08.02.01	ПД.03 Физика 2015	 
264	A207	Кабинет	социально-экономических дисциплин	08.02.01	ПД.02 Информатика 2015	 
1						

Рисунок 2.32 – Паспорта преподавателя

Редактирование выполняется по аналогии с добавлением паспорта с некоторыми отличиями:

- возможность редактирования данных о рабочем месте преподавателя (рис. 2.33);

3.3 Рабочее место		
Наименование	Инвентарный номер	Количество
Доска магнитная	test	3
Карниз	B000468.1	2
Стул офисный	B010000000001516	32
Стол ученический	B010000000001513	23
Кресло офисное	B2101060233	1
Стол компьютерный угловой	B1101060299	1
Штора	B000046.2	3

Рисунок 2.33 – «Рабочее место преподавателя»

- возможность редактирования «Соответствие площади технологическим нормам» (рис. 2.34);

7.1 Соответствие площади технологическим нормам

Количество учебных мест

Количество учебных мест

Рисунок 2.34 – «Соответствие площади технологическим нормам»

- редактирование температурного режима (рис. 2.35);

7.2 Температурный режим

Температура в летний период град. С	Температура в зимний период град. С
<input type="text"/>	<input type="text"/>

Рисунок 2.35 – «Температурный режим»

- добавление источников шума (рис. 2.36);

7.3 Источники шума

Фактический уровень шума, Дб	Источник шума
Данных нет	
<input type="button" value="Добавить"/>	

Рисунок 2.36 – «Источники шума»

- редактирование уровня освещенности (рис. 2.37);

7.4 Уровень освещенности

Вид освещенности	Фактическая освещенность
Естественная, КЕО(%)	<input type="text"/>
Искусственная, Люкс	<input type="text"/>

Рисунок 2.37 – «Уровень освещенности»

- добавление «Обеспечение спецодеждой и индивидуальными защитными средствами» (рис. 2.38);

7.5 Обеспечение спецодеждой и индивидуальными защитными средствами

Наименование СО и индивидуальных ЗС	Обеспеченность (%, шт.)
Данных нет	
<input type="button" value="Добавить"/>	

Рисунок 2.38 – «Обеспечение спецодеждой и индивидуальными защитными средствами»

– добавление и удаление требуемого материально-технического оснащения (рис. 2.39);

8.1 Требуемое материально-техническое оснащение

Наименование	Количество	Примечание	
Молоток	3		<input checked="" type="checkbox"/>

[Добавить](#)

Рисунок 2.39 – «Требуемое материально-техническое оснащение»

– добавление и удаление видов деятельности заведующего лабораторией/мастерской (рис. 2.40).

8.2 Виды деятельности заведующего лабор

Наименование	Примечание	
Покраска стен		<input checked="" type="checkbox"/>

[Добавить](#)

Рисунок 2.40 – «Виды деятельности заведующего лабораторией/мастерской»

2.8 Руководство администратора

После авторизации администратора происходит переход на администраторскую часть сайта (рис. 2.41)

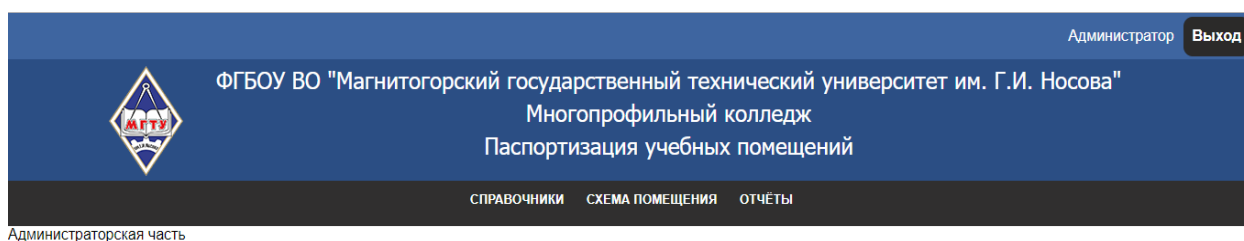


Рисунок 2.41 – «Виды деятельности заведующего лабораторией/мастерской»

У администратора меню отличается от других пользователей, так как имеет другие права доступа.

Основная задача администратора — это работа со справочной информацией, для перехода по справочникам используется подменю «Справочники» (рис. 2.42).

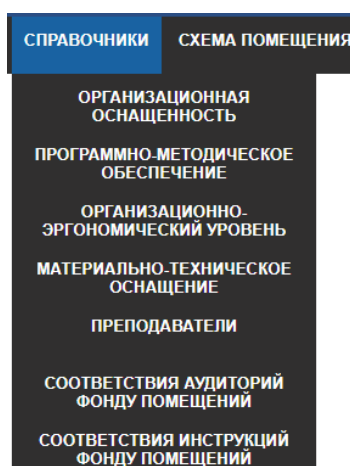


Рисунок 2.42 – Подменю «Справочники»

Администратор переходит на нужный пункт меню и попадает на страницу редактирования справочника (рис. 2.43).

Администратор Выход									
Фильтры		Дисциплины							
ПССЗ		№	ПССЗ	Название	Год ФГОС	Индекс дисциплины	Тип помещения	Фонд помещений	
Фонд помещений		504	22.02.05) Обработка металлов давлением	Автоматизация технологических процессов	2015	МДК.04.01			
Название		687	43.02.02) Парикмахерское искусство	Актуальные тенденции и современные технологии парикмахерского искусства	2016	МДК.03.03			
Год		1302	09.02.03) Программирование в компьютерных системах	Архитектура компьютерных систем	2016	ОП.02	Кабинет	истории	
Индекс дисциплины		25	09.02.03) Программирование в компьютерных системах	Архитектура компьютерных систем	2015	ОП.02			
Применить		610	38.02.01) Экономика и бухгалтерский учет (по отраслям)	Аудит	2016	ОП.09			
Очистить		455	22.02.01) Металлургия черных металлов	Безопасность жизнедеятельности	2015	ОП.10			
Новая запись		614	38.02.01) Экономика и бухгалтерский учет (по отраслям)	Безопасность жизнедеятельности	2016	ОП.10			
Название		383	19.02.10) Технология продукции общественного питания	Безопасность жизнедеятельности	2016	ОП.10			
Индекс дисциплины		682	43.02.02) Парикмахерское искусство	Безопасность жизнедеятельности	2016	ОП.08			
ПССЗ		985	46.02.01) Документационное обеспечение управления и архивоведение	Безопасность жизнедеятельности	2016	ОП.09			
Год ФГОС		233	13.02.11) Техническая эксплуатация и обслуживание электрического и электромеханического оборудования (по отраслям)	Безопасность жизнедеятельности	2015	ОП.12			
Шифр		180	08.02.09) Монтаж, наладка и эксплуатация электрооборудования промышленных и гражданских зданий	Безопасность жизнедеятельности	2015	ОП.05			
Фонд помещений		853	44.02.06.3) Профессиональное обучение. Парикмахерское искусство	Безопасность жизнедеятельности	2015	ОП.13			
Выберите ПССЗ		132	08.02.01) Строительство и эксплуатация зданий и сооружений	Безопасность жизнедеятельности	2015	ОП.07			
Добавить									

Рисунок 2.43 – Подменю «Справочники»

Для удобства работы с большим количеством информации в системе предусмотрены фильтры (рис. 2.44). У каждого справочника свои фильтры, так как разные поля данных.

Фильтры

ППССЗ

Фонд помещений

Название

Год

Индекс дисциплины

Применить

Очистить

Рисунок 2.44 – Панель «Фильтры»

В зависимости от выбранных фильтров меняется содержание таблицы (рис. 2.45).

Фильтры

ППССЗ

09.02.03|Программир

Фонд помещений

Кабинет|социально-э

Название

Год

2016

Индекс дисциплины

Применить

Очистить

Дисциплины

№	ППССЗ	Название	Год ФГОС	Индекс дисциплины	Тип помещения	Фонд помещений	
1283	09.02.03 Программирование в компьютерных системах	Биология	2016	БД.06	Кабинет	социально-экономических дисциплин	✖
1280	09.02.03 Программирование в компьютерных системах	История	2016	БД.03	Кабинет	социально-экономических дисциплин	✖
1292	09.02.03 Программирование в компьютерных системах	История	2016	ОГСЭ.02	Кабинет	социально-экономических дисциплин	✖
1278	09.02.03 Программирование в компьютерных системах	Русский язык и литература	2016	БД.01	Кабинет	социально-экономических дисциплин	✖
1282	09.02.03 Программирование в компьютерных системах	Химия	2016	БД.05	Кабинет	социально-экономических дисциплин	✖
1284	09.02.03 Программирование в компьютерных системах	Экология	2016	БД.07	Кабинет	социально-экономических дисциплин	✖

Рисунок 2.45 – Результат использования фильтров

Для добавления новой записи используется панель «Новая запись» (рис. 2.46).

Новая запись

Название

Индекс дисциплины

ППССЗ

Год ФГОС

Шифр

Фонд помещений

Выберите ППССЗ

Добавить

Рисунок 2.46 – Панель «Новая запись»

Чтобы удалить или редактировать запись нужно нажать кнопки, находящиеся возле каждой записи (рис. 2.47).


12	Балашова	Ирина	Анатольевна	ПЦК Товароведения и экспертизы кач	1) «Информационные технологии и автоматизация технологических процессов»	Штатный	Преподаватель высшей категории	teache	
----	----------	-------	-------------	------------------------------------	--	---------	--------------------------------	--------	---

Рисунок 2.47 – Кнопки «Редактировать» и «Удалить»

Пример формы редактирование изображен на рисунке 2.48.

Аудитория

A104

Тип инструмента

Электрооборудование

Название

Весы

Заказчик

Туркина

Инвентарный номер

T000004131

Количество

0

Дата покупки

Редактировать

Рисунок 2.48 – Форма редактирования

Добавление и удаление пользователей системы также осуществляет администратор, перейдя в раздел «Сотрудники» (рис.2.49) и заполнив необходимую информацию о новом пользователе в форму на панели «Новая запись» (рис. 2.50).

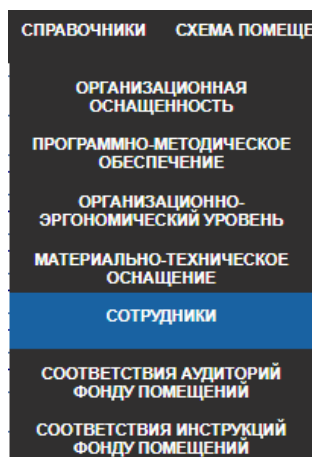


Рисунок 2.49 – Меню «Справочник сотрудников»

Новая запись

Тип инструмента

ПК ПЦК

Тип занятости

Должность

Отделение

Права доступа

Имя

Фамилия

Отчество

Логин

Пароль

Добавить

11	Астахина	Марина	Михайловна	ПК Социально-экономических дисциплин	3) «Механическое, гидравлическое оборудование и транспорт»	Штатный	Преподаватель 1 категории	teacher	
12	Балашова	Ирина	Анатольевна	ПК Товароведения и экспертизы кач	1) «Информационные технологии и автоматизация технологических процессов»	Штатный	Преподаватель высшей категории	teacher	
13	Бахтова	Наталья	Степановна	ПК Строительных и транспортных ма	3) «Механическое, гидравлическое оборудование и транспорт»	Штатный	Преподаватель высшей категории	teacher	
14	Бониковская	Ольга	Сергеевна	ПК Экономики, бухгалтерского учет	4) «Экономика, земельно-имущественные отношения и товароведение»	Штатный	Преподаватель	teacher	
15	Богов	Михаил	Иванович	ПК Товароведения и экспертизы кач	4) «Экономика, земельно-имущественные отношения и товароведение»	Штатный	Преподаватель высшей категории	teacher	
16	Боровских	Ирина	Юрьевна	ПК Строительных и транспортных ма	5) «Строительство и эксплуатация зданий и сооружений»	Штатный	Преподаватель высшей категории	teacher	
17	Булдыгина	Наталья	Васильевна	ПК Технологии продукции обществен	2) «Профессиональное обучение, сервис и документооборот»	Штатный	Преподаватель высшей категории	teacher	
18	Вандышева	Татьяна	Анатольевна	ПК Естественных дисциплин	2) «Профессиональное обучение, сервис и документооборот»	Штатный	Преподаватель 1 категории	teacher	
19	Варкина	Галина	Анатольевна	ПК Строительства и эксплуатации з	5) «Строительство и эксплуатация зданий и сооружений»	Штатный	Преподаватель высшей категории	teacher	
20	Васильева	Елена	Александровна	ПК Информационных технологий прог	1) «Информационные технологии и автоматизация технологических процессов»	Штатный	Преподаватель высшей категории	teacher	
21	Васильева	Марина	Николаевна	ПК Строительных и транспортных ма	3) «Механическое, гидравлическое оборудование и транспорт»	Штатный	Преподаватель высшей категории	teacher	
22	Ветогов	Евгений	Юрьевич	ПК Строительных и транспортных ма	3) «Механическое, гидравлическое оборудование и транспорт»	Штатный	Преподаватель 1 категории	teacher	
23	Вильгаук	Ольга	Анатольевна	ПК Естественных дисциплин	2) «Профессиональное обучение, сервис и документооборот»	Штатный	Преподаватель высшей категории	teacher	

Рисунок 2.50 – Меню «Справочник сотрудников»

3 ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

3.1 Законодательство в области защиты информации

В связи с возрастающим числом компьютерных преступлений проблема защиты информации является приоритетной в настоящих условиях. Защита информации должна придерживаться следующих законов:

- О персональных данных (Закон РФ №152-ФЗ от 27 июля 2006г.) (ред. от 04.06.2014);
- Об электронной цифровой подписи (Закон РФ №63-ФЗ от 30.03.11г.);
- О техническом регулировании (Закон РФ №184-ФЗ от 27.12.02г. с изменениями и дополнениями);
- Федеральный закон "Об информации, информационных технологиях и о защите информации" от 27.07.2006 N 149-ФЗ;
- Федеральный закон от 28 декабря 2010 г. № 390-ФЗ «О безопасности»;
- Федеральный закон от 29 июля 2004 г. № 98-ФЗ «О коммерческой тайне»;
- Федеральный закон от 07 июля 2003 г. № 126-ФЗ «О связи»;
- Федеральный закон от 27 декабря 2002 г. № 184-ФЗ «О техническом регулировании»;
- Трудовой кодекс РФ. Глава 14. «Защита персональных данных работника»;
- Указ Президента Российской Федерации № 260 от 22 мая 2015 года «О некоторых вопросах информационной безопасности Российской Федерации»;
- Указ Президента Российской Федерации № 537 от 12 мая 2009 года «О стратегии национальной безопасности Российской Федерации до 2020 года»;
- Указ Президента Российской Федерации № 351 от 17 марта 2008 года «О мерах по обеспечению информационной безопасности Российской Федерации при использовании информационно-телекоммуникационных сетей международного информационного обмена»;
- Доктрина информационной безопасности Российской Федерации от 9 сентября 2000 г. № Пр-1895.

3.2 Общие положения и уровни информационной безопасности

Информационная безопасность (ИБ) – это состояние информационной системы, при котором она наименее восприимчива к вмешательству и нанесению ущерба со стороны третьих лиц. Безопасность данных также подразумевает управление рисками, которые связаны с разглашением информации или влиянием на аппаратные и программные модули

защиты.

Безопасность информации, которая обрабатывается в организации, – это комплекс действий, направленных на решение проблемы защиты информационной среды в рамках компании. При этом информация не должна быть ограничена в использовании и динамичном развитии для уполномоченных лиц.

Существует 4 уровня информационной безопасности:

1. Программно-технический уровень.

С современной точки зрения информационным системам должны быть доступны следующие механизмы безопасности:

- управление доступом;
- экранирование;
- проверка подлинности пользователей и их идентификация;
- протоколирование и аудит;
- обеспечение высокой доступности;
- криптография.

2. Процедурный уровень.

К нему относятся меры, реализуемые людьми. Опыт, накопленный в отечественных организациях, по реализации процедурных мер пришел из докомпьютерного прошлого и нуждается в существенном пересмотре.

Существуют следующие группы организационных (процедурных) мер:

- управление персоналом;
- поддержание работоспособности;
- планирование восстановительных работ;
- физическая защита;
- реагирование на нарушение безопасного режима.

Для каждой группы должны существовать правила, определяющие действия персонала. Они должны быть учреждены в каждой конкретной организации и отработаны на практике.

3. Административный уровень.

Политика безопасности, предпринимаемая руководством организации, является основой мер административного уровня. Это совокупность документированных решений руководства, которые направлены на защиту информации, а также ресурсов, ассоциированных с ней. Политика безопасности основывается на анализе реальных рисков, угрожающих информационной системе той или иной организации. После анализа разрабатывается стратегия защиты. Это программа, под которую выделяются деньги,

назначаются ответственные, устанавливается порядок контроля ее выполнения и т.д. Поскольку каждая организация имеет свою специфику, бессмысленно переносить практику государственных режимных предприятий на коммерческие структуры, персональные компьютерные системы или учебные заведения. Целесообразнее использовать основные принципы разработки политики безопасности или готовые шаблоны для основных разновидностей организаций.

4. Законодательный уровень.

Это важнейший уровень обеспечения информационной безопасности. В него входит комплекс мер, направленных на создание и поддержание в обществе негативного отношения к нарушителям и нарушениям в этой области. Необходимо создать механизм, который позволил бы согласовывать разработку законов с постоянным совершенствованием информационных технологий. Государство должно выполнять координирующую и направляющую роль в этом вопросе. Российские стандарты информационных технологий и информационной безопасности должны соответствовать международному уровню. Это облегчит взаимодействие с зарубежными компаниями и зарубежными филиалами отечественных компаний. Сейчас эта проблема решается путем разовых разрешений, зачастую в обход действующего законодательства.

Только взаимодействие всех уровней обеспечения информационной безопасности сделают ее максимально эффективной.

3.3 Виды угроз информационной безопасности

Угроза – это возможные или действительные попытки завладеть защищаемыми информационными ресурсами.

Источниками угрозы сохранности конфиденциальных данных являются компании-конкуренты, злоумышленники, органы управления. Цель любой угрозы заключается в том, чтобы повлиять на целостность, полноту и доступность данных.

Угрозы бывают внутренними или внешними. Внешние угрозы представляют собой попытки получить доступ к данным извне и сопровождаются взломом серверов, сетей, аккаунтов работников и считыванием информации из технических каналов утечки (акустическое считывание с помощью жучков, камер, наводки на аппаратные средства, получение виброакустической информации из окон и архитектурных конструкций).

Внутренние угрозы подразумевают неправомерные действия персонала, рабочего отдела или управления фирмы. В результате пользователь системы, который работает с конфиденциальной информацией, может выдать информацию посторонним. На практике

такая угроза встречается чаще остальных. Работник может годами «сливать» конкурентам секретные данные. Это легко реализуется, ведь действия авторизованного пользователя администратор безопасности не квалифицирует как угрозу.

3.4 Организация доступа пользователей к информационным системам

Механизм управления доступом является важным аспектом в безопасности, так как целью такого механизма является предотвращение несанкционированного доступа к компьютерной системе и сетевым ресурсам.

Ненадежное управление доступом пользователей является одной из разновидностей нарушений безопасности информационных систем.

Для предоставления прав доступа к любой информационной системе требуются формальные процедуры. Эти процедуры должны включать в себя все стадии управления доступом пользователей — от начальной регистрации новых пользователей, до удаления учетных записей пользователей, которые больше не нуждаются в доступе к информационным сервисам.

Для управления доступом ко всем многопользовательским информационным системам должна существовать формальная процедура регистрации и удаления учетных записей пользователей.

Доступ к многопользовательским информационным системам необходимо контролировать посредством процесса регистрации пользователей, который должен, например:

- 1) проверять, предоставлено ли пользователю разрешение на использование сервиса владельцем системы;
- 2) проверять, достаточен ли уровень доступа к системе, предоставленного пользователю, для выполнения возложенных на него функций и не противоречит ли он политике безопасности, принятой в организации, например, не компрометирует ли он принцип разделения обязанностей;
- 3) предоставлять пользователям их права доступа в письменном виде;
- 4) потребовать от пользователей подписания обязательства, чтобы показать, что они понимают условия доступа;
- 5) потребовать от поставщиков услуг, чтобы они не предоставляли доступ к системам до тех пор, пока не будут закончены процедуры определения полномочий;
- 6) вести формальный учет всех зарегистрированных лиц, использующих систему;
- 7) немедленно изымать права доступа у тех пользователей, которые сменили работу

или покинули организацию;

8) периодически проверять и удалять пользовательские идентификаторы и учетные записи, которые больше не требуются;

9) проверять, не выданы ли пользовательские идентификаторы, которые больше не нужны, другим пользователям.

Особое внимание следует уделить необходимости управления процессом предоставления привилегированных прав доступа, которые позволяют пользователям обойти средства системного контроля.

Использование специальных привилегий следует ограничить и контролировать так как предоставление и использование излишних системных привилегий зачастую оказывается одним из основных факторов, способствующих нарушению режима безопасности систем.

Для многопользовательских систем, требующих защиты от несанкционированного доступа, предоставление привилегий необходимо контролировать посредством процесса определения полномочий таким образом:

1. Идентифицировать привилегии, связанные с каждым программным продуктом, поддерживаемым системой, например, с операционной системой или СУБД, а также категории сотрудников, которым их необходимо предоставить.

2. Предоставить привилегии отдельным лицам только в случае крайней необходимости и в зависимости от ситуации, т.е. только когда они нужны для выполнения ими своих функций.

3. Реализовать процесс определения полномочий и вести учет всех предоставленных привилегий. Не следует предоставлять привилегии до окончания процесса определения полномочий.

4. Содействовать разработке и использованию системных программ, чтобы избежать необходимость предоставления привилегий пользователям.

5. Пользователи, которым предоставлены большие привилегии для специальных целей, должны использовать другой пользовательский идентификатор для обычной работы.

В настоящее время основным средством подтверждения доступа пользователей к компьютерным системам являются надежные пароли. Назначение паролей необходимо контролировать посредством процесса управления, требования к которому должны быть следующими:

1. Потребовать от пользователей подписания обязательства по хранению персональных паролей в секрете.

2. Передавать временные пароли пользователям надежным способом. Следует избегать передачу паролей через посредников или посредством незашифрованных

сообщений электронной почты. Пользователи должны подтвердить получение паролей.

3. В тех случаях, когда пользователи должны сами выбирать свои пароли, выдать им надежные временные пароли, которые они обязаны немедленно сменить. Временные пароли должны выдаваться только после положительной идентификации пользователя.

В настоящее время уже существуют технологии, такие как, проверка подлинности подписи, которые следует рассмотреть в том случае, если обеспечение более высокого уровня безопасности оправдано.

Для обеспечения эффективного контроля за доступом к данным и информационным системам руководство должно реализовывать процесс пересмотра прав доступа пользователей через регулярные промежутки времени. Такой процесс должен обеспечивать следующее:

- а) пересмотр полномочий доступа пользователей через регулярные промежутки времени; рекомендуется период в 1-3 месяца;
- б) пересмотр разрешения на предоставление специальных привилегированных прав доступа через более короткие промежутки времени; рекомендуется период в 1 месяц;
- в) проверка предоставленных привилегий через регулярные промежутки времени, чтобы не допустить получения пользователями несанкционированных привилегий.

Также важно, чтобы каждый пользователь, для поддержания надлежащего режима безопасности, знал свои обязанности по обеспечению эффективного контроля доступа.

3.5 Обеспечение процедуры резервирования и копирования информации, применение сканеров уязвимости, обеспечение защиты от сбоев в электропитании

Обеспечение процедуры резервирования и копирования информации необходимо для исключения возможности потери информации. Потеря информации происходит в случае поломки оборудования, отвечающего за хранение информации, либо по причине чрезвычайных происшествий (отключение электропитания, пожар, теракт и другие).

Процедуры резервирования и восстановления (Restore) информации разработаны давно и широко используются администраторами информационных систем. Обычный пользователь не знает и не замечает, что на самом деле вся информация, которую он наработал за рабочий день, многократно продублирована на разных физических устройствах и сделано это только с одной единственной целью – обеспечить ее восстановление в случае сбоя.

Различают следующие виды резервирования данных:

- полное;

- дифференциальное (разностное);
- инкрементное.

Полное копирование обеспечивает резервирование всех данных, которые являются критически важными для информационной системы. Оно выполняется с периодичностью, которую определяет администратор. Поскольку информационные базы могут иметь огромный размер, то частое полное резервирование может потребовать значительных вычислительных ресурсов. При сравнительно небольших объемах полное копирование выполняется ежедневно в автоматическом режиме после окончания рабочего дня (например, ночью).

Дифференциальное копирование сохраняет лишь те данные, которые были изменены или добавлены в информационную систему после последнего полного копирования. Поэтому для восстановления требуется два бэкапа – полный и последний дифференциальный. Стремление уменьшить размеры резервного файла привело к созданию инкрементного метода.

При инкрементном подходе каждый раз копируется только та информация, которая изменилась после последнего полного копирования и последнего частичного (инкрементного). В этом случае для восстановления необходимо иметь полный бэкап и все инкрементные бэкапы за прошедший период. Объем памяти под резервные копии меньше, копирование происходит быстрее, чем при дифференциальном копировании, но восстановление сложнее. Управление данными при инкрементном методе усложняется.

Хранение резервных копий – отдельный организационный процесс. Если хранить копии на самом компьютере, то его выход из строя сделает восстановление невозможным или сильно затруднит операцию. Поэтому бэкапы и образы необходимо размещать на других машинах, устройствах и носителях, например, на внешних жестких дисках, на оптических дисках CD, DVD, Blu Ray, в библиотеках магнитных лент, которые используются в стримерах. Чем больше создается копий, тем вероятнее успешное восстановление информации в критических ситуациях, но и тем больше потребность в управлении данными резервных копий, архивы которых не могут вестись бессистемно.

Сканеры защищенности веб-сайтов — это программные (программно-аппаратные) средства, осуществляющие поиск дефектов веб-приложений (уязвимостей), которые приводят к нарушению целостности системных или пользовательских данных, их краже или получению контроля над системой в целом. С помощью сканеров защищенности веб-сайтов можно обнаружить уязвимости следующих категорий: уязвимости этапа кодирования; уязвимости этапа внедрения и конфигурирования веб-приложения; уязвимости этапа эксплуатации веб-сайта. К уязвимостям этапа кодирования относятся уязвимости, связанные

с некорректной обработкой входных и выходных данных (SQL-инъекции, XSS). К уязвимостям этапа внедрения веб-сайта относятся уязвимости, связанные с некорректными настройками окружения веб-приложения (веб-сервера, сервера приложений, SSL/TLS, фреймворк, сторонние компоненты, наличие DEBUG-режима и т. п.). К уязвимостям этапа эксплуатации веб-сайта относятся уязвимости, связанные с использованием устаревшего ПО, простых паролей, хранением архивных копий на веб-сервере в общем доступе, наличием в общем доступе служебных модулей (phpinfo) и т.п.

3.6 Информационная безопасность модуля «Паспортизация помещений»

Защита веб-приложения и защита web-сервера, несомненно, является комплексной задачей, в решении которой может использоваться ряд специализированных программных и программно-аппаратных средств:

- прикладной файрволл (Web Application Firewall - WAF);
- система межсетевого экранирования;
- антивирусное программное обеспечение;
- система управления уязвимостями;
- система обнаружения/предотвращения вторжений;
- система балансировки нагрузки.

Наряду с традиционными средствами защиты, особое внимание следует уделить использованию WAF - системе фильтрации трафика на прикладном уровне, направленной на предотвращение эксплуатации уязвимостей исходного кода web-приложения.

Аппаратно-программный модуль защиты принято разделять на группы, каждая из которых выполняет функцию защиты чувствительной информации:

Уровень идентификации – это комплексная система распознавания пользователей, которая может использовать стандартную или многоуровневую аутентификацию, биометрию (распознавание лица, сканирование отпечатка пальца, запись голоса и прочие приемы).

Уровень шифрования обеспечивает обмен ключами между отправителем и получателем и шифрует/дешифрует все данные системы.

Идентификация – присвоение субъектам и объектам идентификатора и / или сравнение идентификатора с перечнем присвоенных идентификаторов.

Аутентификация – подтверждение подлинности чего-либо или кого-либо.

Авторизация является функцией определения прав доступа к ресурсам и управления этим доступом.

Идентификация — это называние лицом себя системе; аутентификация — это

установление соответствия лица названному им идентификатору; а авторизация — предоставление этому лицу возможностей в соответствии с положенными ему правами или проверка наличия прав при попытке выполнить какое-либо действие.

Информационная безопасность сайта — это реализация авторизации в этой системе, путем аутентификации и идентификации пользователей.

Помимо авторизации для более надежной защиты следует хешировать данные с помощью хеш-функций. В данной автоматизированной информационной системе использовалась хеш-функция MD5.

Хеш-функция предназначена для свертки входного массива любого размера в битовую строку, для MD5 длина выходной строки равна 128 битам. Сравнивая хеши введенных пользователем и содержащихся в базе данные, проверяется совпадают ли эти данные.

Для обеспечения информационной безопасности и разграничения доступа данная система содержит аутентификацию и авторизацию. Для аутентификации пользователя вводится имя и пароль, после чего ему предоставляется вход в систему. После того, как подсистема произвела аутентификацию пользователя, она определяет набор, разрешенных ему действий для доступа к данным, а именно:

- ответственные за помещение имеют доступ к заполнению и редактированию паспорта только закрепленной за ними распоряжением директора лаборатории/мастерской/кабинета;
- ответственные по инвентаризации — права на редактирование, добавление и удаление информации о МТО;
- заведующий учебно-лабораторным комплексом (УЛК) — полный доступ к информации модуля, формирование поисковых запросов для составления отчетной документации;
- администратор — полный доступ к системе.

4 ЭКОНОМИЧЕСКАЯ ЧАСТЬ

Целью экономического раздела дипломной работы является расчет стоимости разработки модуля «Паспортизация помещения» в рамках информатизации деятельности Многопрофильного колледжа. Ведь правильно рассчитанные затраты помогут успешной работе и внедрению программного обеспечения. В данной работе я определил цену программного продукта, рассчитав трудоемкость разработки программного продукта, показал расчет затрат на разработку программного обеспечения и цену интеллектуального труда.

Трудоемкость – это затраты рабочего времени на производство единицы продукции.

Трудоемкость разработки модуля «Паспортизация помещения» в чел.час определяется по формуле (4.1).

$$T = T_o + T_i + T_a + T_{п} + T_{отл} + T_{д}, \quad (4.1)$$

где T – общие затраты труда, чел.час;

T_o – затраты труда на описание задачи;

T_i – затраты на исследование предметной области;

T_a – затраты на разработку алгоритма решения задачи;

$T_{п}$ – затраты на программирование;

$T_{отл}$ – затраты на отладку программы;

$T_{д}$ – затраты на подготовку документации.

Все составляющие определяются через условное число операторов Q по формуле (4.2).

$$Q = q * c * (1 + p), \quad (4.2)$$

где q – число операторов (оператор – предписание о выполнении конкретных действий. Например, в языке программирования Паскаль содержится 11 операторов.);

c – коэффициент сложности задачи (принимается от 1,25 2);

p – коэффициент коррекции программы, учитывающий новизну проекта (для совершенно новой программы равен 0,1).

$$Q = 1000 * 1,7 * (1 + 0,1) = 1870$$

T_o – принимаем в диапазоне от 30 до 40 чел.час (в зависимости от сложности продукта).

T_i определяется по формуле (4.3).

$$T_i = (Q * B) / (S_n * k), \quad (4.3)$$

где B – коэффициент увеличения затрат труда в следствие недостаточного описания задачи (1,2 – 1,5);

S_n – количество операторов, приходящихся на 1 чел.час (75-85);

k – коэффициент квалификации работника (см. таблицу 4.1).

Таблица 4.1 – Квалификация работников

Стаж	Коэффициент
до 2-х лет	0,8
2- 3 года	1,0
3-5 лет	1,1 – 1,2
5 – 7 лет	1,3 – 1,4
свыше 7 лет	1,5 – 1,6

$$T_i = (1870 * 1,2) / (78 * 1,2) = 23,97 \text{ чел. час}$$

T_a рассчитывается по формуле (4.4).

$$T_a = Q / (S_a * k), \quad (4.4)$$

S_a - количество операторов на разработку алгоритма решения задачи, приходящихся на 1 чел.час (20 – 25)

$$T_a = 1870 / (20 * 1,2) = 77,91 \text{ чел.час}$$

Затраты труда на программирование найдем по формуле (4.5).

$$T_{\Pi} = Q / (S_{\Pi} * k), \quad (4.5)$$

S_{Π} - количество операторов на программирование, приходящихся на 1 чел.час (20 – 25)

$$T_{\Pi} = 1870 / (20 * 1,2) = 77,91 \text{ чел.час}$$

Затраты труда на отладку программы находим по формуле (4.6).

$$T_{\text{отл}} = Q / (S_{\text{отл}} * k), \quad (4.6)$$

$S_{\text{отл}}$ - количество операторов на отладку программы, приходящихся на 1 чел.час (4 – 5)

$$T_{\text{отл}} = 1870 / (2,5 * 1,2) = 623,33 \text{ чел.час}$$

Затраты на подготовку документации по задаче определяются по формуле (4.7).

$$T_{\text{д}} = T_{\text{др}} + T_{\text{до}}, \quad (4.7)$$

где $T_{\text{др}}$ – затраты труда на подготовку материала рукописи, рассчитываемые по формуле (4.8);

$T_{\text{до}}$ – затраты труда на редактирование, печать и оформление документов.

$$T_{\text{др}} = Q / (S_{\text{др}} * k), \quad (4.8)$$

$S_{\text{др}}$ - количество операторов на подготовку материала рукописи, приходящихся на 1 чел.час (15 – 20).

$$T_{\text{др}} = 1870 / (17 * 1,2) = 91,66 \text{ чел.час}$$

$T_{\text{до}}$ определяется по формуле (4.9).

$$T_{\text{до}} = 0,75 * T_{\text{др}}, \quad (4.9)$$

$$T_{\text{до}} = 0,75 * 91,66 = 68,74 \text{ чел.час}$$

$$T_{\text{д}} = 91,66 + 68,74 = 160,4 \text{ чел.час}$$

$$T = 35 + 23,97 + 77,91 + 77,91 + 623,33 + 160,4 = 998,52 \text{ чел. час.}$$

Полученное значение общей трудоемкости необходимо скорректировать с учетом уровня языка программирования:

$$T_{\text{кор}} = T * k_{\text{кор}}, \quad (4.10)$$

где $k_{\text{кор}}$ - коэффициент, учитывающий уровень языка программирования (0,8 – 1)

$$T_{\text{кор}} = 998,52 * 0,9 = 898,66 \text{ чел. час.}$$

Заработная плата – это вознаграждение или денежная компенсация за труд в зависимости от квалификации работника, сложности, количества, качества и условий работы.

Формы и системы заработной платы:

- сдельная;
- повременная.

При сдельной системе оплаты труда заработная плата работника зависит от количества изготовленной этим работником продукции (оказанных услуг или выполненных работ).

При простой повременной форме заработной платы оплата труда производится за определенное количество отработанного времени независимо от количества выполненных работ.

Тарифная ставка – это фиксированный размер оплаты труда работника за выполнение нормы труда определенной сложности (квалификации) за единицу времени.

Часовая тарифная ставка инженера технолога определяется по формуле (4.11).

$$T_{\text{ст}} = \text{МРОТ} / \text{Вф}, \quad (4.11)$$

где МРОТ – минимальный размер оплаты труда с 01.01.2019г. (оклад сотрудника), равный 11280 руб./мес;

Вф – фактически отработанное время (168 час. в месяц, т.е 21 смена *8 часов).

$$T_{\text{ст}} = 11280 / 168 = 67,14 \text{ руб.}$$

Время работы персонального компьютера при создании программного продукта определяется по формуле (4.12).

$$\Phi_B = 1,15 * (T_{\text{п}} + T_{\text{д}} + T_{\text{отл}}) * k_{\text{кор}} \quad (4.12)$$

$$\Phi_B = 1,15 * (77,91 + 160,4 + 623,33) * 0,9 = 891,79 \text{ час.}$$

Себестоимость продукции (работ, услуг)- это стоимостная цена используемых в процессе производства продукции (работ, услуг) природных ресурсов, сырья, материалов, топлива, энергии, основных фондов, трудовых ресурсов, а также других затрат на ее производство и реализацию.

Затраты, образующие себестоимость продукции (работ, услуг) группируются в соответствии с их экономическим содержанием по следующим элементам:

- материальные затраты (за вычетом стоимости возвратных отходов);
- затраты на оплату труда;
- отчисления на социальные нужды;
- амортизация основных средств;
- прочие затраты.

Материальные затраты отражают стоимость приобретаемого со стороны материалов; стоимость покупных материалов; стоимость покупных комплектующих изделий; стоимость работ и услуг производственного характера, выплачиваемых сторонним организациям, выработку всех видов энергии, отопления зданий, стоимость покупной энергии всех видов, расходуемой на технологические, энергетические и прочие нужды.

Затраты на оплату труда отражают затраты на оплату труда основного производственного персонала предприятия, включая премии рабочим и служащим за производственные результаты, стимулирующие и компенсирующие выплаты.

Отчисления на социальные нужды до недавнего времени отражали обязательные отчисления от затрат на оплату труда работников, включаемых в себестоимость продукции (работ, услуг). Эти отчисления производили по установленным законодательством нормам органам государственного социального страхования, Пенсионного фонда, государственного фонда занятости и медицинского страхования.

С 1 янв. 2001 г. все отчисления в социальные внебюджетные фонды заменил единый социальный налог.

Прочие затраты – это налоги, сборы, отчисления во внебюджетные фонды, платежи по кредитам в пределах ставок, затраты на командировки, по подготовке и переподготовке кадров, плата за аренду, износ по нематериальным активам, ремонтный фонд, платежи по обязательному страхованию имущества и т.д.

Калькулирование себестоимости - это исчисление себестоимости единицы продукции или услуг по статьям расходов. В отличие от элементов сметы затрат, статьи калькуляции себестоимости объединяют затраты с учетом их конкретного целевого назначения и места образования.

Таблица 4.2 – Затраты на разработку программного продукта

Наименование статьи	Формула для расчета	Сумма затрат
Основная ЗП с учетом коэффициента корректировки	$ЗПо = (Тотл + Тп) * k_{кор} * Тст + (Тдр + Та) k_{кор} * Тст + (Тп + Ти + То) k_{кор} * Тст$ $= (623,33 + 77,91) * 0,9 * 67,14 + (91,66 + 77,91) * 0,9 * 67,14 + (77,91 + 23,97 + 35) * 0,9 * 67,14 = 60890,66 \text{ руб.}$	60890,66 руб.
Дополнительная заработная плата	$ЗПд = 0,2 * ЗПо = 0,2 * 60890,66 = 12178,13 \text{ руб.}$	12178,13 руб
Отчисления на социальные нужды	$ЕСН = (ЗПо + ЗПд) * 0,3 = (60890,66 + 12178,13) * 0,3 = 21920,63$	21920,63 руб.
Расходы на материалы и запчасти (покупка кулера)	Зм от 650 руб. до 1000 руб.	650 руб.
Затраты на ТО и ТР	2 % от балансовой стоимости оборудования	600 руб.
Затраты на потребляемые энергоресурсы	$Зэ = 0,22 \text{ кВт}\cdot\text{ч} * 3,19 \text{ руб.} * 891,79 \text{ ч.} = 625,85 \text{ руб.}$	625,85 руб.
Итого (Зи):		96865,27 руб.

Стоимость услуг за разработку программного продукта по договору с покупателем 125000 руб.

Прибыль -это те деньги, которые остаются у предприятия после того, как из балансовой прибыли вычитаются различные отчисления, налоги и прочие выплаты.

Определим прибыль от реализации программного продукта заказчику по формуле (4.13).

$$Пр = Вр - Зи, \quad (4.13)$$

где $Вр$ – выручка от реализации программного продукта (деньги от заказчика);
 $Зи$ – итоговая сумма затрат на разработку.

$$Пр = 125000 - 96865,27 = 28134,73 \text{ руб.}$$

ЗАКЛЮЧЕНИЕ

Для рационального и эффективного использования рабочего времени ответственных за оформление документации по паспортизации учебных помещений, а также для унификации внешнего представления паспортов в Многопрофильном колледже ФГБОУ ВО «Магнитогорский государственный технический университет им. Г.И. Носова» разработан модуль «Паспортизация помещений», который позволил хранить и редактировать информацию об аудиторном фонде, его материальном, техническом и эргономическом обеспечении, выполнять поисковые запросы административным работникам, формировать паспорта помещений в едином стиле, создавать отчеты.

Модуль «Паспортизация помещений» в рамках информатизации деятельности Многопрофильного колледжа разработан с целью автоматизации процесса формирования паспорта учебного помещения для учебных заведений среднего профессионального образования.

Система выполняет функции создания, редактирования, печати, хранения паспортов учебных помещений, а также формирования отчетов и хранение всей информации об аудиторном фонде учебного заведения.

Модуль разработана в виде web-приложения, что позволяет использовать ее без предварительной подготовки рабочего места. Система является многопользовательской, при этом преподаватель может заполнить паспорт только принадлежащей ему аудитории.

Модуль «Паспортизация помещений» проходит апробацию с сентября 2016 года в Многопрофильном колледже ФГБОУ ВО «Магнитогорский государственный технический университет им. Г.И. Носова». Внедрение модуля в управление образованием позволило выявлять соответствие помещений ФГОС при подготовке специалистов, по наличию материально-технической базы, обеспечивающей проведение всех видов лабораторных работ и практических занятий, дисциплинарной, междисциплинарной и модульной подготовки, учебной практики, рационально и эффективно использовать рабочее время ответственных за оформление паспортов и унифицировать их внешнее представление.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Архитектура и проектирование программных систем : монография / С.В. Назаров. — 2-е изд., перераб. и доп. — М.: ИНФРА-М, 2017. — 374 с. — (Научная мысль). — [www.dx.doi.org/ 10.12737/18292](http://www.dx.doi.org/10.12737/18292). - Режим доступа: <http://znanium.com/catalog/product/907016> (Дата обращения: 11.01.2019)
2. Агальцов, В.П. Базы данных. В 2-х кн. Книга 2. Распределенные и удаленные базы данных [Электронный ресурс]: учебник / В.П. Агальцов. — М.: ИД «ФОРУМ»: ИНФРА-М, 2017. — 271 с. - Режим доступа: <http://znanium.com/catalog/product/652917> (Дата обращения 20.01.2019)
3. Ананьева, Т. Н. Стандартизация, сертификация и управление качеством программного обеспечения: Учебное пособие / Ананьева Т.Н., Новикова Н.Г., Исаев Г.Н. - М.: НИЦ ИНФРА-М, 2016. - 232 с.: [Электронный ресурс], Режим доступа: <http://znanium.com/bookread2.php?book=541003> (Дата обращения: 15.04.2019).
4. Гагарина, Л. Г. Технология разработки программного обеспечения: учеб. пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Виснадул ; под ред. Л.Г. Гагариной. — М.: ИД «ФОРУМ»: ИНФРА-М, 2017. — 400 с. — (Высшее образование). - Режим доступа: <http://znanium.com/catalog/product/768473> (Дата обращения: 22.01.2019).
5. Гагарина, Л.Г. Разработка и эксплуатация автоматизированных информационных систем [Электронный ресурс]: Учебное пособие / Гагарина Л.Г. - М.:ИД ФОРУМ, НИЦ ИНФРА-М, 2017. - 384 с. - Режим доступа: <http://znanium.com/catalog/product/612577> (Дата обращения: 23.04.2019).
6. Дунаев В.В. HTML, скрипты и стили.-4 изд., перераб. и доп.- СПб.: БХВ-Петербург, 2015.-816 с. - Режим доступа: <http://znanium.com/bookread2.php?book=350807> (Дата обращения: 02.03.2019).
7. Исаченко О.В. Программное обеспечение компьютерных сетей: учеб.пособие/О.В. Исаченко.-М.:ИНФРА-М,2017-117с.-(Среднее профессиональное образование).Режим доступа: <http://znanium.com/bookread2.php?book=851518> (Дата обращения 25.01.2019)
8. Колдаев, В. Д. Основы алгоритмизации и программирования: Учебное пособие / В.Д. Колдаев; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ: ИНФРА-М, 2016. - 416 с.: [Электронный ресурс], Режим доступа: <http://znanium.com/bookread2.php?book=537513> (Дата обращения: 20.01.2019).
9. Колдаев, В. Д. Основы алгоритмизации и программирования: Учебное пособие / Колдаев В.Д; Под ред. проф.Л.Г. Гагариной - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2016. - 416

с.: [Электронный ресурс], Режим доступа: <http://znanium.com/bookread2.php?book=537513> (Дата обращения: 11.01.2019).

10. Курсы по программированию: [Электронный ресурс] // URL: <https://htmlacademy.ru/> (Дата обращения 04.03.2019)

11. Немцова, Т.И. Программирование на языке высокого уровня. Программирование на языке C++: учеб. пособие / С.Ю. Голова, А.И. Терентьев; под ред. Л.Г. Гагариной. — М.: ИД «ФОРУМ»: ИНФРА-М, 2018. — 512 с. [Электронный ресурс], Режим доступа: <http://znanium.com/bookread2.php?book=918098> (Дата обращения 20.01.2019)

12. Портал по php, MySQL и другим веб-технологиям [электронный ресурс] : учебное пособие / Дж. Уайттеккер, Дж. Воас // Режим доступа : <http://www.php.ru> (Дата обращения: 06.02.2019)

13. Ресурсы по программированию на php: [Электронный ресурс] // URL: <http://www.php.net> (Дата обращения: 11.01.2019)

14. Сидоров, И. Д. Машинно-ориентированное программирование [Электронный ресурс]: Учебное пособие / Абрамов Е.С., Сидоров И.Д. - Таганрог:Южный федеральный университет, 2016. - 87 с.: ISBN 978-5-9275-2065-7 - Режим доступа: <http://znanium.com/catalog/product/991870> (Дата обращения 20.01.2019)

15. Шустова, Л.И. Базы данных: учебник / Л.И. Шустова, О.В. Тараканов. — М.: ИНФРА-М, 2017. — 304 с. + Доп. материалы [Электронный ресурс]. — Режим доступа: <http://znanium.com/catalog/product/751611> (Дата обращения 20.01.2019)

ПРИЛОЖЕНИЕ А

(обязательное)

Листинг 1 - Widgets.php – класс виджетов

```
class Widgets
{
    //Получить необходимые значений выбранных полей таблиц по id
    public static function getDataTableById($sql,$need,$dataId,$id)
    {
        if(empty($sql) or empty($id) or empty($need) or empty($dataId)){
            return false;
        }
        $db = DataBase::getInstance()->getDb();
        $return = "";
        $sql.=' WHERE '.$dataId.'='.$id;
        $result = $db->prepare($sql);
        $result->execute();
        $row = $result->fetch();
        if (!is_array($need))
        {
            $return = $row[$need];
        }
        else
        {
            for ($i=0; $i < count($need); $i++)
            {
                if ($i != count($need)-1)
                {
                    $return.=$row[$need[$i]].' ';
                } else $return.=$row[$need[$i]];
            }
        }
        return $return;
    }
}
```

```

// Выпадающий список с массивом
public static function dropDownSimple($data,$name,$select = -1,$offset = 0)
{
    echo "<select id='".$name.'" name='".$name.'">
        <option value=""></option>";
    for ($i = 0+$offset; $i < count($data)+$offset; $i++)
    {
        if ($select == $i)
            echo "<option value='".$data[$i]."' selected>".$data[$i]."</option>";
        else
            echo "<option value='".$data[$i]."'>".$data[$i]."</option>";
    }
    echo "</select>";
}

// Выпадающий список с использованием базы данных
public static function dropDownList($sql,$fieldId,$fieldData,$name,$isEmpty =
true,$required = true,$fieldForeignData = array())
{
    $db = DataBase::getInstance()->getDb();
    $result = $db->prepare($sql);
    if (!empty($fieldForeignData)) // Внешние параметры
        $result->execute($fieldForeignData);
    else $result->execute();
    $result->execute();
    if ($result->rowCount() > 0)
    {
        if ($required)
        {
            echo "<select required id='".$name.'" name='".$name.'">";
        }
        else echo "<select id='".$name.'" name='".$name.'">";
        if($isEmpty)
        {
            echo "<option value=""></option>";
        }
    }
}

```

```

    }
    else
        echo "<select id='".$name.'" name='".$name.'" disabled>";
        while ($row = $result->fetch())
        {
            if (is_array($fieldData)) // Если массив с данными
            {
                $data = "";
                for ($i = 0; $i < count($fieldData); $i++)
                    $data .= $row[$fieldData[$i]] . '|';
                echo "<option value='".$row[$fieldId]."'>".$data."</option>";
            }
            else
            {
                echo "<option value='".$row[$fieldId]."'>".$row[$fieldData."</option>";
            }
        }

        echo "</select>";
    }

    // Множественный список с использованием базы данных
    public static function dropDownMultipleList($sql,$fieldId,$fieldData,$name,$isEmpty =
true,$required = true,$fieldForeignData = array())
    {
        $db = DataBase::getInstance()->getDb();
        $result = $db->prepare($sql);
        if (!empty($fieldForeignData)) // Внешние параметры
            $result->execute($fieldForeignData);
        else
            $result->execute();
        $result->execute();
        if ($result->rowCount() > 0)
        {
            if ($required)
            {
                echo "<select required multiple = 'multiple' id='".$name.'" name='".$name.'"[]">";

```

```

    } else
    echo "<select multiple = 'multiple' id='".$name.'" name='".$name.'"[]">";
    if($isEmpty)
    {
        echo "<option value='></option>";
    }
    }
    else
        echo "<select id='".$name.'" name='".$name.'" disabled>";
    while ($row = $result->fetch())
    {
        if (is_array($fieldData)) // Если массив с данными
        {
            $data = "";
            for ($i = 0; $i < count($fieldData); $i++)
                $data .= $row[$fieldData[$i]] . '|';
            echo "<option value='".$row[$fieldId].">".$data."</option>";
        }
        else
        {
            echo "<option value='".$row[$fieldId].">".$row[$fieldData."</option>";
        }
    }
    echo "</select>";
}

// Множественный список с использованием базы данных и заранее выбранный ID
public static function dropDownMultipleSelectIdList
($sql,$fieldId,$fieldData,$fieldSelectData,$name,$isEmpty = false,$fieldForeignData = array())
{
    $db = DataBase::getInstance()->getDb();
    $result = $db->prepare($sql);
    if (!empty($fieldForeignData)) // Внешние параметры
        $result->execute($fieldForeignData);
    else
        $result->execute();
}

```

```

echo "<select multiple id='".$name.'" name='".$name.'">";
if ($isEmpty)
echo "<option value=''></option>";
while ($row = $result->fetch())
{
    if (is_array($fieldData)) // Если массив с данными
    {
        $data = "";
        for ($i = 0; $i < count($fieldData); $i++)
            $data .= $row[$fieldData[$i]] . '|';
        if ($fieldSelectData == $row[$fieldId])
            echo "<option value='".$row[$fieldId]."' selected>".$data."</option>";
        else
            echo "<option value='".$row[$fieldId]."'>".$data."</option>";
    }
    else
    {
        if ($fieldSelectData == $row[$fieldId])
            echo "<option value='".$row[$fieldId]."' selected>".$row[$fieldData]."</option>";
        else
            echo "<option value='".$row[$fieldId]."'>".$row[$fieldData]."</option>";
    }
}
echo "</select>";
}

```

// Выпадающий список с использованием базы данных и заранее выбранный ID

```

public static function dropDownList ($sql, $fieldId, $fieldData, $fieldSelectData, $name,
$isEmpty = false, $required = true, $fieldForeignData = array())
{
    $db = DataBase::getInstance()->getDb();
    $result = $db->prepare($sql);
    if (!empty($fieldForeignData)) // Внешние параметры
        $result->execute($fieldForeignData);
    else    $result->execute();
    if ($required) {

```

```

        echo "<select required id='".$name.'" name='".$name.'">";
    } else
        echo "<select id='".$name.'" name='".$name.'">";
if ($isEmpty)
    echo "<option value='></option>";
while ($row = $result->fetch())
{
    if (is_array($fieldData)) // Если массив с данными
    {
        $data = "";
        for ($i = 0; $i < count($fieldData); $i++)
            $data .= $row[$fieldData[$i]] . '|';
        if ($fieldSelectData == $row[$fieldId])
            echo "<option value='".$row[$fieldId]."' selected>".$data."</option>";
        else
            echo "<option value='".$row[$fieldId]."'>".$data."</option>";
    }
    else
    {
        if ($fieldSelectData == $row[$fieldId])
            echo "<option value='".$row[$fieldId]."' selected>".$row[$fieldData]."</option>";
        else
            echo "<option value='".$row[$fieldId]."'>".$row[$fieldData]."</option>";
    }
}

echo "</select>";
}
}

```


ПРИЛОЖЕНИЕ Б

(обязательное)

Листинг 2 - Model.php— класс модели

```
class Model
{
    private $class;

    function __construct()
    {
        $this->class = mb_strtolower(get_class($this), 'UTF-8');
    }

    //Получение значение полей таблицы по id
    public function getItemById($table_name,$id,$key_table)
    {
        $sql = "SELECT * FROM $table_name WHERE $key_table = :id";
        $db = DataBase::getInstance()->getDb();

        // Используется подготовленный запрос
        $result = $db->prepare($sql);
        $result->bindValue(':id', $id);
        // Выполнение коменды
        $result->execute();
        $row=$result->fetch();
        foreach ($this->options as $option)
        {
            $this->{$option} = $row[$option];
        }
    }

    //Получение количества записей
    public function getTotalItemByTable($table_name,$table_key)
    {
        $db = DataBase::getInstance()->getDb();
        // Текст запроса к БД
        $sql = "SELECT count($table_key) AS count FROM $table_name";
        if (isset($_SESSION['table_filter'][$this->class]) && $_SESSION['table_filter'][$this->class])
```

```

{
    $filter = $_SESSION['table_filter'][$this->class];
    $i = 0;
    $sql .= ' WHERE';
    foreach ($filter as $item => $value)
    {
        if ($i != count($filter)-1)
        {
            $sql .= ' '.$item.' =:'.$item.' AND ';
            $i++;
        }
        else $sql .= ' '.$item.' =:'.$item.' ';
    }
}
$result = $db->prepare($sql);
if (isset($_SESSION['table_filter'][$this->class]) && $_SESSION['table_filter'][$this->class])
{
    foreach ($filter as $item => $value)
    {
        $result->bindValue(":".$item,$value);
    }
}
// Возвращаем значение count - количество
$result->execute();
$row = $result->fetch();
return $row['count'];
}

//Получение массива записей таблицы с учетом номера страницы и примененных
фильтров
public function getListItemByTable($table_name,$arr_name,$table_key,$page)
{
    $limit = $this->class::SHOW_BY_DEFAULT;
    // Смещение (для запроса)
    $offset = ($page - 1) * $this->class::SHOW_BY_DEFAULT;

```

```

// Соединение с БД
$db = DataBase::getInstance()->getDb();
// Текст запроса к БД
$sql = "SELECT * FROM $table_name";
if (isset($_SESSION['table_filter'][$this->class]) && $_SESSION['table_filter'][$this->class])
{
    $filter = $_SESSION['table_filter'][$this->class];
    $i = 0;
    $sql .= ' WHERE';
    foreach ($filter as $item => $value)
    {
        if ($i != count($filter)-1)
        {
            $sql .= ' '.$item.' = :'.$item.' AND ';
            $i++;
        }
        else $sql .= ' '.$item.' = :'.$item.' ';
    }
    $sql .= " ORDER BY $table_key ASC LIMIT :limit OFFSET :offset";
    // Используется подготовленный запрос
    $result = $db->prepare($sql);
    if (isset($_SESSION['table_filter'][$this->class]) && $_SESSION['table_filter'][$this->class])
    {
        foreach ($filter as $item => $value)
        {
            $result->bindValue(":".$item,$value);
        }
    }

    $result->bindParam(':limit', $limit, PDO::PARAM_INT);
    $result->bindParam(':offset', $offset, PDO::PARAM_INT);
    // Выполнение запроса
    $result->execute();

```

```

    $i = 0;
    ${$arr_name} = array();
    $row = $result->fetchall();
    for ($i=0; $i < count($row) ; $i++)
    {
        foreach ($row[$i] as $key => $value)
        {
            ${$arr_name}[$i][$key] = $value;
        }
    }
    return ${$arr_name};
}

//Удаление записи таблицы
public function deleteTableItemById ($table_name,$table_key,$id)
{
    // Соединение с БД
    $db = DataBase::getInstance()->getDb();
    // Текст запроса к БД
    $sql = "DELETE FROM $table_name WHERE $table_key = :id";
    // Получение и возврат результатов. Используется подготовленный запрос
    $result = $db->prepare($sql);
    $result->bindValue(':id', $id);
    $result->execute();
}

//редактирование записи таблицы
public function editTableItemById($table_name,$table_key,$id)
{
    // Соединение с БД
    $db = DataBase::getInstance()->getDb();
    // Текст запроса к БД
    $sql = "UPDATE $table_name SET ";
    $i = 0;
    foreach ($this->options as $option)
    {
        if ($option != $table_key)

```

```

    {
        if ($i < count($this->options)-1)
        {
            $sql .= '.$option.' = '.$option.';
            $i++;
        }
        else
        {
            $sql .= '.$option.' = '.$option.';
        }
    }
    else $i++;
}
$sql .= "WHERE $table_key = :id";
// Получение и возврат результатов. Используется подготовленный запрос
$result = $db->prepare($sql);
$result->bindValue(':id', $id);
foreach ($this->options as $option)
{
    if ($option != $table_key)
    {
        $result->bindValue('.$option, $_POST[$option]);
    }
}
return $result->execute();
}
//Добавление записи в таблицу
public function addTableItem($table_name, $table_key)
{
    // Соединение с БД
    $db = DataBase::getInstance()->getDb();
    // Текст запроса к БД
    $sql = "INSERT INTO $table_name SET";
    $i = 0;
    foreach ($this->options as $option)

```

```

{
    if ($option != $table_key)
    {
        if ($i < count($this->options)-1)
        {
            $sql .= ' '.$option.' = ':' '.$option.' ';
            $i++;
        }
        else
        {
            $sql .= ' '.$option.' = ':' '.$option.' ';
        }
    }
    else $i++;
}
$result = $db->prepare($sql);
foreach ($this->options as $option)
{
    if ($option != $table_key)
    {
        $result->bindValue(':', $option, $_POST[$option]);
    }
}
return $result->execute();
}}

```