**Реферат**

Дипломный проект содержит !!!!! страницы пояснительной записки, !!! таблицы, !!!! рисунков, !!!! источников литературы, !!!!! приложения.

АВТОМАТИЗАЦИЯ,Лабораторные работы, Учёт, ASP.NET, Деканат, ВЕБ-СЕРВЕР, БАЗА ДАННЫХ

Целью данного дипломного проекта является автоматизация процесса учёта выполнения лабораторных работ и анализа успеваемости.

В первой главе проводится обзор существующих технологий, удобных для написания дипломного проекта а также выбор из них тех, которые будут применяться впоследствии.

Во второй главе указаны данные о проектировании приложения, схемы базы данных и диаграммы работы

В третьей главе описываетсяструктура приложения, средства разработки, реализации основного функционала.

В четвёртой главе приведены скриншоты приложения с обзором его работы и руководством пользователя.

В пятой главе приводится расчет экономических параметров и себестоимость данного программного средства

В шестой главе данного дипломного проекта содержатся требования по охране труда и безопасности жизнедеятельности.

В заключении — анализ проделанной работы.

Изм.

№ докум.

Подпись

Дата

Лист

1

ДП 00.00.ПЗ

Разраб.

Прокопович Д.В.

Провер.

Смелов В.В.

Консульт.

Смелов В.В.

.

Н. Контр.

Жиляк Н.А.

Утверд.

Смелов В.В.

Реферат

Лит.

Листов

1

*БГТУ 64419006,2017*

У

Содержание

Изм.

№ докум.

Подпись

Дата

Лист

1

ДП 00.00.ПЗ

Разраб.

Прокопович Д.В.

Провер.

Смелов В.В.

Консульт.

Смелов В.В.

.

Н. Контр.

Жиляк Н.А.

Утверд.

Смелов В.В.

# Содержание

Лит.

Листов

1

*БГТУ 64419006,2017*

У

[Содержание 2](file:///D:\Дмитрий\Программы\диплом\репозиторий\документация\Пояснительная%20записка\текст.docx#_Toc482481858)

[Введение 4](#_Toc482481859)

[1 Анализ и выбор технологий и инструментария для разработки автоматизированной системы «Электронный деканат» 5](#_Toc482481860)

[1.1 Стек технологий LAMP 5](#_Toc482481861)

[1.2 Стек технологий Java 6](#_Toc482481862)

[1.3.1 JEE 6](#_Toc482481863)

[1.3.2 SpringFramework 7](#_Toc482481864)

[1.3 Стек технологий MEAN 11](#_Toc482481865)

[1.4 Стек технологий.NET 12](#_Toc482481866)

[1.5 Патентный поиск и аналоги программного обеспечения 15](#_Toc482481867)

[1.5.3 Автоматизация процесса составления деканатом рейтинга студентов 16](#_Toc482481868)

[1.5.4 Модуль «Деканат» интегрированной автоматизированной информационной системы «ВУЗ» 17](#_Toc482481869)

[1.5.5 Система управления учебным процессом «Магеллан» 17](#_Toc482481870)

[1.5.6 Система автоматизации учебного процесса «GS-Ведомости» 18](#_Toc482481871)

[1.5.7 Информационно-аналитическая система управления деканатомвысшего учебного заведения 18](#_Toc482481872)

[1.5.8 Программный комплекс «Электронный документооборот деканата» 19](#_Toc482481873)

[1.5.9 Информационная система «Деканат» 19](#_Toc482481874)

[1.5.10 Модуль "ДЕКАНАТ JTSQL" 19](#_Toc482481875)

[1.5.11 Программный комплекс «Деканат» КГПУ им. В.П. Астафьева 20](#_Toc482481876)

[1.6 Технологии и инструментарий для разработки автоматизированной системы «Электронный деканат» 21](#_Toc482481877)

[1.6.1 OracleClient 21](#_Toc482481878)

[1.6.2 OWIN 21](#_Toc482481879)

[1.6.3 EntityFramework 22](#_Toc482481880)

[1.6.4 Паттерн «репозиторий» 23](#_Toc482481881)

[1.6.5 Bootstrap 24](#_Toc482481882)

[1.6.6 jQuerry 24](#_Toc482481883)

[1.6.7 jQueryDataTables 25](#_Toc482481884)

[1.7 Заключение 25](#_Toc482481885)

[2 Проектирование автоматизированной системы «Электронный деканат» 27](#_Toc482481886)

[2.1 Архитектура 27](#_Toc482481887)

[2.2 Диаграммы вариантов использования и последовательностей 28](#_Toc482481888)

[2.3 Структура базы данных 31](#_Toc482481889)

[2.4 Заключение 36](#_Toc482481890)

[3 Реализация программного обеспечения автоматизированной системы «Электронный деканат» 37](#_Toc482481891)

[3.1 Принципы построения по автоматизированной системы 37](#_Toc482481892)

[Реализация программного обеспечения автоматизированной системы «Электронный деканат» 37](file:///D:\Дмитрий\Программы\диплом\репозиторий\документация\Пояснительная%20записка\текст.docx#_Toc482481893)

[3.2 MVC – Контроллеры 38](#_Toc482481894)

[3.3 MVC – Модели 39](#_Toc482481895)

[3.4 MVC – Представления 40](#_Toc482481896)

[3.4 Маршрутизация 45](#_Toc482481897)

[3.5 Заключение 46](#_Toc482481898)

[4 Описание контрольного примера 47](#_Toc482481899)

[4.1 Администратор 47](#_Toc482481900)

[4.2 Преподаватель 56](#_Toc482481901)

[4.3 Студент 59](#_Toc482481902)

[4.4 Деканат 61](#_Toc482481903)

[5 Экономический раздел 64](#_Toc482481904)

[Экономический раздел 64](file:///D:\Дмитрий\Программы\диплом\репозиторий\документация\Пояснительная%20записка\текст.docx#_Toc482481905)

[6 Мероприятия по охране труда и безопасности жизнедеятельности 65](#_Toc482481906)

[6.1 Анализ потенциально опасных и вредных производственных факторов 65](#_Toc482481907)

[6.2 Инженерные решения по обеспечению санитарно–гигиенических условий труда 66](#_Toc482481908)

[6.3 Инструкция по безопасной эксплуатации ЭВМ 69](#_Toc482481909)

[6.3.1 Требования по охране труда перед началом работы на ЭВМ 70](#_Toc482481910)

[6.3.2 Требования по охране труда во время работы на ЭВМ 70](#_Toc482481911)

[6.3.3 Требования по охране труда по окончанию работы на ЭВМ 72](#_Toc482481912)

[6.4 Требования по охране труда в аварийных ситуациях 73](#_Toc482481913)

[6.5 Организация и безопасность работы на ПЭВМ 73](#_Toc482481914)

[6.6 Технические решения, обеспечивающие взрыво– и пожаробезопасность объекта 74](#_Toc482481915)

[6.7 Мероприятия по безопасности жизнедеятельности 74](#_Toc482481916)

[6.7.1 Анализ потенциально опасных источников возникновения ЧС 75](#_Toc482481917)

[6.7.2 Мероприятия, направленные на предотвращение потерь персонала от возникновения ЧС 76](#_Toc482481918)

[Заключение 77](#_Toc482481919)

[Заключение 77](file:///D:\Дмитрий\Программы\диплом\репозиторий\документация\Пояснительная%20записка\текст.docx#_Toc482481920)

[Список использованных источников 78](#_Toc482481921)

# Введение

Изм.

№ докум.

Подпись

Дата

Лист

1

ДП 00.00.ПЗ

Разраб.

Прокопович Д.В.

Провер.

Смелов В.В.

Консульт.

Смелов В.В.

.

Н. Контр.

Жиляк Н.А.

Утверд.

Смелов В.В.

Введение

Лит.

Листов

1

*БГТУ 64419006,2017*

У

В деканате информационных технологий Белорусского государственного технологического университета осуществляется учёт выполнения лабораторных работ студентов, анализ текущей успеваемости студентов. Учёт выполнения лабораторных работ производится путём получения от преподавателей отчетов по сданным студентами лабораторным работам в бумажном виде. Для создания сводной ведомости по успеваемости студентов группы, ответственный за успеваемость в конце каждого месяца (семестра) берет ведомость у заведующего отделения. Затем он, используя список студентов, список предметов и оценки по каждому предмету заполняет её.

**Актуальность темы:** часть полезного времени преподавателей тратится на составление отчёта по текущей успеваемости студентов, и, если заменить это взаимодействие на единую электронную систему, то производительность преподавателей улучшится, деканат будет иметь инструментарий для анализа успеваемости, для своевременного обнаружения проблем в какой-либо дисциплине а, студенты будут иметь возможность просмотреть, какие лабораторные работы по каким дисциплинам запланированы, и вовремя их выполнить.

**Цель дипломного проекта:** автоматизация процесса учёта выполнения лабораторных работ и анализа успеваемости.

Для достижения цели сформулированы следующие **задачи**.

1. Исследование процесса документооборота в деканате.
2. Проектирование архитектуры автоматизированной системы.
3. Проектирование программного обеспечения автоматизированной системы.
4. Проектирование логической схемы базы данных.
5. Разработка программного обеспечения.
6. Тестирование автоматизированной системы.

**Предмет исследования**: автоматизированная система учёта выполнения лабораторных работ и анализа успеваемости.

**Практическая значимость:** внедрение автоматизированной системы позволит уменьшить трудозатраты по учёту успеваемости студентов.

# Анализ и выбор технологий и инструментария для разработки автоматизированной системы «Электронный деканат»

Изм.

№ докум.

Подпись

Дата

Лист

1

ДП 00.00.ПЗ

Разраб.

Прокопович Д.В.

Провер.

Смелов В.В.

Консульт.

Смелов В.В.

.

Н. Контр.

Жиляк Н.А.

Утверд.

Смелов В.В.

Анализ и выбор технологий и инструментария для разработки автоматизированной системы «Электронный деканат»

Лит.

Листов

1

*БГТУ 64419006,2017*

У

## Стек технологий LAMP

Рассмотрим свободно распространяемый и практически бесплатный набор программного обеспечения под названием LAMP. Как правильно расшифровывается данная аббревиатура? Она состоит из 4-ех популярных технологий, которые состоят в данной связке: Linux — операционная система; Apache — веб-сервер; MySQL — СУБД; PHP — язык программирования, с помощью которого создают веб-ресурсы.

Данный набор ПО устанавливают на сервер для отображения динамических веб-сайтов и веб-приложений. Эта платформа позволяет установить (развернуть) большинство типов веб-сайтов и сетевого программного обеспечения на вашем сервере.

Комбинация LAMP – это:

* Серверная ОС Linux для выполнения нужных задач;
* веб-сервер Apache, для которого создано множество дополнительных модулей, которые решают вопрос совместной работы веб-сервера и сценариев, написанных на самых разных языках программирования;
* серверный язык динамических сценариев PHP (либо Perl);
* СУБД MySQL, которая демонстрирует отличную скорость выполнения SQL-запросов и идеально подходит для малых и средних проектов. MySQL работает на Unix и Windows, отличается особенным удобством в работе.

Для чего может понадобиться LAMP?

1. в случае необходимости окружения для веб-разработки и тестирования приложений, которые написаны для LAMP. Это как и для собственных приложений, так и для любых необходимых Вам CMS. Вы можете работать на Windows и на Ubuntu;
2. если Вам необходимо сделать на виртуальном VPS сервере или на выделенном сервере производительный веб-сервер для хостинга своих проектов;
3. если Вам нужен сервер для систем контроля версий;
4. для самообучения администрирования;
5. в экономических целях, для создания собственного сервера.

Как весомое преимущество данной сборки: она замечательно подходит для быстрого развертывания приложения, из-за простой конфигурации, но все же она дает мало функций в плане масштабируемости и изоляции компонентов.

Недостаток подобной сборки: приложение и база данных используют одни и те же ресурсы сервера (CPU, память, I/O и т.д.), что дает низкую производительность и затрудняет определение источника (приложение или база данных) этой проблемы. Так же есть помехи в осуществлении горизонтального масштабирования[7].

## Стек технологий Java

### JEE

Сервер приложений (англ. applicationserver) — это программная платформа (фреймворк), предназначенная для эффективного исполнения процедур (программ, скриптов), на которых построены приложения. Сервер приложений действует как набор компонентов, доступных разработчику программного обеспечения через API (Интерфейс прикладного программирования), определённый самой платформой [8].

Для веб-приложений основная задача компонентов сервера — обеспечивать создание динамических страниц. Однако современные серверы приложений включают в себя и поддержку кластеризации, повышенную отказоустойчивость, балансировку нагрузки, позволяя, таким образом, разработчикам сфокусироваться только на реализации бизнес-логики.

В случае сервера приложений Java, сервер приложений ведёт себя как расширенная виртуальная машина для запуска приложений, прозрачно управляя соединениями с базой данных с одной стороны и соединениями с веб-клиентом с другой.

Преимущества серверов приложений

1. Целостность данных и кода

Выделяя бизнес логику на отдельный сервер, или на небольшое количество серверов, можно гарантировать обновления и улучшения приложений для всех пользователей. Отсутствует риск, что старая версия приложения получит доступ к данным или сможет их изменить старым несовместимым образом.

1. Централизованная настройка и управление

Изменения в настройках приложения, таких как изменение сервера базы данных или системных настроек, могут производиться централизованно.

1. Безопасность

Сервер приложений действует как центральная точка, используя которую, поставщики сервисов могут управлять доступом к данным и частям самих приложений, что считается преимуществом защиты. Её наличие позволяет переместить ответственность за аутентификацию с потенциально небезопасного уровня клиента на уровень сервера приложений, при этом дополнительно скрывая уровень базы данных.

1. Поддержка транзакций

Транзакция представляет собой единицу активности, во время которой большое число изменений ресурсов (в одном или различных источниках) может быть выполнено атомарно (как неделимая единица работы). Конечные пользователи при этом могут выиграть от стандартизованного поведения системы, от уменьшения времени на разработку и от снижения стоимости. В то время как сервер приложений выполняет массу нужного генерирования кода, разработчики могут сфокусироваться на бизнес-логике.

Примеры реализации

1. Под сервером приложений в случае Java EE подразумевается комплекс программ, реализующих концепцию Java EE и позволяющих запускать в себе Java EE приложения. К классу серверов приложений относятся такие продукты как SunGlassFish, IBM WebSphere, RedHatJBossApplicationServer, AppleWebObjects (англ.), OracleWeblogicServer и др.
2. Zope, развитый сервер web-приложений.
3. Терминальные серверы, например, поставляемые компанией Citrix.

### SpringFramework

SpringFramework (или коротко Spring) — универсальный фреймворк с открытым исходным кодом для Java-платформы. Также существует форк для платформы .NET Framework, названный Spring.NET [10].

Первая версия была написана Родом Джонсоном, который впервые опубликовал её вместе с изданием своей книги «ExpertOne-on-OneJava EE DesignandDevelopment

Фреймворк был впервые выпущен под лицензией Apache 2.0 license в июне 2003 года. Первый стабильный релиз 1.0 был выпущен в марте 2004. Spring 2.0 был выпущен в октябре 2006, Spring 2.5 — в ноябре 2007, Spring 3.0 в декабре 2009, и Spring 3.1 в декабре 2011. Текущая версия — 4.3.7.

Несмотря на то, что Spring не обеспечивал какую-либо конкретную модель программирования, он стал широко распространённым в Java-сообществе главным образом как альтернатива и замена модели EnterpriseJavaBeans. Spring предоставляет бо́льшую свободу Java-разработчикам в проектировании; кроме того, он предоставляет хорошо документированные и лёгкие в использовании средства решения проблем, возникающих при создании приложений корпоративного масштаба.

Между тем, особенности ядра Spring применимы в любом Java-приложении, и существует множество расширений и усовершенствований для построения веб-приложений на JavaEnterprise платформе. По этим причинам Spring приобрёл большую популярность и признаётся разработчиками как стратегически важный фреймворк.

Spring обеспечивает решения многих задач, с которыми сталкиваются Java-разработчики и организации, которые хотят создать информационную систему, основанную на платформе Java. Из-за широкой функциональности трудно определить наиболее значимые структурные элементы, из которых он состоит. Spring не всецело связан с платформой JavaEnterprise, несмотря на его масштабную интеграцию с ней, что является важной причиной его популярности.

Spring, вероятно, наиболее известен как источник расширений (features), нужных для эффективной разработки сложных бизнес-приложений вне тяжеловесных программных моделей, которые исторически были доминирующими в промышленности. Ещё одно его достоинство в том, что он ввел ранее неиспользуемые функциональные возможности в сегодняшние господствующие методы разработки, даже вне платформы Java.

Этот фреймворк предлагает последовательную модель и делает её применимой к большинству типов приложений, которые уже созданы на основе платформы Java. Считается, что Spring реализует модель разработки, основанную на лучших стандартах индустрии, и делает её доступной во многих областях Java.

**Модули**

Spring может быть рассмотрен как коллекция меньших фреймворков или фреймворков во фреймворке. Большинство этих фреймворков может работать независимо друг от друга, однако они обеспечивают большую функциональность при совместном их использовании. Эти фреймворки делятся на структурные элементы типовых комплексных приложений:

* + InversionofControl-контейнер: конфигурирование компонентов приложений и управление жизненным циклом Java-объектов.
  + Фреймворк аспектно-ориентированного программирования: работает с функциональностью, которая не может быть реализована возможностями объектно-ориентированного программирования на Java без потерь.
  + Фреймворк доступа к данным: работает с системами управления реляционными базами данных на Java-платформе, используя JDBC- и ORM-средства и обеспечивая решения задач, которые повторяются в большом числе Java-basedenvironments.
  + Фреймворк управления транзакциями: координация различных API управления транзакциями и инструментарий настраиваемого управления транзакциями для объектов Java.
  + Фреймворк MVC: каркас, основанный на HTTP и сервлетах, предоставляющий множество возможностей для расширения и настройки (customization).
  + Фреймворк удалённого доступа: конфигурируемая передача Java-объектов через сеть в стиле RPC, поддерживающая RMI, CORBA, HTTP-based протоколы, включая web-сервисы (SOAP).
  + Фреймворк аутентификации и авторизации: конфигурируемый инструментарий процессов аутентификации и авторизации, поддерживающий много популярных и ставших индустриальными стандартами протоколов, инструментов, практик через дочерний проект SpringSecurity (ранее известный как Acegi).
  + Фреймворк удалённого управления: конфигурируемое представление и управление Java-объектами для локальной или удалённой конфигурации с помощью JMX.
  + Фреймворк работы с сообщениями: конфигурируемая регистрация объектов-слушателей сообщений для прозрачной обработки сообщений из очереди сообщений с помощью JMS, улучшенная отправка сообщений по стандарту JMS API.
  + Тестирование: каркас, поддерживающий классы для написания модульных и интеграционных тестов.

**Inversion of Control**

Центральной частью Spring является контейнер Inversion of Control, который предоставляет средства конфигурирования и управления объектами Java с помощью рефлексии. Контейнер отвечает за управление жизненным циклом объекта: создание объектов, вызов методов инициализации и конфигурирование объектов путём связывания их между собой.

Объекты, создаваемые контейнером, также называются управляемыми объектами (beans). Обычно конфигурирование контейнера осуществляется путём загрузки XML-файлов, содержащих определение bean’ов и предоставляющих информацию, необходимую для создания bean’ов.

Объекты могут быть получены одним из двух способов:

* Поиск зависимости — шаблон проектирования, в котором вызывающий объект запрашивает у объекта-контейнера экземпляр объекта с определённым именем или определённого типа.
* Внедрение зависимости — шаблон проектирования, в котором контейнер передает экземпляры объектов по их имени другим объектам с помощью конструктора, свойства или фабричного метода.

**MVC**

Spring имеет собственную MVC-платформу веб-приложений, которая не была первоначально запланирована. Разработчики Spring решили написать её как реакцию на то, что они восприняли как неудачность конструкции (тогда) популярного Apache Struts, а также других доступных веб-фреймворков. В частности, по их мнению, было недостаточным разделение между слоями представления и обработки запросов, а также между слоем обработки запросов и моделью.[4]

Класс Dispatcher Servlet является основным контроллером фрэймворка и отвечает за делегирование управления различным интерфейсам, на всех этапах выполнения HTTP-запроса. Об этих интерфейсах следует сказать более подробно.

Как и Struts, Spring MVC является фреймворком, ориентированным на запросы. В нем определены стратегические интерфейсы для всех функций современной запросно-ориентированной системы. Цель каждого интерфейса — быть простым и ясным, чтобы пользователям было легко его заново имплементировать, если они того пожелают. MVC прокладывает путь к более чистому front-end-коду. Все интерфейсы тесно связаны с Servlet API. Эта связь рассматривается некоторыми как неспособность разработчиков Spring предложить для веб-приложений абстракцию более высокого уровня. Однако эта связь оставляет особенности Servlet API доступными для разработчиков, облегчая все же работу с ним. Наиболее важные интерфейсы, определенные Spring MVC, перечислены ниже:

* HandlerMapping: выбор класса и его метода, которые должны обработать данный входящий запрос на основе любого внутреннего или внешнего для этого запроса атрибута или состояния.
* HandlerAdapter: вызов и выполнение выбранного метода обработки входящего запроса.
* Controller: включен между Моделью (Model) и Представлением (View). Управляет процессом преобразования входящих запросов в адекватные ответы. Действует как ворота, направляющие всю поступающую информацию. Переключает поток информации из модели в представление и обратно.
* View: ответственно за возвращение ответа клиенту в виде текстов и изображений. Некоторые запросы могут идти прямо во View, не заходя в Model; другие проходят через все три слоя.
* ViewResolver: выбор, какое именно View должно быть показано клиенту.
* HandlerInterceptor: перехват входящих запросов. Сопоставим, но не эквивалентен сервлет-фильтрам (использование не является обязательным и не контролируется DispatcherServlet-ом).
* LocaleResolver: получение и, возможно, сохранение локальных настроек (язык, страна, часовой пояс) пользователя.
* MultipartResolver: обеспечивает Upload — загрузку на сервер локальных файлов клиента.

Spring MVC предоставляет разработчику следующие возможности:

* Ясное и прозрачное разделение между слоями в MVC и запросах.
* Стратегия интерфейсов — каждый интерфейс делает только свою часть работы.
* Интерфейс всегда может быть заменен альтернативной реализацией.
* Интерфейсы тесно связаны с Servlet API.
* Высокий уровень абстракции для веб-приложений.
* В веб-приложениях можно использовать различные части Spring, а не только Spring MVC.

**Доступ к данным**

Spring предоставляет свой слой доступа к базам данных и поддерживает все популярные СУБД.

JDBC, iBatis / MyBatis, Hibernate, JDO, JPA, Oracle TopLink, Apache OJB, Apache Cayenne ит. д.

Для всех этих фреймворковSpring предоставляет такие особенности:

* Управление ресурсами — автоматическое получение и освобождение ресурсов базы данных
* Обработка исключений — перевод исключений при доступе к данным в исключенияSpring-а
* Транзакционность — прозрачные транзакции в операциях с данными
* Распаковка ресурсов — получение объектов базы данных из пула соединений
* Абстракция для обработки BLOB и CLOB

**Управление транзакциями**

Фреймворк управления транзакциями в Spring привносит механизм абстракций для платформы Java. Основные возможности этих абстракций:

* работа с локальными и глобальными транзакциями
* работа с вложенными транзакциями
* работа с точками сохранения в транзакциях

**«Соглашение прежде конфигурации» в разработке приложения**

SpringRoo — фреймворк, поддерживающий принцип «соглашение прежде конфигурации» для быстрого создания бизнес-приложений.

Может использоваться со:

* Spring Framework
* Spring Security
* Spring Web Flow

**Интеграция**

SpringIntegration — фреймворк для JEE, который предоставляет функции, необходимые для отправки сообщений или для построения событийно-ориентированной архитектуры.

* роутеры
* трансформеры
* адаптеры для интеграции с другими технологиями и системами (HTTP, AMQP, JMS, XMPP, SMTP, IMAP, FTP (FTPS/SFTP), файловые системы и т. д.)
* фильтры
* активаторы сервисов
* аудит и управление

## Стек технологий MEAN

В последнее время в веб-сообществе много говорят о новом стеке MEAN: MongoDB, Express, AngularJS, Node.js. Стек технологий MEAN отражает современный подход к веб-разработке: когда на каждом уровне приложения, от клиента до сервера и персистентности, применяется один и тот же язык (JavaScript).

MEAN — это больше, чем простая перестановка начальных букв и новых технологий. Смещение базовой платформы с ОС (Linux) к среде исполнения JavaScript (Node.js) несет с собой независимость от ОС: Node.js работает на Windows® и OS X так же, как и на Linux.

Node.js заменяет Apache из стека LAMP. Но Node.js — это гораздо больше, чем просто веб-сервер. На самом деле готовое приложение не развертывается на отдельном веб-сервере; вместо этого сам веб-сервер включается в приложение и автоматически устанавливается в составе стека MEAN. В результате процесс развертывания значительно упрощается, так как требуемая версия веб-сервера явно определена вместе с остальными зависимостями времени выполнения.

Переход от традиционной базы данных, такой как MySQL, к бессхемному, документо-ориентированному NoSQL-хранилищу, такому как MongoDB, представляет собой фундаментальный сдвиг в стратегии персистенции. Программист тратит меньше времени на написание операторов SQL и больше — на написание функций map/reduce на JavaScript. При этом исключаются огромные пласты логики преобразования, так как MongoDB изначально выдает формат JavaScriptObjectNotation (JSON). В результате предельно упрощается написание веб-сервисов REST.

Но главный сдвиг между LAMP и MEAN заключается в переходе от традиционного генерирования страниц на стороне сервера к ориентации на одностраничные приложения (SPA) на стороне клиента. Express позволяет управлять и маршрутизацией / генерацией страниц на стороне сервера, но теперь — благодаря AngularJS — упор делается на представления на стороне клиента. Эта перемена означает не просто перенос ваших артефактов модель-представление-контроллер (MVC) с сервера в клиентское устройство. Это также скачок от менталитета синхронности к менталитету, носящему событийно-управляемый, принципиально асинхронный характер. И, пожалуй, самое главное, — это движение от странично-ориентированных приложений к компонентно-ориентированным.

Стек MEAN не «заточен» на мобильные приложения — AngularJS одинаково хорошо работает на настольных компьютерах и ноутбуках, смартфонах и планшетах и даже на смарт-телевизорах, — но он и не относится к мобильным устройствам как к гражданам второго сорта. И тестирование больше не откладывается на потом: с помощью платформ тестирования мирового класса, таких как MochaJS, JasmineJS и KarmaJS, можно писать тщательные и всеобъемлющие наборы тестов для своих MEAN-приложений [9].

## Стек технологий.NET

.NET Framework — это программная платформа, выпущенная компанией Microsoft, которая подходит для разных языков программирования. ASP.NET — технология создания веб-приложений и веб-сервисов от компании Microsoft. Она является составной частью платформы Microsoft. NET и развитием более старой технологии Microsoft ASP.

Считается, что платформа. NET Framework явилась ответом компании Microsoft на набравшую к тому времени большую популярность платформу Java. ASP.NET основывается на CommonLanguageRuntime: разработчики могут писать код для ASP.NET, используя практически любые языки программирования, некоторые из которых входят в комплект. NET Framework (C#, Visual Basic.NET и JScript. NET), а другие могут быть установлены дополнительно (IronRuby, IronPython, PHP, Perl, Smalltalk, Haskell и др.). Тем не менее, нередко используется не вполне корректный термин «.NET-программист», который, видимо, характеризует разработчика на любом языке программирования, который может быть использован в. NET-проекте.

Некоторые особенности ASP.NET:

* Компилируемый код выполняется быстрее, а большинство ошибок отлавливается ещё на стадии разработки.
* Расширяемый набор элементов управления и библиотек классов, ускоряющий разработку.
* Возможность кэширования всей страницы, её частей или данных, используемых на странице.
* Возможность разделения визуальной части и бизнес-логики по разным файлам, есть возможность выделять часто используемые шаблоны пользовательских элементов управления, таких как меню сайта, наличие master-страниц для задания шаблонов оформления, поддержка AJAX (расширение ASP.NET AJAX).
* Расширяемые модели событий, обработки запросов и серверных элементов управления.
* Поддержка CRUD-операций при работе с таблицами через GridView.
* Возможно создание веб-приложений, которые реализуют шаблон Model-View-Controller (ASP.NET MVC Framework).

.NET достаточно широко распространён в сфере разработки внутрикорпоративных программных продуктов, но в веб-разработке всё же встречается относительно редко, как и другие программные продукты корпорации Microsoft. Поэтому найти разработчиков для веб-проекта бывает достаточно непросто. Использование. NET «тянет» за собой покупку и иного ПО от корпорации Microsoft (серверной ОС, СУБД и т.п.). Технология достаточно дорогая в разработке и сопровождении: кроме затрат на покупку лицензий на необходимое ПО существенный вклад в бюджет проектов вносят высокие зарплаты разработчиков. [11].

Преимущества ASP.NET:

* Надежность и стабильность вашего сайта

Огромное значение для современного бизнеса имеет отсутствие простоев, не важно, будет это на час или несколько часов в день. Простой приводит к серьезным убыткам, удару по репутации в деловом мире. Именно поэтому огромную роль в работе сайта играет его надежность и устойчивость к хакерским атакам.

Встроенная защита от различных видов нападений, предоставляет следующие возможности: SQL Injection, переполнение буфера, XSS, изменение скрытых полей и прочие. Технология ASP.NET повышает степень устойчивости к вредоносным действиям и различным видам хакерских атак сайтов, построенных на ней.

* Производительность и скорость работы сайта

Строение ASP.NET как технологии позволяет компилировать программный код и все страницы сайта. Код интерпретируется в PHP значительно медленнее и не дает нужного эффекта производительности.

Особенно это касается активного использования в разработке сайта концепции ООП. Благодаря тому, что в ASP.NET встроено функционирование сайта на кластере сервера, при увеличении посещаемости на сайте достигается масштабируемость.

* Интеграция с приложениями и прочими информационными системами

Платформа Microsoft .NET имеет множество встроенных технологий для интеграции информационных систем и приложений, таких как службы web, WCF, JSON, remoting, XML и пр., при этом ASP.NET существует как ее часть.

Наличие таких многочисленных решений дает возможность выбора оптимальной технологии для каждого отдельного случая. Это обеспечит отменную производительность, масштабируемость и, самое главное, - безопасность.

* Скорость разработки на высшем уровне

Для разработки на платформе ASP.NET предлагается среда MS VisualStudio от 2010 года, которая признана одним из лучших средств. Данная среда упростит и ускорит создание web-приложений за счет применения усовершенствованного конструктора web-форм, поддержки ASP.NET AJAX, неограниченного набора коммерческих и стандартных элементов управления.

В целях эффективной совместной работы группы и организации взаимодействия, возможна интеграция VisualStudio с TFS и SS - средствами коллективной разработки. При этом обеспечивается наглядность управления функционированием приложений и процесса разработки, а также повышается качество создаваемого программного обеспечения.

Надежная и безопасная работа приложений обеспечена и повышается также с помощью применения встроенного модульного тестирования. Для среды MicrosoftVisualStudio придается встроенная поддержка продвинутых технологий AJAX, WCF, Silverlight, WPF, а также WWF [20].

Среди основных свойств СУБД Oracle следует отметить такие, как [21]:

1. Высочайшая надежность.
2. Возможность разбиения крупных баз данных на разделы (large-databasepartition), что дает возможность эффективно управлять гигантскими гигабайтными базами;
3. Наличие универсальных средств защиты информации;
4. Эффективные методы максимального повышения скорости обработки запросов;
5. Индексация по битовому отображению;
6. Свободные таблицы (в других СУБД все таблицы заполняются сразу при создании);
7. Распараллеливание операций в запросе;
8. Наличие широкого спектра средств разработки, мониторинга и администрирования;
9. Поддержка известных платформ: Windows, AIX, CompaqTru64 UNIX, HP 9000 SeriesHP-UX, LinuxIntel, SunSolaris
10. Ориентация на интернет технологии.
11. Поддержка XML в хранимых процедурах, позволяющая разработчикам (традиционных) баз данных непосредственно использовать преимущества языка XML, применяя привычный механизм хранимых процедур
12. Доступ по протоколу HTTP, поддерживающий отправку SQL-запросов к БД с применением URL-адресов
13. Интегрированное средство выявления закономерностей, применяемое для отбора важной, но не обязательно очевидной бизнес-информации из больших наборов данных. Оно обеспечиваюет сбор, хранение, управление и анализ потока данных о действиях пользователей при посещении ими Web-узла
14. Связанные базы данных OLAP - это, в первую очередь, кубы OLAP, применяемые для реализации новых возможностей анализа данных; они позволяют повысить ценность данных за счет предоставления возможностей анализа OLAP через Web
15. Доступ к кубам данных по протоколу HTTP, обеспечивающий просмотр и управление многомерными кубами данных в Web и через брандмауэры при помощи встроенного в БД средства поддержки протокола HTTP
16. Средство передачи журналов, позволяющее синхронизировать отдельные системы посредством автоматизированного обновления журнала транзакций, что повышает надежность системы и устойчивость к авариям
17. Параллельные операции при создании индексов и выполнении поиска на компьютерах с симметричной многопроцессорной обработкой, ускоряющие работу приложений
18. Улучшенные средства отказоустойчивости, упрощающие создание и управление отказоустойчивыми кластерами, переключение между узлами кластера при сбое и восстановление узла кластера, не оказывающее влияния на остальные узлы
19. Поддержка большого объема памяти и симметричной многопроцессорной обработки,позволяющая управлять приложениями с высокими нагрузками, связанными с большим объемом транзакций
20. Поддержка протокола VI SAN (VirtualInterfaceSystemAreaNetwork), позволяющая эффективно использовать сети SAN (SystemAreaNetwork - системная сеть) со встроенным интерфейсом VI Consumer, который позволяет устанавливать прямую связь (в режиме пользователя) с дисковыми устройствами SAN
21. Поддержка службы единого каталога, позволяющая в крупных организациях упростить и централизовать управление именованием серверов, данными OLAP, репликациями и хронологией архивирования
22. Инструментальные средства, включающие графические средства разработки схем, генерации запросов и кода.

## Патентный поиск и аналоги программного обеспечения

Патентные исследования – это исследования технического уровня и тенденций развития объектов техники, их патентоспособности, патентной чистоты, конкурентоспособности на основе патентной и иной информации.

По своему характеру и содержанию патентные исследования относятся к прикладным научно-исследовательским работам. Патентные исследования проводятся как в виде самостоятельной научно-исследовательской работы, так и в составе работ субъекта хозяйствования.

Патентный поиск - это процесс отбора соответствующих запросу документов или сведений по одному или нескольким признакам из массива патентных документов или данных, при этом осуществляется процесс поиска из множества документов и текстов только тех, которые соответствуют теме или предмету запроса.

Предмет поиска определяют исходя из конкретных задач патентных исследований, особенностей объекта (устройство, способ, вещество), а так же из того, какие его элементы, параметры, свойства и другие характеристики предполагается исследовать.

Основные виды патентного поиска: предметный, именной (или фирменный), нумерационный, патентов-аналогов. Выбор типа патентного поиска определяется как необходимой глубиной поиска и временными ограничениями, так и поисковыми возможностями лица или организации, проводящих поиск.

Предметный поиск – является основным и чаще всего применяемым. При этом виде поиска формулируется техническая задача (предмет поиска), выбором рубрики (рубрик) патентной классификации ограничивается тематическая область поиска, выявляются и анализируются патентные материалы, относящиеся к ней за необходимый временной промежуток.

Именной (или фирменный) – поиск проводится в том случае, когда известны имя (имена) изобретателя (изобретателей) или названия фирм. Этот вид поиска дополняет предметный поиск.

Нумерационный поиск – осуществляется, когда известен номер охранного документа и по его номеру требуется узнать другие данные об изобретении, полезной модели, промышленном образце.

Поиск патентов-аналогов – проводится для выявления патентов, выданных в какой-либо стране и запатентованных затем в других странах, т.е. выявляются патенты, выданные в каждой стране патентования на одно и то же изобретение. К этому виду поиска целесообразно прибегать, если найден патент, интересующий специалиста, на редком языке, а патенты-аналоги позволяют ознакомиться с описанием этого изобретения на других более доступных языках. Кроме того, этот вид поиска дополняет предметный и проводится на стадии подробного ознакомления с полными описаниями к патентам.

Среди патентных исследований следует особо выделить исследование патентной чистоты. Патентная чистота – это юридическое свойство технического изделия или технологии, заключающееся в том, что они могут использоваться в определенной стране без нарушения действующих на ее территории патентов.

Установление патентной чистоты продукции позволяет избежать ответственности за возможное нарушение прав, вытекающих из патентов, действующих на территории той страны, в которой предполагается реализация данной продукции. Исследование на патентную чистоту заключается в отыскании действующих в данной стране патентов, имеющих отношение к продукции, их анализе, а также изучении обстоятельств, которые могли бы способствовать беспрепятственной реализации продукции на данном рынке [12].

В соответствии с темой дипломного проекта был проведен патентный поиск в области программных систем для анализа успеваемости студентов, и автоматизации отчётности в деканате высшего учебного заведения[19].

В качестве аналогов можно выделить:

1. Автоматизация процесса составления деканатом рейтинга студентов
2. Модуль «Деканат» интегрированной автоматизированной информационной системы «ВУЗ»
3. Система управления учебным процессом «Магеллан»
4. Система автоматизации учебного процесса «GS-Ведомости»
5. Информационно-аналитическая система управления деканатом высшего учебного заведения
6. Программный комплекс «Электронный документооборот деканата»
7. Информационная система «Деканат»
8. МОДУЛЬ "ДЕКАНАТ JTSQL"
9. Программный комплекс «Деканат» КГПУ им. В.П. Астафьева

### Автоматизация процесса составления деканатом рейтинга студентов

Язык программирования:Visual Basic for Applications.

Номер регистрации (свидетельства): 2015610242.

Дата регистрации: 12.01.2015.

Реферат: Программа служит для автоматизации процесса составления сотрудниками деканата сводных ведомостей успеваемости студентов факультета. Программа подготавливает формы для отчетности сотрудниками деканата о рейтинговых оценках студентов за определенные учебные периоды. Формы отчетности содержат списки групп, студентов, учебных дисциплин и рейтинговых оценок. На основании получаемых данных производится анализ, определяются проблемные места учебного процесса, формируются сводные таблицы, производятся сравнения различных структурных подразделений.

### Модуль «Деканат» интегрированной автоматизированной информационной системы «ВУЗ»

Язык программирования Java, frameworkStruts.

Номер регистрации (свидетельства): 2014611450.

Дата регистрации: 03.02.2014.

Реферат: Программа предназначена для ввода, хранения и модификации данных о студентах с целью дальнейшей обработки в ходе процесса обучения, а также для автоматизированного создания, хранения и редактирования документов, используемых сотрудниками деканатовв отчетах, для упрощения процедуры мониторинга успеваемости студентов. Область применения - программа может быть использована ввысшихучебныхзаведениях как средство мониторинга учебногопроцесса и автоматизациидокументооборота сотрудниками деканатов и отдела кадров. Программа является «клиент-серверным» приложением, выполняющим следующие функции: оператор системы. Функции оператора системы: ввод и корректировка личной карточки студента (учебная информация, личные данные, сведения о работе и образовании и прочее); поиск личных карточек по различным критериям, создание списков студентов по группам; создание и печать экзаменационных ведомостей; создание, хранение, редактирование и мониторинг оценок студентов в период сессии (по экзаменам, зачетам, курсовым работам и т.д.); оформление продления сессии студенту, создание и редактирование информации о дипломниках; получение информации о расписании занятий; формирование и учет приказов о перемещении и отчислении студентов; формирование, просмотр, удаление и повторная печать отчетов с данными; формирование бланков, ведомостей, справок, форм и сводок (по специальностям, факультетам и т.д.); формирование и просмотр выписок из дипломов; генерация текущей отчетной документации с целью упрощения контроля за успеваемостью студентов.

### Система управления учебным процессом «Магеллан»

Язык программирования Delphi, С++, PHP, JavaScript, P-SQL, SQL.

Номер регистрации (свидетельства): 2014615939.

Дата регистрации: 06.06.2014.

Реферат:Программа предназначена для автоматизации учебного процесса и административной деятельности в учебных заведениях среднего, высшего и дополнительного профессионального образования. Основные функциональные возможности программы: автоматизация работы сотрудников приёмной комиссии, деканата , отдела кадров, учебно-методического управления, кафедр, отдела составления расписаний (позволяет формировать расписание занятий в ручном и полностью автоматическом режимах), научных библиотек, общежитий; автоматизация работы, связанной с ведением НИОКР, договорной работы, ведения электронного документооборота, планирования деятельности сотрудников, проведения автоматизированного тестирования.

### Система автоматизации учебного процесса «GS-Ведомости»

Язык программирования Delphi, С++, PHP, SQL, P-SQL.

Номер регистрации (свидетельства): 2014610950.

Дата регистрации:21.01.2014.

Реферат: Программа предназначена для автоматизации работы учебных заведений среднего профессионального, высшего профессионального и дополнительного профессионального образования. Функциональные возможности: автоматизация работы приёмной комиссии; подача заявлений абитуриентами в электронном виде; автоматизация работы учебной части и деканатов ; ведение базы данных по учащимся и их успеваемости; автоматизация работы отдела кадров образовательного учреждения; ведение базы данных по сотрудникам; автоматизация работы учебно-методического управления; автоматизация работы кафедр и предметно-цикловых комиссий; обеспечивает возможность распределения нагрузки по преподавателям; автоматизация процесса создания расписания занятий в учебном заведении ; автоматизация прохождение учащимися тестов для проверки уровня знаний; автоматизация работы с контрагентами учебного заведения ; реализует ведение базы данных по заключаемым договорам. Программа реализует простейшие функции системы электронного документооборота . Программа автоматизирует процесс формирования планов работы сотрудников учебного заведения , распределение заданий и поручений; реализует сетевое унифицированное хранилище файлов и документов в произвольных форматах; реализует возможность быстрого формирования и просмотра различных статистических и аналитических отчётов руководителями учебного заведения ; автоматизирует работу библиотеки учебного заведения , ведения библиотечного фонда; автоматизирует работу общежитий учебного заведения , вселения и выселения постояльцев; реализует функциональность по настройке системы, управления пользователями системы, назначения прав доступа; реализует функциональность по созданию и редактированию шаблонов печатных документов, формируемых из системы, а также созданию и редактированию статистических и аналитических отчётов, формируемых из системы; реализует внутрисистемный чат для общения пользователей между собой.

### Информационно-аналитическая система управления деканатомвысшего учебного заведения

Язык программирования С#.

Номер регистрации (свидетельства): 2015660921.

Дата регистрации:13.10.2015.

Реферат: Программа предназначена для управления деканатом высшего учебного заведения с реализацией ведения справочной информации и документов, на основе которых происходит составление статистических и аналитических отчетов об успеваемости и движении контингента учащихся.

### Программный комплекс «Электронный документооборот деканата»

Язык программирования РНР.

Номер регистрации (свидетельства): 2016614900.

Дата регистрации:11.05.2016.

Реферат: Программный комплекс предназначен для автоматизации документооборота деканата вуза. Программный комплекс выполняет следующие основные функции: отражение основных документов, касающихся учебного процесса; формирование, вывод на экран и печать отчетов; регистрация входящей, исходящей и внутренней документаций; предварительное рассмотрение и распределение документов; оповещения о сроках выполнения того или иного приказа; контроль исполнения приказов.

### Информационная система «Деканат»

Язык программирования ObjectPascal (в среде Delphi).

Номер регистрации (свидетельства): 2014616013.

Дата регистрации:09.06.2014

Реферат: Программа предназначена для развития электронного документооборота вуза, связанного с успеваемостью студентов. Программа состоит из совокупности модулей, обеспечивающих ведение в базе данных информации об успешности освоения студентами образовательных программ, автоматического формирования ведомостей, направлений, личных карточек студентов. Специальный модуль, предназначенный для автоматического формирования академических справок, приложений к диплому, европейских приложений к диплому, основан на системе шаблонов, обеспечивающих гибкую настройку на различные принтеры. В программу включен модуль, автоматически формирующий итоговую справочную информацию по успеваемости контингента студентов в различных разрезах по внутренним запросам и данные для внешних мониторингов.

### Модуль "ДЕКАНАТ JTSQL"

Язык программирования JAVA, XML, JSON

Номер регистрации (свидетельства): 2017612930

Дата регистрации:06.03.2017

Реферат: Программа предназначена для автоматизированного сбора и обработки данных деканатов ВУЗа. Она представляет собой «тонкий» клиент, позволяющий создавать, открывать, редактировать и сохранять в базу данных информацию о студентах и о цикле обучения, а также в форматы JSON, XML. В состав программы входят компоненты по вводу и редактированию карточки студента, модуль миграции данных зачисленных абитуриентов в группы обучения студентов и компоненты ведения документооборота секретариатом деканатов. Программа поддерживает многопользовательский режим работы. Программа поддерживает следующие роли: наблюдатель, оператор, администратор.

### Программный комплекс «Деканат» КГПУ им. В.П. Астафьева

Язык программирования Python, Odoo ERP

Номер регистрации (свидетельства): 2016612958

Дата регистрации:14.03.2016

Реферат: Программа предназначена для автоматизированного сбора и обработки данных деканатов ВУЗа. Она представляет собой «тонкий» клиент, позволяющий создавать, открывать, редактировать и сохранять в базу данных информацию о студентах и о цикле обучения, а также в форматы JSON, XML. В состав программы входят компоненты по вводу и редактированию карточки студента, модуль миграции данных зачисленных абитуриентов в группы обучения студентов и компоненты ведения документооборота секретариатом деканатов. Программа поддерживает многопользовательский режим работы. Программа поддерживает следующие роли: наблюдатель, оператор, администратор.

## Технологии и инструментарий для разработки автоматизированной системы «Электронный деканат»

### OracleClient

Пакет из репозитория пакетов Nuget, предназначен для взаимодействия с СУБД Oracle. Устанавливается командой в npm консоли:

*Install-PackageOracle.OracleClient*

Основной класс, с помощью которого устанавливается соединение, это OracleConnection из пространства имён Oracle.ManagedDataAccess.Client. Из экземпляра объекта этого класса можно создать объект OracleCommand, он используется для выполнения как sql запросов, так и вызова хранимых процедур. Поддерживает вставку параметров в плейсхолдеры, которые помечаются двоеточиями в строке sql запроса, и позднее, заполняются методом Add, в который передаются объекты класс OracleParameter, инициализированные необходимыми значениями. Это помогает обеспечить защиту от sql-иньекций, исключая конкатенацию строк. Если выполнение OracleCommand подразумевает получение результирующего набора данных, то вызывается метод ExecuteReader, если получение данных не подразумевается, то запрос выполняется методом ExecuteNonQuery. ExecuteReader возвращает объект OracleDataReader, который представляет из себя итератор. Считать из него данные можно как по названию столбца результирующего набора, так и по его номеру, для перемещения к следующей записи используется метод Read.

### OWIN

Вдохновленные Rack, несколько разработчиков .NET решили создать свою абстракцию между веб сервером и компонентами инфраструктуры. Разработчики преследовали две цели:

* + Новые компоненты будут просты в разработке и использовании.
  + Приложения могут быть легко переносимы с хоста на хост, возможно даже на разных платформах

И вот что вышло.

OWIN (The OpenWeb Interface for .NET) — это спецификация определяющая интерфейс и описывающая взаимодействие между всеми компонентами.

Основой работы OWIN является словарь IDictionary<string, object>, который используется для получения доступа к HTTP запросам, заголовку запроса и окружению хоста. Все ключи описаны в спецификации OWIN.

Owin-совместимый сервер отвечает за заполнение этого словаря данными.

Следующий элемент OWIN это делегат:

Func<IDictionary<string, object>, Task>;

В качестве входных данных он принимает выше описанный словарь, а возвращает объект Task по завершению процесса.

Katana — это OWIN-совместимый хост написанный Microsoft. Помимо реализации спецификации OWIN, Katana содержит различные вспомогательные классы и обертки для упрощения разработки, содержащиеся в библиотеке Owin.Types. Например два класса-адаптера упрощающие работу со словарём: OwinRequest и OwinResponse, реализация WebSocket для Owin — OwinWebSocket, вспомогательный класс для работы с заголовками и другими параметрами запросов — OwinHelpers.

Важный момент: Katana не завязана на использовании каких либо интерфейсов или базовых типов, вместо этого определены соглашения, которым должны следовать разработчики[13].

В данной работе используется классы GoogleAuthenticationExtensions и GoogleOAuth2AuthenticationOptions из пространства имён Owin и Microsoft.Owin.Security.Google соответственно, которые позволяют организовать OAuth2 авторизацию с помощью учётной записи Google.

### EntityFramework

EntityFramework представляет специальную объектно-ориентированную технологию на базе фреймворка .NET для работы с данными. Если традиционные средства ADO.NET позволяют создавать подключения, команды и прочие объекты для взаимодействия с базами данных, то EntityFramework представляет собой более высокий уровень абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища. Если на физическом уровне мы оперируем таблицами, индексами, первичными и внешними ключами, но на концептуальном уровне, который нам предлагает EntityFramework, мы уже работает с объектами [14].

Центральной концепцией EntityFramework является понятие сущности или entity. Сущность представляет набор данных, ассоциированных с определенным объектом. Поэтому данная технология предполагает работу не с таблицами, а с объектами и их наборами.

Любая сущность, как и любой объект из реального мира, обладает рядом свойств. Например, если сущность описывает человека, то мы можем выделить такие свойства, как имя, фамилия, рост, возраст, вес. Свойства необязательно представляют простые данные типа int, но и могут представлять более комплексные структуры данных. И у каждой сущности может быть одно или несколько свойств, которые будут отличать эту сущность от других и будут уникально определять эту сущность. Подобные свойства называют ключами.

При этом сущности могут быть связаны ассоциативной связью один-ко-многим, один-ко-одному и многие-ко-многим, подобно тому, как в реальной базе данных происходит связь через внешние ключи.

Отличительной чертой EntityFramework является использование запросов LINQ для выборки данных из БД. С помощью LINQ мы можем не только извлекать определенные строки, хранящие объекты, из бд, но и получать объекты, связанные различными ассоциативными связями.

Другим ключевым понятием является EntityDataModel. Эта модель сопоставляет классы сущностей с реальными таблицами в БД.

EntityDataModel состоит из трех уровней: концептуального, уровень хранилища и уровень сопоставления (маппинга).

На концептуальном уровне происходит определение классов сущностей, используемых в приложении.

Уровень хранилища определяет таблицы, столбцы, отношения между таблицами и типы данных, с которыми сопоставляется используемая база данных.

Уровень сопоставления (маппинга) служит посредником между предыдущими двумя, определяя сопоставление между свойствами класса сущности и столбцами таблиц.

Таким образом, мы можем через классы, определенные в приложении, взаимодействовать с таблицами из базы данных.

Способы взаимодействия с БД

EntityFramework предполагает три возможных способа взаимодействия с базой данных:

* Database first: Entity Framework создает набор классов, которые отражают модель конкретной базы данных
* Model first: сначала разработчик создает модель базы данных, по которой затем EntityFramework создает реальную базу данных на сервере.
* Code first: разработчик создает класс модели данных, которые будут храниться в бд, а затем EntityFramework по этой модели генерирует базу данных и ее таблицы

### Паттерн «репозиторий»

Основное преимущество репозиториев — это абстрактный механизм хранения для коллекций сущностей.

Предоставляя интерфейс MemberRepository мы развязываем руки разработчику, который уже сам решит, как и где хранить данные.

Посредничает между уровнями области определения и распределения данных (domain and data mapping layers), используя интерфейс, схожий с коллекциями для доступа к объектам области определения.

Система со сложной моделью области определения может быть упрощена при помощи дополнительного уровня, например DataMapper, который бы изолировал объекты от кода доступа к БД. В таких системах может быть полезным добавление ещё одного слоя абстракции поверх слоя распределения данных (DataMapper), в котором бы был собран код создания запросов. Это становится ещё более важным, когда в области определения множество классов или при сложных, тяжелых запросах. В таких случаях добавление этого уровня особенно помогает сократить дублирование кода запросов.

Паттерн репозиторий посредничает между слоем области определения и слоем распределения данных, работая, как обычная колекция объектов области определения. Объекты-клиенты создают описание запроса декларативно и направляют их к объекту-репозиторию (Repository) для обработки. Объекты могут быть добавлены или удалены из репозитория, как будто они формируют простую коллекцию объектов. А код распределения данных, скрытый в объекте Repository, позаботится о соответствующих операциях в незаметно для разработчика. В двух словах, паттерн Repository инкапсулирует объекты, представленные в хранилище данных и операции, производимые над ними, предоставляя более объектно-ориентированное представление реальных данных. Repository также преследует цель достижения полного разделения и односторонней зависимости между уровнями области определения и распределения данных [15].

### Bootstrap

Bootstrap — это CSS/HTML фреймворк для создания сайтов. Другими словами, это набор инструментов для вёрстки. В нём есть ряд преимуществ, благодаря которым BS считается самым популярным из себе подобных[16]. Преимущества бутстрапа:

* Скорость работы — благодаря множеству готовых элементов вёрстка с бутстрапом занимает значительно меньше времени;
* Масштабируемость — добавление новых элементов не нарушает общую структуру;
* Лёгкая настраиваемость — редактирование стилей производится путём создания новых css-правил, которые исполняются вместо стандартных. При этом не требуется использовать атрибуты типа “!important”;
* Большое количество шаблонов;
* Огромное сообщество разработчиков;
* Широкая сфера применения — Bootstrap используется в создании тем для практически любой CMS (OpenCart, Prestashop, Magento, Joomla, Bitrix, WordPress и любые другие), в том числе для одностраничных приложений.

Особой популярностью пользуется Bootstrap для создания одностраничников или “лендингов” (landingpage).

Шаблоны Bootstrap

Шаблоны в бутстрапе позволяют менять уже изменённые элементы под свои потребности. Множество разработчиков предлагают свои шаблоны (как платно, так и бесплатно).

Подключить шаблон в Bootstrap очень легко: после подключения бутстрапа нужно только добавить вызов CSS вашего шаблона.

### jQuerry

jQuery — библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API для работы с AJAX.

Возможности:

* Движок кросс-браузерных CSS-селекторов Sizzle, выделившийся в отдельный проект;
* Переход по дереву DOM, включая поддержку XPath как плагина;
* События;
* Визуальные эффекты;
* AJAX-дополнения;
* JavaScript-плагины.

Точно так же, как CSS отделяет визуализацию от структуры HTML, JQuery отделяет поведение от структуры HTML. Например, вместо прямого указания на обработчик события нажатия кнопки, управление передаётся JQuery, которая идентифицирует кнопки и затем преобразует его в обработчик события клика. Такое разделение поведения и структуры также называется принципом ненавязчивого JavaScript.

Библиотека jQuery содержит функциональность, полезную для максимально широкого круга задач. Тем не менее, разработчиками библиотеки не ставилась задача совмещения в jQuery функций, которые подошли бы всюду, поскольку это привело бы к большому коду, бо́льшая часть которого не востребована. Поэтому была реализована архитектура компактного универсального ядра библиотеки и плагинов. Это позволяет собрать для ресурса именно ту JavaScript-функциональность, которая на нём была бы востребована[17].

### jQueryDataTables

**DataTables** это мощный плагин jQuery для создания таблиц и списков. Плагин jQueryDataTables способен сделать таблицу с поиском, сортировкой и разбитие на страницы без каких-либо особых настроек.

Сначала нужно скачать библиотеку DataTables. Если Вы не хотите, скачивать файлы, можете воспользоваться Microsoft CDN. Его необходимо подключить как и плагин jQuery и тоже желательно последней версии.

DataTables позволяет работать с данными напрямую в HTML, указать данные в виде массива во время инициализации. Или можем работать с данными, источник которых поступает из Ajax и MySQL.

Таблицу можно просматривать на маленьких экранах мобильных устройств. Для этого нужно дополнительно подключить файл CSS и JavaScript. Для таблицы HTML добавить необходимые классы[18].

## Заключение

1. Автоматизированная система «электронный деканат» включает в себя следующие основные компоненты: база данных, веб-сервер и веб-приложение.
2. Для реализации автоматизированной системы целесообразно применить платформу Windows Server 2012,ASP.NET 4.5.
3. Для разработки веб-приложения целесообразно использовать технологию ASP.NETMVC 5, язык программирования С# 6.0.
4. Для организации базы данных целесообразно использовать Oracle 12c.
5. Для доступа к базе данных Oracle используется Oracle.OracleClient версии 4.121.2
6. Для хранения информации о пользователях предполагается использовать Microsoft SQL Server 2012
7. Для доступа к базе данных Microsoft SQL Server используется технология Object Relation Mapping(ORM) Code First.
8. В качестве сервера приложений предлагается применить IIS8.0.
9. Осуществляется поддержка 2 форм авторизации: прямая аутентификация (по имени и паролю) и OAuth2 авторизация.
10. Для разработки клиентской части приложения будут использованы javascript - библиотеки jQuarry версии 1.10.2 и DataTables версии 1.10.15 и CSS-фреймворк Bootstrap версии 3.0.0.
11. Для разработки программного обеспечения будет использовано Visual Studio 2013.

### Проектирование автоматизированной системы «Электронный деканат»

Изм.

№ докум.

Подпись

Дата

Лист

1

ДП 02.00.ПЗ

Разраб.

Прокопович Д.В.

Провер.

Смелов В.В.

Консульт.

Смелов В.В.

.

Н. Контр.

Жиляк Н.А.

Утверд.

Смелов В.В.

Проектирование автоматизированной системы «Электронный деканат»

Лит.

Листов

1

*БГТУ 64419006,2017*

У

## Архитектура

На рисунке 2.1 изображена архитектура автоматизированной системы «электронный деканат».

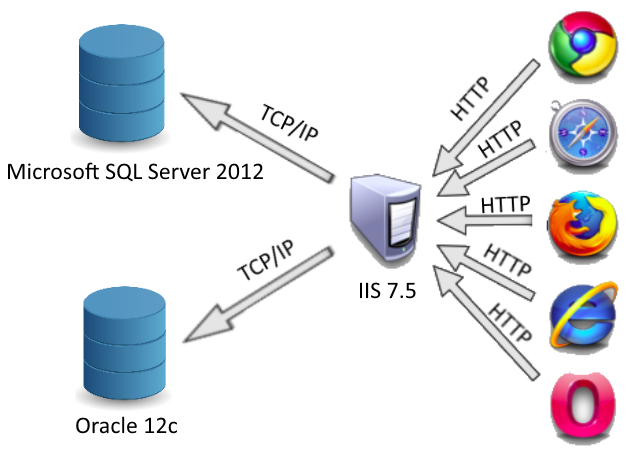


Рисунок 2.1 – Архитектура решения

Приложение состоит из 3 частей: база данных Oracle, база данных MSSQL и IIS сервера, на котором работает ASP.NET приложение.

И базы данных, и IIS сервер работают в виртуальной машине под управлением Windows 7, доступной по внешнему IP адресу 195.50.2.89.

Само приложение развёрнуто на IIS сервере и доступно по адресу *http://195.50.2.89:40001/*

## Диаграммы вариантов использования и последовательностей

Для описания пользователей системы, а так же действий, которые может выполнять каждый пользователь в системе, была разработана диаграмма вариантов использования, представленная на рисунке 2.2.

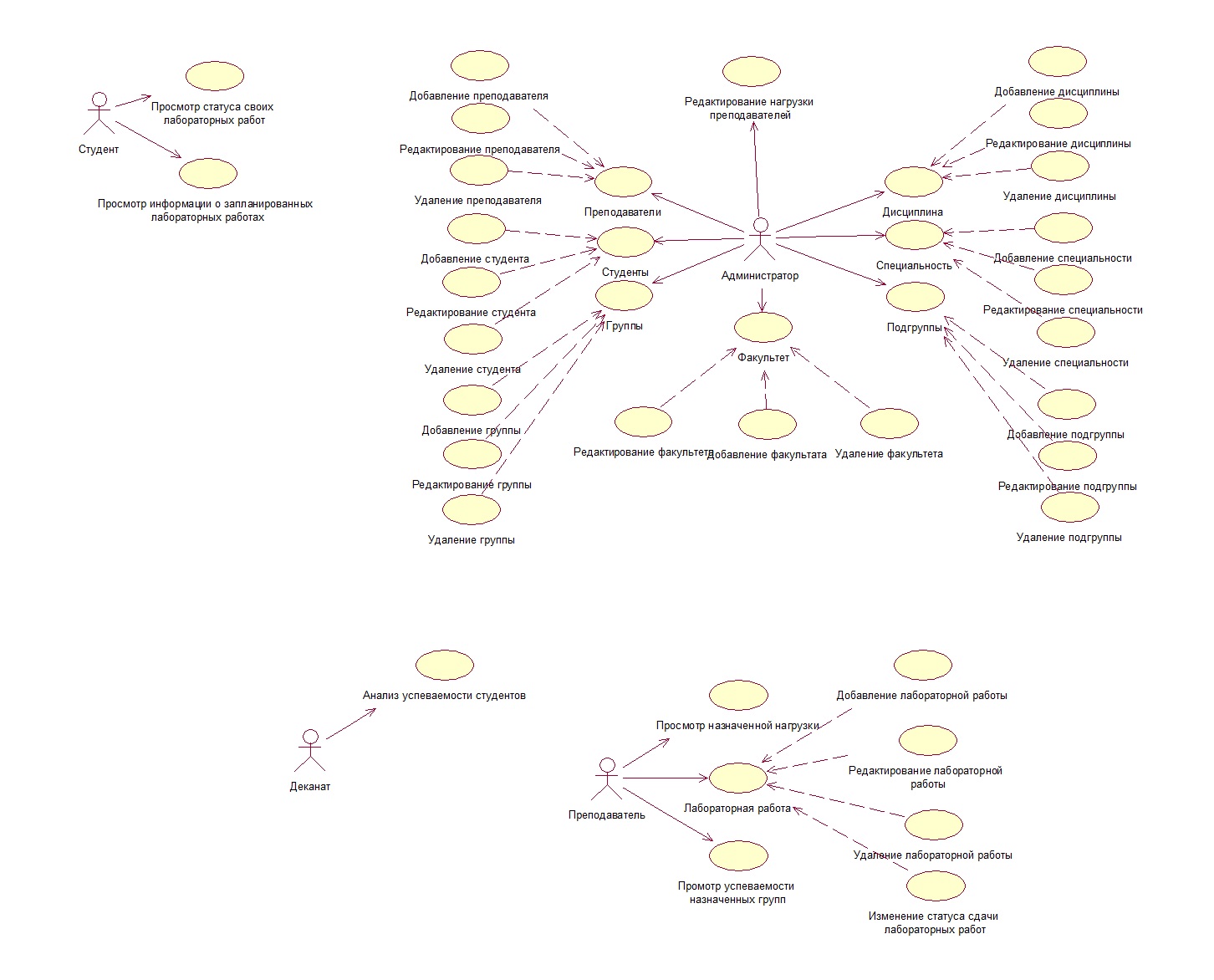


Рисунок 2.2 –Диаграмма вариантов использования

Автоматизированная система предполагает 4 роли пользователей, которые перечислены в таблице 2.1.

Таблица 2.1 – Роли в системе

|  |  |
| --- | --- |
| Роли пользователей | Назначение |
| Администратор | Пользователь с этой ролью оперирует основными данными в базе, такими как факультеты, специальности, дисциплины, группы, подгруппы, студенты, преподаватели. |
| Преподаватель | Пользователь с этой ролью выполняет функции преподавателя, имеет доступ к лабораторным работам и успеваемости на преподаваемых им дисциплинах. |
| Деканат | Пользователь с этой ролью может анализировать успеваемость как групп, так и отдельных студентов. |
| Студент | Пользователь с этой ролью может просматривать информацию о запланированных лабораторных работах. |

Для каждого пользователя очерчен круг решаемых им задач. Перечень задач и их краткое описание содержится в таблице 2.2.

Таблица 2.2 – Задачи ролей

|  |  |  |
| --- | --- | --- |
| Роль | Задача | Назначение |
| Администратор | Преподаватель (Добавление, редактирование, удаление) | Операции с записями о преподавателях. |
| Студент (Добавление, редактирование, удаление) | Операции с записями о студентах. |
| Факультет (Добавление, редактирование, удаление) | Операции с записями о факультетах. |
| Дисциплина (Добавление, редактирование, удаление) | Операции с записями о дисциплинах. |
| Специальность (Добавление, редактирование, удаление) | Операции с записями о специальностях. |
| Группы (Добавление, редактирование, удаление) | Операции с записями о группах. |
| Подгруппы (Добавление, редактирование, удаление) | Операции с записями о подгруппах. |
| Редактирование нагрузки преподавателей. | Назначение и удаление нагрузки преподавателям |
| Деканат | Анализ успеваемость студентов. | Представление данных, для удобного анализа успеваемости студентов, и обнаружения возникающих проблем. |
| Преподаватель | Лабораторные работы (Добавление, редактирование, удаление) | Операции с записями о лабораторных работах на преподаваемых дисциплинах. |
| Просмотр назначенной нагрузки | Отображение преподаваемых преподавателем дисциплин и подгрупп, у которых они преподаются |
| Изменение статуса сдачи лабораторных работ | Возможность отобразить в системе факт сдачи студентом лабораторной работы. |
| Просмотр успеваемости назначенных групп | Представление данных, обеспечивающее возможность оперативно просмотреть успеваемость подгруппы, и конкретных студентов в ней. |
| Студент | Просмотр статуса своих лабораторных работ | Используя представление данных, отображающее успеваемость, студент может оценить, по каким дисциплинам он отстаёт, какие лабораторные работы выполнять в первую очередь, и сколько работ по какой дисциплине необходимо выполнить. |
| Просмотр информации о запланированных лабораторных работах | Используя информационную страничку, студент может получить актуальную информацию о том, какие лабораторные работы запланированы по преподаваемым у него дисциплинам. |

На рисунке 2.3 представлена диаграмма последовательностей.

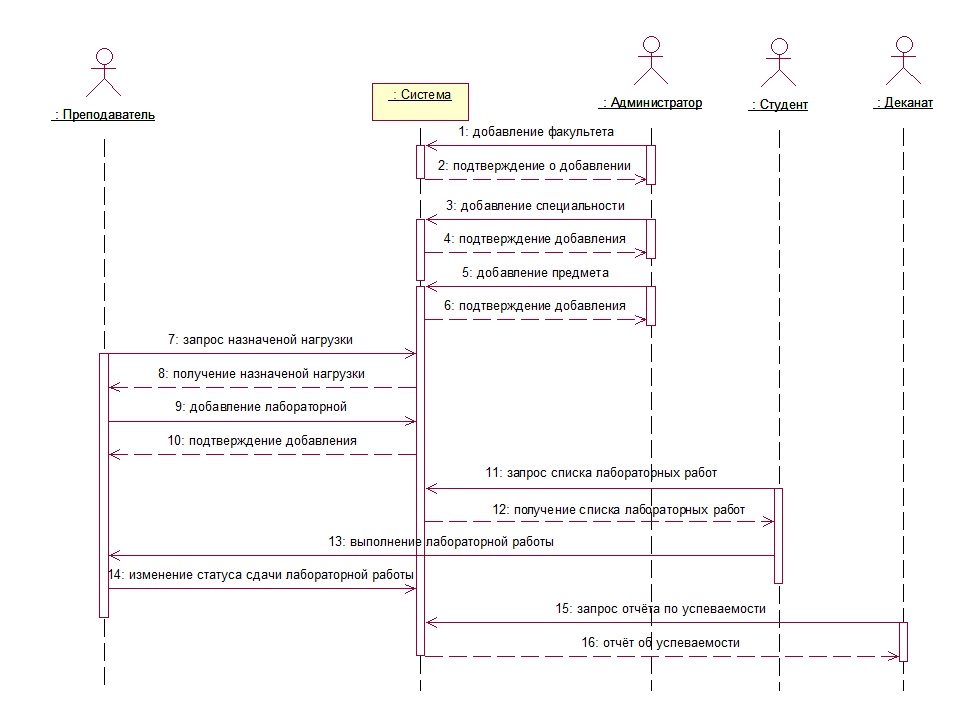


Рисунок 2.3 – Диаграмма последовательностей

Работа системы начинается с того, что администратор заполняет необходимую инфраструктуру базы, а именно: факультеты, специальности, дисциплины, группы, подгруппы, студентов. Далее, администратор назначает, какие преподаватели и у каких групп преподают ранее занесённые в базу дисциплины. После этого, преподаватель, авторизовавшись, видит все дисциплины, которые он преподаёт, и может добавлять на них необходимые лабораторные работы. Далее, студент, запрашивая информацию о запланированных лабораторных работах, получает список работ, запланированных для его специальности. Далее, студент выполняет одну или несколько работ, защищает их преподавателю. Преподаватель, отмечает факт сдачи студентом работы в системе. После этого, информация о том, что студент защитил эту лабораторную работу, будет доступна как самому студенту, так и преподавателю и деканату.

## Структура базы данных

Логическая схема базы данных представлена на рисунке 2.4.

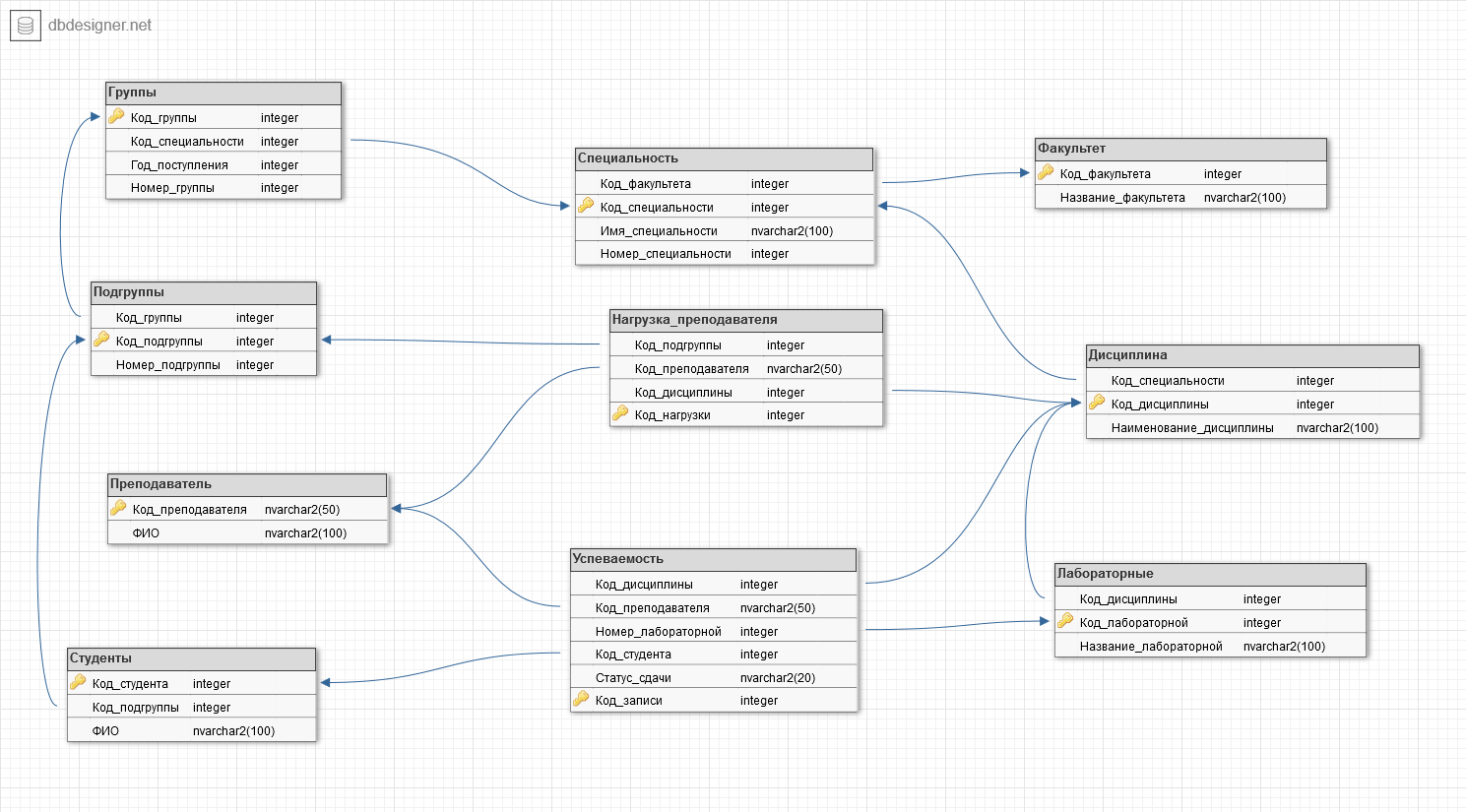


Рисунок 2.4 – Логическая схема базы данных

Описание таблиц базы данных представлено в таблице 2.3.

Таблица 2.3 – Таблицы в базе данных

|  |  |
| --- | --- |
| Наименование таблицы | Назначение |
| Факультеты | Перечень факультетов вуза |
| Специальность | Перечень кафедр вуза |
| Дисциплина | Перечень дисциплин, привязка их к кафедрам |
| Группы | Перечень групп, содержащий их специальность и год поступления |
| Подгруппы | Перечень подгрупп, содержащий их номера и привязку к группам |
| Студенты | Перечень студентов, содержащий ФИО студентов и их привязку к подгруппе |
| Преподаватель | Перечень преподавателей вуза |
| Нагрузка преподавателя | Таблица, в которой хранится закрепление подгруппы за преподавателем |
| Лабораторные | Перечень всех лабораторных с привязкой к дисциплине |
| Успеваемость | Таблица, которая хранит статус сдачи лабораторных студентами |

Назначение столбцов таблиц представлено в таблице 2.4.

Таблица 2.4 – Таблицы в базе данных с описанием полей

|  |  |  |
| --- | --- | --- |
| Таблица | Столбец | Назначение |
| Факультеты | Название\_факультета | Содержит название факультета в буквенном сокращении (ФИТ, ИДиП…) |
| Код\_факультета | Уникальный числовой идентификатор факультета |
| Специальность | Код\_факультета | Содержит название факультета в буквенном сокращении (ФИТ, ИДиП…) |
| Номер\_специальности | Цифровой код, описывающий эту специальность (например, 400501) |
| Имя\_специальности | Полное название этой специальности |
| Код\_специальности | Уникальный числовой идентификатор специальности |
| Дисциплина | Код\_специальности | Цифровой код, описывающий эту специальность (см. таблицу Специальность столбец 2) |
| Код\_дисциплины | Уникальный числовой идентификатор дисциплины |
| Наименование\_дисциплины | Полное название этой дисциплины |
| Группы | Код\_группы | Уникальный числовой идентификатор группы |
| Код\_специальности | Цифровой код, описывающий эту специальность (см. таблицу Специальность столбец 2) |
| Год\_поступления | Год зачисления студентов группы в вуз |
| Номер\_группы | Порядковый номер группы |
| Подгруппы | Код\_группы | Привязка подгруппы к группе |
| Код\_подгруппы | Уникальный числовой идентификатор подгруппы |
| Номер\_подгруппы | Порядковый номер подгруппы |
| Студенты | Код\_студента | Уникальный числовой идентификатор студента |
| Код\_подгруппы | Привязка студента к подгруппе |
| ФИО | Полное ФИО студента |
| Преподаватели | Код\_преподавателя | Уникальный числовой идентификатор преподавателя |
| ФИО | Полное ФИО преподавателя |
| Нагрузка преподавателя | Код\_подгруппы | Идентификатор обучаемой подгруппы |
| Код\_преподавателя | Идентификатор обучающего преподавателя |
| Код\_дисциплины | Идентификатор преподаваемой дисциплины |
| Лабораторные | Код\_дисциплины | Идентификатор дисциплины, за которой закреплена лабораторная работа |
| Код\_лабораторной | Уникальный числовой идентификатор лабораторной работы |
| Название\_лабораторной | Полное название лабораторной работы |
| Успеваемость | Код\_дисциплины | Идентификатор дисциплины, за которой закреплена лабораторная работа |
| Код\_преподавателя | Уникальный числовой идентификатор лабораторной работы |
| Номер\_лабораторной | Полное название лабораторной работы |

Для каждого поля, в котором присутствует уникальный идентификатор, были созданы sequence и BeforeInsert триггер уровня строки, который получает идентификаторы из этого sequence.

Помимо таблиц, для удобного представления данных из базы, были созданы представления, описанные в таблице 2.5.

Таблица 2.5 – Представления

|  |  |
| --- | --- |
| Представление | Назначение |
| FACULTY\_LIST | Перечень названий всех факультетов |
| KOORS\_LIST | Перечень всех групп, с указанием факультета, кафедры и текущего курса обучения. |
| STUDENT\_LIST | Перечень всех студентов, с указанием специальности, номера группы и подгруппы |
| LAB\_LIST | Текущая успеваемость студентов, статус сдачи ими лабораторных, в удобном для восприятия виде |
| TEACHER\_DISCIPLINE\_LIST | Список привязок преподавателей к подгруппам, в удобном для чтения виде |
| DISCIPLINE\_LIST | Перечень информации о дисциплинах |
| LABS\_DISCIPLINE | Перечень лабораторных работ на дисциплинах |
| SPECIALITY\_LIST | Перечень специальностей на факультете |
| STUDENT\_DISCIPLINE | Перечень связки студент-дисциплина-преподаватель |
| STUDENT\_LAB\_DISCIPLINE | Успеваемость студентов, отфильтрованная по дисциплинам |
| STUDENT\_LABS | Успеваемость по студентам, суммарная по всем дисциплинам |
| STUDENT\_LIST | Перечень всех студентов |
| SUBGROUP\_LIST | Перечень всех подгрупп |
| TEACHER\_DISCIPLINE | Перечень связки дисциплина-преподаватель |

Помимо создания всех необходимых таблиц и представлений, для обеспечения эффективной работы с базой данных, были созданы функции и процедуры, представленные в таблице 2.6. Все процедуры, кроме SET\_PROGRESS, представлены парами ADD/REMOVE, позволяющие удалить или добавить соответствующий логический объект в базу данных.

Таблица 2.6 – Процедуры

|  |  |
| --- | --- |
| Название | Назначение |
| REMOVE/EDIT/ADD\_FACULTY | Добавление/редактирование/удаление факультета |
| REMOVE/ EDIT/  ADD\_SPECIALISTIS | Добавление/ редактирование/ удаление кафедры, и привязка её к факультету |
| REMOVE/ EDIT/  ADD\_DISCIPLINE | Добавление/ редактирование/ удаление дисциплины, и привязка её к кафедре |
| REMOVE/EDIT/ADD\_TEACHER | Добавление/ редактирование/ удаление преподавателя |
| REMOVE/ EDIT/ADD\_LAB | Добавление/ редактирование/ удаление лабораторной работы на указанную дисциплину |
| REMOVE/ADD\_GROUP | Добавление/удаление группы студентов, и привязка её за специальностью |
| REMOVE/ADD\_SUBGROUP | Добавление/удаление подгруппы студентов, и привязка её к конкретной группе |
| REMOVE/ EDIT/ADD\_STUDENT | Добавление/ редактирование/ удаление студента, привязка его к конкретной группе |
| REMOVE/ADD\_WORK | Добавление/удаление нагрузки преподавателя |
| SET\_PROGRESS | Изменение статуса сдачи лабораторной конкретного студента |
| LOAD\_LABS | Служебная процедура, вызываемая для студента, при его добавлении, и при добавлении лабораторной работы, проверяющая таблицу успеваемости, и добавляющая необходимые записи в неё. |
| LOAD\_LABS\_SUBGROUP | Процедура, вызывающая LOAD\_LABS для всех студентов в подгруппе. |

Для вычисления курса по году поступления была разработана функция GET\_KOORS\_BY\_YEAR, которая считает разницу между годом поступления студента и 1 августа текущего года, определяя курс, на котором студент сейчас находится.

## Заключение

1. В ходе проектирования, была разработана архитектура базы данных, включающая в себя 10 таблиц, 27 хранимых процедур, 9 триггеров, 1 функцию, 13 представлений.
2. Была разработана схема взаимодействия пользователей с системой, распределены роли и их функции.
3. Автоматизированная система использует 4 роли: администратор, преподаватель, деканат, студент, общее количество задач, выполняемых пользователями, 31.

### Реализация программного обеспечения автоматизированной системы «Электронный деканат»

Изм.

№ докум.

Подпись

Дата

Лист

1

ДП 03.00.ПЗ

Разраб.

Прокопович Д.В.

Провер.

Смелов В.В.

Консульт.

Смелов В.В.

.

Н. Контр.

Жиляк Н.А.

Утверд.

Смелов В.В.

# Реализация программного обеспечения автоматизированной системы «Электронный деканат»

Лит.

Листов

1

*БГТУ 64419006,2017*

У

## Принципы построения по автоматизированной системы

Система имеет следующие особенности реализации и технические решения:

* Приложение разработано в соответствии с паттерном MVC.
* Используется принцип «тонкий клиент» - вся работа возложена на Oracleserver, комбинация из процедур, триггеров, ограничений целостности, функций и представлений выполняют всю логику, и поддерживают базу в согласованном состоянии.
* Для доступа к данным применяется паттерн «репозиторий» который позволяет получить доступ к данным единообразным образом.
* Репозитории объединены с помощью паттерна UnitOfWork (связь включением), который обеспечивает единообразное использование репозиториев.
* Так как необходимо отображать множество различных таблиц, была применена JQuerrydatatable библиотека, обеспечивающая разбивку на страницы, сортировку и поиск в каждой таблице.
* Для соединения с базой данных была написана обёртка в виде singleton-класса, с помощью которого все репозитории получают доступ, использую единый объект connection.
* Для сущностей, применяемых в системе, реализованы CRUD операции создания, чтения, редактирования и удаления.
* Для большинства сущностей применяется схема, где для редактирования сущности применяется модель, наследуемая от базовой, с префиксом New, что позволяет корректно использовать внутренние механизмы ASP.NET, такие как построение страницы по модели, и валидация. Это необходимо, так как при редактировании сущности, необходимо показать как старые, так и новые значения, и базовый класс для этого не подходит.
* База ASP.NET аккаунтов физически отделена от базы сущностей предметной области, и находится на MSSQL сервере, что позволяет использовать встроенные в ASP.NETмеханизмы авторизации, аутентификации, ролей, OAuth2 авторизацию, защищённое хранилище паролей.
* Стандартная регистрация была доработана, чтобы обеспечить соответствие между пользователями ASP.NET и преподавателями в системе.

## MVC – Контроллеры

Таблица 3.1 – Контроллеры

|  |  |
| --- | --- |
| Контроллер | Описание |
| Account | Этот контроллер содержит действия по управлению аккаунтом, регистрация (по логину и паролю, и по OAuth2), авторизация, смена пароля. |
| Admin | Этот контроллер содержит все действия, соответствующие роли «администратор» в системе, а именно: создание, редактирование и удаление факультетов, кафедр, специальностей, групп, подгрупп, студентов, управление нагрузкой преподавателя. |
| Teacher | Этот контроллер содержит все действия, соответствующие роли «преподаватель» в системе, а именно: создание, редактирование и удаление лабораторных работ по преподаваемым им дисциплинам. Так же, ему доступна страница успеваемости, с помощью которой, преподаватель может отмечать факт защиты студентом лабораторной работы. |
| Dekan | Этот контроллер содержит все действия, соответствующие роли «декан» в системе, а именно: просмотр успеваемости студентов, анализ данных, содержащихся в базе данных, оперативное выявление проблем с успеваемостью. |
| Student | Этот контроллер содержит все действия, соответствующие роли «студент» в системе, а именно: просмотр запланированных и защищённых лабораторных работ. |
| Main | Содержит страничку приветствия. |

## MVC – Модели

Таблица 3.2 – Модели

|  |  |
| --- | --- |
| Модель | Описание |
| Discipline | Модель, представляющая дисциплину на специальности |
| DisciplineStudy | Наследуется от модели Study, представляет успеваемость студента по конкретной дисциплине |
| Faculty | Модель, представляющая факультет |
| Group | Модель, представляющая группу |
| Lab | Модель, представляющая лабораторную работу |
| LabProgress | Модель, представляющая запись о статусе защиты студентом лабораторной работы. |
| NewDiscipline | Наследуется от модели Discipline, используется при редактировании дисциплины. |
| NewFaculty | Наследуется от модели Faculty, используется при редактировании факультета. |
| NewLab | Наследуется от модели Lab, используется при редактировании лабораторной работы. |
| NewSpeciality | Наследуется от модели Speciality, используется при редактировании специальности. |
| NewStudent | Наследуется от модели Student, используется при редактировании студента. |
| NewTeacher | Наследуется от модели Teacher, используется при редактировании преподавателя. |
| NewUserRole | Наследуется от модели Teacher, используется при добавлении роли пользователю. |
| NewWork | Наследуется от модели Work, используется при добавлении нагрузки преподавателю. |
| Speciality | Модель, представляющая специальность на факультете |
| Stydent | Модель, представляющая студента |
| Study | Модель, представляющая суммарную успеваемость студента |
| Subgroup | Модель, представляющая подгруппу |
| Teacher | Модель, представляющая преподавателя |
| TeacherWork | Модель, представляющая нагрузку преподавателя для преподавателя. |
| UserRole | Модель, представляющая роль пользователя |
| Work | Модель, представляющая нагрузку преподавателя для администратора. |

Диаграмма классов, показывающая отношения между моделями:

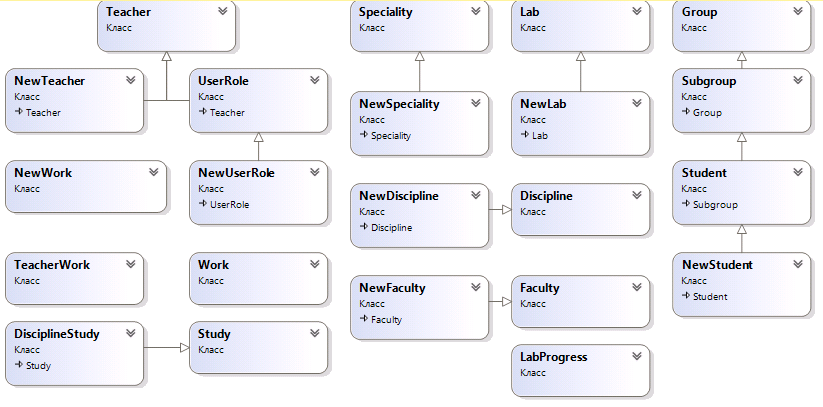


Рисунок 3.1 – Диаграмма классов

## MVC – Представления

Таблица 3.3 – Разработанные представления

|  |  |  |
| --- | --- | --- |
| Контроллер | Представление | Описание |
| Teacher | AddLab | Форма для добавления лабораторной работы на дисциплину |
| DeleteLab | Форма для удаления выбранной лабораторной работы |
| EditLab | Форма для редактирования выбранной лабораторной работы |
| Index | Страница, позволяющая преподавателю выбрать дисциплину и подгруппу, для возможности последующего редактирования статуса защиты лабораторных работ у студентов выбранной подгруппы. |
| Labs | Страница, позволяющая выбирать студентов, и смотреть, какие лабораторные работы ими сданы, какие нет, и отмечать факт защиты работ. |
| LabsList | Страница, на которой преподаватель может просмотреть все преподаваемые им дисциплины, а так же, перейти к списку лабораторных работ на выбранной дисциплине. |
| LabOnDisciplineList | Страница, отображающая для выбранной ранее дисциплины лабораторные работы с возможностью удалять, добавлять, и редактировать их. |
| ChangeLabStatus | Форма, с помощью которой можно отметить факт сдачи лабораторной работы. |
| Student | Index | Страница, позволяющая найти информацию о студенте в базе данных |
| Student | Страница, отображающая успеваемость студента по всем преподаваемым у него дисциплинам |
| Main | Index | Страница приветствия, так же отображающая авторство этой дипломной работы. |
| Dekan | Facultes | Страница со списком факультетов |
| Specialitis | Страница со списком специальностей на факультете |
| Subgroups | Страница со списком групп на специальности |
| Studing | Страница, отображающая список студентов подгруппы, суммарное количество защищённых, запланированных и не защищённых студентами лабораторных работ |
| DisciplineStuding | Страница, отображающая список дисциплин у студента, суммарное количество защищённых, запланированных и не защищённых им лабораторных работ по каждой дисциплине |
| Admin | Facultes | Страница со списком факультетов с возможностью удалять, добавлять, и редактировать их. |
| AddFaculty | Форма для добавления факультета |
| EditFaculty | Форма для переименования факультета |
| DeleteFaculty | Форма для удаления факультета |
| Specialitis | Страницасо списком специальностей на факультете,с возможностью удалять, добавлять, и редактировать их. |
| AddSpeciality | Форма для добавления специальности на факультет |
| EditSpeciality | Форма для редактирования выбранной специальности |
| DeleteSpeciality | Форма для удаления выбранной специальности |
| Disciplines | Страницасо списком дисциплин на специальности,с возможностью удалять, добавлять, и редактировать их. |
| AddDiscipline | Форма для добавления дисциплины на специальность |
| EditDiscipline | Форма для редактирования выбранной дисциплины |
| DeleteDiscipline | Форма для удаления выбранной дисциплины |
| Groups | Страницасо списком групп на специальности,с возможностью добавлять и удалять их. |
| AddGroup | Форма для добавления группы на специальность |
| DeleteGroup | Форма для удаления выбранной группы |
| Subgroups | Страницасо списком подгрупп в группе,с возможностью добавлять и удалять их. |
| AddSubgroup | Форма для добавления подгруппы в группу |
| DeleteSubgroup | Форма для удаления выбранной подгруппы |
| Students | Страницасо списком студентов в подгруппес возможностью удалять, добавлять, и редактировать их. |
| AddStudent | Форма для добавления студента в подгруппу |
| EditStudent | Форма для редактирования выбранного студента |
| DeleteStudent | Форма для удаления выбранного студента |
| Works | Страница, отображающая, кто и какую дисциплину преподаёт у выбранной подгруппы, с возможностью добавления и удаления нагрузки преподавателям |
| AddWork | Форма для добавления нагрузки преподавателя на выбранную подгруппу |
| DeleteWork | Форма для удаления нагрузки преподавателя |
| Teachers | Страница, отображающая информацию о зарегистрированных в системе преподавателях |
| EditTeacher | Форма для редактирования выбранного преподавателя |
| DeleteTeacher | Форма для удаления выбранного преподавателя |
| Users | Страница, отображающая информацию о зарегистрированных в системе пользователях, включая преподавателей, деканов и администраторов |
| Roles | Страница, отображающая информацию о ролях, назначенных пользователям, с возможностью добавить и удалить роли |
| AddRole | Форма для добавления пользователю роли |

## Маршрутизация

Маршрутизация стандартная, по шаблону {controller}/{action}/{id}, где:

controller– имя контроллера

action– имя действия в контроллере

id – необязательный параметр идентификатора

Контроллер по умолчанию Main, действие по умолчанию –Index

Метод, регистрирующий описанное поведение:

public static void RegisterRoutes(RouteCollection routes)

{

routes.IgnoreRoute("{resource}.axd/{\*pathInfo}");

routes.MapRoute(

name: "Default",

url: "{controller}/{action}/{id}",

defaults: new { controller = "Main", action = "Index",

id = UrlParameter.Optional }

);

}

## Заключение

1. В соответствии с разработанной ранее архитектурой, было написано ASP.NET приложение, реализующее запланированный функционал.
2. Для разработки использовался язык C#, а так же, паттерны, такие как MVC и Repozitory.
3. Архитектура приложения позволяет добавлять новый функционал, интегрируясь с реализованными классами.
4. Логика работы с базой данных вынесена в один класс, поэтому приложение не сложно адаптировать к изменяющимся требованиям к содержимому таблиц, форм и списков.
5. В соответствии с MVC паттерном, было разработано 7 контроллеров и 74 представления, которые обеспечивают весь запланированный функционал приложения.

Таблица 3.4 – Количественные характеристики кода

|  |  |  |
| --- | --- | --- |
| Расширение файла | Количество файлов | Строк кода(суммарно) |
| .cs | 63 | 4667 |
| .cshtml | 146 | 6210 |
| .js | 28 | 34977(включая библиотеки) |
| .css | 10 | 13929(включая библиотеки) |

### Описание контрольного примера

Изм.

№ докум.

Подпись

Дата

Лист

1

ДП 04.00.ПЗ

Разраб.

Прокопович Д.В.

Провер.

Смелов В.В.

Консульт.

Смелов В.В.

.

Н. Контр.

Жиляк Н.А.

Утверд.

Смелов В.В.

*Описание контрольного примера*

Лит.

Листов

1

*БГТУ 64419006,2017*

У

Пример состоит из 4 частей-вкладок меню: администратор, преподаватель, студент, деканат.

## Администратор

Для администратора была создана учётная запись (Логин admin@gmail.com, пароль 12345). Администратор, помимо контроля за содержимым базы данных, может выдавать полномочия администратора или деканата любому зарегистрированному пользователю. Страница, открывающяяся после авторизации, представлено на рисунке 4.1. С этой страницы, можно создать, удалить или переименовать факультет, а так же зайти в подробную информацию о факультете.

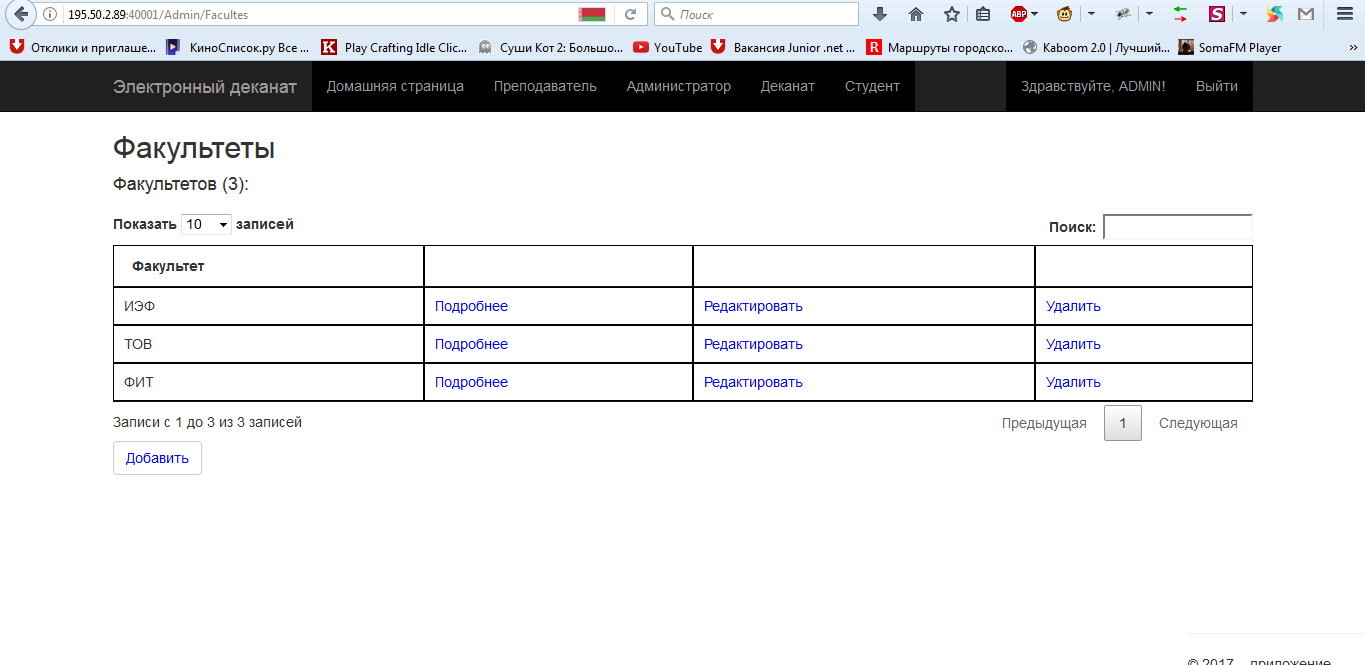


Рисунок 4.1 – Администратор, страница факультетов

Форма добавления факультета представлена на рисунке 4.2.

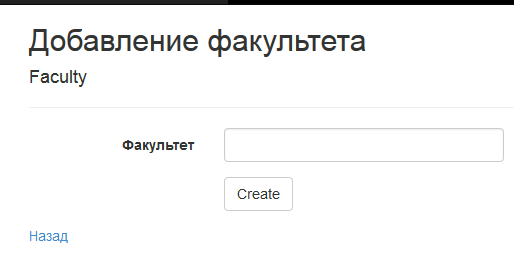


Рисунок 4.2 – Администратор, добавление факультета

Форма удаления факультета представлена на рисунке 4.3.

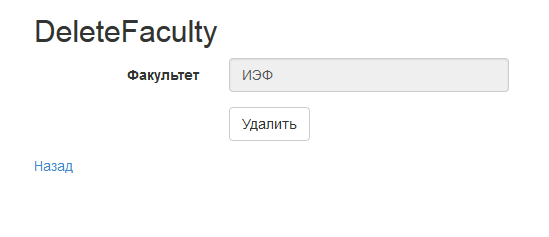


Рисунок 4.3 – Администратор, удаление факультета

Форма редактирования факультета представлена на рисунке 4.4

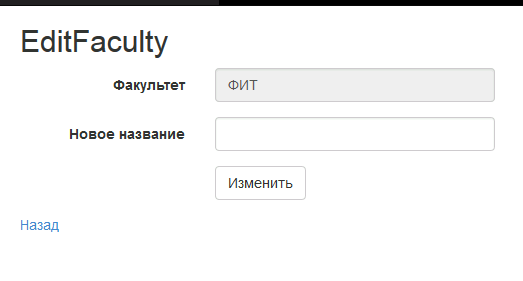


Рисунок 4.4 – Администратор, редактирование факультета

Страница, после нажатия на «подробнее» напротив факультета ФИТ, представлена на рисунке 4.5.

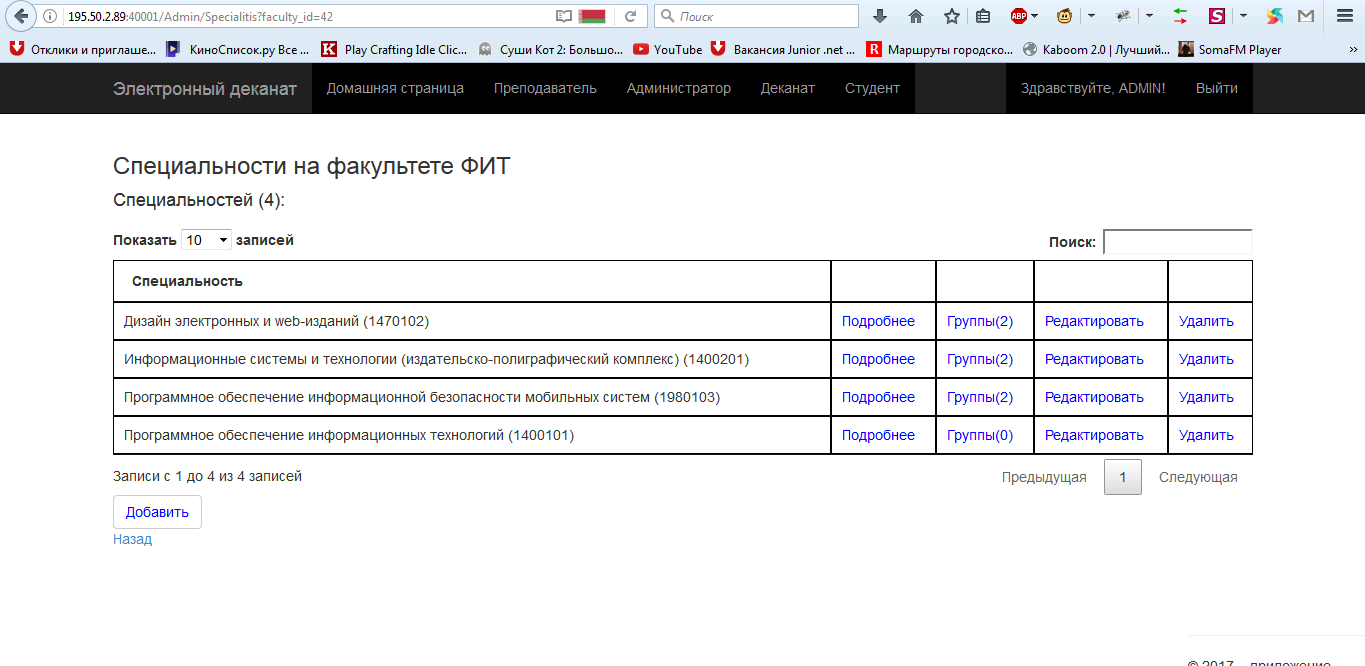


Рисунок 4.5 – Администратор, страница специальностей

С этой страницы, можно создать, удалить или переименовать специальность, а так же зайти в подробную информацию о специальности, и посмотреть какие группы учатся на этой специальности.

Форма добавления специальности представлена на рисунке 4.6.

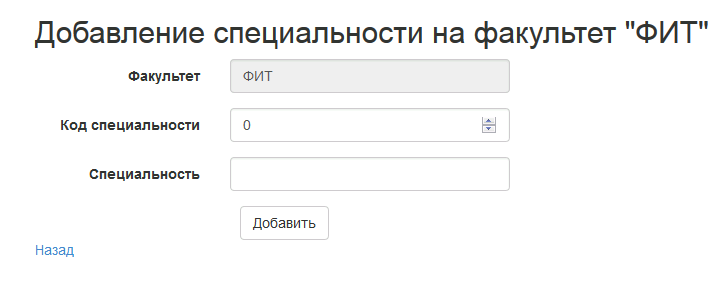


Рисунок 4.6 – Администратор, добавление специальности

Форма редактирования специальности, представлена на рисунке 4.7.

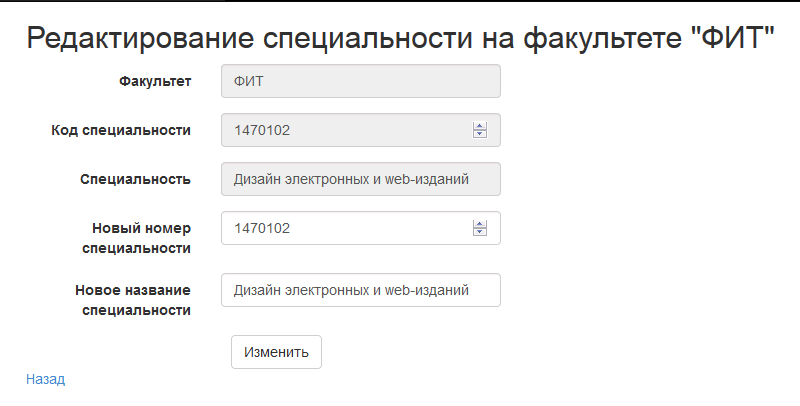


Рисунок 4.7 – Администратор, редактирование специальности

Форма удаления специальности, представлена на рисунке 4.8.

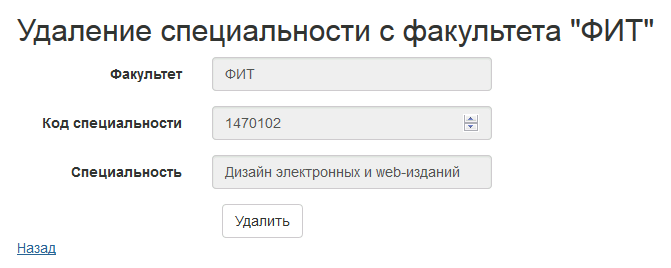


Рисунок 4.8 – Администратор, удаление специальности

После нажатия на «подробнее» напротив специальности «ИСиТ», откроется страница, представленная на рисунке 4.9.

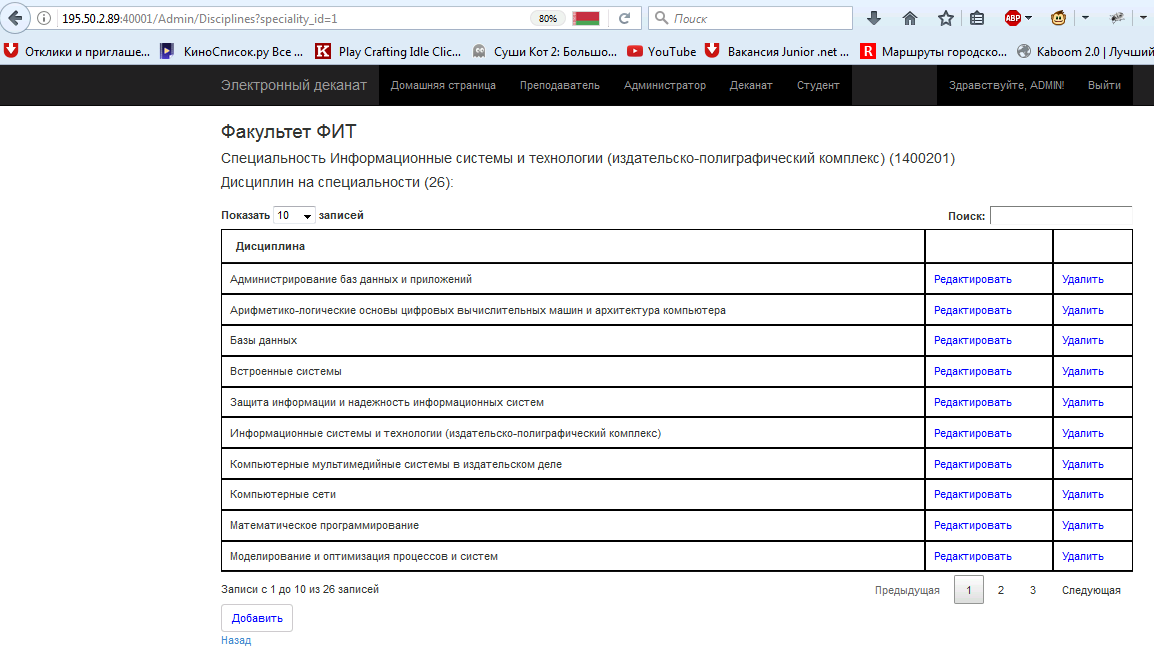


Рисунок 4.9 – Администратор, дисциплины

С этой страницы, можно создать, удалить или переименовать дисциплины.

Вернёмся на экран «специальности», и нажмём на «группы» напротив специальности ИСиТ(число в скобках – количество групп на специальности).

Попадаем на экран «группы». С этой страницы, можно создать и удалить группы. На кнопку «подробнее» можно переместиться на экран «подгруппы», представленный на рисунке 4.10 (число в скобках показывает количество подгрупп):

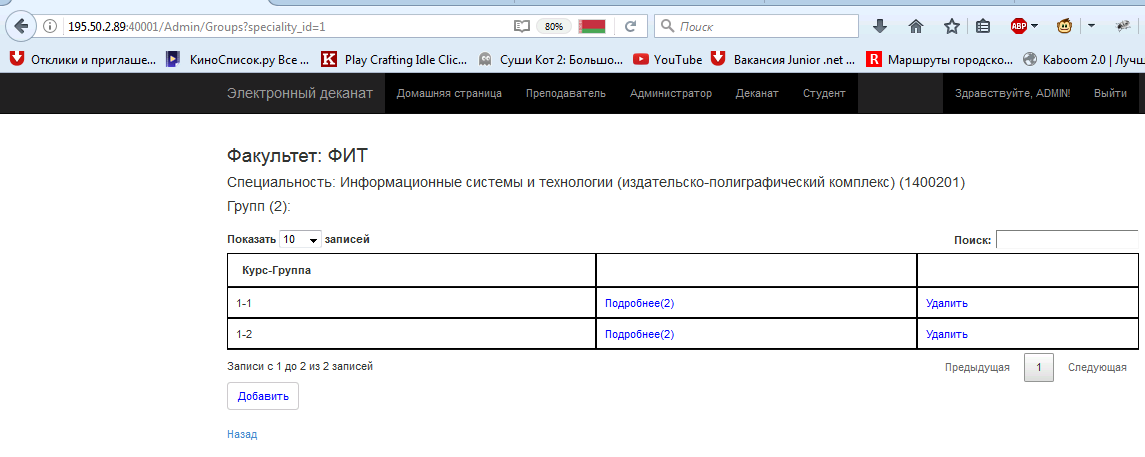


Рисунок 4.10 – Администратор, страница групп

После нажатия на кнопку «подробнее» напротив группы 1-1, откроется страница, представленная на рисунке 4.11.

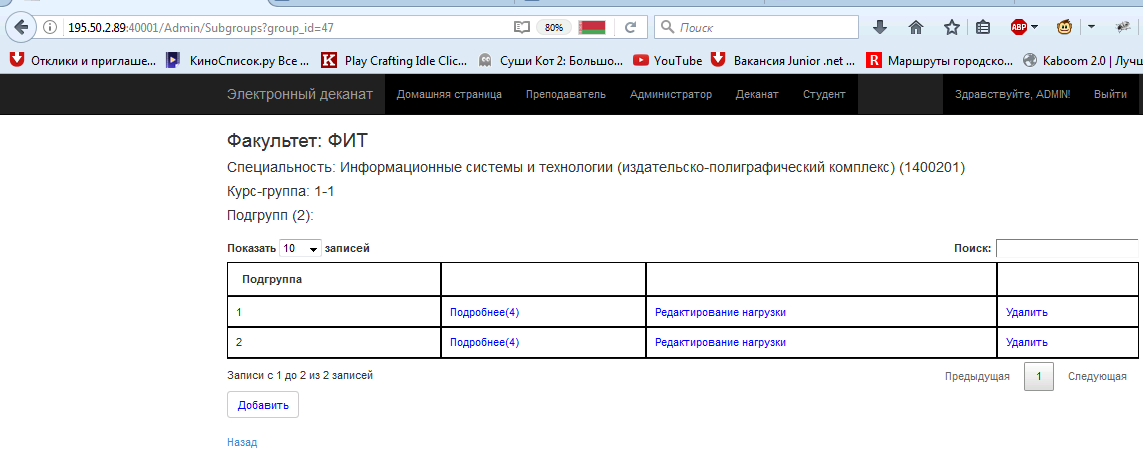


Рисунок 4.11 – Администратор, страница подгрупп

Попадаем на экран «группы». С этой страницы, можно создать и удалить подгруппы. На кнопку «подробнее» можно переместиться на экран «студенты» (число в скобках показывает количество студентов), на кнопку «редактирование нагрузки» можно просмотреть, кто и что преподаёт у подгруппы, и отредактировать эти данные:

Нажимаем «подробнее» напротив подгруппы 1. Откроется страница, представленная на рисунке 4.12. С этой страницы, можно создать, редактировать и удалить студентов.

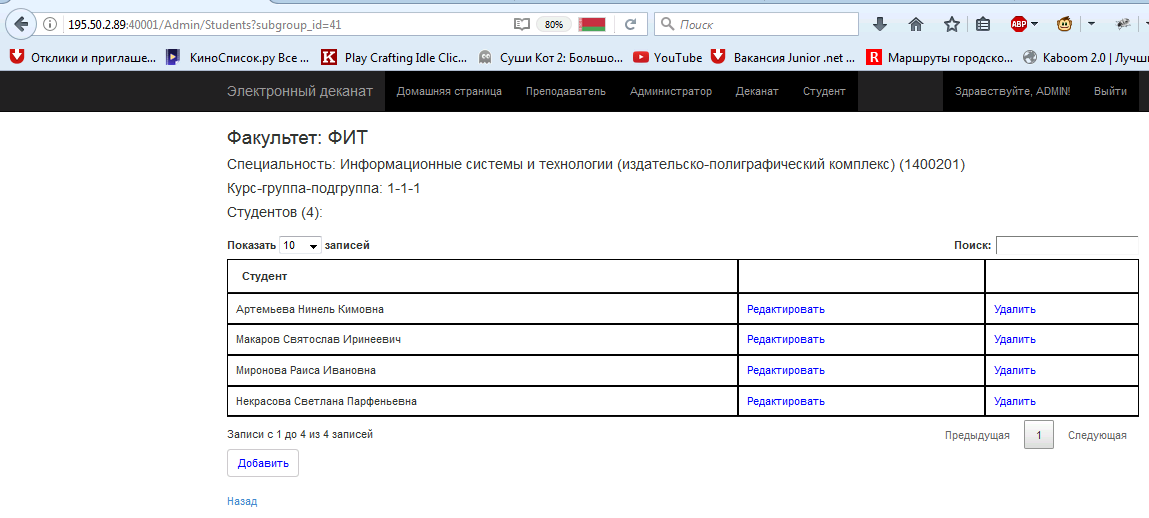


Рисунок 4.12 – Администратор, страница студентов

Вернувшись назад, и нажав на «редактирование нагрузки», попадаем на страницу, представленную на рисунке 4.13.

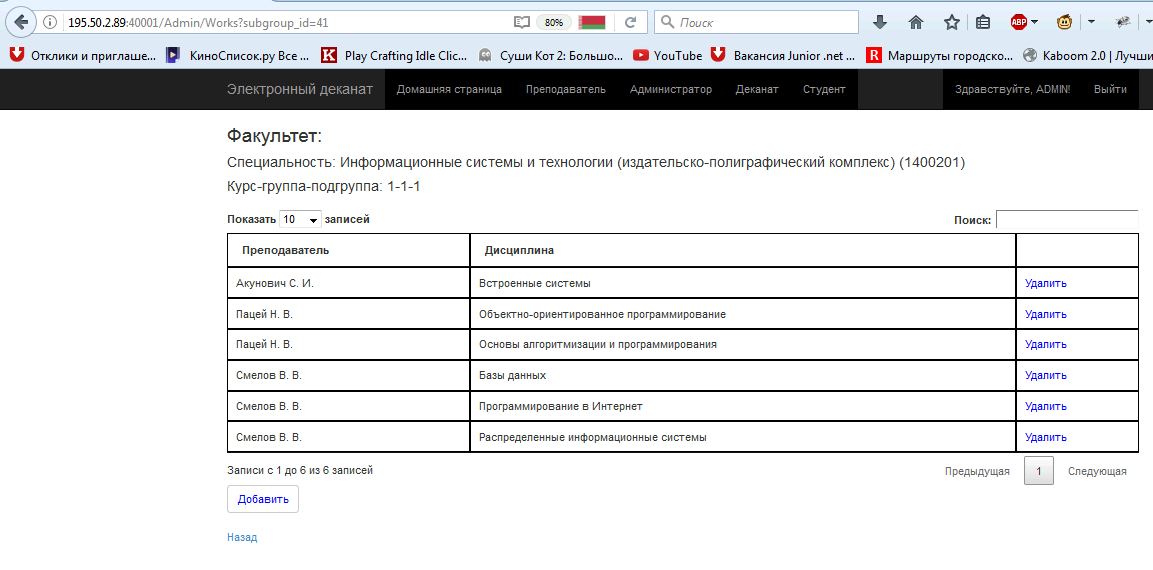


Рисунок 4.13 – Администратор, список нагрузок преподавателей

На этой странице, можно добавить и удалить нагрузку преподавателей для этой подгруппы. Форма добавления нагрузки представлена на рисунке 4.14.

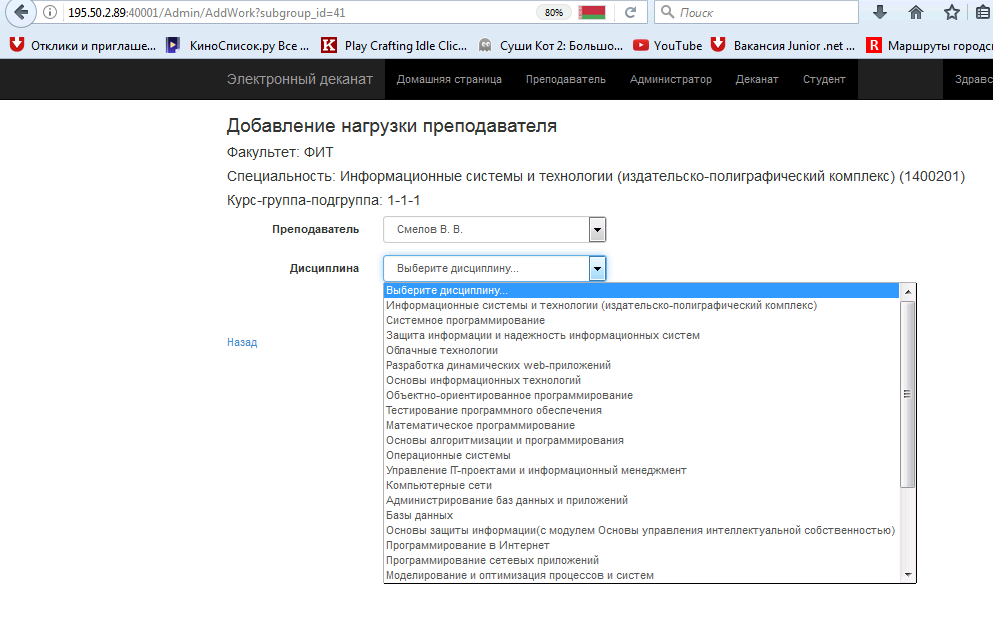


Рисунок 4.14 – Администратор, добавление нагрузки преподавателя

Перейдём на вкладку меню «преподаватели» (расположение этой вкладки приведено на рисунке 4.15). Откроется страница, представленная на рисунке 4.16.

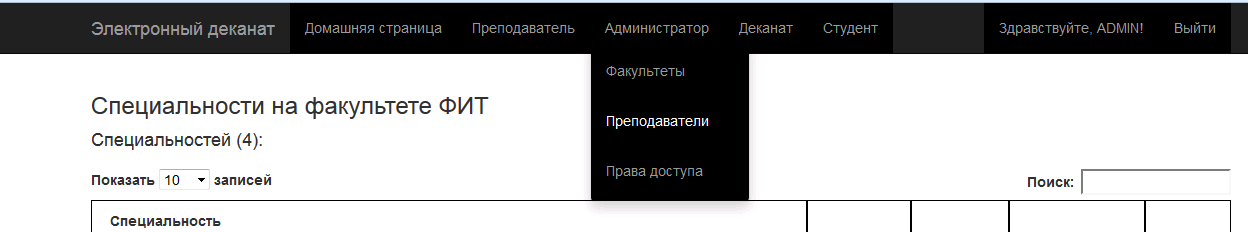


Рисунок 4.15 – Администратор, горизонтальное меню

На этой странице, можно увидеть всех зарегистрированных преподавателей, редактировать и удалять записи.

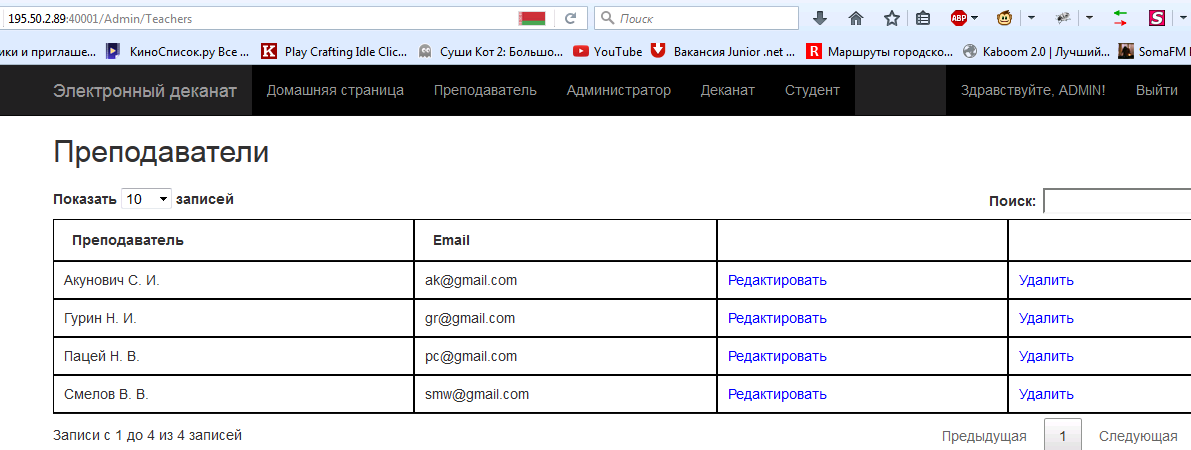


Рисунок 4.16 – Администратор, список преподавателей

Перейдём на вкладку «права доступа» (расположение этой вкладки приведено на рисунке 4.17). Откроется страница, представленная на рисунке 4.18.

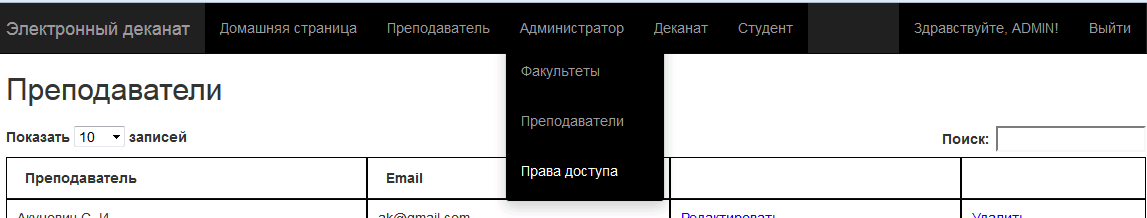


Рисунок 4.17 – Администратор, горизонтальное меню

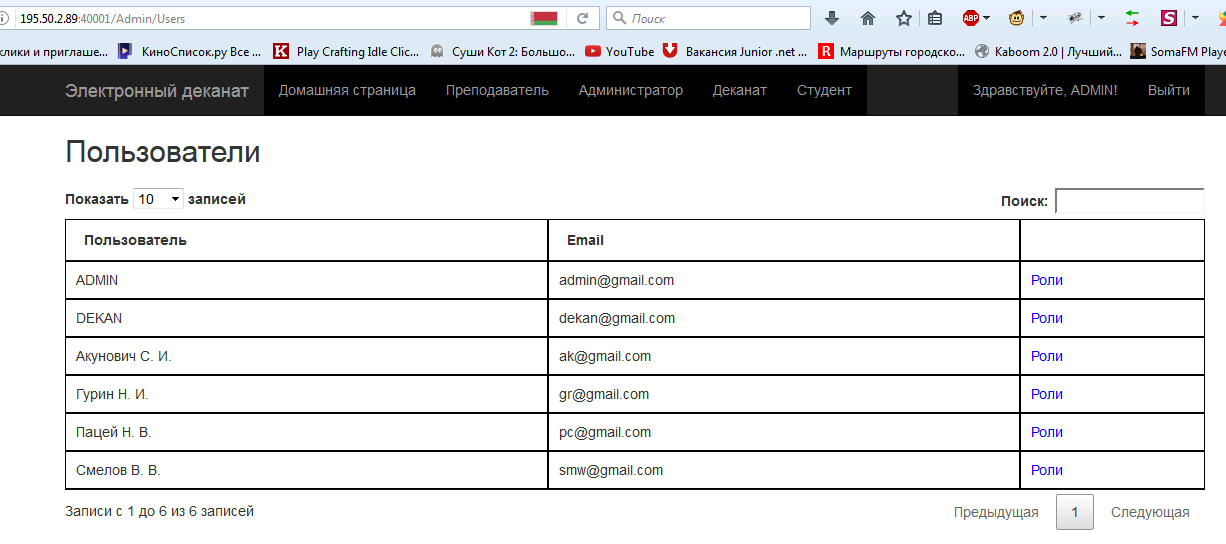


Рисунок 4.18 – Администратор, страница пользователей

На этой станице, видно всех пользователей, включая деканов и администраторов, а так же, можно просмотреть роли каждого из них. Нажмём на ссылку «роли» напротив Смелов В.В. Откроется страница, представленная на рисунке 4.19.

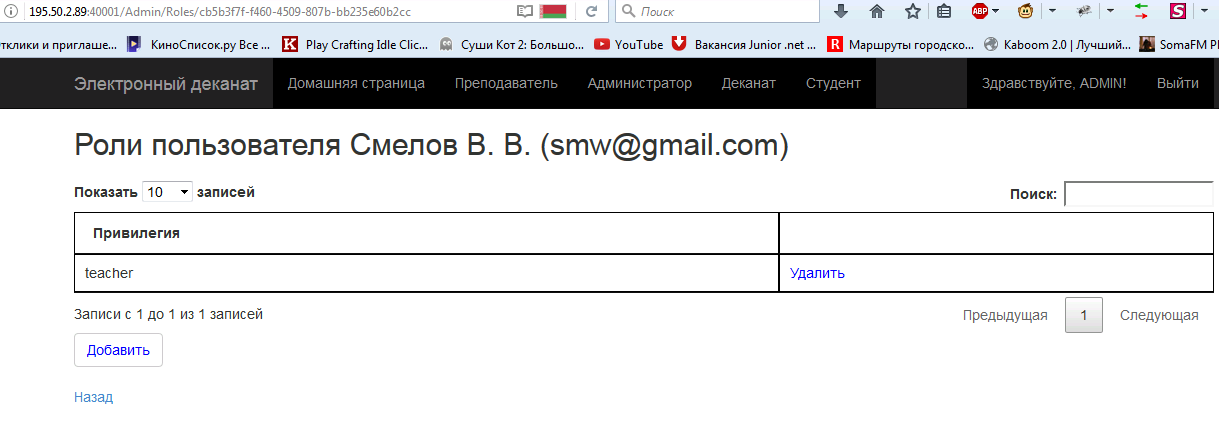


Рисунок 4.19 – Администратор, страница ролей

На этой странице, мы видим, что единственная назначенная роль этому пользователю в системе на текущий момент, это «преподаватель», а так же, есть кнопки для добавления и удаления ролей.

## Преподаватель

Для демонстрации работы преподавателя была создана учётная запись (Логин smw@gmail.com, пароль 123). Преподаватель может добавлять, удалять и редактировать лабораторные работы на всех преподаваемых им дисциплинах, а так же, отмечать факт защиты студентом лабораторной работы. Страница, открывающаяся после авторизации, представлена на рисунке 4.20.

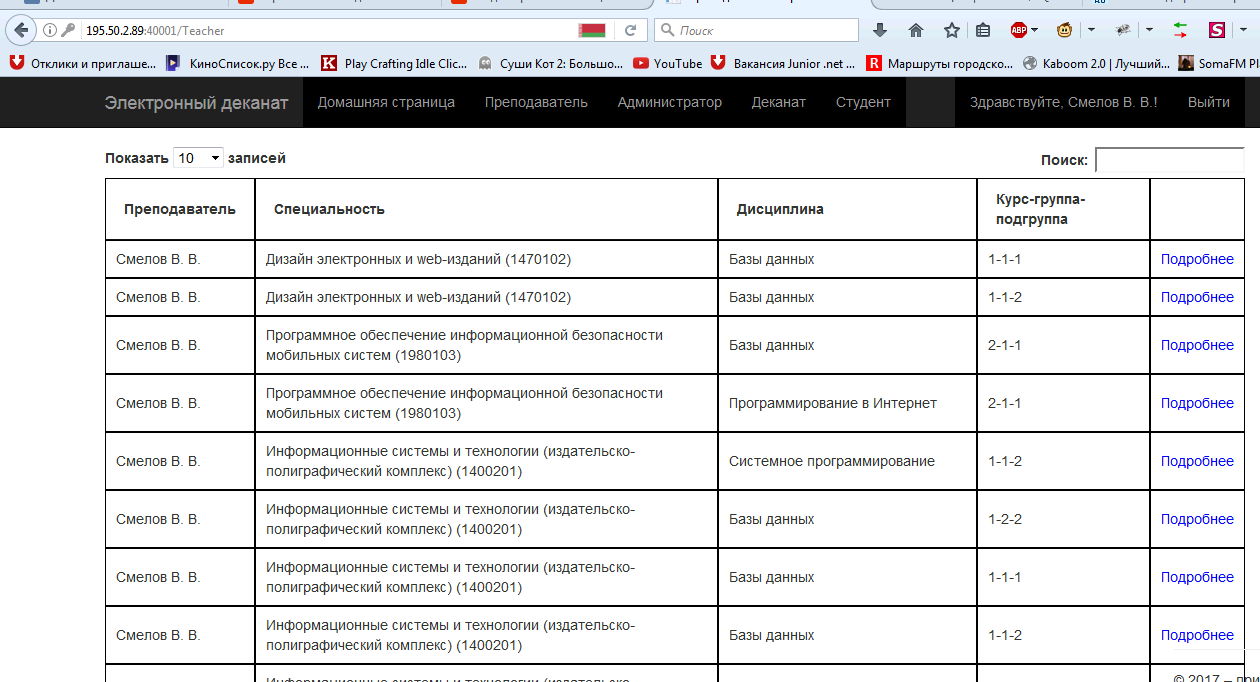


Рисунок 4.20 – Преподаватель, страница выбораподгруппы и дисциплины

На этой странице можно увидеть, какие дисциплины, и у каких групп преподаются, с помощью поиска отфильтровать данные, а так же, перейти к подробному представлению успеваемости конкретной подгруппы. Нажимаем на одну из них, получаем окошко, представленное на рисунке 4.21.

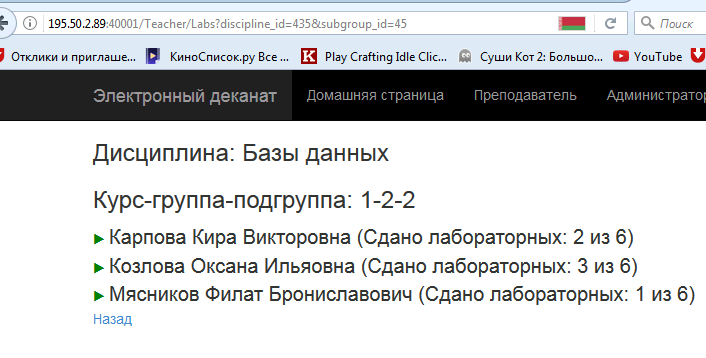


Рисунок 4.21 – Преподаватель, выбранная подгруппа

Нажав на любую из записей, она разворачивается в таблицу, представленную на рисунке 4.22.

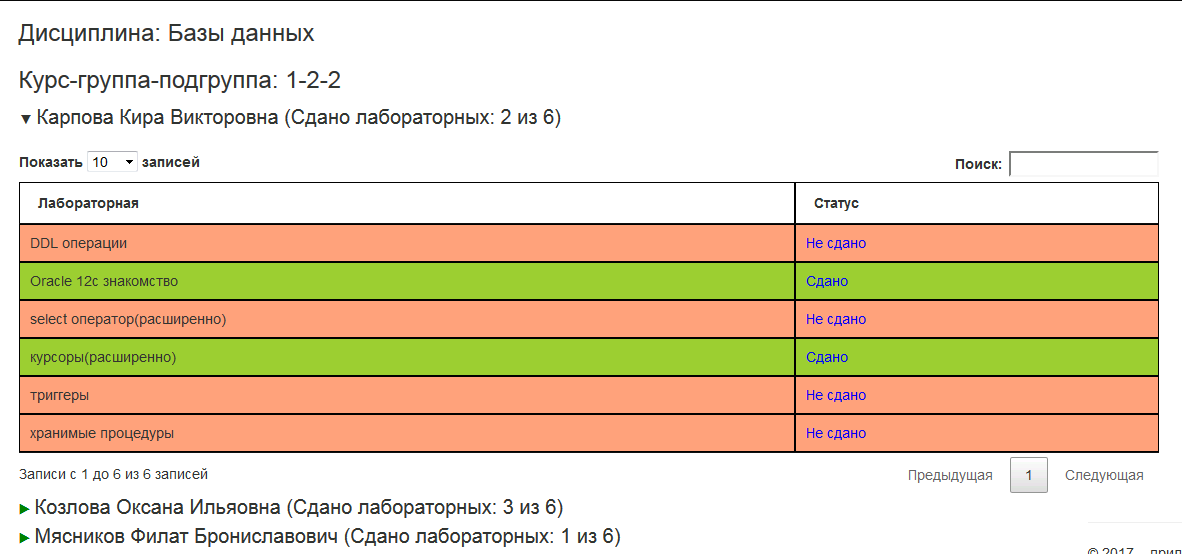


Рисунок 4.22 – Преподаватель, информация о студенте

Нажав на «статус», переходим на страницу, представленную на рисунке 4.23, в котором можно поменять статус защиты лабораторной для выбранного студента.

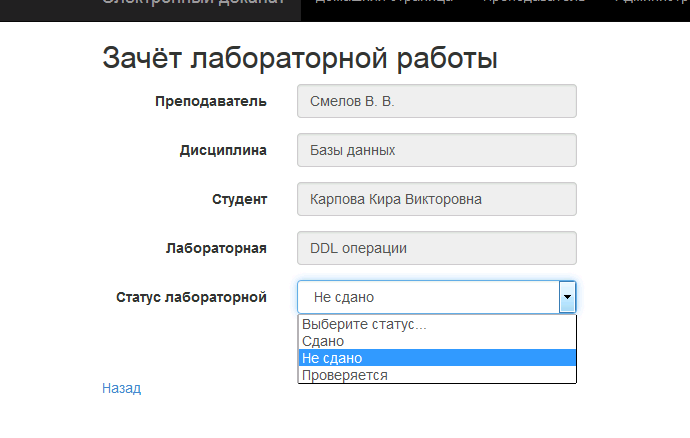


Рисунок 4.23 – Преподаватель, защита лабораторной работы

Нажав на вкладку «редактирование лабораторных» в горизонтальном меню (расположение этой вкладки приведено на рисунке 4.24). Попадаем на страницу, представленную на рисунке 4.25.

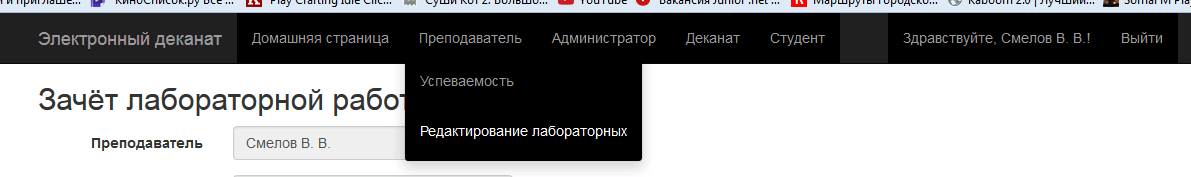


Рисунок 4.24 – Преподаватель, горизонтальное меню

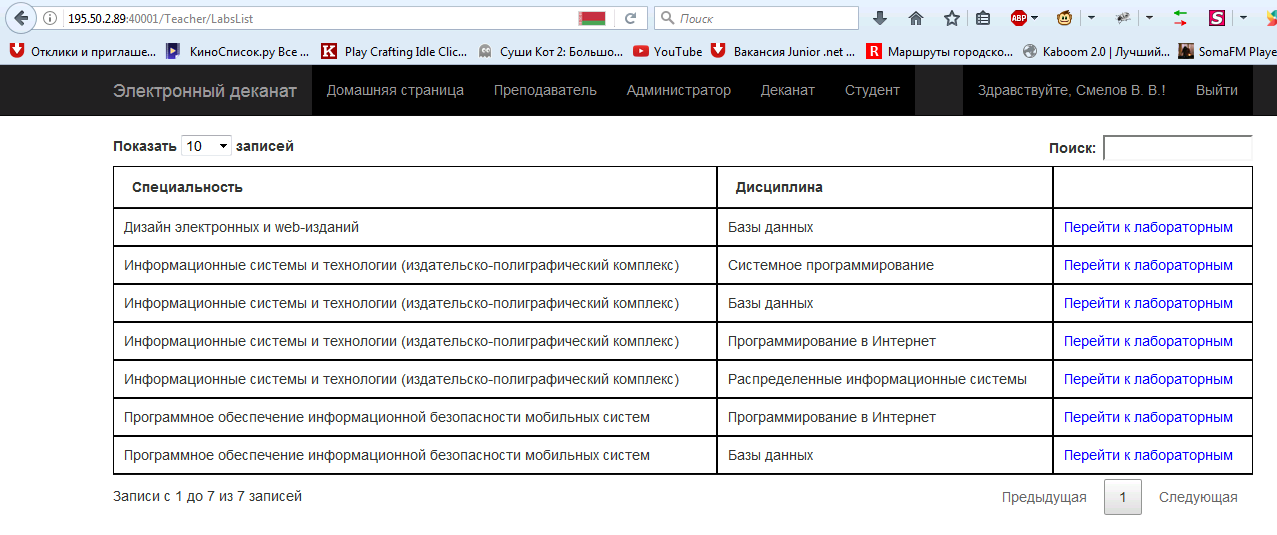


Рисунок 4.25 – Преподаватель, список дисциплин

На этой странице, можно просмотреть, на каких специальностях какие дисциплины ведёт преподаватель, который авторизован на данный момент в системе, а так же, посмотреть лабораторные работы на выбранной дисциплине, нажав на «перейти к лабораторным». При этом откроется страница, представленная на рисунке 4.26.

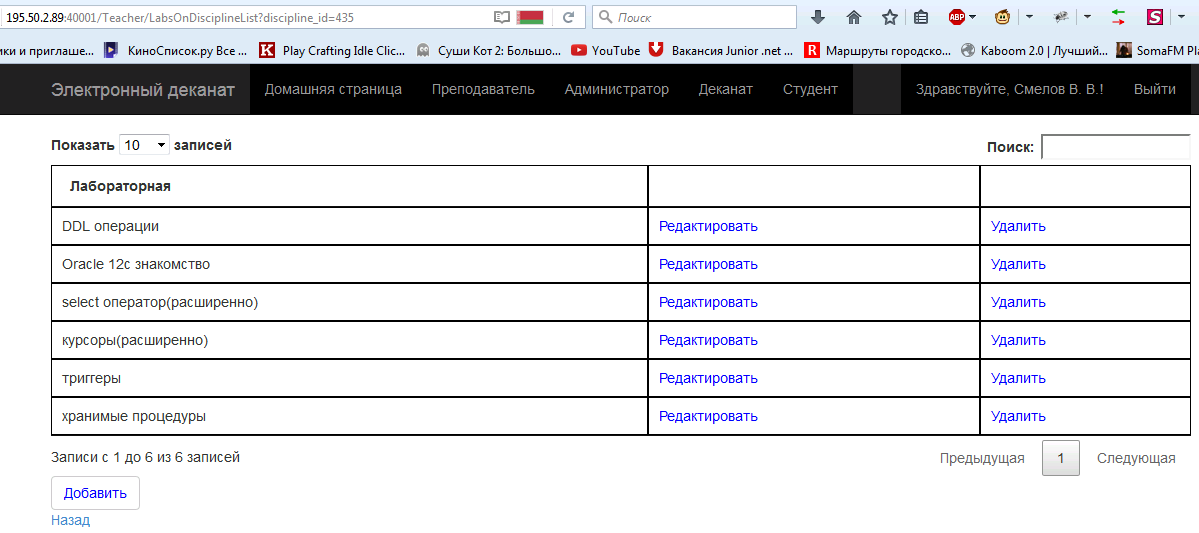


Рисунок 4.26 – Преподаватель, список лабораторных работ

С этой страницы, можно создавать, добавлять и удалять лабораторные работы.

## Студент

На странички, относящиеся к студентам, можно попасть без авторизации. Их всего 2, это поиск по студентам, и подробная информация о студенте. Страница поиска представлена на рисунке 4.27. Поиск осуществляется по части строки, например, введём «Раиса», видим, что нашло 2 студентов с таким именем, внешний вид страницы после поиска представлен на рисунке 4.28. Нажав «выбрать» открывается страничка, представленная на рисунке 4.29, на которой можно увидеть успеваемость выбранного студента, а так же, какие лабораторные нужно защитить этому студенту.

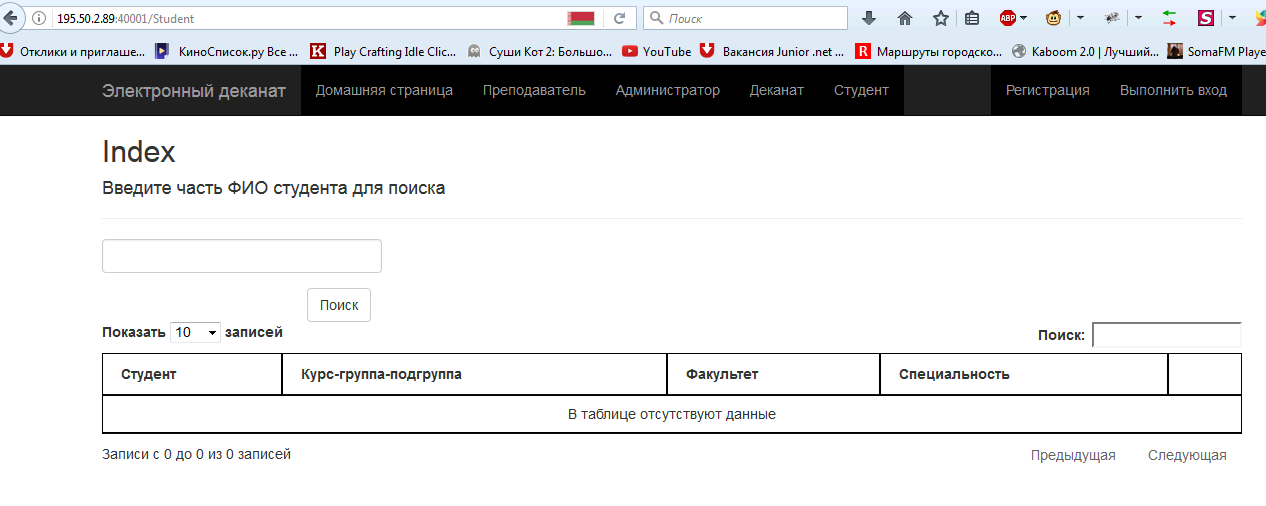


Рисунок 4.27 – Форма поиска студентов

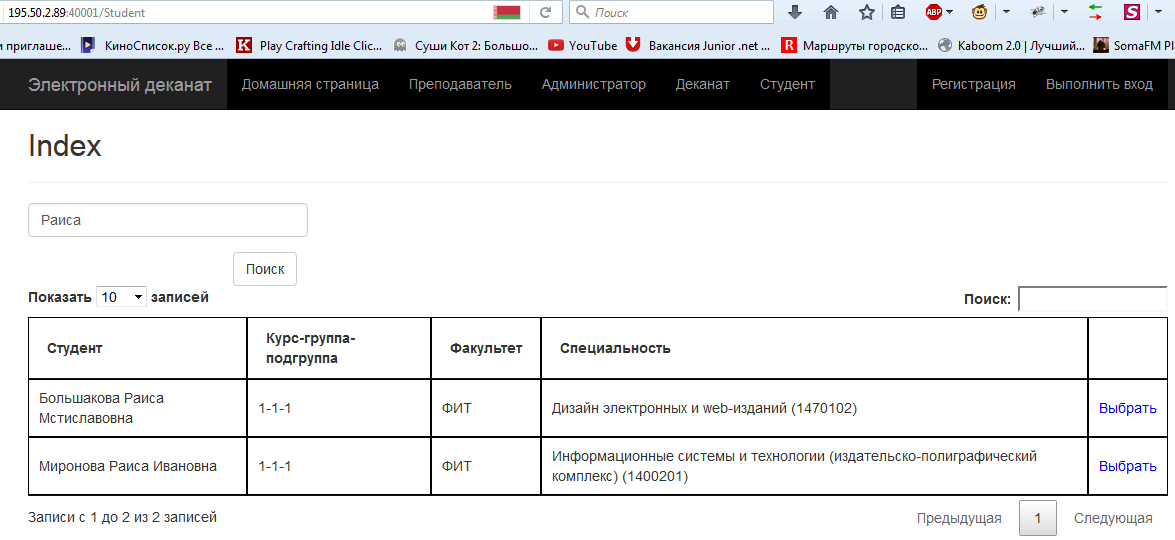


Рисунок 4.28 – Список найденных студентов

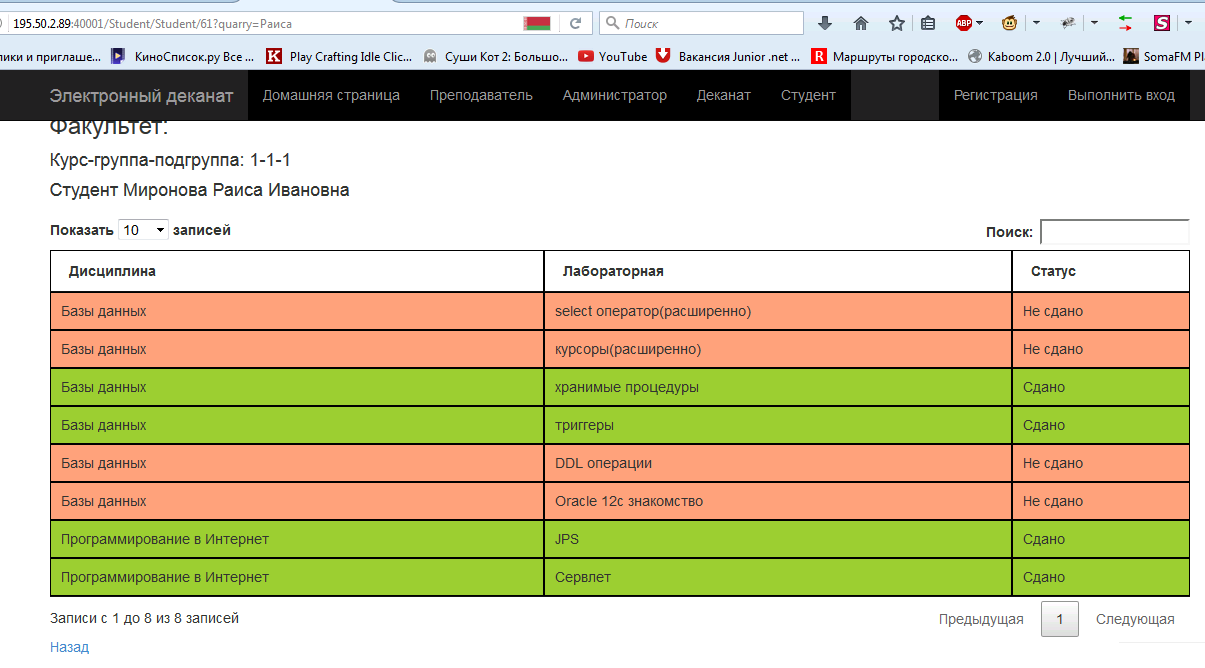


Рисунок 4.29 – Информация о студенте

## Деканат

Для деканата была создана учётная запись (Логин dekan@gmail.com, пароль 12345). Деканат может просматривать успеваемость, как по всем группам, так и по конкретным студентам. Страница, открывающаяся после авторизации, представлена на рисунке 4.30.

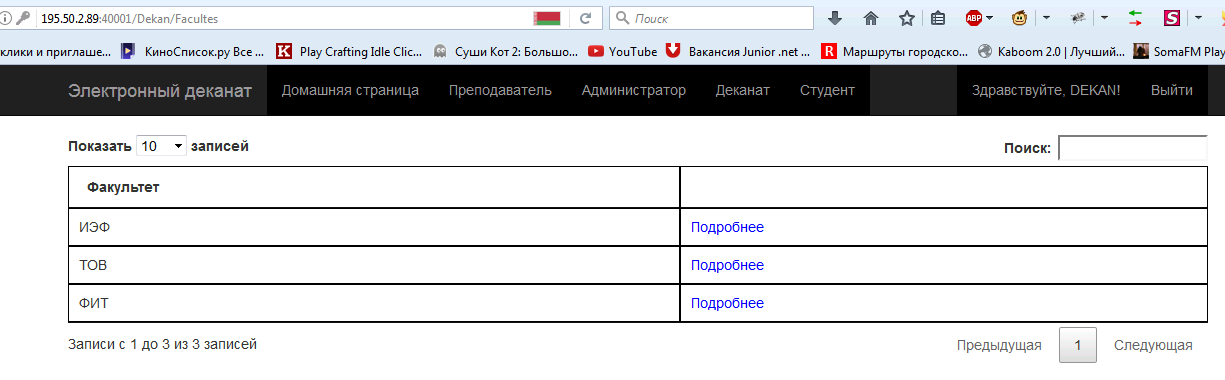


Рисунок 4.30– Деканат, список факультетов

На этой странице, можно выбирать факультет кнопкой «подробнее». Нажимаем на ссылку напротив факультета «ФИТ». Открывается страница, представленная на рисунке 4.31.

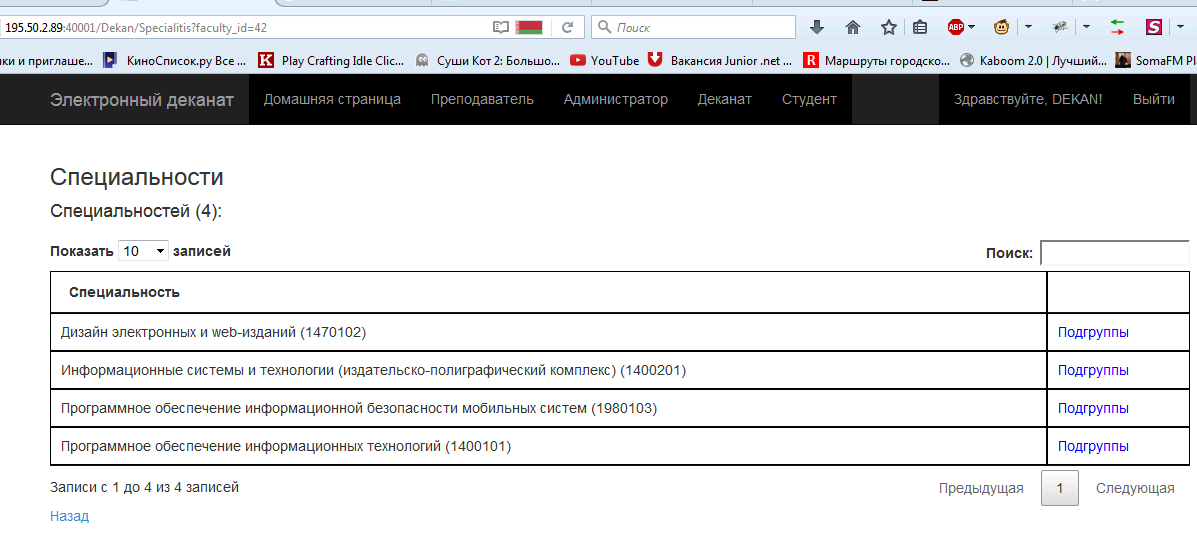


Рисунок 4.31 – Деканат, список специальностей

На этой странице, видны все специальности на факультете, нажимаем на «подгруппы» напротив специальности «ИСиТ». Открывается страница, представленная на рисунке 4.32.

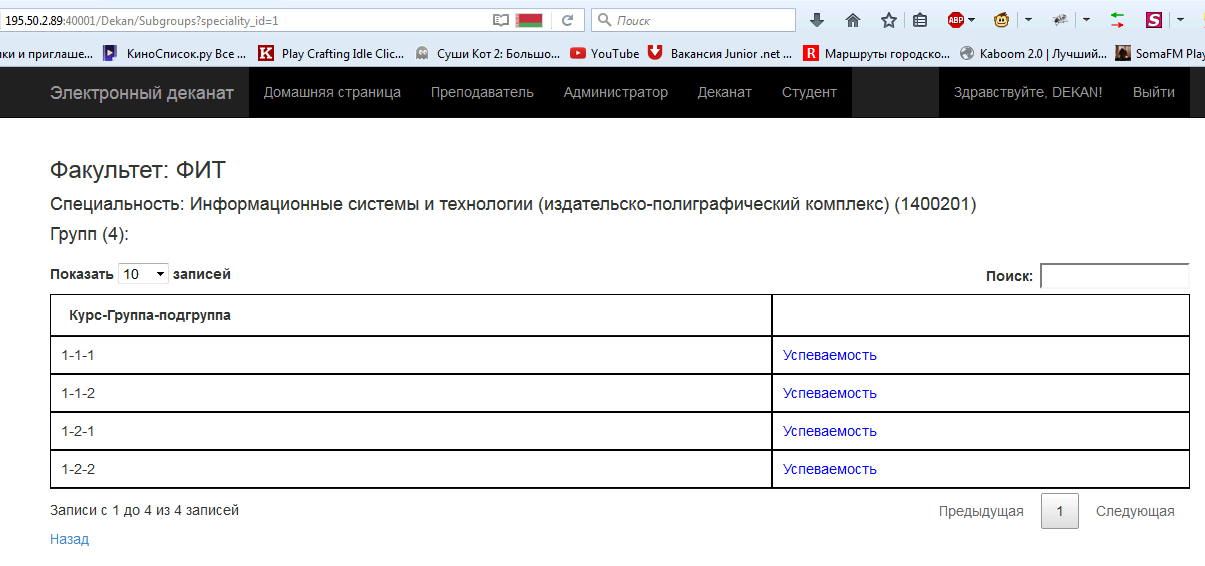


Рисунок 4.32 – Деканат, список групп

На этой странице, видны все подгруппы на специальности, нажимаем на «успеваемость» подгруппы 1-1-2. Открывается страница, представленная на рисунке 4.33.

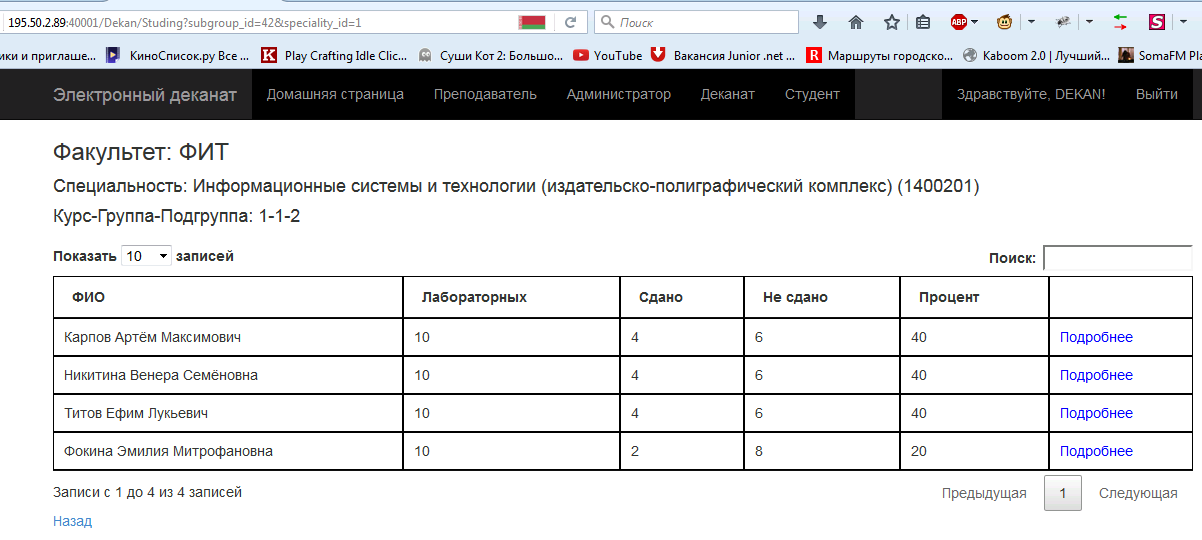


Рисунок 4.33 – Деканат, список успеваемости студентов

На этой странице, видны все студенты в выбранной подгруппе, а так же, суммарная успеваемость каждого студента, при необходимости посмотреть подробнее, нажимаем на «успеваемость» нужного студента в списке. Открывается страница, представленная на рисунке 4.34.

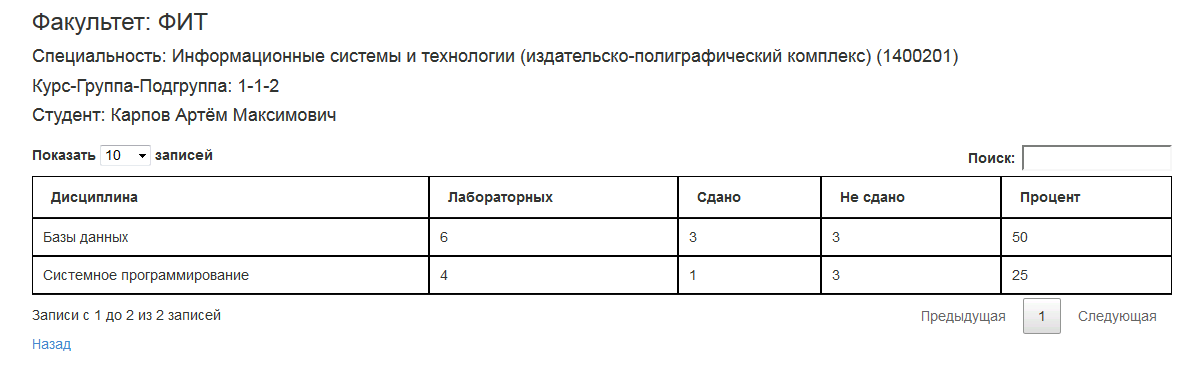


Рисунок 4.34– Деканат, список успеваемости студента

На этой странице мы видим успеваемость студента по каждой дисциплине по отдельности.

### Экономический раздел

Изм.

№ докум.

Подпись

Дата

Лист

1

ДП 05.00.ПЗ

Разраб.

Прокопович Д.В.

Провер.

Смелов В.В.

Консульт.

Смелов В.В.

.

Н. Контр.

Жиляк Н.А.

Утверд.

Смелов В.В.

# Экономический раздел

Лит.

Листов

1

*БГТУ 64419006,2017*

У

### Мероприятия по охране труда и безопасности жизнедеятельности

Изм.

№ докум.

Подпись

Дата

Лист

1

ДП 06.00.ПЗ

Разраб.

Прокопович Д.В.

Провер.

Смелов В.В.

Консульт.

Смелов В.В.

.

Н. Контр.

Жиляк Н.А.

Утверд.

Смелов В.В.

Мероприятия по охране труда и безопасности жизнедеятельности

Лит.

Листов

1

*БГТУ 64419006,2017*

У

## Анализ потенциально опасных и вредных производственных факторов

Эксплуатирующий средства вычислительной техники и периферийное оборудование персонал может подвергаться опасным и вредным воздействия, которые по природе действия подразделяются на следующие группы:

– физические: повышенные уровни электромагнитного излучения, статического электричества, запыленности воздуха рабочей зоны; повышенный или пониженный уровень освещенности рабочей зоны и др.;

– химические: содержание в воздухе рабочей зоны оксида углерода, озона, аммиака, фенола, формальдегида и полихлорированных фенилов;

– психофизиологические: напряжение зрения, памяти, внимания; длительное статическое напряжение; монотонность труда; нерациональная организация рабочего места; эмоциональные перегрузки;

– биологические: повышенное содержание в воздухе рабочей зоны микроорганизмов.

Труд работающих на ПЭВМ с использованием видеодисплейного терминала (ВДТ) относится к категории умственного труда [22].

Различают 3 группы деятельности: группа А – работа по считыванию информации с предварительным запросом; группа Б – работа по вводу информации; группа В – творческая работа в режиме диалога с ЭВМ. Для групп трудовой деятельности различают 3 категории тяжести и напряженности работы, которые описаны таблицей 6.1.

Таблица 6.1 – Категории работы ВДТ, ЭВМ и ПЭВМ

|  |  |  |  |
| --- | --- | --- | --- |
| Категория | Уровень нагрузки за рабочую смену при видах работ с ВДТ | | |
| группа А, количество знаков | группа Б, количество знаков | группа В, ч |
| I | до 20000 | до 15000 | до 2,0 |
| II | до 40000 | до 30000 | до 4,0 |
| III | до 60000 | до 40000 | до 6,0 |

Деятельность по созданию данного дипломного проекта можно отнести к группе В и категории III. Поскольку компьютер является электрическим устройством, то существует опасность поражения электрическим током

Опасность поражения электрическим током может возникнуть при прикосновении к открытым токоведущим частям с нарушенной изоляцией или к находящемуся под напряжением оборудованию при отсутствии или нарушении изоляции.

По степени опасности поражения электрическим током согласно правил устройства электроустановок (ПУЭ) рабочее помещение относится к классу помещений с повышенной опасностью, т. к. имеется возможность одновременного прикосновения человека к имеющим соединения с землей металлоконструкциям здания с одной стороны и металлическим корпусам электрооборудования с другой.

Несоблюдение мер пожарной безопасности, неисправное электрооборудование, может привести к пожару. Источниками зажигания могут быть электронные схемы от ЭВМ, приборы, применяемые для технического обслуживания, устройства электропитания, кондиционирования воздуха, где в результате различных нарушений образуются перегретые элементы, электрические искры и дуги, способные вызвать загорания горючих материалов.

## Инженерные решения по обеспечению санитарно–гигиенических условий труда

Выполняемые работы по энергозатратам относятся к категории 1а с наиболее низкими энергозатратами. Однако работа оператора ЭВМ характеризуется большим нервно–эмоциональным напряжением, в деле снижения которого не последнюю роль играют санитарно–гигиенические условия производственного помещения.

Для обеспечения нормальных условий труда в санитарных нормах устанавливают на одного работающего объем производственного помещения не менее 18 м3. Площадь машинного зала соответствует площади, необходимой по заводским техническим условиям для данного типа ЭВМ:

– высота зала под техническим полом до подвесного потолка 3 – 3,5 м;

– расстояние между подвесным и основным потолком при этом должно быть 0,5 – 0,8 м;

– габариты дверей машинного зала принимаются не менее 1,8 х1,1 м.

С целью обеспечения комфортных условий для обслуживающего персонала и надежности технологического процесса [23] устанавливают требования к микроклиматическим условиям, указанные в таблице 6.2.

Таблица 6.2 – Параметры воздушной среды на рабочих местах

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Период года | Категория работ | Параметры воздушной среды на рабочих местах | | |
| Температура, ºС | Относительная влажность, % | Скорость движения воздуха, м/с, не более |
| Холодный | Легкая – Iа | 22 – 24 | 40 – 60 | 0,1 |
| Теплый | Легкая – Iа | 23 – 25 | 40 – 60 | 0,1 |

Для поддержания нормального микроклимата необходим достаточный объем вентиляции, для чего в вычислительном центре предусматривается кондиционирование воздуха, осуществляющее поддержание постоянных параметров микроклимата в помещении независимо от наружных условий. В вычислительном центре установлены кондиционеры неполного кондиционирования марки БК-2500 (чистота воздуха, t, W).

Параметры микроклимата поддерживаются в указанных пределах в холодное время за счет системы водяного отопления с нагревом воды до 100º С, в теплый – за счет кондиционирования.

В дневное время в вычислительном центре используется естественное одностороннее освещение, в вечернее время или при недостаточных нормах освещения – искусственное общее равномерное.

Помещения для работы с дисплеями и видеотерминалами [24] относятся к I разряду по задачам зрительной работы. Нормированный уровень освещенности для работы с ЭВМ – 400 лк. Для работы в дневное время предусмотрено естественное освещение, боковое, одностороннее, отвечающее требованиям. Значение коэффициента естественного освещения (КЕО) должно быть не менее 1,5%.

В помещениях, оборудованных ЭВМ, предусматриваются меры для ограничения слепящего воздействия светопроемов, имеющих высокую яркость (8000 кд/м2 и более), и прямых солнечных лучей для обеспечения благоприятного распределения светового потока в помещении и исключения на рабочих поверхностях ярких и темных пятен, засветки экранов посторонним светом, а так же для снижения теплового эффекта от инсоляции. Это достигается путем соответствующей ориентации светопроемов, правильного размещения рабочих мест и использования солнцезащитных средств.

Требования к снижению зеркального отражения в экранах удовлетворяются путем использования светильников с комбинированным прямым и отраженным направлением света, которое осуществляется с помощью двойной крестовой оптики. Часть прямого светового потока лампы направляется через параболический зеркальный растр таким образом, что ограничивается слепящее действие прямого и отраженного света: отраженная часть излучения лампы направляется широким потоком на потолок.

Благодаря такому светораспределению в верхней полусфере яркость потолка в любом месте, в том числе и непосредственно под светильником, не превышает 200 кд/м2. Габаритная яркость светильников в зоне углов излучения более 50° от вертикали ограничивается в обеих плоскостях 200 кд/м2.

Для искусственного освещения помещений отдела разработки программного обеспечения следует использовать главным образом люминесцентные лампы белого света (ЛБ) мощностью 40 или 80 Вт.

Шум ухудшает условия труда оказывая вредное действие на организм человека. При выполнении основной работы на ПК уровень шума на рабочем месте не должен превышать 50 дБА.

Снизить уровень шума в помещениях с ВДТ, ЭВМ и ПЭВМ можно использованием звукопоглощающих материалов для отделки помещений.

Уровень вибрации в помещениях вычислительных центров может быть снижен путем установки оборудования на специальные виброизоляторы.

В производственных помещениях, в которых работа с ВДТ, ЭВМ и ПЭВМ является основной, вибрация на рабочих местах не должна превышать допустимых значений, приведенных в таблице 6.3.

Таблица 6.3 – Допустимые уровни вибрации

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Среднегеометрические частоты октавных полос, Гц | Допустимые значения | | | |
| по виброускорению | | по виброскорости | |
| м/с2 | дБ | м/с | дБ |
| оси X, Y, Z | | | |
| 2 | 5,3х10-3 | 25 | 4,5х10-4 | 79 |
| 4 | 5,3х10-3 | 25 | 2,2х10-4 | 73 |
| 8 | 5,3х10-3 | 25 | 1,1х10-4 | 67 |
| 16 | 1,1х10-2 | 31 | 1,1х10-4 | 67 |
| 31,5 | 2,1х10-2 | 37 | 1,1х10-4 | 67 |
| 63 | 4,2х10-2 | 43 | 1,1х10-4 | 67 |
| Корректированные значения и их уровни в дБ | 9,5х10-3 | 30 | 2,0х10-4 | 72 |

Уровни положительных и отрицательных аэроионов в воздухе помещений с ВДТ, ЭВМ и ПЭВМ должны соответствовать нормам, приведенным в таблице 6.4.

Таблица 6.4 – Уровни ионизации воздуха помещений при работе на ВДТ, ЭВМ и ПЭВМ

|  |  |  |
| --- | --- | --- |
| Уровни | Число ионов в 1 см3 воздуха | |
| n+ | n– |
| Минимально необходимые | 400 | 600 |
| Оптимальные | 1500 – 3000 | 3000 – 5000 |
| Максимально допустимые | 50000 | 50000 |

Эксплуатация вычислительной техники связана с применением электриче­ской энергии. Опасность поражения электрическим током возникает при прикосновении к открытым токоведущим частям с нарушенной изоляцией или к оборудованию, находящемуся под напряжением при отсутствии или нарушении изоляции. По степени поражения людей электрическим током вычислительный центр относится к классу помещений без повышенной опасности.

При работе с дисплеем возникают следующие опасные факторы: электромагнитные поля и электростатические поля.

Основным способом защиты от статического электричества в машинных залах является заземление оборудования.

Воздействие электромагнитных полей на человека зависит от напряженностей электрического и магнитного полей, потока энергии, частоты колебаний, размера облучаемой поверхности тела и индивидуальных особенностей организма.

В диапазоне частот от 60 кГц до 300 МГц интенсивность электромагнитных полей характеризуется напряженностью электрических (Е) и магнитных (Н) полей. На рабочих местах персонала в течение рабочего дня напряженность не должна превышать установленных предельно допустимый уровень (ПДУ), приведенным в таблице 6.5.

Таблица 6.5– ПДУ электромагнитных излучений на рабочих местах

|  |  |
| --- | --- |
| По электрической составляющей, А/м | Диапазон |
| 50 | 60 кГц – 3 МГц |
| 20 | 3 МГц – 30 МГц |
| 10 | 30 МГц – 50 МГЦ |
| 5 | 50 МГЦ – 300 МГЦ |
| По магнитной составляющей, В/м | Диапазон |
| 5 | 60 кГЦ – 1,5 МГц |
| 0,3 | 3 МГц – 50 МГц |

Наиболее эффективным и часто применяемым из названных методов защиты от электромагнитных излучений является установка экранов. Экранируют либо источник излучения, либо рабочее место. Экраны бывают отражающие и поглощающие. Индивидуальные экранирующие комплексы предназначены для защиты от воздействия электростатического поля, напряженность которого не превышает 60 кВ/м.

Для обеспечения безопасности работ с источниками электромагнитных волн (в данном случае с ПЭВМ) производится систематический контроль фактических значений нормируемых параметров на рабочих местах. Контроль осуществляется измерением напряжения электрического и магнитного полей, а также измерением плотности потока энергии. В целях предосторожности ограничивается продолжительность работы с экраном персонального компьютера. ЭВМ размещают концентрированно в рабочей зоне и выключают их, если на них не работают, используют защитные экраны.

## Инструкция по безопасной эксплуатации ЭВМ

Организация рабочих мест, труда и отдыха пользователей средств вычислительной техники (СВТ) осуществляется в соответствии с СанПиН 9-131 РБ 2000 [36].

Проведение уборки (в т.ч. влажной) рабочего места допустимо только после обесточивания устройств ЭВМ.

Пользователи ЭВМ обязаны выполнять только ту работу, которая им пору­чена руководителем работ.

Площадь на 1 рабочее место пользователей ЭВМ должна составлять не менее 6 м2.

Для обеспечения оптимальной работоспособности и сохранения здоровья пользователей устройств ЭВМ при 8–ми часовой рабочей смене должны устанав­ливаться регламентированные перерывы [25]:

– для I категории работ: через 2 часа от начала рабочей смены и через 2 часа после обеденного перерыва – 15 минут каждый;

– для II категории работ: через 2 часа от начала рабочей смены и через 1,5 – 2 часа после обеденного перерыва – 15 минут каждый или 10 минут через каждый час работы;

– для III категории работ: через 1,5 – 2 часа от начала рабочей смены и через 1,5 – 2 часа после обеденного перерыва – 20 минут каждый или 15 минут через

каждый час работы.

Профессиональные пользователи ЭВМ должны проходить обязательные предварительные (при поступлении на работу) и периодические медицинские осмотры в порядке и в сроки, установленные постановлением Минздрава республики Беларусь №33от 8.08.2000 г.

Женщины со времени установления беременности и в период кормления ребенка грудью к выполнению всех видов работ на ЭВМ не допускаются.

Запрещается появляться на рабочем месте в состоянии алкогольного, нарко­тического, токсического опьянения, приносить и распивать спиртные напитки.

### 6.3.1 Требования по охране труда перед началом работы на ЭВМ

Удостовериться (визуальным осмотром) в том, что СВТ находится в рабочем состоянии, т.е. кожуха устройств одеты, нет запрещающих плакатов, рабочее место укомплектовано необходимыми устройствами. В случае сомнения в исправности СВТ следует получить информацию у ответственного специалиста отдела информационных технологий (ИТ–отдела).

Включение ПЭВМ проводят в следующем порядке: все периферийные устройства (монитор, принтер и т.д.); системный блок.

После включения тумблера (кнопки) "Сеть" на системном блоке необходимо убедиться в работе вентилятора – по звуку или потоку воздуха, исходящего от задней панели блока.

Кнопки и тумблеры "Сеть" допускается включать только руками, но не ручками, карандашами и т.д. Включение проводится плавным нажатием на кнопку до ее фиксации.

Настроить монитор согласно индивидуальным особенностям зрения (яркость, контрастность). При индивидуальной настройке следует избегать выхода устройства за границы нормального режима, т.е. появления излишней яркости экрана, полос и линий, появления гудения и т.д. В таком случае необходимо вернуться к предыдущей настройке.

Категорически запрещается включать СВТ, сетевое коммуникационное оборудование, на которых установлены запрещающие плакаты или иные предупредительные надписи установленного образца. Снимать указанные плакаты и допускать пользователей к работе на этих рабочих местах разрешено только лицу, установившему запрещающий плакат.

Личные (или находящиеся в его ведении) носители информации пользователь обязан хранить в чистоте, при заданных климатических условиях и согласно техническим требованиям на данный вид носителя. Все носители должны быть документированы и датированы.

Запрещаются несанкционированные перенастройки CMOS и BIOS систем.

### 6.3.2 Требования по охране труда во время работы на ЭВМ

В процессе работы пользователь обязан соблюдать в рабочем помещении тишину и порядок.

Во время работы пользователь СВТ должен контролировать техническое состояние устройств: устойчивость изображения на мониторе, цветовую палитру, реакцию устройств на команды, работу вентиляторов охлаждения СВТ и других устройств, отсутствие нехарактерного шума и вибрации, признаков возгорания (запах, дым и т.д.).

При нарушении нормальной работы устройства следует немедленно прекратить работу, выключить устройство и сообщить о неисправности руководителю работ и техническому персоналу ИТ–отдела.

Пользователям СВТ запрещается:

– оставлять на верхней панели монитора и процессорного блока любые предметы, бумагу, книги и т.д., тем самым перекрывать воздушные потоки воздухообмена компьютера;

– прикасаться к экрану или приклеивать к экрану предметы, которые могут повредить экран;

– производить любые ремонты устройств (подключение к сети, коммутация сигнальных проводов и т.д.), а также перемещать оборудование без контроля технических специалистов ИТ–отдела.

Пользователь работает только с известными ему программами и данными.

Ему запрещены все работы с потенциально опасными программами, которые могут повлечь разрушение данных, а также с неизвестными ему программами. При появлении неизвестных сообщений от ранее используемых программ пользователь обязан задокументировать свои действия и сообщения программы и передать эти материалы техническому персоналу ИТ–отдела.

Организация копирования данных текущего сеанса, а также контроль за архивом возлагается на пользователя. Пользователь обязан контролировать свои действия по обновлению (стиранию) информации и в случае порчи информации объективно рассказать ответственному лицу о своих действиях.

Пользователю запрещены любые работы, связанные с изменением информации MBR (загрузочная запись) дисковых разделов и разделов логических дисков.

Пользователь обязан хранить в секрете свой пароль для работы в сети, не допускать ситуации, позволяющей воспользоваться его правами доступа к сети, передавать право работы под своим именем. Работа в сети с использованием чужого права доступа считается нарушением технологической дисциплины.

Строго соблюдать правила антивирусной безопасности:

– запрещается самостоятельно устанавливать любые программные системы (в том числе и игровые);

– при поступлении любой информации извне, необходимо поставить в известность ответственных лиц в ИТ–отделе, также совместно провести антивирусный контроль принесенных носителей;

– проводить антивирусный контроль ежедневно с помощью программы-сканера. В случае аварийного сообщения такой программы немедленно прекратить работу и сообщить в ИТ–отдел. Работникам запрещается препятствовать работе программы–сканера.

Обо всех особенностях, странном поведении обычных программ немедленно ставить в известность администратора сети или ответственное лицо. В случае появления на мониторе неизвестных или провокационных сообщений (на русском или иных языках) неизвестного происхождения записать их, прекратив работу, оповестить сотрудников ИТ–отдела.

На протяжении рабочего дня запрещается обесточивать сетевое коммуникационное оборудование. Каждое такое устройство снабжается соответствующей табличкой с напоминанием.

Пользователь обязан пресекать любые попытки распространения коммерческих программ путем несанкционированного их копирования или нарушения лицензионного соглашения на ПС (программные средства).

Запрещается несанкционированная или негласная перенастройка программных продуктов совместного пользования, установленных на СВТ подразделений.

При необходимости оставить рабочее место на непродолжительный срок пользователь обязан принять меры к уменьшению выгорания люминофора экрана (снизить яркость, перевести монитор в менее напряженный режим).

При возникновении сбоев принтеров (или графопостроителей), связанных с замятием бумаги или красящей ленты, обратиться к ответственному лицу, отвечающему за исправность техники.

Категорически запрещается перемещать системный блок СВТ во включенном состоянии, изменять расположение рабочих мест (столов), оборудованных СВТ при работе последних, прилагать чрезмерные усилия при работе с клавиатурой и устройством типа "Мышь", а также принимать пищу и курить на рабочем месте (так как это ведет к засорению этих устройств).

Запрещается помещать в дисководы какие–либо предметы, кроме дискет.

Не рекомендуется оставлять дискеты в дисководах после окончания работы. При затруднениях с извлечением малоформатных (3,5) дискет запрещено применять усилия. Если дискета свободно не удаляется, то это свидетельствует о неисправности. В этом случае необходимо обратиться к инженерам ИТ–отдела.

Запрещена эксплуатация дискет, имеющих видимые физические дефекты: царапины, задиры, изгибы и т.д., что приводит к порче дисковода.

Категорически запрещается любая реконфигурация СВТ во включенном состоянии. Запрещается растягивать кабели клавиатуры, изгибать и деформировать провод мыши. Пользователям запрещается наносить на корпуса устройств СВТ надписи или иные отметки. Они обязаны выполнять распоряжения инженерного персонала, обслуживающего сетевую инфраструктуру предприятия, относительно правил работы на СВТ при проведении срочных ремонтных или штатных профилактических работ в сети.

### 6.3.3 Требования по охране труда по окончанию работы на ЭВМ

По окончании работы на ПК пользователь обязан отключится от сети, завершить все текущие программы.

По окончании работы (или при длительном перерыве) пользователь выключает ПК и сетевое оборудование. Выключения ПК проводят в порядке, обратном порядку включения, запрещается обесточивать сетевое коммуникационное оборудование ранее, чем истечет официальное рабочее время предприятия. Контроль за соблюдением данного положения возлагается на старшего по должности в соответствующем помещении.

## Требования по охране труда в аварийных ситуациях

При возникновении аварийной ситуации (внезапное отключение питания, короткое замыкание, появление дыма или специфического запаха горящей изоляции и т.п.) необходимо немедленно отключить все устройства СВТ и доложить о случившемся руководителю работ и техническим специалистам.

При возникновении пожара в помещении расположения СВТ необходимо обесточить оборудование, вызвать пожарную службу по тел. 101, оповестить руководство подразделения и приступить к тушению возгорания имеющимися средствами пожаротушения: пожарный гидрант, огнетушители и т.д.

## Организация и безопасность работы на ПЭВМ

Обычно видеотерминалы имеют экраны размером 25х20 см: размер экрана по диагонали предусмотрен не менее 38 см. Однако экраны большого размера (68 см по диагонали) обладают рядом недостатков: повышенная возможность появления бликов, деформация знаков на периферии экрана, сложность размещения экрана в нормальном поле зрения.

Предпочтительным является плоский экран, позволяющий избегать наличия на нем ярких пятен за счет отражения световых потоков. Необходимо избегать насыщенных цветов, особенно красного, синего, ярко–зеленого и т.д.

Для экрана допустимы ненасыщенные тона: светло–зеленый, желто–зеленый, желто–оранжевый, желто–коричневый. Желательно с раздельной регулировкой яркости фона и информации.

В отношении количества передаваемых цветов следует придерживаться монохроматического изображения, при передаче графической информации можно использовать четырехцветное изображение.

О качестве экрана судят также по отсутствию мерцания и постоянству яркости, независимо от характера передаваемой информации при прямом контрасте (темные сигналы на светлом фоне). Частота мельканий предусмотрена не менее 80 Гц.

Оптимальная высота расположения экрана соответствует направлению взгляда оператора в секторе 5 – 35º. по отношению к вертикали. Большой наклон экрана может привести к появлению бликов от светильников.

Условия зрительного восприятия информации на экране зависят от параметров знаков, плотности их размещения, контраста, соотношения яркости символов и фона экрана. При расстоянии от глаз до экрана 60 – 80 см высота знака предусмотрена не менее 3 – 4 мм, а оптимальное соотношение ширины и высоты составляет 3:4, расстояние между знаками 15 – 20% от его высоты.

Нижней комфортной границей уровня яркости светящихся символов (знаков) считается 30 кд/(м2), верхняя комфортная граница определяется значением следящей яркости. Яркость символа на экране согласуется с яркостью фона экрана и окружающим освещением.

При прямом контрасте яркостный контраст составляет 75 – 80 % с возможностью регулировки яркости фона экрана, а при обратном контрасте (светлые символы на темном фоне) — 85 – 90% с возможностью регулирования яркости знака. Коэффициент контрастности считается благоприятным в пределах 5 – 10 для обратного контраста и в пределах 8 – 12 для прямого.

Для обеспечения оптимальной работоспособности и сохранения здоровья пользователей устройств ЭВМ при 8–ми часовой рабочей смене для I категории работ должны устанавливаться следующие регламентированные перерывы: через 2 часа от начала рабочей смены и через 2 часа после обеденного перерыва – 15 минут каждый.

## Технические решения, обеспечивающие взрыво– и пожаробезопасность объекта

Одной из важных задач пожарной профилактики является защита строительных конструкций от разрушений и обеспечение их достаточной прочности в условиях воздействия высоких температур при пожаре.

Комнаты отдела разработки программного обеспечения и части зданий другого назначения, в которых предусмотрено размещение ЭВМ, относятся к 1 или 2 степени огнестойкости [27]. Для изготовления строительных конструкций используют, как правило, кирпич, железобетон, стекло и другие негорючие материалы.

Для предотвращения распространения огня во время пожара с одной части здания на другую устраивают противопожарные преграды в виде стен, перегородок, дверей, окон, люков, клапанов.

Двери устанавливают металлические или деревянные, обитые листовым железом по войлоку, смоченному раствором глины, или асбесту.

Особое требование предъявляется к устройству и размещению кабельных коммуникаций.

Для ликвидации пожаров в начальной стадии применяются первичные средства пожаротушения: внутренние пожарные водопроводы, огнетушители типа ОУ‑2, асбестовые одеяла и др.

## Мероприятия по безопасности жизнедеятельности

В соответствии с Законом Республики Беларусь «О защите населения и территорий от чрезвычайных ситуаций природного и техногенного характера», чрезвычайная ситуация – это обстановка, сложившаяся в результате аварии, катастрофы, стихийного или иного бедствия, которые повлекли или могут повлечь за собой человеческие жертвы, вред здоровью людей или окружающей среде, значительные материальные потери и нарушение условий жизнедеятельности людей.

Предупреждение чрезвычайных ситуаций – комплекс мероприятий, проводимых заблаговременно и направленных на максимально возможное уменьшение риска возникновения чрезвычайных ситуаций, а также на сохранение здоровья людей, снижение размеров вреда окружающей среде и материальных потерь в случае их возникновения.

Зона чрезвычайной ситуации – территория, на которой возникла чрезвычайная ситуация.

Ликвидация чрезвычайных ситуаций – аварийно–спасательные и другие неотложные работы, проводимые при возникновении чрезвычайных ситуаций и направленные на спасение жизни и сохранение здоровья людей, снижение размеров вреда окружающей среде и материальных потерь, а также на локализацию зон чрезвычайных ситуаций.

По характерам происхождения ситуации, которые могут обусловить возникновение чрезвычайной ситуации (ЧС) на территории Республики Беларусь, различают:

ЧС техногенного характера – транспортные аварии (катастрофы), пожары, неспровоцированные взрывы или их угроза, аварии с выбросом (угрозой выброса) опасных химических, радиоактивных, биологических веществ и т.д.;

ЧС природного характера – опасные геологические, метеорологические, гидрологические явления, деградация грунтов или недр, природные пожары, изменение состояния воздушного бассейна, инфекционная заболеваемость людей, сельскохозяйственных животных, массовое поражение растений и др.

### 6.7.1 Анализ потенциально опасных источников возникновения ЧС

Университет БГТУ находится по адресу ул. Свердлова, 13а. Университет готовит инженеров-программистов, web-дизайнеров, редакторов-технологов, инженеров-технологов, инженеров-экономистов, менеджеров, маркетологов, инженеров по стандартизации и сертификации, инженеров-экологов, специалистов по автоматизации процессов производств и в сфере туризма, инженеров-энергоменеджеров, инженеров-механиков; специалистов для следующих отраслей промышленности: химической и нефтехимической; фармацевтической и целлюлозно-бумажной; строительных материалов; ландшафтной архитектуры, лесной и деревообрабатывающей, лесному хозяйству и др.. С наружной стороны здание оштукатурено. Внутренние стены между помещениями здания панельной кладки, так же оштукатурены. Водоснабжение осуществляется из городской водопроводной сети. Электроснабжение по городской сети электропередач. В непосредственной близости от здания предприятия находятся жилые дома, административные здания, транспортные коммуникации.

Существует вероятность возникновения пожара в помещениях университета. Причинами возгорания могут быть короткие замыкания в электрических сетях, курение и разведение огня в неположенных местах, несанкционированное пользование электронагревательными и другими бытовыми электроприборами.

Помимо пожара в процессе повседневной деятельности могут возникнуть следующие чрезвычайные ситуации: бури, проливные дожди, авария на АЭС, аварии на промышленных предприятиях и др. Все рассмотренные факторы могут повлечь за собой повреждение или уничтожение здания, материальных ценностей, оборудования, травмирование людей.

Последствия стихийных бедствий могут быть существенно уменьшены или предотвращены при прогнозировании данных бедствий и их последствий, своевременного предупреждения населения.

Для предотвращения производственных аварий руководители предприятий разрабатывают и осуществляют организационно технические мероприятия. Обязательное условие безаварийной работы предприятия – строгое соблюдение требований техники безопасности. Также существует необходимость осуществления систематических проверок состояния самого здания, сооружений и оборудования.

### 6.7.2 Мероприятия, направленные на предотвращение потерь персонала от возникновения ЧС

Для предотвращения производственных аварий и катастроф разрабатывается и осуществляется комплекс мероприятий по обеспечению безопасности жизнедеятельности персонала объекта хозяйствования и населения, который включает в себя два направления.

Первое направление состоит в разработке технических и организационных мероприятий, уменьшающих вероятность реализации опасного поражающего потенциала современных технических систем. Второе направление содержит комплекс мероприятий по защите обслуживающего персонала, формирований гражданской обороны и населения при возникновении ЧС природного и техногенного характера.

Большое значение имеет своевременное оповещение людей о произошедшей аварии. Для этого создаются системы оповещения, организуются внутренняя система оповещения и связи. Наличие подобных систем имеет большое значение, так как они позволяют в минимально короткие сроки предупреждать работающих об аварии. Ответственность за организацию оповещения населения возлагается на штабы гражданской обороны объектов и районов созданных при отмеченных объектах и районах.

При возникновении пожаров необходима срочная эвакуация людей из районов, которым угрожает опасность. Действия людей, оказавшихся в очаге пожара, должны быть строго подчинены указаниям личного состава пожарных команд.

При невозможности выхода через неохваченные огнем помещения необходимо накрыться с головой одеялом, пальто, куском ткани, облиться водой и продвигаться очень быстро через уже горящее помещение.

Выйдя на улицу, загоревшееся покрывало немедленно сбрасывают на землю и тушат его. Наиболее доступными средствами тушения загораний и пожаров является вода, песок, земля, ручные огнетушители, пожарные краны, гидропульты, асбестовые и брезентовые покрывала.

При минировании здания необходима эвакуация людей из зоны возможного взрыва, приводятся в готовность органы управления, охраны общественного порядка, медицинская служба. Специализированные невоенизированные формирования оперативно должны проверить наличие в здании источника взрыва и обезвредить его.

При возникновении ЧС решается целый комплекс задач по ликвидации их последствий: ведется разведка, выявляются объекты и населенные пункты, которым непосредственно угрожает опасность, определяется группировка сил, средств для предотвращения и локализации угрожающей опасности, производится спасение людей, материальных ценностей, оказание медицинской помощи пострадавшим, организовывается комендантская служба для обеспечения порядка, восстанавливается жизнедеятельность населенных пунктов.

# Заключение

Изм.

№ докум.

Подпись

Дата

Лист

1

ДП 00.00.ПЗ

Разраб.

Прокопович Д.В.

Провер.

Смелов В.В.

Консульт.

Смелов В.В.

.

Н. Контр.

Жиляк Н.А.

Утверд.

Смелов В.В.

# Заключение

Лит.

Листов

1

*БГТУ 64419006,2017*

У

# Список использованных источников

1. Кафедра Интеллектуальных Информационных Технологий ИнФОУрФУ[Электронный ресурс] – Режим доступа:<http://lecturesdb.readthedocs.io/databases/dbms.html>. Дата доступа: 15.04.2017.
2. Частное Боровское исследовательское учреждение по внедрению новых технологий [Электронный ресурс] – Режим доступа:<http://bourabai.ru/dbt/servers/oracle2.htm>. Дата доступа: 15.04.2017.
3. Абитуриент БГТУ[Электронный ресурс] – Режим доступа:<https://abiturient.belstu.by/foreign-abiturient.html>. Дата доступа: 15.04.2017.
4. Википедия [Электронный ресурс] – Режим доступа:<https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio>. Дата доступа: 15.04.2017.
5. MozillaDeveloperNetwork [Электронный ресурс] – Режим доступа:<https://developer.mozilla.org/ru/docs/Learn/Что_такое_веб_сервер>. Дата доступа: 15.04.2017.
6. Википедия [Электронный ресурс] – Режим доступа:<https://ru.wikipedia.org/wiki/Microsoft_SQL_Server>. Дата доступа: 15.04.2017.
7. HyperHost хостинг [Электронный ресурс] – Режим доступа:<https://hyperhost.ua/info/?p=3007>. Дата доступа: 15.04.2017.
8. Википедия [Электронный ресурс] – Режим доступа:<https://ru.wikipedia.org/wiki/Сервер_приложений>.Дата доступа: 15.04.2017.
9. developerWorks[Электронный ресурс] – Режим доступа:<https://www.ibm.com/developerworks/ru/library/wa-mean1/>. Дата доступа: 15.04.2017.
10. Википедия [Электронный ресурс] – Режим доступа:<https://ru.wikipedia.org/wiki/Spring_Framework>. Дата доступа: 15.04.2017.
11. WebCreator [Электронный ресурс] – Режим доступа:<https://web-creator.ru/articles/dot_net_and_asp>.Дата доступа: 15.04.2017.
12. РегФармПатент[Электронный ресурс] – Режим доступа:<http://regpharmpat.by/ru/patentnye-issledovaniya>. Дата доступа: 15.04.2017.
13. Хабрахабр[Электронный ресурс] – Режим доступа:<https://habrahabr.ru/post/202018/>.Дата доступа: 15.04.2017.
14. metanit.com[Электронный ресурс] – Режим доступа:<https://metanit.com/sharp/entityframework/1.1.php>. Дата доступа: 15.04.2017.
15. Справочник по паттернам проектирования [Электронный ресурс] – Режим доступа:<http://design-pattern.ru/patterns/repository.html>. Дата доступа: 15.04.2017.
16. dedushka.org[Электронный ресурс] – Режим доступа:<http://dedushka.org/uroki/6901.html>. Дата доступа: 15.04.2017.
17. Википедия [Электронный ресурс] – Режим доступа:<https://ru.wikipedia.org/wiki/JQuery>. Дата доступа: 15.04.2017.
18. htmlhook.ru[Электронный ресурс] – Режим доступа:<https://htmlhook.ru/rabota-s-jquery-datatables.html>. Дата доступа: 15.04.2017.
19. Федеральный институт промышленной собственности [Электронный ресурс] – Режим доступа:<http://www1.fips.ru>. Дата доступа: 15.04.2017.
20. webdom.net[Электронный ресурс] – Режим доступа:<http://www.webdom.net/inf/asp>. Дата доступа: 15.04.2017.
21. orastack.narod.ru[Электронный ресурс] – Режим доступа:<http://orastack.narod.ru/index_file/onebd.html>. Дата доступа: 15.04.2017.
22. Здания, строительные конструкции, материалы и изделия. Правила пожарно-технической классификации: ТКП 45–2.02–142–2011. – Введ. 01.01.10. – Минск: Министерство архитектуры и строительства Республики Беларусь, 2010. – 110 с.
23. Гигиенические требования к микроклимату производственных помещений: СанПиН № 9–80–98. – Введ. 01.07.98. – Минск: Министерство здравоохранения Республики Беларусь, 1998. – 12 с.
24. Общие санитарно–гигиенические требования к воздуху рабочей зоны: ГОСТ 12.1.005–88 ССБТ. – Введ. 01.01.89. – Минск: Государственный комитет по стандартам, 1989. – 52 с.
25. Гигиенические требования к видеодисплейным терминалам, электронно-вычислительным машинам и организации работ: СанПиН № 9–131 РБ 2000. – Введ. 10.11.00. – Минск: Министерство здравоохранения Республики Беларусь, 2000. – 32 с.
26. Естественное и искусственное освещение. Строительные нормы проектирования: ТКП 45–2.04–153–2009. – Введ. 01.01.10. – Минск: Министерство архитектуры и строительства Республики Беларусь, 2010. – 106 с.
27. Электробезопасность. Защитное заземление, зануление: ГОСТ 12.1.030–81 ССБТ. – Введ. 01.07.82. – Минск: Государственный комитет по стандартам, 1982. – 16 с.