

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

Направление подготовки: 09.03.01 «Информатика и вычислительная техника»

Профиль подготовки: «Программное обеспечение средств ВТ и автоматизированных систем»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

БАКАЛАВРА

на тему

Система учета пользователей ЛВС ПГУ.

Поиск и редактирование

Студент

 03.06.2020 г.
(подпись, дата)

Духанов Илья Валерьевич
(ФИО полностью)

Руководитель

 05.06.20
(подпись, дата)

К.В. Попов
(фамилия, инициалы)

Нормоконтролёр

 (подпись, дата)

А.В. Кучин
(фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 03.06.2020 г. № 12)

Заведующий кафедрой

 (подпись)

М.А. Митрохин
(фамилия, инициалы)

Работа защищена с отметкой отлично (протокол заседания ГЭК от 11.06.2020 № 4)

Секретарь ГЭК

 (подпись)

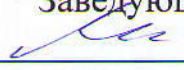
Е.И. Гурин
(фамилия, инициалы)

Пенза 2020

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
 М.А. Митрохин
«30» апреля 2020 г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ
РАБОТУ БАКАЛАВРА**

1. Студент Духанов Илья Валерьевич гр.16ВВ2 факультета вычислительной техники
2. Тема работы: «Система учета пользователей ЛВС ПГУ. Поиск и редактирование»

Тема утверждена приказом ПГУ № 0335 а/о от " 30 " апреля 20 20 г.

3. Руководитель выпускной квалификационной работы бакалавра к.т.н., доцент кафедры «ВТ» Попов К.В.

4. Задание на работу (назначение разработки, исходные данные и т.п.)

Назначение разработки - приложение предназначено для учета пользователей ЛВС ПГУ.

Разработать систему запросов, позволяющих взаимодействовать с базой данных, связанной с веб-интерфейсом.

5. Перечень подлежащих разработке вопросов.

Требуется связать веб-интерфейс и базу данных для системы, осуществляющей учет пользователей ЛВС ПГУ, а также обеспечить возможность поиска, добавления, редактирования и удаления ее элементов. База данных будет хранить в себе данные о строениях, подразделениях, аудиториях и ip-адресах университета.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Анализ предметной области	10	30.04.2020 – 03.05.2020	
2.	Выбор средств разработки	10	03.05.2020 – 05.05.2020	
3.	Изучение основных характеристик СУБД	10	05.05.2020 – 08.05.2020	
4.	Изучение языка php	10	08.05.2020 – 11.05.2020	
5.	Разработка SQL-запросов	15	11.05.2020 – 16.05.2020	
6.	Внедрение поисковой логики в веб-интерфейс	20	16.05.2020 – 31.05.2020	
7.	Оформление выпускной работы	15	31.05.2020 – 03.06.2020	

Дата выдачи задания " 30 " апреля 2020 г.

Руководитель выпускной квалификационной работы бакалавра


(подпись, дата)

Задание к исполнению принял студент

 30.04.2020 Духанов И.В.
(подпись, дата)

Работу к защите допустить

Декан факультета


(подпись, дата)

РЕФЕРАТ

Пояснительная записка содержит 75 листов, 42 рисунка, 16 источников, 3 приложения.

PHPMYADMIN, БАЗА ДАННЫХ, BOOTSTRAP, MAMP, СУБД MYSQL, SQL, ПОИСКОВАЯ СИСТЕМА, SQL-ЗАПРОСЫ.

В рамках данной выпускной квалификационной работы были созданы функции, входящие в состав веб-приложения и включающие в себя запросы к базе данных - MYSQL, спроектированной с использованием phpMyAdmin.

Объект исследования - база данных компьютерного фонда ПГУ.

Предмет исследования - особенности составления алгоритмов работы с базой данных на базе платформы MYSQL.

Цель выпускной квалификационной работы - разработка и внедрение полученной системы учета пользователей ЛВС ПГУ, и реализация этого проекта с использованием базы данных – MYSQL.

Задача выпускной квалификационной работы есть разработка функций для работы с базой данных учета компьютерного фонда университета, в рамках веб-приложения.

В результате выпускной квалификационной работы было реализовано и протестировано приложение, удовлетворяющее поставленной задаче.

Графическая часть включает в себя: структуру взаимодействия компонентов веб-приложения; UML – диаграммы последовательности и модель базы данных.

Разработка проводилась посредством использования компонентов локального веб-сервера MAMP. Использовались языки: программирования PHP и JavaScript, структурированных запросов SQL, гипертекстовой разметки HTML, каскадных таблиц стилей CSS. Для хранения данных выбрана СУБД MySQL, для ее администрирования применялся инструмент phpMyAdmin.

					ПГУ 2.090301.11.001 ПЗ							
Изм.	Лист	№ докум.	Подпись	Дата	Система учета пользователей ЛВС ПГУ. Поиск и редактирование. Пояснительная записка			Лит.	Лист	Листов		
Разраб.		Духанов И.В.								4	75	
Провер.		Попов К.В.						ФВТ, каф. ВТ гр. 16ВВ2				
Н. Контр.		Кучин А.В.										
Утверд.		Митрохин М.А.										

СОДЕРЖАНИЕ

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ	7
ВВЕДЕНИЕ	8
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	9
1.1 Постановка задачи	9
1.2 Анализ существующих разработок в предметной области	10
1.2.1 Nagios	10
1.2.2 Icinga	11
1.2.3 NeDi	11
1.2.4 Ntop	12
1.2.5 Zabbix	12
1.2 Базы данных	13
1.4 Реляционная модель данных	14
1.5 СУБД	15
1.6 SQL	15
1.7 Сценарные языки	16
1.8 Актуальность выбранной темы	18
2 ТРЕБОВАНИЯ К РАЗРАБОТКЕ ПРИЛОЖЕНИЯ	19
2.1 Выбор и обоснование средств разработки	19
2.2 MAMP	19
2.3 PHPMyAdmin	20
2.4 СУБД MySQL	20
2.5 Язык PHP	21
3 ПРИКТИЧЕСКАЯ РЕАЛИЗАЦИЯ	25
3.1 Алгоритм работы	25
3.2 Формирование ролей	25
3.3 Разработка SQL запросов	26
3.3.1 Авторизация пользователя	27
3.3.2 Добавление данных в базу	29

3.3.3 Удаление записей базы данных	32
3.3.4 Редактирование записей базы данных.....	34
3.3.5 Выборка данных из базы.....	36
3.3.6 Поиск записей в базе данных	37
4 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ	42
4.1 Введение.....	42
4.2 Вредные факторы	42
4.3 Требования к организации рабочих мест	43
4.4 Требования к освещению рабочих мест.....	44
4.5 Соблюдение режима труда и отдыха	45
ЗАКЛЮЧЕНИЕ.....	48
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	49
ПРИЛОЖЕНИЯ.....	51
Приложение А Листинг программы	51
Приложение Б Модель базы данных	68
Приложение В Презентация	69

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ЭИОС – электронная информационно-образовательная среда;

ЛВС – локальная вычислительная сеть;

СУБД – система управления базами данных;

БД – база данных;

SQL (англ. Structured Query Language) – язык структурированных запросов;

ЭВМ – электронно-вычислительная машина;

ПЭВМ – персональная электронно-вычислительная машина;

IP (англ. Internet Protocol) – межсетевой протокол;

PHP (англ. Personal Home Page Tools) – препроцессор гипертекста или инструмент для создания персональных веб-страниц.

					ПГУ 2.090301.11.001 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

ВВЕДЕНИЕ

В настоящее время стремительно развиваются технологии, появляются новые механизмы, а старые постоянно улучшаются. То же самое происходит и в сфере информационных технологий, поэтому повсеместно работают или внедряются различные информационные системы.

Университеты и школы тоже используют информационные технологии для упрощения образовательных процессов и создания благоприятной среды для студентов и учеников. Примерами таких технологий можно считать различные сайты школ и университетов, помогающие наблюдать за жизнью и деятельностью учреждения, автоматизированные системы управления, упрощающие процедуру принятия абитуриентов, а также ЭИОС (Электронная Информационно-Образовательная Среда), предоставляющая студентам образовательные материалы, сведения об успеваемости и многое другое.

Многие из этих технологий выполнены на базе ЛВС (Локальная Вычислительная Сеть), которая позволяет использовать ресурсы и устройства всех компьютеров, подключенных к ней.

Предложенная система упрощает работу системных администраторов. В их задачи входит содержание ЛВС в работоспособном состоянии, устранение утечек трафика и других проблем. Программа позволяет определить в каком корпусе и кабинете находится неисправность, избавляя администратора от необходимости обходить большое количество узлов. Так же приложение помогает вовремя обнаружить ряд комплексных проблем с сетью, позволяя решить их без усугубления ситуации.

К основным задачам выпускной квалификационной работы можно отнести: охарактеризовать предметную область решаемой задачи; провести анализ существующих разработок в предметной области; рассмотреть основные функции системы учета пользователей; разработать требований к базе данных; выбрать среду разработки; разработать базы данных для сайта; сформировать запросы к базе данных средствами MySQL; провести верификацию проекта; провести тестирование.

					ПГУ 2.090301.11.001 ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Постановка задачи

В соответствии с заданием, необходимо разработать приложение для администрирования компьютеров университета.

Приложение должно состоять из двух взаимосвязанных частей:

1) Клиентская часть.

Эта часть включает в себя графический интерфейс с базовыми элементами ввода-вывода (кнопка, таблица, поле и т.п.) и программу для обработки запросов к базе данных, связанную с элементами интерфейса и транслирующую запросы клиента в базу данных.

2) База данных.

большая структура, представленная в объективной форме, созданная для хранения и произведения различных операций над данными, построенная в соответствии с реляционной моделью представления данных.

На рисунке 1 изображена структура взаимодействия компонентов приложения.

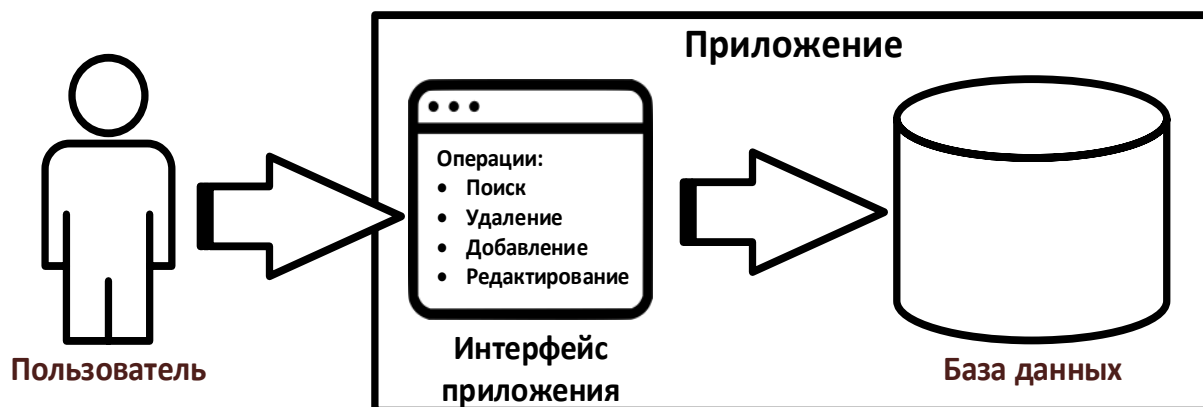


Рисунок 1 - Структура взаимодействия компонентов.

1.2 Анализ существующих разработок в предметной области

Существует немало различных программных решений для мониторинга локальной вычислительной сети. Данные программы незаменимы для системных администраторов, в чьи обязанности входит поддержание сети в рабочем состоянии. Эти приложения помогают отслеживать все сетевые взаимодействия и избавляют от рутинной работы, тем самым упрощая решение даже самых тяжелых сетевых проблем.

1.2.1 Nagios

Nagios — это довольно популярная система для мониторинга сети, которая активно развивается уже много лет. Она написана на языке С и позволяет делать почти все, что может понадобится системным администраторам. Веб-интерфейс этой программы прост, быстр и понятен а серверная часть — надежна.

Существуют различные плагины для аппаратных средств и программного обеспечения. От простых ping-запросов до интеграции со сложными программными решениями. Эта программа может постоянно проводить мониторинг состояния серверов, интернет-сервисов, сетевых каналов, приложений и многого другого, что поддерживает протокол IP. Есть возможность контроля доступного для использования дискового пространства на сервере, загруженности процессора, задержки в WAN и прочее.

В программе также предусмотрен механизм уведомления пользователей по электронной почте, популярным мессенджерам, а также СМС-сообщениям. Данный механизм поддается настройке и есть возможность указать схему уведомлений, которая представляет собой список пользователей, которых требуется оповестить и соответствующие ситуации, в случае которых необходимо произвести рассылку.

Кроме того, функция отображения демонстрирует все контролируемые устройства в логическом представлении их размещения в сети, что дает возможность узнавать о проблемах по мере их возникновения.

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 2.090301.11.001 ПЗ

1.2.2 Icinga

Изначально это приложение предполагалось как новая версия программы Nagios, для мониторинга сети, но в итоге была выделена в самостоятельное приложение под названием Icinga 2. Icinga совместима с большим количеством плагинов и конфигураций Nagios и сейчас разработка этих программ производится параллельно. Приложение Icinga является менее громоздким, в отличие от Nagios, с упором на удобство и производительность. Она предлагает модульную архитектуру и многопоточный дизайн, которых нет в Nagios.

Несмотря на свою легковесность приложение может понимать все, что имеет связь с IP протоколом, к тому же есть возможность использования SNMP, а также различных плагинов.

Icinga имеет несколько вариантов веб-интерфейса, а также возможность конфигурации через него, без использования специальных файлов конфигурации через командную строку.

1.2.3 NeDi

NeDi — это бесплатное приложения для мониторинга сети, которое регулярно просматривает MAC-адреса и таблицы ARP в коммутаторах вашей сети, добавляя в каталог все обнаруженные устройства в локальной базе данных. Таким образом приложение позволяет определить физическое местоположение сетевого оборудования.

Через веб-интерфейс NeDi есть возможность найти любые устройства, порты, точки доступа и многое другое, используя MAC-адрес, IP-адрес или DNS-имя. Приложение запоминает всю информацию о найденных устройствах: серийные номера, версии прошивки, текущие параметры, версии программного обеспечения, конфигурации и многое другое. NeDi также может использоваться для отметания MAC-адресов устройств, которые были потеряны или украдены, при их появлении в сети программа уведомит пользователя об этом

Обнаружение происходит с помощью скан с заданными интервалами. Очень простая конфигурация одним файлом позволяет значительно повысить количество настроек, в том числе возможность пропускать устройства на

					ПГУ 2.090301.11.001 ПЗ	Лист 11
Изм.	Лист	№ докум.	Подпись	Дата		

основе регулярных выражений или заданных сетевых границ. NeDi может использовать как протоколы Cisco Discovery Protocol, так и Link Layer Discovery Protocol для обнаружения новых коммутаторов и маршрутизаторов, а затем подключается к ним для сбора их информации. После установки начальной конфигурации, обнаружение устройств происходит довольно быстро.

1.2.4 Ntop

Проект Ntop — также известный как Ntopng — программа на языке C, предназначенная для мониторинга сетевого трафика. Ntop также обладает простым и понятным веб-интерфейсом. Пользователь запускает один процесс, настроенный на определенный сетевой интерфейс и этого достаточно для полноценной работы приложения.

Ntop предоставляет удобные графики и таблицы, показывающие сетевой трафик: протоколы, источники, назначение и историю транзакций, в том числе в реальном времени. Кроме того, приложение поддерживает большое количество надстроек, например, добавление мониторов NetFlow и sFlow. Программа также предоставляет возможность использования аппаратного монитора.

Ntop может сохранять данные в файлах RRD для непрерывного сбора данных. Также есть возможность контроля трафика в определенном месте на карте сети.

1.2.5 Zabbix

Zabbix — это полномасштабный инструмент для сетевого и системного мониторинга сети, который объединяет множество функций в одной веб-консоли. Он может производить мониторинг и сбор данных с серверов и сетевых устройств.

Zabbix может работать с программными агентами, SNMP и включает поддержку различных гипервизоров виртуализации и может собирать данные об их активности, конфигурации и производительности, а также есть

					ПГУ 2.090301.11.001 ПЗ	Лист 12
Изм.	Лист	№ докум.	Подпись	Дата		

возможности для осуществления мониторинга серверов приложений Java, веб-сервисов и баз данных.

Хосты могут добавляться вручную или автоматически. Имеется большое количество шаблонов для серверов Linux, FreeBSD и Windows, а также служб SMTP, HTTP, ICMP и IPMI, используемых в основном при подробном мониторинге аппаратной части сети. Кроме того, в приложении есть возможность интеграции пользовательских проверок на Perl, Python и многих других языках.

Zabbix предоставляет возможность гибкой настройки панели мониторинга и веб-интерфейса и отображает в виде графиков различные данные, например, пропускную способность сети или загрузка процессора. Программа предоставляет автоматический анализ собранных данных, выделяя проблемы и может ранжировать их по степени важности, а в крайнем случае проводить оповещение по электронной почте.

Zabbix может быть сложным для развертывания на начальном этапе, но возможности автоматического обнаружения и различных шаблонов помогают упростить интеграцию.

1.2 Базы данных

База данных представляет собой набор независимых данных представленных в особой форме, это позволяет легко найти и обработать их с помощью электронно-вычислительной машины. Форма представления данных в таблице – есть модель данных, существует несколько видов таких моделей:

- Иерархическая - это модель данных, где данные представляются в виде древовидной структуры и имеют строгую иерархию.
- Объектно-ориентированная - данные представлены в виде объектов классов, их атрибутов и методов.
- Объектно-реляционная - эта модель является реляционной, но поддерживает принципы объектно-ориентированной модели.

- Реляционная – подразумевает собой данные, связанные между собой, структура которых приведена к «нормальной форме» (устранение избыточности).
- Сетевая – эта модель является улучшенным вариантом иерархической, данные в ней должны быть строго структурированы, целостны и обрабатываться определенным образом.
- Функциональная – подразумевает многомерную иерархию в структуре данных и интерактивные вычисления.

1.4 Реляционная модель данных

Реляционная модель представления данных опирается на следующие аспекты:

- Структурный — данные в базе представляются в виде набора отношений.
- Целостности — отношения должны соответствовать условиям целостности, которые к тому же, могут быть заданы декларативно на уровне типа данных, отношения или базы данных.
- Обработки — в базе данных должна поддерживаться реляционная алгебра.
- Нормализации — структура представления данных должна быть лишена избыточности, так как потенциально это может привести к нарушению целостности данных.

Для лучшего понимания реляционной модели следует отметить:

- Это логическая модель, это означает, что отношения в ней абстрактны и не имеют физической структуры.
- Вся информация в базе представлена в виде заданных значений атрибутов в кортежах отношений.
- Отсутствие указателей, связывающих значения между собой.
- Реляционная алгебра, поддерживаемая в базе данных, позволяет использовать программирование на декларативных языках.

Альтернативами реляционной модели можно назвать иерархическую и сетевую модели представления данных. Некоторые системы, построенные на

					ПГУ 2.090301.11.001 ПЗ	Лист 14
Изм.	Лист	№ докум.	Подпись	Дата		

этих архитектурах, до сих пор находятся эксплуатации, несмотря на то, что модели считаются устаревшими.

1.5 СУБД

Специалисты обращают внимание на то, что многие люди не видят разницы между терминами «база данных» и «система управления базами данных», и подчеркивают, что это является заблуждением, призывая разграничивать эти понятия.

Система управления базами данных (СУБД) - комплекс специализированных программ, необходимый для эффективного администрирования баз данных. В качестве структурной формы СУБД может быть использована любая из перечисленных выше моделей.

1.6 SQL

Язык структурированных запросов SQL является декларативным языком программирования и используется для выполнения операций с данными в реляционной базе данных, под руководством соответствующей системы управления базами данных.

SQL предназначен для удаления, редактирования, поиска и добавления элементов в базу данных, однако это далеко не полный список операций, который он способен выполнять. Когда SQL был относительно молодым языком он уже мог выполнять следующие операции:

- Создание таблицы в рамках базы данных.
- Добавление данных в таблицу.
- Изменение данных.
- Удаление данных.
- Выборка данных из таблицы, или нескольких таблиц.
- Изменение структур таблиц.

Со временем SQL усложнялся, в нем появлялись новые функции, например, появилась возможность управления индексами, триггерами и хранимыми процедурами.

Таким образом даже сейчас язык SQL является одним из самых распространенных решений, позволяющих программному обеспечению взаимодействовать с базами данных.

В SQL содержится большое количество различных операторов, инструкций и функций.

Операторы языка можно разделить на четыре категории:

- операторы определения данных (Data Definition Language, DDL):
 - CREATE создаёт объект базы данных (саму базу, таблицу, представление, пользователя и так далее),
 - ALTER изменяет объект,
 - DROP удаляет объект;
- операторы манипуляции данными (Data Manipulation Language, DML):
 - SELECT выбирает данные, удовлетворяющие заданным условиям,
 - INSERT добавляет новые данные,
 - UPDATE изменяет существующие данные,
 - DELETE удаляет данные;
- операторы определения доступа к данным (Data Control Language, DCL):
 - GRANT предоставляет пользователю (группе) разрешения на определённые операции с объектом,
 - REVOKE отзывает ранее выданные разрешения,
 - DENY задаёт запрет, имеющий приоритет над разрешением;
- операторы управления транзакциями (Transaction Control Language, TCL):
 - COMMIT применяет транзакцию,
 - ROLLBACK откатывает все изменения, сделанные в контексте текущей транзакции,
 - SAVEPOINT делит транзакцию на более мелкие участки.

1.7 Сценарные языки

Сценарный язык является высокоуровневым языком программирования, на котором пишутся небольшие программы, которые также называют сценариями или скриптами.

					ПГУ 2.090301.11.001 ПЗ	Лист
						16
Изм.	Лист	№ докум.	Подпись	Дата		

Высокие языки программирования можно условно разделить на системные языки и языки сценарные, последние обычно интерпретируются, в отличие от компилируемых системных языков.

Сценарные языки имеют множество различных параметров для классификации, например, по быстродействию они делятся на:

- Языки динамического набора (например, sh) – программа компилируется и исполняется небольшими блоками.
- Предварительно компилируемые (Perl) – программа переводится в байт код и исполняется целиком.

По применению:

- Командно-сценарные – выполняется оболочкой операционной системы (bash, sh).
- прикладные сценарные – языки, специально разработанные для конкретной задачи (GML, ERM).
- Универсальные сценарные – являются универсальными, однако чаще всего используются в веб-программировании (PHP, Perl).

Преимущества использования сценарных языков:

- Они интерпретируются, а не компилируются, поэтому даже некорректно написанный сценарий может привести только к выводу диагностического сообщения, но не может привести к системному краху.
- Сценарий может быть написан исходя из другой концепции программирования, нежели код в основной программе. Например, основная приложение будет являться однопоточным и монолитным, а сценарии будут выполняться параллельно.
- Одна строка сценария может заменить множество строк кода на системном языке, так как сценарии содержат в себе множество небольших проблемно-ориентированных функций.
- Сценарные языки обладают низким порогом вхождения.
- Подавляющее большинство сценарных языков обладают кроссплатформенностью.

1.8 Актуальность выбранной темы

На рынке программного обеспечения есть большое количество программ для учета пользователей и компьютеров в сети организаций. Однако большинство таких программных решений продвигаются на коммерческой основе. Также следует учесть, что богатый функционал таких программ универсален, но большинство возможностей будет использоваться далеко не всегда.

В частности, при интегрировании такого рода системы в среду образовательного учреждения, организация будет выделять необоснованно большие денежные средства на оплату неиспользуемых, в рамках локальной вычислительной сети возможностей.

Исходя из вышесказанного было принято решение, в рамках дипломного проекта, разработать приложение для учета пользователей и компьютеров локальной вычислительной сети ПГУ, разработка которого будет сопровождаться внедрением исключительно необходимых функций.

					ПГУ 2.090301.11.001 ПЗ	Лист
						18
Изм.	Лист	№ докум.	Подпись	Дата		

2 ТРЕБОВАНИЯ К РАЗРАБОТКЕ ПРИЛОЖЕНИЯ

Разрабатываемое приложение предназначено для учета пользователей и компьютеров локальной вычислительной сети университета. Программное решение предоставляет пользователям доступ к базе данных, возможности по ее редактированию и администрированию и представляет собой веб-интерфейс. Использовать приложение могут как администраторы ЛВС ПГУ, так и операторы.

2.1 Выбор и обоснование средств разработки

Исходя из поставленных задач были выбраны следующие средства разработки:

- MAMP – локальный веб сервер для запуска веб-интерфейса;
- СУБД MySQL;
- PMA (phpMyAdmin) – веб-интерфейс для написания и тестирования

SQL-запросов к базе данных.

И следующие языки:

- SQL – для формирования запросов к базе данных.
- PHP + PHP Data Object – для создания сценариев отправки запросов к базе данных. Использованный интерфейс предлагает универсальный способ работы с базами данных.
- HTML – для разметки веб-интерфейса.

Приведенный стек технологий выбирался исходя из их некоммерческого распространения и функциональности. А также немаловажным фактом была распространенность и доступность большого количества теоретических и практических материалов по теме.

2.2 MAMP

В процессе создания сайта часто возникает необходимость протестировать его и проверить работоспособность. Для этого был использован локальный веб-сервер сервер MAMP. Во время локального тестирования нет ограничения на размер сайта или использование ресурсов компьютера, а также

					ПГУ 2.090301.11.001 ПЗ	Лист 19
Изм.	Лист	№ докум.	Подпись	Дата		

загрузка любой страницы происходит настолько быстро, насколько позволяет вычислительная мощность ЭВМ, то есть практически мгновенно, поэтому такой способ очень удобен.

Существует две версии приложения: бесплатная MAMP и платная MAMP PRO. Функциональность MAMP PRO гораздо больше, однако в данной работе была бы избыточна. Приложение не требует установки, для его использования достаточно просто перенести его в папку на компьютере.

Управление настройками производится через окно программы, а работа с веб-сервером производится через веб-интерфейс.

В настройках имеется возможность установить порты Apache и MySQL, выбрать версию PHP, оптимизатор (XCache, APC, eAccelerator) и задать Root-папку для проекта.

Преимуществами данного программного продукта является:

- Бесплатное распространение.
- Простота использования.
- Интуитивно понятный интерфейс.

2.3 PHPMyAdmin

PHPMyAdmin данное приложение выполнено на языке PHP и представляет собой веб-интерфейс. Программа позволяет выполнять различные действия с таблицами и их содержимым, например, добавление, удаление, поиск или редактирование элементов, а также создавать таблицы и базы данных. Одной из основных причин использования данного программного продукта является возможность моделирования SQL запросов к базе данных.

Помимо прочего эта программа обладает удобным и интуитивно понятным интерфейсом, позволяющим администрировать базу данных без непосредственного ввода SQL команд.

2.4 СУБД MySQL

Это универсальная реляционная система управления базами данных, она очень популярна среди разработчиков. Спектр ее возможностей довольно обширен, особенно удобно ее использование в веб-разработке.

					ПГУ 2.090301.11.001 ПЗ	Лист 20
Изм.	Лист	№ докум.	Подпись	Дата		

К плюсам этой СУБД можно отнести:

- Открытый исходный код.

Приложение свободно распространяется, поэтому его использование и обслуживание, по сравнению с платными аналогами, обходится намного дешевле.

- Большое и активное сообщество.

Благодаря большому сообществу, в интернете всегда можно найти решение любой возникшей проблемы.

- Стабильность и надежность.
- Частые обновления.

Регулярные обновления и поддержка разработчиков позволяют системе оставаться надежной и работать корректно, а также получать новые функции.

- Масштабируемое хранилище.
- Скорость.

MySQL имеет меньше функциональных возможностей, в сравнении с другими СУБД, однако это позволяет ему работать намного быстрее.

2.5 Язык PHP

PHP — универсальный сценарный язык, обладает огромной популярностью и часто применяется для разработки веб-приложений. Этот язык имеет большое количество модулей, которые удобно применять в разработке веб-приложений:

- автоматическое извлечение POST- и GET-параметров, а также переменных окружения веб-сервера в предопределённые массивы.
- взаимодействие с большим количеством различных СУБД через дополнительные модули.
- автоматизированная отправка HTTP-заголовков.
- работа с HTTP-авторизацией.
- работа с cookies и сессиями.
- работа с локальными и удалёнными файлами, сокетами.
- обработка файлов, загружаемых на сервер.

					ПГУ 2.090301.11.001 ПЗ	Лист 21
Изм.	Лист	№ докум.	Подпись	Дата		

PHP Си-подобный синтаксис. Некоторые элементы, такие как ассоциативные массивы и цикл foreach, заимствованы из Perl.

Для написания простейшего скрипта не требуется описывать какие-либо переменные или используемые модули. Любой скрипт может начинаться непосредственно с оператора PHP.

PHP язык программирования с динамической типизацией и, следовательно, не требует объявления переменных или их типа. Преобразования между типами переменных происходят автоматически, но могут быть произведены и вручную. В PHP имеются следующие типы переменных:

- целочисленный тип (integer)
- число с плавающей точкой (float, double)
- логический тип (boolean)
- строковый тип (string)
- массив (array)
- объект (object)
- внешний ресурс (resource)
- неопределенное значение (null)
- mixed любой тип
- number число (integer либо float)
- callback (анонимная функция, string или массив)
- void отсутствие параметров

Также в PHP есть специальные массивы переменных, названные суперглобальными. Большая часть этих массивов содержит входные данные запроса пользователя (параметры GET-запроса, поля форм при отправке методом POST, куки и т. п.):

`$GLOBALS` - массив всех глобальных переменных.

`$_SERVER` - содержит переменные окружения, которые операционная система передает серверу.

`$_ENV` - текущие переменные среды.

					ПГУ 2.090301.11.001 ПЗ	Лист 22
Изм.	Лист	№ докум.	Подпись	Дата		

`$_GET` - содержит параметры GET-запроса, переданные в URI после знака вопроса «?».

`$_POST` - ассоциативный массив значений полей HTML-формы при отправке методом POST.

`$_FILES` - ассоциативный массив со сведениями об отправленных методом POST файлах. В него входят:

- `['name']` — исходное имя файла на компьютере пользователя.
- `['type']` — указанный агентом пользователя MIME-тип файла.
- `['size']` — размер файла в байтах.
- `['tmp_name']` — полный путь к файлу во временной папке.
- `['error']` — код ошибки.

`$_COOKIE` - ассоциативный массив с переданными агентом пользователя значениями куки.

`$_REQUEST` - содержит элементы из массивов `$_GET`, `$_POST`, `$_COOKIE`.

`$_SESSION` - содержит данные сессии.

PHP-скрипты обычно обрабатываются интерпретатором в порядке, обеспечивающем кроссплатформенность разработанного приложения:

- лексический анализ исходного кода и генерация лексем,
- синтаксический анализ полученных лексем,
- генерация байт-кода,
- выполнение байт-кода интерпретатором (без создания исполняемого файла).

Для увеличения быстродействия приложений возможно использование специального программного обеспечения, так называемых акселераторов. Принцип их работы заключается в кэшировании однажды сгенерированного байт-кода в памяти и/или на диске, таким образом, из процесса работы приложения исключаются этапы 1—3, что в общем случае ведёт к значительному ускорению работы.

Важной особенностью является то, что разработчику нет необходимости заботиться о распределении и освобождении памяти. Ядро РНР реализует средства для автоматического управления памятью; вся выделенная память возвращается системе после завершения работы скрипта.

					ПГУ 2.090301.11.001 ПЗ	Лист
						24
Изм.	Лист	№ докум.	Подпись	Дата		

3 ПРИКТИЧЕСКАЯ РЕАЛИЗАЦИЯ

3.1 Алгоритм работы

Разрабатываемый программный продукт включает в себя две функциональные части:

- Клиентское приложение, представленное в виде веб-интерфейса.
- База данных, содержащая информацию о компьютерах и пользователях локальной вычислительной сети ПГУ.

Пользователь должен пройти авторизацию, после чего он получает возможность взаимодействовать с базой данных. Это происходит следующим образом:

1. Пользователь взаимодействует с необходимым элементом интерфейса.
2. Отправляется SQL запрос к базе данных.
3. Приложение получает результат запроса.
4. Визуализация результата запроса через веб-интерфейс.

Если на любом из этапов происходит ошибка, то пользователю выводится оповещение о произошедшем сбое.

3.2 Формирование ролей

Роли, предусмотренные в приложении:

- Администратор – имеет весь спектр возможностей взаимодействия с базой данных, а именно:
 - Добавление данных
 - Просмотр данных
 - Редактирование данных
 - Удаление данных
 - Резервное копирование и восстановление
- Оператор – имеет немного меньший список возможностей взаимодействия:
 - Добавление записей ip в аудитории
 - Редактирование записей ip в аудиториях
 - Удаление записей ip в аудиториях

					ПГУ 2.090301.11.001 ПЗ	Лист
						25
Изм.	Лист	№ докум.	Подпись	Дата		

- Просмотр данных

На рисунках 2 и 3 изображены UML диаграммы прав ролей:

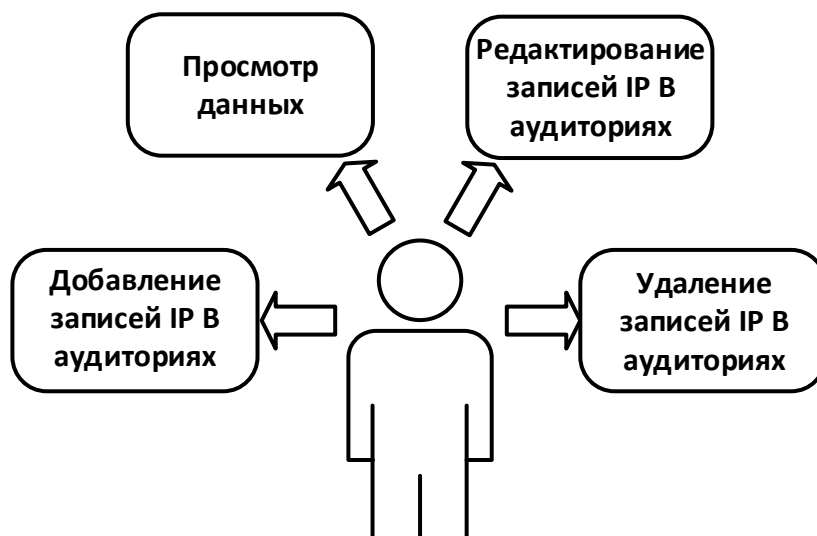


Рисунок 2 - UML-диаграмма прав оператора.

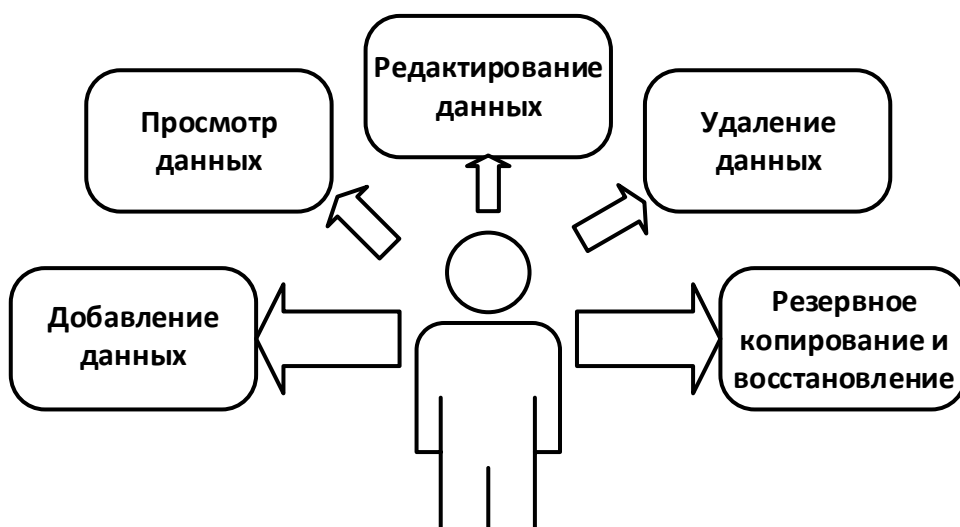


Рисунок 3 - UML-диаграмма прав администратора.

3.3 Разработка SQL запросов.

Перед началом разработки SQL запросов к базе данных необходимо подробно ознакомиться с ее структурой. Структура базы данных разрабатываемого приложения изображена на рисунке 4:

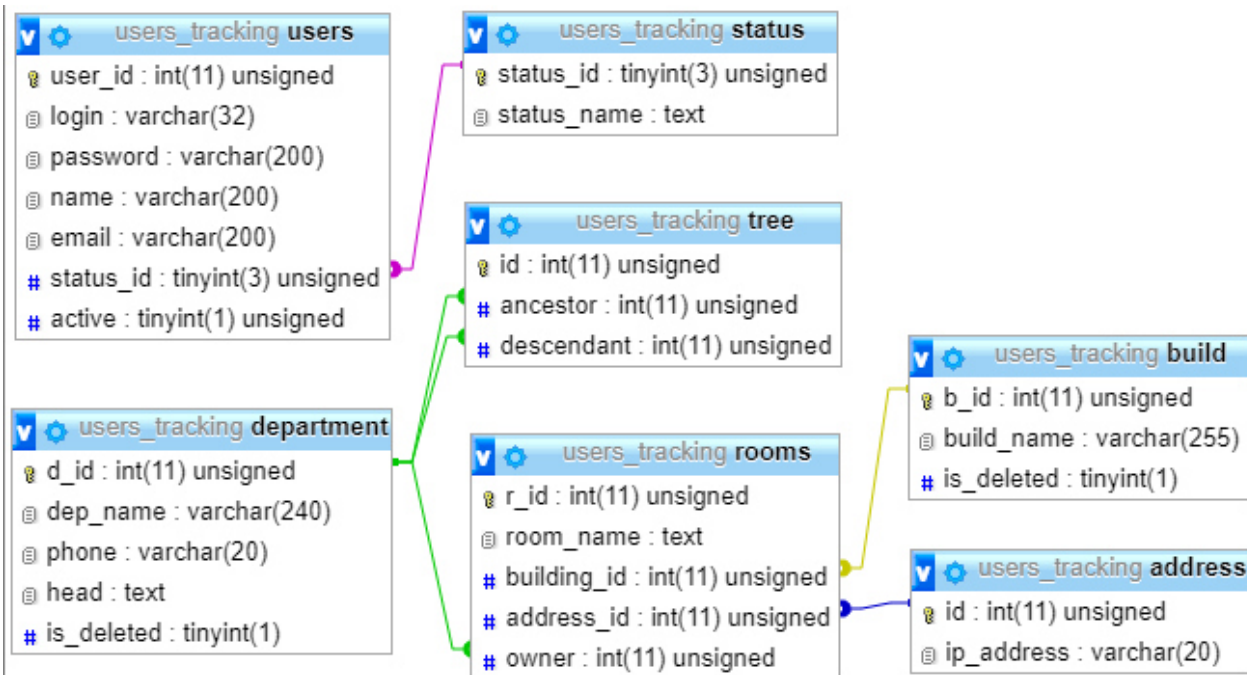


Рисунок 4 - структура базы данных.

Любой разрабатываемый запрос сначала необходимо протестировать в приложении РНРМуAdmin, для того, чтобы удостовериться в правильности его составления.

После тестирования запрос необходимо обернуть в РНР-код и интегрировать в интерфейс приложения.

3.3.1 Авторизация пользователя.

Разработанный веб-интерфейс включает в себя авторизацию пользователя. Это реализовано с помощью функций getAllUsers и getAllStatus, которые обращаются к таблицам Users и Status.

Значения в таблице Status представлены на рисунке 5 представляют собой роли.

status_id	status_name
1	Оператор
2	Администратор

Рисунок 5 - содержимое таблицы Status.

В таблице Users, представленной на рисунке 6 содержится информация о логинах и паролях пользователей, а также обозначены права пользователей.

user_id	login	password	name	email	status_id	active
44	admin	b58438775e431c88164e48b8a225f3af	test	email	2	1
45	admin11	71ea76dfbf3d90f26ee2c486ea94745b	FIO	email	1	1
46	oksaalla@yandex.ru	0eaa2793389d4fae91b25126db4e7cfc	FIO	email	2	1
47	qqq	\$2y\$10\$0jSatveYQvz9FojJh9VrNehjeYq.ROgcXtgzw.9hckt...	eee	qwe	1	1

Рисунок 6 - содержимое таблицы Users.

Таким образом, если введенный логин и пароль пользователя совпадают с данными, возвращенными функцией getAllUsers, а также значение уровня прав, возвращаемое функцией getAllStatus соответствует роли администратора, то авторизованный пользователь может выполнять взаимодействия с базой данных, в рамках прав администратора. Аналогично, если функция getAllStatus вернет значение прав оператора, то пользователю будут доступны только права, предусмотренные для операторов системы.

Для того, чтобы протестировать составленный SQL запрос была использована соответствующая функция веб-интерфейса PHPMyAdmin, результаты тестирования, представленные на рисунках 7 и 8 подтверждают корректность составленного запроса.

Отображение строк 0 - 1 (2 всего, Запрос занял 0.0008 сек.)

SELECT * FROM status

Профилирование [Построение редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице | Сортировать по индексу: Ниодного

Параметры

	status_id	status_name
☐ Изменить ☐ Копировать ☐ Удалить	1	Оператор
☐ Изменить ☐ Копировать ☐ Удалить	2	Администратор

Рисунок 7 - результаты тестирования запроса.

Отображение строк 0 - 3 (4 всего, Запрос занял 0.0012 сек.)

SELECT * FROM users LEFT JOIN status ON users.status_id=status.status_id

Профилирование [Построение редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице | Сортировать по индексу: Ниодного

Параметры

user_id	login	password	name	email	status_id	active	status_id	status_name
45	admin11	71ea76dfbf3d90f26ee2c486ea94745b	FIO	email	1	1	1	Оператор
47	qqq	\$2y\$10\$0jSatveYQvz9FojJh9VrNehjeYq.ROgcXtgzw.9hckt...	eee	qwe	1	1	1	Оператор
44	admin	b58438775e431c88164e48b8a225f3af	test	email	2	1	2	Администратор
46	oksaalla@yandex.ru	0eaa2793389d4fae91b25126db4e7cfc	FIO	email	2	1	2	Администратор

Рисунок 8 - результаты тестирования запроса.

3.3.2 Добавление данных в базу.

Если авторизированный пользователь обладает правами администратора, то у него есть возможность добавление записей в базу данных. Добавление происходит с помощью веб-интерфейса, представленного на рисунке 9.

ПГУ Главная Подразделения					
#	Логин	Пароль	ФИО	e-mail	Тип пользователя
☐	admin11	71ea76dfb3d90f26ee2c486ea94745b	FIO	email	Оператор
☐	qqq	\$2y\$10\$0jSatveYQvz9FojJh9VrNehjeYq.ROgcXtIgzW.9hckt2jsfQVvmzTq	eee	qwe	Оператор
☐	admin	b58438775e431c88164e48b8a225f3af	test	email	Администратор
☐	oksaalla@yandex.ru	0eaa2793389d4fae91b25126db4e7cfc	FIO	email	Администратор
Добавить пользователя Удалить пользователя Редактировать пользователя					

Рисунок 9 - веб-интерфейс: добавление пользователя.

После нажатия на кнопку «Добавить пользователя» открываются текстовые окна, в которые необходимо ввести данные добавляемого пользователя, это изображено на рисунке 10.

☐	qqq	\$2y\$10\$0jSatveYQvz9FojJh9VrNehjeYq.ROgcXtIgzW.9hckt2jsfQVvmzTq	eee	qwe	Оператор
☐	admin	b58438775e431c88164e48b8a225f3af	test	email	Администратор
☐	oksaalla@yandex.ru	0eaa2793389d4fae91b25126db4e7cfc	FIO	email	Администратор
Добавить пользователя Удалить пользователя Редактировать пользователя					
Логин					
<input type="text" value="superUser"/>					
Пароль					
<input type="text" value="password"/>					
ФИО					
<input type="text" value="Илья"/>					
E-mail					
<input type="text" value="iliaduh@mail.ru"/>					
Выберите тип пользователя					
<input type="text" value="Администратор"/>					
Добавить					

Рисунок 10 - веб-интерфейс: добавление пользователя.

После нажатия на кнопку «добавить» происходит вызов функции addUser, которая отправляет запрос на добавление элемента в таблицу пользователей базы данных, в результате в базу добавляется новая запись и пользователь может видеть ее в таблице пользователей, отображаемой в веб-интерфейсе. Это показано на рисунке 11.

#	Логин	Пароль	ФИО	e-mail	Тип пользователя
☐	admin11	71ea76dfb3d90f26ee2c486ea94745b	PIO	email	Оператор
☐	qqq	\$2y\$10\$0j\$atveYQvz9FojJh9VrNehjeYq.ROgcXtgow.9hckt2jsfQVvmzTq	eee	qwe	Оператор
☐	admin	b58438775e431c88164e48b8a225f3af	test	email	Администратор
☐	oksaalla@yandex.ru	0eaa2793389d4fae91b25126db4e7cfc	PIO	email	Администратор
☐	superUser	password	Илья	iliaduh@mail.ru	Администратор

Добавить пользователя

Удалить пользователя

Редактировать пользователя

Рисунок 11 - веб-интерфейс: добавление пользователя.

Аналогично пользователь может добавить подразделение университета, исключение составляет только вызываемая функция. В данном случае будет вызвана функция AddDepartment, которая содержит запрос на добавление элемента в таблицу подразделений, процесс добавления проиллюстрирован на рисунках 12 - 14.

Название подразделения	Руководитель	Телефон
Кафедра "Общая психология"	Иванова Наталья Ивановна	987654128
Кафедра "Информационно-вычислительные системы"	Бобрышева Галина Владимировна	987654123
Кафедра "Вычислительная техника"	Митрохин Максим Александрович	784123682
Кафедра "Информационное обеспечение управления и производства"	Катышева Марина Александровна	+7 (8412) 36-82-39
Кафедра "Математическое обеспечение и применение ЭВМ"	Макарычев Петр Петрович	+7(412) 36-82-26
Кафедра "Высшая и прикладная математика"	Бойков Илья Владимирович	+7(8412)36-82-59
Кафедра "Математика и суперкомпьютерное моделирование"	Смирнов Юрий Геннадьевич	+7(8412)36-80-96
Факультет вычислительной техники	Фионова Людмила Римовна	+7(8412)36-82-48
Кафедра "Компьютерные технологии"	Горбаченко Владимир Иванович	+7(8412)36-80-68
Управление информатизации		0
Политехнический институт	Козлов Геннадий Васильевич	+7(8412)36-84-56

Добавить подразделение

Удалить подразделение

Редактировать подразделение

Рисунок 12 - веб-интерфейс: добавление подразделения.

Кафедра "Математическое обеспечение и применение ЭВМ"	Макарычев Петр Петрович	+7(412) 36-82-26
Кафедра "Высшая и прикладная математика"	Бойков Илья Владимирович	+7(8412)36-82-59
Кафедра "Математика и суперкомпьютерное моделирование"	Смирнов Юрий Геннадьевич	+7(8412)36-80-96
Факультет вычислительной техники	Фионова Людмила Римовна	+7(8412)36-82-48
Кафедра "Компьютерные технологии"	Горбаченко Владимир Иванович	+7(8412)36-80-68
Управление информатизации		0
Политехнический институт	Козлов Геннадий Васильевич	+7(8412)36-84-56

Выберите над подразделение

Кафедра "Вычислительная техника" ▼

Название подразделения

Проверочная кафедра

Руководитель подразделения

Проверочников В. И.

Телефон

88002000122

Рисунок 13 - веб-интерфейс: добавление подразделения.

Название подразделения	Руководитель	Телефон
Кафедра "Общая психология"	Иванова Наталья Ивановна	987654128
Кафедра "Информационно-вычислительные системы"	Бобрышева Галина Владимировна	987654123
Кафедра "Вычислительная техника"	Митрохин Максим Александрович	784123682
Кафедра "Информационное обеспечение управления и производства"	Катышева Марина Александровна	+7 (8412) 36-82-39
Кафедра "Математическое обеспечение и применение ЭВМ"	Макарычев Петр Петрович	+7(412) 36-82-26
Кафедра "Высшая и прикладная математика"	Бойков Илья Владимирович	+7(8412)36-82-59
Кафедра "Математика и суперкомпьютерное моделирование"	Смирнов Юрий Геннадьевич	+7(8412)36-80-96
Факультет вычислительной техники	Фионова Людмила Римовна	+7(8412)36-82-48
Кафедра "Компьютерные технологии"	Горбаченко Владимир Иванович	+7(8412)36-80-68
Управление информатизации		0
Политехнический институт	Козлов Геннадий Васильевич	+7(8412)36-84-56
Проверочная кафедра	Проверочников В. И.	88002000122

Рисунок 14 - веб-интерфейс: добавление подразделения.

Результаты тестирования запросов в RНРМуAdmin приведены на рисунках 15 и 16.

✓ Добавлена 1 строка.

Идентификатор вставленной строки: 50 (Запрос занял 0.0936 сек.)

```
INSERT INTO users(login, password, name, email, status_id) VALUES("test", "test", "test", "test", 1)
```

[Построчное редактирование] [Изменить] [Создать PHP-код]

Рисунок 15 - тестирование запроса.

✓ Добавлена 1 строка.

Идентификатор вставленной строки: 352685476 (Запрос занял 0.0575 сек.)

```
INSERT INTO department(dep_name, head, phone) VALUES("test", "test", 88888888888)
```

[Построчное редактирование] [Изменить] [Создать PHP-код]

Рисунок 16 - тестирование запроса.

					ПГУ 2.090301.11.001 ПЗ	Лист 31
Изм.	Лист	№ докум.	Подпись	Дата		

3.3.3 Удаление записей базы данных

Удаление записей в базе данных происходит в том же окне веб-интерфейса, в котором происходит добавление элементов (Рисунок 9, 12). Авторизированный пользователь с правами администратора помечает галочкой в таблице тот элемент, который требуется удалить и нажимает кнопку «Удалить пользователя» или «Удалить подразделение», после чего вызывается соответствующая функция. При удалении пользователя будет вызвана функция `deleteUser`, отправляющая запрос на удаление к таблице пользователей, при удалении подразделения вызывается функция `deleteDepartment`, предусматривающая отправку запроса к таблице подразделений. Результаты удаления записей предоставлены на рисунках 17-20.

ПГУ Главная Подразделения

#	Логин	Пароль	ФИО	e-mail	Тип пользователя
☐	admin11	71ea76dfb3d90f26ee2c486ea94745b	PIO	email	Оператор
☐	qqq	\$2y\$10\$0jSatieYQvz9FojJh9VrNehjeYq.ROgcXtIgzW.9hckl2jsfQVvmzTq	eee	qwe	Оператор
☒	test	test	test	test	Оператор
☐	admin	b58438775e431c88164e48b8a225f3af	test	email	Администратор
☐	oksaalla@yandex.ru	0eaa2793389d4fae91b25126db4e7cfc	PIO	email	Администратор
☐	superUser	password	Илья	iliaduh@mail.ru	Администратор

Добавить пользователя

Удалить пользователя

Редактировать пользователя

Рисунок 17 - веб-интерфейс: удаление пользователя.

ПГУ Главная Подразделения

#	Логин	Пароль	ФИО	e-mail	Тип пользователя
☐	admin11	71ea76dfb3d90f26ee2c486ea94745b	PIO	email	Оператор
☐	qqq	\$2y\$10\$0jSatieYQvz9FojJh9VrNehjeYq.ROgcXtIgzW.9hckl2jsfQVvmzTq	eee	qwe	Оператор
☐	admin	b58438775e431c88164e48b8a225f3af	test	email	Администратор
☐	oksaalla@yandex.ru	0eaa2793389d4fae91b25126db4e7cfc	PIO	email	Администратор
☐	superUser	password	Илья	iliaduh@mail.ru	Администратор

Добавить пользователя

Удалить пользователя

Редактировать пользователя

Рисунок 18 - веб-интерфейс: удаление пользователя.

3.3.4 Редактирование записей базы данных

Редактирование записей может проводить авторизованный пользователь, обладающий правами администратора. Для этого необходимо в веб-интерфейсе выбрать необходимый для редактирования элемент с помощью галочки и нажать кнопку «Редактировать подразделение» или «Редактировать пользователя», после этого пользователь получит возможность заполнения полей формы новыми данными для записи, затем будет вызвана соответствующая функция: `changeDepartment` для подразделения и `changeUser` для пользователя. Обе функции отправляют запрос к базе данных с требованием изменить запись, однако в запросе фигурируют разные таблицы. Причем функции и запросы спроектированы таким образом, что нет необходимости вводить значение для всех полей, если нужно поменять только имя – пользователь вводит только его. Результаты редактирования записей приведены на рисунках 23 – 26.

ПГУ Главная Подразделения

#	Логин	Пароль	ФИО	e-mail	Тип пользователя
☐	admin11	71ea76dfbf3d90f26ee2c486ea94745b	FIO	email	Оператор
☒	qqq	\$2y\$10\$0jSatveYQvz9FojJh9VrNehjeYq.ROgcXtIgzW.9hckt2jsfQVvmzTq	eee	qwe	Оператор
☐	admin	b58438775e431c88164e48b8a225f3af	test	email	Администратор
☐	oksaalla@yandex.ru	0eaa2793389d4fae91b25126db4e7cfc	FIO	email	Администратор

Добавить пользователя

Удалить пользователя

Редактировать пользователя

Рисунок 23 - веб-интерфейс: редактирование пользователя.

ПГУ Главная Подразделения

#	Логин	Пароль	ФИО	e-mail	Тип пользователя
☐	admin11	71ea76dfbf3d90f26ee2c486ea94745b	FIO	email	Оператор
☐	Илья	\$2y\$10\$0jSatveYQvz9FojJh9VrNehjeYq.ROgcXtIgzW.9hckt2jsfQVvmzTq	eee	qwe	Оператор
☐	admin	b58438775e431c88164e48b8a225f3af	test	email	Администратор
☐	oksaalla@yandex.ru	0eaa2793389d4fae91b25126db4e7cfc	FIO	email	Администратор

Добавить пользователя

Удалить пользователя

Редактировать пользователя

Рисунок 24 - веб-интерфейс: редактирование пользователя.

№	Название подразделения	Руководитель	Телефон
☐	Кафедра "Общая психология"	Иванова Наталья Ивановна	987654128
☐	Кафедра "Информационно-вычислительные системы"	Бобрышева Галина Владимировна	987654123
☐	Кафедра "Вычислительная техника"	Митрохин Максим Александрович	784123682
☐	Кафедра "Информационное обеспечение управления и производства"	Катышева Марина Александровна	+7 (8412) 36-82-39
☐	Кафедра "Математическое обеспечение и применение ЭВМ"	Макарычев Петр Петрович	+7(412) 36-82-26
☐	Кафедра "Высшая и прикладная математика"	Бойков Илья Владимирович	+7(8412)36-82-59
☐	Кафедра "Математика и суперкомпьютерное моделирование"	Смирнов Юрий Геннадьевич	+7(8412)36-80-96
☐	Факультет вычислительной техники	Фионова Людмила Римовна	+7(8412)36-82-48
☐	Кафедра "Компьютерные технологии"	Горбаченко Владимир Иванович	+7(8412)36-80-68
☐	Управление информатизации		0
☐	Политехнический институт	Козлов Геннадий Васильевич	+7(8412)36-84-56
☒	Проверочная кафедра	Проверочников В. И.	88002000122

Добавить подразделение

Удалить подразделение

Редактировать подразделение

Рисунок 25 - веб-интерфейс: редактирование подразделения.

№	Название подразделения	Руководитель	Телефон
☐	Кафедра "Общая психология"	Иванова Наталья Ивановна	987654128
☐	Кафедра "Информационно-вычислительные системы"	Бобрышева Галина Владимировна	987654123
☐	Кафедра "Вычислительная техника"	Митрохин Максим Александрович	784123682
☐	Кафедра "Информационное обеспечение управления и производства"	Катышева Марина Александровна	+7 (8412) 36-82-39
☐	Кафедра "Математическое обеспечение и применение ЭВМ"	Макарычев Петр Петрович	+7(412) 36-82-26
☐	Кафедра "Высшая и прикладная математика"	Бойков Илья Владимирович	+7(8412)36-82-59
☐	Кафедра "Математика и суперкомпьютерное моделирование"	Смирнов Юрий Геннадьевич	+7(8412)36-80-96
☐	Факультет вычислительной техники	Фионова Людмила Римовна	+7(8412)36-82-48
☐	Кафедра "Компьютерные технологии"	Горбаченко Владимир Иванович	+7(8412)36-80-68
☐	Управление информатизации		0
☐	Политехнический институт	Козлов Геннадий Васильевич	+7(8412)36-84-56
☐	Кафедра проверки	Проверочников В. И.	88002000122

Добавить подразделение

Удалить подразделение

Редактировать подразделение

Рисунок 26 - веб-интерфейс: редактирование подразделения.

Результаты тестирования запросов приведены на рисунках 27 и 28.

✓ Затронута 1 строка. (Запрос занял 0.0872 сек.)

```
UPDATE users SET login = (SELECT IFNULL("test",login)), password = (SELECT IFNULL(NULL,password)), name = (SELECT IFNULL(NULL,name)), email = (SELECT IFNULL(NULL,email)), status_id = (SELECT IFNULL(NULL, status_id)) WHERE users.user_id = 47
```

[Построение редактирование] [Изменить] [Создать PHP-код]

Рисунок 27 - тестирование запроса.

✓ Затронута 1 строка. (Запрос занял 0.0514 сек.)

```
UPDATE `department` SET dep_name = (SELECT IFNULL("test",dep_name)), phone = (SELECT IFNULL(111,phone)), head = (SELECT IFNULL("test",head)) WHERE department.d_id = 352685475
```

[Построение редактирование] [Изменить] [Создать PHP-код]

Рисунок 28 - тестирование запроса.

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 2.090301.11.001 ПЗ

Лист

35

3.3.5 Выборка данных из базы

Пользователь обладает возможностью посмотреть на все компьютеры ПГУ в одной таблице, такая возможность реализована с помощью функций getAllBuildings, getAllRooms и getAllDepartments, которые доставляют базе данных запрос на выборку всех данных из соответствующих таблиц. После чего данные визуализируются в виде таблицы в пользовательском интерфейсе, что показано на рисунке 29.

#	Адрес	Аудитория	Корпус	Подразделение
☐	172.16.7.10	7-108	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-112	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-117	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-209	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-211	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-307	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-407	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-425	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-502	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-503	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-505	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-508	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-527	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-102	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-103	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-104	7 корпус	Факультет вычислительной техники

Рисунок 29 - веб-интерфейс: главная таблица.

Тестирование запросов приведено на рисунках 30-32.

Отображение строк 0 - 24 (44 всего, Запрос занял 0.0044 сек.)

SELECT * FROM build

Профилирование

[Построчное редактирование]

[Изменить]

[Анализ SQL запроса]

[Создать PHP-код]

[Обновить]

1

>

>>

☐

Показать все

Количество строк:

25

Фильтровать строки:

Поиск в таблице

Сортировать по индексу:

Ниодного

+ Параметры

		b_id	build_name	is_deleted
☐	Изменить	1	1 корпус	0
☐	Изменить	2	2 корпус	0
☐	Изменить	3	3 корпус	0
☐	Изменить	4	4 корпус	0
☐	Изменить	5	5 корпус	0
☐	Изменить	6	6 корпус	0
☐	Изменить	7	7 корпус	0
☐	Изменить	8	8 корпус	0
☐	Изменить	9	9 корпус	0
☐	Изменить	10	10 корпус	0

Рисунок 30 - тестирование запросов.

✔ Отображение строк 0 - 24 (75 всего, Запрос занял 0.0183 сек.)

```
SELECT * FROM rooms LEFT JOIN address ON rooms.address_id=address.id RIGHT JOIN build ON rooms.building_id=build.b_id INNER JOIN department ON rooms.owner=department.d_id
```

☐ Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

1 > >> | ☐ Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице | Сортировать по индексу: Ни одного

✔ Отображение строк 0 - 11 (12 всего, Запрос занял 0.0044 сек.)

SELECT * FROM department

☐ Профилитрование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

☐ Показать все | Количество строк: | Фильтровать строки: | Сортировать по индексу:

+ Параметры		d_id	dep_name	phone	head	is_deleted
☐	Изменить Копировать Удалить	1	Кафедра "Общая психология"	987654128	Иванова Наталья Ивановна	0
☐	Изменить Копировать Удалить	544152	Кафедра "Информационно-вычислительные системы"	987654123	Бобрышева Галина Владимировна	0
☐	Изменить Копировать Удалить	544170	Кафедра "Вычислительная техника"	784123682	Митрохин Максим Александрович	0
☐	Изменить Копировать Удалить	544206	Кафедра "Информационное обеспечение управления и п..."	+7 (8412) 36-82-39	Катышева Марина Александровна	0
☐	Изменить Копировать Удалить	544242	Кафедра "Математическое обеспечение и применение Э..."	+7(412) 36-82-26	Макарычев Петр Петрович	0
☐	Изменить Копировать Удалить	544368	Кафедра "Высшая и прикладная математика"	+7(8412)36-82-59	Бойков Илья Владимирович	0
☐	Изменить Копировать Удалить	544842	Кафедра "Математика и суперкомпьютерное моделирова..."	+7(8412)36-80-96	Смирнов Юрий Геннадьевич	0
☐	Изменить Копировать Удалить	31429808	Факультет вычислительной техники	+7(8412)36-82-48	Фионова Людмила Римовна	0
☐	Изменить Копировать Удалить	123595545	Кафедра "Компьютерные технологии"	+7(8412)36-80-68	Горбаченко Владимир Иванович	0

Кроме добавления, удаления, редактирования и просмотра элементов у пользователя есть возможность поиска записей в базе данных. Для этого необходимо зайти в пользовательский интерфейс и перейти на вкладку «Поиск», после чего выбрать необходимый тип поиска.

При выборе «Поиск по подразделению» пользователю откроется форма для заполнения, в которой необходимо выбрать подразделение, а затем и корпус. Все подразделения выводятся с помощью функции `getAllDepartment`, а после выбора одного из подразделений вызывается `findDepartment`, с помощью

					ПГУ 2.090301.11.001 ПЗ	Лист
						37
Изм.	Лист	№ докум.	Подпись	Дата		

этой функции формируется запрос по поиску всей информации о выбранном подразделении, полученная информация появляется в таблице веб-интерфейса. Затем вызывается функция findBuildings, осуществляющая поиск всех зданий, принадлежащих выбранному подразделению и подконтрольным ему подразделениям. После выбора здания срабатывает следующая функция – getAllPCInBuilding, эта функция осуществляет поиск всех компьютеров указанного подразделения и дочерних ему подразделений в выбранном здании. Полученный список компьютеров заключается в таблицу на интерфейсе. Весь процесс поиска по подразделению проиллюстрирован на рисунках 33-36.

ПГУ Главная Пользователи Подразделения

Поиск по подразделению

Поиск по ip-адресу

Рисунок 33 - веб-интерфейс: поиск.

ПГУ Главная Пользователи Подразделения

Поиск по подразделению

Поиск по ip-адресу

Выберите подразделение

Кафедра "Общая психология"
Кафедра "Информационно-вычислительные системы"
Кафедра "Вычислительная техника"
Кафедра "Информационное обеспечение управления и производства"
Кафедра "Математическое обеспечение и применение ЭВМ"
Кафедра "Высшая и прикладная математика"
Кафедра "Математика и суперкомпьютерное моделирование"
Факультет вычислительной техники
Кафедра "Компьютерные технологии"
Управление информатизации
Политехнический институт
test

Рисунок 34 - веб-интерфейс: поиск.

Поиск по подразделению

Поиск по ip-адресу

Выберите подразделение

▼

Подразделение	Глава	Номер
Политехнический институт	Козлов Геннадий Васильевич	+7(8412)36-84-56

Выберите корпус

▼

7 корпус

8 корпус

9 корпус

Рисунок 35 - веб-интерфейс: поиск.

Поиск по подразделению

Поиск по ip-адресу

Выберите подразделение

▼

Подразделение	Глава	Номер
---------------	-------	-------

Выберите корпус

▼

#	Адрес	Аудитория	Корпус	Подразделение
☐	172.16.7.10	7-223	8 корпус	Факультет вычислительной техники
☐	172.16.7.10	8-703а	8 корпус	Факультет вычислительной техники
☐	172.16.7.10	8-703б	8 корпус	Факультет вычислительной техники
☐	172.16.7.12	8-705	8 корпус	Факультет вычислительной техники
☐	172.16.7.10	8-706	8 корпус	Факультет вычислительной техники

Рисунок 36 - веб-интерфейс: поиск.

При выборе «Поиск по ip-адресу» появляется форма для ввода IP – адреса искомого компьютера, после чего вызывается функция getPCByIP, которая осуществляет поиск требуемой информации по адресу компьютера в сети и выводит результат на экран. Процесс изображен на рисунках 37 и 38.

Поиск по подразделению

Поиск по ip-адресу

Введите IP

172.16.7.12

Найти

#	Адрес	Аудитория	Корпус	Подразделение
---	-------	-----------	--------	---------------

Рисунок 37 - веб-интерфейс: поиск.

Поиск по подразделению

Поиск по ip-адресу

Введите IP

Ipaddress

Найти

#	Адрес	Аудитория	Корпус	Подразделение
☐	172.16.7.12	8-705	8 корпус	Факультет вычислительной техники

Рисунок 38 - веб-интерфейс: поиск.

Результаты тестирования запросов приведены на рисунках 39 - 41.

Отображение строк 0 - 2 (3 всего, Запрос занял 0.0152 сек.)

```
SELECT DISTINCT build.b_id, build.build_name FROM department,build,rooms WHERE department.d_id=rooms.owner AND rooms.building_id=build.b_id AND (department.d_id=352685446 OR department.d_id IN (SELECT tree.descendant FROM department JOIN tree ON (department.d_id = tree.descendant) WHERE tree.ancestor = 352685446))
```

☐ Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

☐ Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице

+ Параметры

	b_id	build_name
☐ Изменить ☐ Копировать ☐ Удалить	7	7 корпус
☐ Изменить ☐ Копировать ☐ Удалить	8	8 корпус
☐ Изменить ☐ Копировать ☐ Удалить	9	9 корпус

Рисунок 39 - тестирование запроса.

Отображение строк 0 - 4 (5 всего, Запрос занял 0.0198 сек.) [room_name: [BLOB - 5 БАЙТ]... - [BLOB - 5 БАЙТ]...]

```
SELECT * FROM 'rooms' RIGHT JOIN address ON rooms.address_id = address.id RIGHT JOIN build ON rooms.building_id = build.b_id INNER JOIN department ON department.d_id = rooms.owner WHERE build.b_id = 8 AND (department.d_id = 352685446 OR department.d_id IN (SELECT tree.descendant FROM department JOIN tree ON (department.d_id = tree.descendant) WHERE tree.ancestor = 352685446)) ORDER BY room_name ASC
```

☐ Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

☐ Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице | Сортировать по индексу: Ниодного

+ Параметры

r_id	room_name	building_id	address_id	owner	id	ip_address	b_id	build_name	is_deleted	d_id	dep_name	phone	head	is_c
6894	7-223	8	1	31429808	1	172.16.7.10	8	8 корпус	0	31429808	Факультет вычислительной техники	+7(8412)36-82-48	Фионова Людмила Римовна	
6953	8-703a	8	1	31429808	1	172.16.7.10	8	8 корпус	0	31429808	Факультет вычислительной техники	+7(8412)36-82-48	Фионова Людмила Римовна	
6954	8-7036	8	1	31429808	1	172.16.7.10	8	8 корпус	0	31429808	Факультет вычислительной техники	+7(8412)36-82-48	Фионова Людмила Римовна	
6912	8-705	8	3	31429808	3	172.16.7.12	8	8 корпус	0	31429808	Факультет вычислительной техники	+7(8412)36-82-48	Фионова Людмила Римовна	
6937	8-706	8	1	31429808	1	172.16.7.10	8	8 корпус	0	31429808	Факультет вычислительной техники	+7(8412)36-82-48	Фионова Людмила Римовна	

Рисунок 40 - тестирование запроса.

✓ Отображение строк 0 - 0 (1 всего, Запрос занял 0.0114 сек.) [room_name: [BLOB - 5 БАЙТ]... - [BLOB - 5 БАЙТ]...]

```
SELECT * FROM rooms RIGHT JOIN address ON rooms.address_id = address.id RIGHT JOIN build ON rooms.building_id = build.b_id INNER JOIN department ON rooms.owner = department.d_id WHERE address.ip_address = "172.16.7.12" ORDER BY room_name ASC
```

☐ Профилирование [[Построчное редактирование](#)] [[Изменить](#)] [[Анализ SQL запроса](#)] [[Создать PHP-код](#)] [[Обновить](#)]

☐ Показать все | Количество строк: 25 | Фильтровать строки:

+ Параметры

r_id	room_name	building_id	address_id	owner	id	ip_address	b_id	build_name	is_deleted	d_id	dep_name	phone	head	is_de
6912	8-705	8	3	31429808	3	172.16.7.12	8	8 корпус	0	31429808	Факультет вычислительной техники	+7(8412)36- 82-48	Фионова Людмила Римовна	

Рисунок 41 - тестирование запроса.

3.3.6 Поиск записей в базе данных

Также в приложении были реализованы функции для резервного копирования и восстановления, названные соответственно backupDataBase и recoveryDataBase. Они вызываются при нажатии соответствующих кнопок, которые называются «Создать резервную копию базы данных» и «Восстановить базу данных». Интерфейс резервного копирования и восстановления представлен на рисунке 42.

☐	172.16.7.10	7-212	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-212a	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-223	8 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-224	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-225	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-225a	7 корпус	Факультет вычислительной техники
☐	172.16.7.12	8-705	8 корпус	Факультет вычислительной техники
☐	172.16.7.10	8-706	8 корпус	Факультет вычислительной техники
☐	172.16.7.10	8-703a	8 корпус	Факультет вычислительной техники
☐	172.16.7.10	8-703б	8 корпус	Факультет вычислительной техники
☐	172.16.7.10	8-703в	8 корпус	Кафедра "Высшая и прикладная математика"
☐	172.16.7.10	8-703г	8 корпус	Кафедра "Высшая и прикладная математика"
☐	172.16.7.10	8-704	8 корпус	Кафедра "Высшая и прикладная математика"
☐	172.16.7.10	9-503	9 корпус	Факультет вычислительной техники
☐	192.168.10.11	7-100	4 корпус	Кафедра "Математическое обеспечение и применение ЭВМ"

Рисунок 42 - веб-интерфейс: резервное копирование и восстановление.

4 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

4.1 Введение

Современный человек привык к различного рода вычислительной технике и просто не может представить свою жизнь без персональных компьютеров, которые не только помогают в работе и сложных расчетах, но и скрашивают досуг. Эти устройства безусловно имеет множество преимуществ. Однако, если не соблюдать ряда предостережений, касающихся времени работы за компьютером, освещенности рабочего места и эргономики рабочего пространства, можно заметить влияние на организм множества вредных факторов.

4.2 Вредные факторы

Вредные факторы бывают физическими, химическими и психофизиологическими.

К физическим вредным факторам чаще всего относят:

- повышенный уровень электромагнитного излучения.
- повышенный уровень рентгеновского излучения.
- повышенный уровень ультрафиолетового излучения.
- повышенный уровень инфракрасного излучения.
- повышенный уровень статического электричества и запыленности

воздуха рабочей зоны.

- повышенный уровень светового загрязнения.
- неравномерность распределения яркости в поле зрения.
- повышенная яркость экрана монитора.

К химическим факторам относятся:

- повышенное содержание в воздухе рабочей зоны двуокиси углерода.
- повышенное содержание в воздухе рабочей зоны озона.
- повышенное содержание в воздухе рабочей зоны аммиака.

- повышенное содержание в воздухе рабочей зоны фенола.

Лист

42

ПГУ 2.090301.11.001 ПЗ

- повышенное содержание в воздухе рабочей зоны формальдегида.

К психофизиологическим относятся:

- напряжение зрения и внимания.
- длительные эмоциональные и интеллектуальные нагрузки.
- монотонность труда.
- большой поток обрабатываемой информации.
- Плохая эргономика рабочего места.

4.3 Требования к организации рабочих мест

- Необходимо расположить рабочее место таким образом, чтобы естественный свет, падающий на него был сбоку.

- Не допускается располагать рабочие места таким образом, чтобы естественный свет падал на них со стороны спины или лица пользователя.

- Расстояние между рабочими столами должны составлять не менее 2 метров перед пользователем и за ним, а слева и справа – 1,2 метров.

- Проходы между рабочими местами должны быть достаточно широкими, чтобы обеспечить свободное перемещение персонала, не касаясь оборудования или материалов, расположенных на рабочем месте. Требуемая минимальная ширина 0,6 м, оптимальная 0,9 м.

- Рабочие места с компьютерами при выполнении творческой работы, требующей значительного умственного напряжения или высокой концентрации внимания, должны быть изолированы друг от друга непрозрачными перегородками высотой 1,5-2,0 м.

- Рекомендуется использование специальных компьютерных столов в сочетании с письменными столами.

- Высота рабочей поверхности стола должна регулироваться в пределах 680-800 мм, либо составлять 725 мм.

- Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной – не менее 500 мм, глубиной на уровне колен – не менее 450 мм и на уровне вытянутых ног – не менее 650 мм.

					ПГУ 2.090301.11.001 ПЗ	Лист
						43
Изм.	Лист	№ докум.	Подпись	Дата		

- Конструкция рабочего стула должна обеспечивать поддержание эргономичной рабочей позы, позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления.

- Рабочее место должно быть оборудовано подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20 градусов. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

- Экран видеомонитора должен находиться от глаз пользователя на оптимальном расстоянии, составляющем 600-700 мм.

- Клавиатуру следует располагать на поверхности стола на расстоянии 100-300 мм от края, обращенного к пользователю или на специальной, регулируемой по высоте рабочей поверхности, отделенной от основной столешницы.

4.4 Требования к освещению рабочих мест

- Помещения, предназначенные для размещения рабочих мест пользователей персональных компьютеров, должны иметь как естественное, так и искусственное освещение.

- Естественное освещение помещений должно осуществляться через световые проемы, ориентированные преимущественно на север и северо-восток.

- В случае иной ориентации световых проемов, необходимо предусматривать эффективные средства регулирования интенсивности естественного освещения.

- Оконные проемы необходимо оборудовать жалюзи или занавесями.

- Искусственное освещение в помещениях должно осуществляться системой общего равномерного освещения.

- Для использования в качестве источников местного освещения пригодны лампы, позволяющие избежать возникновения ослепления и бликов.

					ПГУ 2.090301.11.001 ПЗ	Лист
						44
Изм.	Лист	№ докум.	Подпись	Дата		

- Местное освещение на рабочих местах пользователей персональных компьютеров не должно создавать бликов на поверхности экрана.

- Применять светильники без сертифицированных рассеивателей и экранирующих решеток не допускается.

- Общее освещение следует выполнять в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения пользователя при рядном расположении рабочих мест. При периметральном расположении рабочих мест линии светильников должны располагаться локализовано над рабочим столом ближе к его переднему краю, обращенному к оператору.

- Для обеспечения нормируемых значений освещенности в помещениях следует проводить чистку стекол оконных рам и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

- Так как при работе на компьютере основная нагрузка ложится на глаза, поэтому большие требования предъявляются к экранам устройств. Предпочтительным является плоский экран, позволяющий избежать наличие на нем ярких пятен за счет отражения световых потоков. Особенно важен цвет экрана. Он должен быть нейтральным. Допустимы ненасыщенные светло-зеленые, желто-зеленые, желто-оранжевые, желто-коричневые тона.

- Для устранения бликов и снижения влияния электромагнитного излучения экраны должны быть снабжены защитными фильтрами.

4.5 Соблюдение режима труда и отдыха

Режим труда и отдыха при работе с компьютером должен организовываться в зависимости от вида и категории деятельности. Виды деятельности подразделяются на следующие группы:

- группа А - работа по считыванию информации с монитора с предварительным запросом;

- группа Б - работа по вводу информации;

					ПГУ 2.090301.11.001 ПЗ	Лист 45
Изм.	Лист	№ докум.	Подпись	Дата		

- группа В - творческая работа в режиме диалога.
- Для преподавателей высших и средних специальных учебных заведений, учителей общеобразовательных школ устанавливается длительность работы в компьютерных классах и кабинетах информатики и вычислительной техники не более 4 часов в день.
- Для инженеров, обслуживающих учебный процесс в кабинетах (аудиториях) с персональными компьютерами, продолжительность работы не должна превышать 6 часов в день.
- Для пользователей в возрасте до 18 лет длительность работы с использованием персонального компьютера не должна превышать 3 часов в день.
- Для обеспечения оптимальной работоспособности и сохранения здоровья профессиональных пользователей, на протяжении рабочей смены должны устанавливаться регламентированные перерывы.
- Время регламентированных перерывов в течение рабочей смены следует устанавливать в зависимости от ее продолжительности, вида и категории трудовой деятельности. Режим труда и отдыха операторов, работающих с ЭВМ, должен быть следующим: через каждый час интенсивной работы необходимо устраивать 15 - минутный перерыв, при менее интенсивной через каждые 2 - часа.
- Эффективность регламентируемых перерывов повышается при их сочетании с производственной гимнастикой. Производственная гимнастика должна включать комплекс упражнений, направленных на восполнение дефицита двигательной активности, снятие напряжения мышц шеи, спины, снижение утомления зрения. Она проводится в течение 5 - 7 мин. 1 - 2 раза в смену.
- При 12-ти часовой рабочей смене регламентированные перерывы должны устанавливаться в первые 8 часов работы аналогично перерывам при 8-ми часовой рабочей смене, а в течение последних 4 часов работы, независимо от категории и вида работ, каждый час продолжительностью 15 минут.

- Пользователи, использующие персональные компьютеры в качестве основного производственного средства, должны проходить обязательные предварительные (при поступлении на работу) и периодические медицинские осмотры в порядке и в сроки, установленные Минздравом России.

- К непосредственным работам, связанным с использованием персонального компьютера, допускаются лица, не имеющие медицинских противопоказаний.

- Женщины со времени установления беременности и в период кормления ребенка грудью к выполнению всех видов работ, связанных с использованием компьютеров и копировально-множительной техники, не допускаются (СанПиН 2.2.2-542-96 п. 10.3., СанПиН 2.2.0.555-96 п. 4.1.10.).

- Трудоустройство беременных женщин следует осуществлять в соответствии с «Гигиеническими рекомендациями по рациональному трудоустройству беременных женщин».

					ПГУ 2.090301.11.001 ПЗ	Лист
						47
Изм.	Лист	№ докум.	Подпись	Дата		

ЗАКЛЮЧЕНИЕ

В результате данной выпускной квалификационной работы было реализовано приложение для учета пользователей и компьютеров локальной вычислительной сети пензенского государственного университета. В рамках разработки этого приложения была создана таблица, веб-интерфейс и функции для их беспрепятственного взаимодействия, а также оформлена документация на приложение. Проведенное тестирование показало, что часть приложения, разработанная в ходе выпускной квалификационной работы, работает правильно.

В дальнейшем приложение может быть улучшено за счет расширения базы данных, усложнения структуры и добавления новых функций взаимодействия.

В ходе выполнения выпускной квалификационной работы был значительно повышен уровень практических навыков по созданию веб-приложений, освоен процесс разработки баз данных для приложения. Были рассмотрены такие средства, как: MAMP, СУБД MySQL, PhpMyAdmin, Bootstrap. Получены навыки работы с множеством языков программирования: PHP, HTML, CSS, JavaScript и SQL – запросов.

					ПГУ 2.090301.11.001 ПЗ	Лист
						48
Изм.	Лист	№ докум.	Подпись	Дата		

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Л. Томсон, Л. Веллинг. Разработка web-приложений на PHP и MySQL, 5-е изд. Москва, Санкт-Петербург, Киев, 2017. 652 с.
2. Р. Никсон. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5, 4-е изд. 2016. 766 с.
3. К. Янк. PHP и MySQL. От новичка к профессионалу, 5-е изд. Москва, 2013. 585 с.
4. К. Шмитт. CSS. Рецепты программирования, 3-е изд. Санкт – Петербург, 2011. 672 стр.
5. А. Бьюли. Изучаем SQL. Санкт-Петербург – Москва, 2007. 309 с.
6. Н. Прохоренок, В. Дронов. HTML, JavaScript, PHP и MySQL. Джентельменский набор web-мастера, 5-е изд. Санкт – Петербург, 2019. 912 с.
7. Д. Дакетт. HTML и CSS. Разработка и дизайн веб-сайтов. Санкт – Петербург, 2013. 480 с.
8. Д. Макфарланд. Новая большая книга CSS. Санкт – Петербург, 2018. 720 с.
9. Н. Закас. JavaScript для профессиональных веб-разработчиков. Санкт – Петербург, 2015. 960 с.
10. Э. Фримен, Э. Робсон. Изучаем программирование на JavaScript. Санкт – Петербург, 2015. 640 с.
11. MySQL [Электронный ресурс]. – Электрон.дан. URL: <http://www.mysql.com/>
12. Д. Флэнаган. Javascript. Подробное руководство, 6-е изд. СанктПетербург – Москва, 2012. 1080 с. 11. Д. Флэнаган. JavaScript. Карманный справочник, 3-е изд. Москва, Санкт-Петербург, Киев, 2013. 314 с.
13. Организация труда на предприятии [Электронный ресурс] // Организация труда на предприятии Режим доступа URL <http://www.ur-pro.ru/encyclopedia/organizaciya-truda.html> [Дата обращения 10.05.20 20 г.].

14. Опасные и вредные факторы при работе с ПЭВМ [электронный ресурс] // Опасные и вредные факторы при работе с ПЭВМ Режим доступа URL <http://rgrtu-640.ru/bezopasnost-zhiznedeyatelnosti/45.html> [Дата обращения 18.05.2020 г.]

					ПГУ 2.090301.11.001 ПЗ	Лист
						50
Изм.	Лист	№ докум.	Подпись	Дата		

ПРИЛОЖЕНИЯ

Приложение А Листинг программы

```
<?php

function getAllStatus($db) {

    $sql = "SELECT * FROM status";

    $result = array();

    $stmt = $db->prepare($sql);

    $stmt->execute();

    while($row = $stmt->fetch(PDO::FETCH_ASSOC)) {

        $result[$row['status_id']] = $row;

    }

    return $result;
}

function getAllUsers($db) {

    $sql = "SELECT * FROM users LEFT JOIN status ON
users.status_id=status.status_id

    ";

    $result = array();

    $stmt = $db->prepare($sql);

    $stmt->execute();

    //print_r($stmt);

    while($row = $stmt->fetch(PDO::FETCH_ASSOC)) {

        $result[$row['user_id']] = $row;

        // echo "<pre>";

        //print_r($result[$row['user_id']]);
    }
}
```

					ПГУ 2.090301.11.001 ПЗ	Лист
						51
Изм.	Лист	№ докум.	Подпись	Дата		

```

        //echo "</pre>";

    }

    return $result;

}

function getAllAddress($db) {

    $sql = "SELECT * FROM address";

    $result = array();

    $stmt = $db->prepare($sql);

    $stmt->execute();

    while($row = $stmt->fetch(PDO::FETCH_ASSOC)) {

        $result[$row['id']] = $row;

    }

    return $result;

}

function getAllBuildings($db) {

    $sql = "SELECT * FROM build";

    $result = array();

    $stmt = $db->prepare($sql);

    $stmt->execute();

    while($row = $stmt->fetch(PDO::FETCH_ASSOC)) {

        $result[$row['b_id']] = $row;

    }

    return $result;

}

function getAllRooms($db) {

```

					ПГУ 2.090301.11.001 ПЗ	Лист
						52
Изм.	Лист	№ докум.	Подпись	Дата		

```

$sql = "SELECT * FROM rooms
LEFT JOIN address ON rooms.address_id=address.id
RIGHT JOIN build ON rooms.building_id=build.b_id
INNER JOIN department ON rooms.owner=department.d_id
" ;

```

```

$result = array();

```

```

$stmt = $db->prepare($sql);

```

```

$stmt->execute();

```

```

while($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
    $result[$row['r_id']] = $row;
}

```

```

return $result;

```

```

}

```

```

function getAllDepartment($db) {

```

```

    $sql = "SELECT * FROM department";

```

```

    $result = array();

```

```

    $stmt = $db->prepare($sql);

```

```

    $stmt->execute();

```

```

while($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
    $result[$row['d_id']] = $row;
}

```

```

return $result;

```

```

}

```

```

function addUser($db, $login, $password, $name, $email, $status_id){

```

					ПГУ 2.090301.11.001 ПЗ	Лист
						53
Изм.	Лист	№ докум.	Подпись	Дата		

```

$password=password_hash( $password,PASSWORD_DEFAULT);

$sql = "INSERT INTO users(login, password, name, email, status_id)
        VALUES(:login, :password, :name, :email, :status_id)";

$stmt = $db->prepare($sql);

$stmt->bindValue(':login', $login, PDO::PARAM_STR);

$stmt->bindValue(':password', $password, PDO::PARAM_STR);

$stmt->bindValue(':name', $name, PDO::PARAM_STR);

$stmt->bindValue(':email', $email, PDO::PARAM_STR);

$stmt->bindValue(':status_id', $status_id, PDO::PARAM_INT);

$stmt->execute();

}

```

```

function addTree($db, $ancestor, $head, $phone) {

    $sql = "INSERT INTO tree(ancestor)
            VALUES(:ancestor)

    ";

    $stmt = $db->prepare($sql);

    $stmt->bindValue(':ancestor', $ancestor, PDO::PARAM_INT);

    $stmt->execute();

}

```

```

function getDepartmentById($db, $id) {

    $sql = "SELECT * FROM department
            WHERE d_id = :d_id;

    ";

    $stmt = $db->prepare($sql);

    $stmt->bindValue('d_id', $id, PDO::PARAM_INT);

```

					ПГУ 2.090301.11.001 ПЗ	Лист
						54
Изм.	Лист	№ докум.	Подпись	Дата		

```

$stmt->execute();

$row = $stmt->fetch(PDO::FETCH_ASSOC);

return $row;

}

function getUserById($db, $user_id) {

    $sql = "SELECT * FROM users

        WHERE user_id = :user_id;";

    $stmt = $db->prepare($sql);

    $stmt->bindValue('user_id', $user_id, PDO::PARAM_INT);

    $stmt->execute();

    $row = $stmt->fetch(PDO::FETCH_ASSOC);

    return $row;

}

function deleteUser($db, $u_id)

{

    $sql = "DELETE FROM users

    WHERE users.user_id = :u_id";

    $stmt = $db->prepare($sql);

    $stmt->bindValue('u_id', $u_id, PDO::PARAM_INT);

    $stmt->execute();

}

function changeUser($db, $user_id, $login = NULL, $password = NULL, $name =
NULL, $email = NULL, $status_id = NULL)

{

    $password=password_hash( $password,PASSWORD_DEFAULT);

    $sql = "UPDATE users

    SET login = (SELECT IFNULL(:login,login)),

    password = (SELECT IFNULL(:password,password)),

    name = (SELECT IFNULL(:name,name)),

    email = (SELECT IFNULL(:email,email)),

    status_id = (SELECT IFNULL(:status_id,status_id))

    WHERE user_id = :user_id;";

    $stmt = $db->prepare($sql);

    $stmt->bindValue('login', $login, PDO::PARAM_STR);

    $stmt->bindValue('password', $password, PDO::PARAM_STR);

    $stmt->bindValue('name', $name, PDO::PARAM_STR);

    $stmt->bindValue('email', $email, PDO::PARAM_STR);

    $stmt->bindValue('status_id', $status_id, PDO::PARAM_INT);

    $stmt->bindValue('user_id', $user_id, PDO::PARAM_INT);

    $stmt->execute();

}

```

```

        name = (SELECT IFNULL(:name,name)),
        email = (SELECT IFNULL(:email,email)),
        status_id = (SELECT IFNULL(:status_id, status_id))
WHERE users.user_id = :user_id";

$stmt = $db->prepare($sql);

$stmt->bindValue(':user_id', $user_id, PDO::PARAM_INT);
$stmt->bindValue(':login', $login, PDO::PARAM_STR);
$stmt->bindValue(':password', $password, PDO::PARAM_STR);
$stmt->bindValue(':name', $name, PDO::PARAM_STR);
$stmt->bindValue(':email', $email, PDO::PARAM_STR);
$stmt->bindValue(':status_id', $status_id, PDO::PARAM_INT);

$stmt->execute();
}

```

```
function deleteDepartment($db, $d_id)
```

```

{
    $sql = "DELETE FROM department
WHERE department.d_id = :d_id";

    $stmt = $db->prepare($sql);

    $stmt->bindValue('d_id', $d_id, PDO::PARAM_INT);

    $stmt->execute();
}

```

```
function deleteMainTableById($db, $r_id)
```

```

{
    $sql = "DELETE FROM rooms
WHERE rooms.r_id = :r_id";

    $stmt = $db->prepare($sql);

    $stmt->bindValue('r_id', $r_id, PDO::PARAM_INT);

    $stmt->execute();
}

```

					ПГУ 2.090301.11.001 ПЗ	Лист
						56
Изм.	Лист	№ докум.	Подпись	Дата		


```

UPDATE `address`,`rooms`,`department`,`build`

SET `rooms`.`address_id`=`address`.`id`,

`rooms`.`owner` = `department`.`d_id`,

`rooms`.`building_id` = `build`.`b_id`,

`rooms`.`room_name` = (SELECT IFNULL(:room_name,room_name))

WHERE `address`.`ip_address` = (SELECT IFNULL(:ip_address,ip_address))

AND `build`.`build_name` = (SELECT IFNULL(:build_name,build_name))

AND `department`.`dep_name` = (SELECT IFNULL(:dep_name,dep_name))

AND `rooms`.`r_id` = :r_id";

$stmt = $db->prepare($sql);

$stmt->bindValue('ip_address', $ip_address, PDO::PARAM_INT);

$stmt->bindValue('room_name', $room_name, PDO::PARAM_INT);

$stmt->bindValue('build_name', $build_name, PDO::PARAM_INT);

$stmt->bindValue('dep_name', $dep_name, PDO::PARAM_INT);

$stmt->bindValue('r_id', $r_id, PDO::PARAM_INT);

$stmt->execute();

}

function findDepartment($db, $d_id) {

    $sql = "SELECT * FROM department WHERE d_id=:d_id";

    $result = array();

    $stmt = $db->prepare($sql);

    $stmt->execute(array('d_id' => $d_id));

    while($row = $stmt->fetch(PDO::FETCH_ASSOC)) {

        $result[$row['d_id']] = $row;

    }

    return $result;

}

```

```

function findBuildings($db, $d_id) {

    $sql = "SELECT DISTINCT build.b_id, build.build_name
            FROM department,build,rooms
            WHERE department.d_id=rooms.owner
            AND rooms.building_id=build.b_id
            AND (department.d_id=:d_id
            OR department.d_id
            IN (SELECT tree.descendant
            FROM department JOIN tree
            ON (department.d_id = tree.descendant)
            WHERE tree.ancestor = :d_id))";

    $result = array();

    $stmt = $db->prepare($sql);

    $stmt->bindValue(':d_id', $d_id, PDO::PARAM_INT);

    $stmt->execute();

    while($row = $stmt->fetch(PDO::FETCH_ASSOC)) {

        $result[$row['b_id']] = $row;

    }

    return $result;
}

function getAllPCInBuilding($db, $b_id, $d_id) {

    $sql = "SELECT * FROM `rooms`

            RIGHT JOIN address ON rooms.address_id = address.id

            RIGHT JOIN build ON rooms.building_id = build.b_id

            INNER JOIN department ON department.d_id = rooms.owner

            WHERE build.b_id = :b_id AND (department.d_id = :d_id OR
            department.d_id

```

					ПГУ 2.090301.11.001 ПЗ	Лист
						59
Изм.	Лист	№ докум.	Подпись	Дата		

```

        IN (SELECT tree.descendant

        FROM department JOIN tree

        ON (department.d_id = tree.descendant)

        WHERE tree.ancestor = :d_id))

        ORDER BY room_name ASC" ;

$result = array();

$stmt = $db->prepare($sql);

$stmt->execute(array('b_id' => $b_id, 'd_id' => $d_id));

while($row = $stmt->fetch(PDO::FETCH_ASSOC)) {

    $result[$row['r_id']] = $row;

    // echo "<pre>";

    //print_r($result[$row['r_id']]);

    //echo "</pre>";

}

return $result;

}

function getPCByIP($db, $ip) {

    $sql = "SELECT * FROM rooms

            RIGHT JOIN address ON rooms.address_id = address.id

            RIGHT JOIN build ON rooms.building_id = build.b_id

            INNER JOIN department ON rooms.owner = department.d_id

            WHERE address.ip_address = :ip

            ORDER BY room_name ASC" ;

    $result = array();

    $stmt = $db->prepare($sql);

    $stmt->execute(array('ip' => $ip));

```

					ПГУ 2.090301.11.001 ПЗ	Лист
						60
Изм.	Лист	№ докум.	Подпись	Дата		

```

while($row = $stmt->fetch(PDO::FETCH_ASSOC)) {

    $result[$row['r_id']] = $row;

}

return $result;

}

function addMainTable($db, $ip, $room_name, $building_id, $owner){

    $sql="INSERT INTO address(ip_address) VALUES(:ip)";

    $stmt = $db->prepare($sql);

    $stmt->bindValue(':ip', $ip, PDO::PARAM_STR);

    $stmt->execute();

    $address_id = $db->lastInsertId();

    $query = "INSERT INTO rooms(room_name, building_id, address_id, owner)

        VALUES(:room_name, :building_id, :address_id, :owner )";

    $stmt1 = $db->prepare($query);

    $stmt1->bindValue(':room_name', $room_name, PDO::PARAM_STR);

    $stmt1->bindValue(':building_id', $building_id, PDO::PARAM_INT);

    $stmt1->bindValue(':address_id', $address_id, PDO::PARAM_INT);

    $stmt1->bindValue(':owner', $owner, PDO::PARAM_INT);

    $stmt1->execute();

}

function addDepartment($db, $ancestor, $dep_name, $head, $phone){

    $sql = "INSERT INTO department(dep_name, head, phone)

        VALUES(:dep_name, :head, :phone)

    ";

    $stmt = $db->prepare($sql);

    $stmt->bindValue(':dep_name', $dep_name, PDO::PARAM_STR);

    $stmt->bindValue(':head', $head, PDO::PARAM_STR);

    $stmt->bindValue(':phone', $phone, PDO::PARAM_INT);

```

					ПГУ 2.090301.11.001 ПЗ	Лист
						61
Изм.	Лист	№ докум.	Подпись	Дата		

```

$stmt->execute();

$descendant = $db->lastInsertId();

//var_dump($descendant);

//var_dump($ancestor);

$query = "INSERT INTO tree(ancestor, descendant)

        VALUES(:ancestor, :descendant)";

$stmt1 = $db->prepare($query);

$stmt1->bindValue(':ancestor', $ancestor, PDO::PARAM_INT);

    $stmt1->bindValue(':descendant', $descendant, PDO::PARAM_INT);

$stmt1->execute();

}

function backupDataBase($db) {

    $tables = array();

    $db->setAttribute(PDO::ATTR_ORACLE_NULLS, PDO::NULL_TO_STRING );

    //Script Variables

    $compression = false;

    $BACKUP_PATH = "";

    $backupname = "backup";

    //unlink('backup.sql');

    //create/open files

    if ($compression) {

        $zp = gzopen($BACKUP_PATH.$backupname.'.sql.gz', "w9");

    } else {

        $handle = fopen($BACKUP_PATH.$backupname.'.sql', 'w+');

    }
}

```

```

//array of all database field types which just take numbers

$numtypes=array('tinyint','smallint','mediumint','int','bigint','float','double','decimal','real');

//get all of the tables

if(empty($tables)) {

    $pstmt1 = $db->query('SHOW TABLES');

    while ($row = $pstmt1->fetch(PDO::FETCH_NUM)) {

        $tables[] = $row[0];

    }

} else {

    $tables = is_array($tables) ? $tables : explode(',',$tables);

}

//cycle through the table(s)

foreach($tables as $table) {

    $result = $db->query('SELECT * FROM '.$table);

    $num_fields = $result->columnCount();

    $num_rows = $result->rowCount();

    $return="";

    //uncomment below if you want 'DROP TABLE IF EXISTS' displayed

    //$return.= 'DROP TABLE IF EXISTS `'.$table.'`;';

    //table structure

    $pstmt2 = $db->query('SHOW CREATE TABLE '.$table);

    $row2 = $pstmt2->fetch(PDO::FETCH_NUM);

    $ifnotexists = str_replace('CREATE TABLE', 'CREATE TABLE IF NOT EXISTS', $row2[1]);

    $return.= "\n\n".$ifnotexists."; \n\n";

```

```

if ($compression) {
    gzwrite($zp, $return);
} else {
    fwrite($handle,$return);
}

$return = "";

//insert values
if ($num_rows){
    $return= 'INSERT INTO `'.$table.'"` (';
    $pstm3 = $db->query('SHOW COLUMNS FROM '.$table);
    $count = 0;
    $type = array();

    while ($rows = $pstm3->fetch(PDO::FETCH_NUM)) {

        if (stripos($rows[1], '(')) {
            $type[$table][] = stristr($rows[1], '(', true);
        } else
            $type[$table][] = $rows[1];

        $return.= $rows[0];
        $count++;
        if ($count < ($pstm3->rowCount())) {
            $return.= ", ";
        }
    }

    $return.= ")". ' VALUES';

    if ($compression) {

```

					ПГУ 2.090301.11.001 ПЗ	Лист
						64
Изм.	Лист	№ докум.	Подпись	Дата		

```

        gzwrite($zp, $return);

    } else {

        fwrite($handle,$return);

    }

    $return = "";

    $count = 0;

}

while($row = $result->fetch(PDO::FETCH_NUM)) {

    $return= "\n\t(";

    for($j=0; $j<$num_fields; $j++) {

        $row[$j] = addslashes($row[$j]);

        // $row[$j] = preg_replace("\n","\\n",$row[$j]);

        if (isset($row[$j])) {

            //if number, take away ". else leave as string

            if (in_array($type[$table][$j], $numtypes))

                $return.= $row[$j] ; else $return.= "'".$row[$j]."' " ;

        } else {

            $return.= '""';

        }

        if ($j<($num_fields-1)) {

            $return.= ',';

        }

    }

    $count++;

    if ($count < ($result->rowCount())) {

        $return.= "),";

    } else{

        $return.= ");";

    }

```

					ПГУ 2.090301.11.001 ПЗ	Лист
						65
Изм.	Лист	№ докум.	Подпись	Дата		

```

        if ($compression) {

            gzwrite($zp, $return);

        } else {

            fwrite($handle,$return);

        }

        $return = "";

    }

    $return="\n\n-- -----
\n\n";

    if ($compression) {

        gzwrite($zp, $return);

    } else {

        fwrite($handle,$return);

    }

    $return = "";

}

$error1= $pstm2->errorInfo();
$error2= $pstm3->errorInfo();
$error3= $result->errorInfo();

echo $error1[2];
echo $error2[2];
echo $error3[2];

if ($compression) {

    gzclose($zp);

} else {

    fclose($handle);

}

}

```

					ПГУ 2.090301.11.001 ПЗ	Лист
						66
Изм.	Лист	№ докум.	Подпись	Дата		

```

function recoveryDataBase($db)
{
    $sql1 = "DROP TABLE rooms, tree, users, address, build, department,
status";

    $sql2 = file_get_contents('backup.sql');

    $stmt = $db->prepare($sql1);

    $stmt->execute();

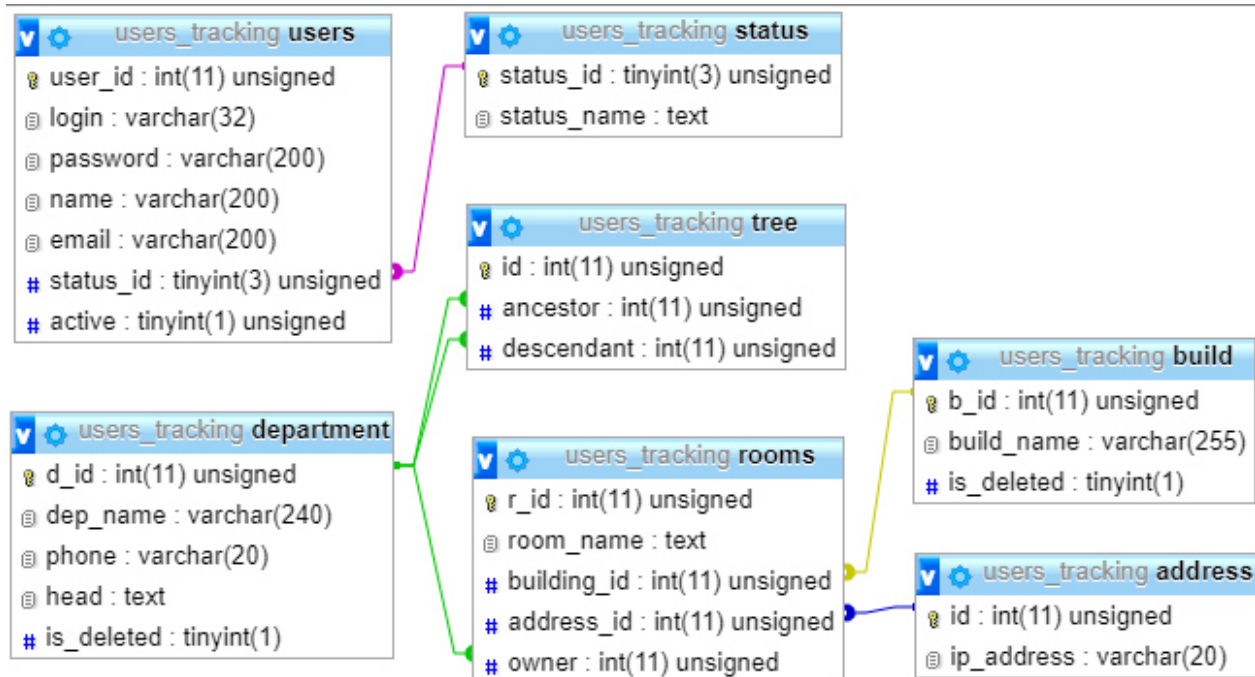
    $stmt = $db->prepare($sql2);

    $stmt->execute();
}

```

					ПГУ 2.090301.11.001 ПЗ	Лист
						67
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение Б Модель базы данных



Система учета пользователей ЛВС ПГУ. Поиск и редактирование

НАУЧНЫЙ РУКОВОДИТЕЛЬ: К.Т.Н., ДОЦЕНТ КАФЕДРЫ «ВТ» ПОПОВ К. В.

СТУДЕНТ: ДУХАНОВ И. В.

ГРУППА: 16ВВ2

Разработка и внедрение.

Разработка и внедрение системы учета пользователей ЛВС ПГУ, и реализация этого проекта с использованием базы данных – MYSQL.



Задачи

Задача выпускной квалификационной работы есть разработка функций для работы с базой данных учета компьютерного фонда университета, в рамках веб-приложения.



SQL

```
function getAllUsers($db) {  
  
    $sql = "SELECT * FROM users  
        LEFT JOIN status  
        ON users.status_id=status.status_id";  
  
    $result = array();  
  
    $stmt = $db->prepare($sql);  
    $stmt->execute();  
  
    while($row = $stmt->fetch(PDO::FETCH_ASSOC)) {  
        $result[$row['user_id']] = $row;  
    }  
    return $result;  
}
```



Исследование

Объектом исследования выступает база данных компьютерного фонда ПГУ, а предметом исследования особенности составления алгоритмов работы с базой данных на базе платформы MYSQL.



Результаты работы

Логин

Пароль

Пользователи с указанным логином и паролем не существуют

Войти



ПГУ

Главная

Просмотр

Поиск

Подразделения

Пользователи

Создать резервную копию базы данных

Восстановить базу данных



ПГУ Главная Просмотр Поиск Подразделения Пользователи				
Добавить				
Удалить				
#	Адрес	Аудитория	Корпус	Подразделение
☐	172.16.7.10	7-112	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-117	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-209	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-211	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-307	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-407	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-425	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-502	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-503	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-505	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-508	7 корпус	Факультет вычислительной техники

ПГУ	Главная	Просмотр	Поиск	Подразделения	Пользователи
-----	---------	----------	-------	---------------	--------------

Добавить

Адрес

Аудитория

Добавить

Удалить

#	Адрес	Аудитория	Корпус	Подразделение
☐	172.16.7.10	7-112	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-117	7 корпус	Факультет вычислительной техники
☐	172.16.7.10	7-209	7 корпус	Факультет вычислительной техники

ПГУ Главная Просмотр Поиск Подразделения Пользователи

Адрес
 172.16.7.10

Аудитория
 7-112

Подразделение
 Факультет вычислительной техники

Корпус
 7 корпус

Сохранить

ПГУ Главная Просмотр Поиск Подразделения Пользователи

№	Название подразделения	Руководитель	Телефон
☐	Кафедра "Вычислительная техника"	Митрохин Максим Александрович	784123682
☐	Кафедра "Математическое обеспечение и применение ЭВМ"	Макарычев Петр Петрович	+7(8412) 36-82-26
☐	Кафедра "Высшая и прикладная математика"	Бойков Илья Владимирович	+7(8412)36-82-59
☐	Кафедра "Математика и суперкомпьютерное моделирование"	Смирнов Юрий Геннадьевич	+7(8412)36-80-96
☐	Факультет вычислительной техники	Фисенко Людмила Римовна	+7(8412)36-82-48
☐	Кафедра "Компьютерные технологии"	Горбаненко Владимир Иванович	+7(8412)36-80-68
☐	Политехнический институт	Козлов Геннадий Васильевич	+7(8412)36-84-56

Удалить подразделение

Добавить подразделение

#	Логин	Пароль	ФИО	e-mail	Тип пользователя
☐	admin11	71ea76dfb3d9026ee2c486ea54745b	ФИО	email	Оператор
☐	iladuh@mail.ru	pass	test	iladuh@mail.ru	Администратор
☐	oksasifa@yandex.ru	0eaa2793389d4fae91b25125db4e7cfc	ФИО	email	Администратор



#	Адрес	Аудитория	Корпус	Подразделение
Поиск по подразделению				
Выберите подразделение				
<div> <div>Подразделение</div> <div>Глава</div> <div>Номер</div> </div>				
Выберите корпус				
Поиск по IP-адресу				
Введите IP				
<input type="text" value="ipaddress"/> <input type="button" value="Найти"/>				
#	Адрес	Аудитория	Корпус	Подразделение



Конец.

Спасибо за внимание!



					ПГУ 2.090301.11.001 ПЗ	Лист
						75
Изм.	Лист	№ докум.	Подпись	Дата		