МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет	*Кафедра*
вычислительной техники	«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. Графическое представление расположения

Студент 03.06.2020 Воронков Евгений Витальевич
(подпись, дата) (ФИО полностью)

Руководитель 03.06.20 К.В. Попов
(подпись, дата) (фамилия, инициалы)

Нормоконтролёр А.В. Кучин
(подпись, дата) (фамилия, инициалы)

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

Заведующий кафедрой М.А. Митрохин
(подпись) (фамилия, инициалы)

Работа защищена с отметкой отлично (протокол заседания ГЭК от 11.06.2020 № 3)

Секретарь ГЭК Е.И. Гурин
(подпись) (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 2020г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ БАКАЛАВРА

1. Студент Воронков Евгений Витальевич гр.16ВВ2 факультета вычислительной техники

2. Тема работы: «Система учета пользователей ЛВС ПГУ. Графическое представление расположения»

Тема утверждена приказом ПГУ № 0335а/о от "30" апреля 2020г.

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 веб-интерфейс, связанный с базой данных, который позволит пользователю получить быстрый доступ к хранимой информации.

5. Перечень подлежащих разработке вопросов.
Требуется разработать веб-интерфейс для системы, осуществляющей учет пользователей ЛВС ПГУ. Он будет предоставлять доступ к данным о строениях, подразделениях, аудиториях и ip-адресах университета. Также, он предоставит администратору интерфейс для удаления, добавления и редактирования необходимой информации, а также средства авторизации для разграничения доступа для сотрудников университета, использующих систему.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Анализ предметной области	10	30.04.2020	
2.	Выбор средств разработки	10	08.05.2020	
3.	Изучение основных характеристик веб-интерфейсов	10	13.05.2020	
4.	Проектирование веб-интерфейса	20	18.05.2020	
5.	Разработка веб-интерфейса	40	26.05.2020	
6.	Оформление выпускной работы	20	28.05.2020	

Дата выдачи задания "30" апреля 2020 г.

Руководитель выпускной квалификационной работы бакалавра _____
(подпись, дата)

Задание к исполнению принял студент _____ 30.04.2020
(подпись, дата)

Работу к защите допустить
Декан факультета _____
(подпись, дата)

РЕФЕРАТ

Пояснительная записка содержит 87 листов, 22 рисунков, 12 источников, 2 приложения.

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

В рамках данной выпускной квалификационной работы был создан веб-интерфейс, с использованием языков HTML и PHP.

Объект исследования – веб-интерфейс системы учета пользователей ЛВС ПГУ.

Предмет исследования – особенности проектирования веб-интерфейса на базе библиотеки BOOTSTRAP.

Цель выпускной квалификационной работы – разработка и внедрение полученной системы учета пользователей ЛВС ПГУ, и реализация ее веб-интерфейса.

Задача выпускной квалификационной работы есть разработка веб-интерфейса системы учета пользователей ЛВС ПГУ.

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 разработки и администрирования применялся инструмент phpMyAdmin.

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....9
1.2.1 Spiceworks....10
1.2.2 Zabbix....10
1.2.3 Zabbix....11
1.2.4 IPplan....13
1.2.5 NeDi....14
1.3 Характерные черты пользовательских интерфейсов....15
1.3.1 Общие принципы проектирования пользовательских интерфейсов.15
1.3.2 Ключевые компоненты современных графических пользовательских интерфейсов....17
1.3.3 Критерии качества пользовательского интерфейса....18
1.4 Этапы проектирования пользовательского интерфейса....19
1.5 Актуальность выбранной темы....21
2 ТРЕБОВАНИЯ К РАЗРАБОТКЕ ПРИЛОЖЕНИЯ....22
2.1 Выбор и обоснование средств разработки....22
2.2 MAMP....22
2.3 СУБД MySQL....24
2.4 PHPMyAdmin....26
2.5 Язык PHP....27
2.6 Язык HTML....27
3 ПРИКТИЧЕСКАЯ РЕАЛИЗАЦИЯ....29
3.1 Алгоритм работы....29
3.2 Формирование ролей....29
3.3 Проектирование веб-интерфейса....30
3.3.1 Определение структуры страниц....30
3.3.2 Определение функциональных возможностей страниц....31

3.3.3 Создание страниц веб-интерфейса.....38
3.3.4 Функциональная связь страниц с базой данных.....41
3.3.5 Разделение функционала страниц по ролям.....42
4 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ.....44
4.1 Введение.....44
4.2 Требования при работе с компьютером.....44
4.2.1 Требования к параметрам воздушной среды.....44
4.2.2 Требования к освещению помещений и рабочих мест.....45
4.2.3 Требования к уровню шума и вибрации.....46
4.2.4 Требования к производственному оборудованию.....46
4.2.5 Требования к размещению производственного оборудования и организации рабочих мест пользователей компьютеров.....47
4.2.6 Требования к применению средств защиты.....47
4.2.7 Режимы труда и отдыха при работе с компьютером.....48
ЗАКЛЮЧЕНИЕ.....50
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....51
ПРИЛОЖЕНИЯ.....52

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

БД – база данных

ВДТ – видеодисплейный терминал

ЛВС – локальная вычислительная сеть

ПК – персональный компьютер

ПО – программное обеспечение

ПЭВМ – персональная электронно-вычислительная машина

СУБД – система управления базами данных

ЭВМ – электронно-вычислительная машина

ВВЕДЕНИЕ

Развитие информационных технологий привело к тому, что практически во всех современных организациях, как частных, так и государственных, используется огромное количество всевозможных информационных систем.

Образовательные учреждения, как и другие организации, используют современные средства для управления учебным процессом, например, информационные системы университета, обеспечивающие возможности для обмена информацией между студентами и преподавателями.

Для связи множества различных устройств организации используется локальная вычислительная сеть, которая позволяет использовать многочисленные ресурсы компьютеров, подключенных к единой сети. Естественно, учет пользователей ЛВС приобретает огромное практическое значение.

Разработка носит прикладной характер. Предложенная система учета пользователей ЛВС ПГУ поможет решить ряд задач при возникновении трудноразрешимых проблем с сетью или при утечке трафика. Кроме того, качественно спроектированный графический пользовательский интерфейс позволит быстро найти необходимую информацию, для скорейшего устранения возникших проблем, возникших в ЛВС.

Основными задачами, решаемыми автором в выпускной квалификационной работе, были:

- охарактеризовать предметную область решаемой задачи;
- провести анализ существующих разработок в предметной области;
- рассмотреть основные функции системы учета пользователей;
- разработать требований к веб-интерфейсу;
- выбрать среду разработки;
- разработать веб-интерфейс приложения;

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Постановка задачи

Требуется разработать приложение, позволяющее выполнять учет компьютерного фонда университета.

Для реализации приложения должна быть использована структура на основе двух компонентов:

- клиентская часть;
- база данных.

Клиентская часть – представляет собой графический интерфейс приложения, а также программу для обработки запросов к базе данных. Доступ пользователей и взаимодействие осуществляется при помощи браузера и различных элементов управления: окон, кнопок, полей ввода и других.

База данных – организованная структура, предназначенная для хранения, изменения и обработки взаимосвязанной информации преимущественно больших объемов. Она выдается по запросу веб-приложения. Структура взаимодействия компонентов для одного из вариантов использования представлена на рисунке 1.

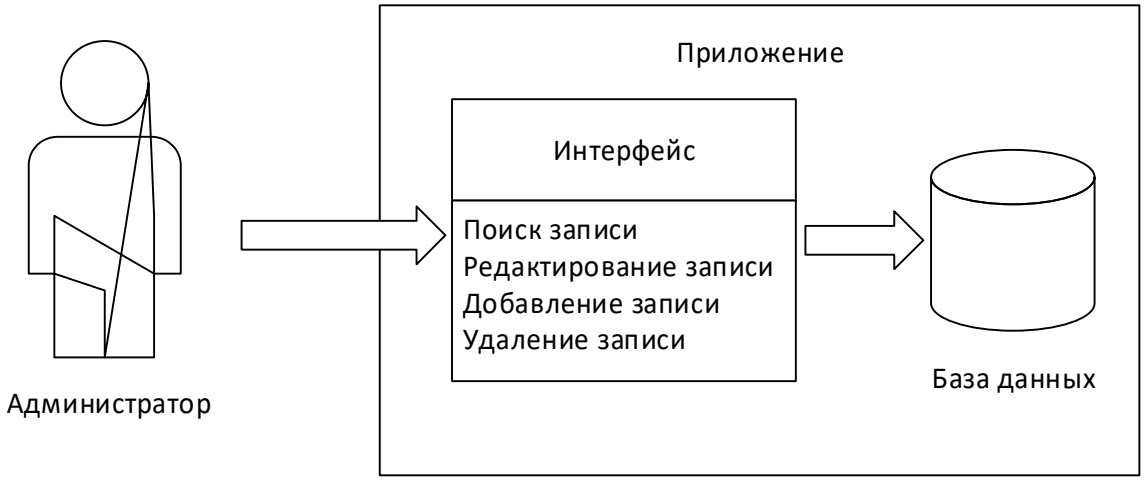


Рисунок 1 - Структура взаимодействия компонентов

1.2 Анализ существующих разработок в предметной области

Программы мониторинга сети - основной инструмент каждого системного администратора. Они позволяют быть в курсе всех сетевых процессов и оперативно реагировать на аномальную деятельность в пределах

локальной вычислительной сети. Так, появляется возможность автоматизировать часть деятельности администратора: прежде всего связанной с обеспечением сетевой безопасности. Также такие программы позволяют узнать месторасположение компьютера и быстро устранить неполадку. Некоторые имеют возможность устранять часть проблем самостоятельно.

1.2.1 Spiceworks

Это удобная и мощная система, позволяющая собрать и структурировать информацию о сети. Используется различными предприятиями для мониторинга и инвентаризации сети.

Она позволяет получить готовую систему управления активами.

Ее достоинствами являются:

- Быстрое сканирование подсетей на наличие любого оборудования с IP-адресом, включая компьютеры Windows и Mac, маршрутизаторы, коммутаторы, принтеры и многое другое.
- Предоставление полученной информации в виде таблиц и ее фильтрация. Также есть возможность сравнения устройств.
- Имеет понятный веб-интерфейс
- Имеет многопользовательский доступ к данным
- Предоставление инструментов для мониторинга и поддержки пользователей

К недостаткам можно отнести только то, что ввиду использования веб-интерфейса есть некоторые ограничения на визуализацию процессов, но это компенсируется всеми достоинствами программы.

1.2.2 Zabbix

Zabbix –это универсальная система мониторинга. Основными функциями которой являются:

- Сбор данных
- Автоматическое обнаружение проблем. Данная система анализирует собранные данные, выделяя причины проблем, имеет насколько

уровней критичности, а также позволяет получать оповещения о проблемах высокой важности по СМС и e-mail

- Визуализация. Программа имеет несколько вариантов представления веб-интерфейса для обзора IT-инфраструктуры;
- Безопасность и аутентификация. Позволяет защищать данные на всех уровнях благодаря надежному шифрованию между компонентами и нескольким уровням аутентификации
- Простота внедрения. Эта система позволяет создавать пользовательские шаблоны, что позволяет эффективно управлять тысячами устройств. Также возможно осуществлять экспорт и импорт узлов сети. Данные экспортируются в формате XML, который легко читать и модифицировать
- Автоматизация. Осуществление мониторинга больших инфраструктур может стать ночным кошмаром для администратора без использования автоматизации. В Zabbix есть 3 подхода для автоматического обнаружения и управления элементами инфраструктуры сетевое обнаружение, низкоуровневое обнаружение и авторегистрация агентов, каждый подход покрывает свое поле деятельности;
- Выполнение команд удаленно через брандмауэр
- Мониторинг состояния веб-сайта с помощью автоматического выполнения сценария

Zabbix довольно мощная система, она имеет множество функций, которые упрощают наблюдение за сетью. Система написана на C и PHP. Веб-интерфейс Zabbix написан на PHP. Для хранения данных используются различные СУБД, такие как MySQL, PostgreSQL и множество других;

В настоящее время эта система используется Пензенским государственным университетом для мониторинга сети;

1.2.3 Zabbix

Zabbix –это универсальная система мониторинга. Основными функциями которой являются:

- Сбор данных;
- Автоматическое обнаружение проблем. Данная система анализирует собранные данные, выделяя причины проблем, имеет насколько уровней критичности, а также позволяет получать оповещения о проблемах высокой важности по СМС и e-mail;
- Визуализация. Программа имеет несколько вариантов представления веб-интерфейса для обзора IT-инфраструктуры;
- Безопасность и аутентификация. Позволяет защищать данные на всех уровнях благодаря надежному шифрованию между компонентами и нескольким уровням аутентификации;
- Простота внедрения. Эта система позволяет создавать пользовательские шаблоны, что позволяет эффективно управлять тысячами устройств. Также возможно осуществлять экспорт и импорт узлов сети. Данные экспортируются в формате XML, который легко читать и модифицировать;
- Автоматизация. Осуществление мониторинга больших инфраструктур может стать ночным кошмаром для администратора без использования автоматизации. В Zabbix есть 3 подхода для автоматического обнаружения и управления элементами инфраструктуры сетевое обнаружение, низкоуровневое обнаружение и авторегистрация агентов, каждый подход покрывает свое поле деятельности;
- Выполнение команд удаленно через брандмауэр;
- Мониторинг состояния веб-сайта с помощью автоматического выполнения сценария;

Zabbix довольно мощная система, она имеет множество функций, которые упрощают наблюдение за сетью. Система написана на C и PHP. Веб-интерфейс Zabbix написан на PHP. Для хранения данных используются различные СУБД, такие как MySQL, PostgreSQL и множество других;

В настоящее время эта система используется Пензенским государственным университетом для мониторинга сети;

1.2.4 IPplan

IPplan – это доступное, веб-многоязычное программное обеспечение для управления IP-адресами и инструмент отслеживания, написанный на php4, упрощающий администрирование пространства IP-адресов.

IPplan выходит за рамки управления адресами TCP/IP, он также включает в себя возможность администрирования DNS, управление файлами конфигурации, управление расположением и хранение информации об оборудовании, которое настраивается дополнительно.

Эта система может обслуживать одну сеть или обслуживать несколько сетей и клиентов. Она делает управление IP-адресами простыми и легкими.

IPplan работает со многими базами данных и был протестирован с MySQL, Postgress и Oracle. Также известно, что MSSQL работает так же, как и большинство других баз данных, хотя они не были напрямую протестированы на IPplan. Система может работать на любой операционной системе, которая поддерживает язык PHP и одну из вышеуказанных баз данных.

Основными функциями и достоинствами IPplan являются:

- поддержка IPv6
- интернационализация
- импорт определений сети из таблиц маршрутизации
- импорт определений из TAB файлов с разделителями в XML - формат
- несколько администраторов с разными профилями доступа (для каждой группы, разрешающей доступ для каждого клиента, для каждой сети.)
- определение границы полномочий адресного пространства для каждой группы
- поиск свободное адресное пространство в диапазоне
- разделение и объединение сетей, чтобы сделать их все меньше и меньше (определенные ip остаются неизменными)

- отображение перекрывающегося адресного пространство между сетями
- возможности поиска
- журнал аудита
- статистика
- администрирование DNS (прямая и обратная зоны, импорт существующих зон с помощью передачи зон)
- система шаблонов для расширения IPplan, содержащая специфичную для сайта информацию, например, данные схемы, данные конфигурации хоста, и другое.
- управление файлами конфигурации устройства
- позволяет связывать адреса вместе через указатели
- имеет внешнюю таблицу стилей для изменения внешнего вида дисплея
- наличие триггеров - каждое пользовательское событие может вызывать пользовательскую функцию
- external poller - проверка подсети на наличие активных адресов для сбора статистики использования
- система запроса IP-адресов - позволяет пользователям запрашивать статические IP-адреса из базы данных.

Доступны два метода аутентификации - либо собственная схема внутренней аутентификации IPplan, либо в качестве альтернативы использование любого внешнего модуля аутентификации Apache. Такие как системы единого входа, например, SiteMinder или собственные схемы, основанные на LDAP, или любые другие системы, совместимые с Apache.

1.2.5 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 отмечания MAC-адресов устройств, которые были потеряны или украдены, при их появлении в сети программа уведомит пользователя об этом

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

1.3 Характерные черты пользовательских интерфейсов

1.3.1 Общие принципы проектирования пользовательских интерфейсов

В условиях использования современных информационных технологий становятся актуальны вопросы организации взаимодействия между человеком и техническими и программными средствами. Обеспечить подобное взаимодействие позволяет пользовательский интерфейс.

Интерфейс пользователя – это элементы и компоненты программы, которые способны оказывать влияние на взаимодействие пользователя с программным обеспечением.

Для создания интерфейса, который позволил бы пользователю работать с программой более эффективно, разработчику необходимо понимать, какие задачи пользователи будут решать, используя данное ПО. Выделяют общие принципы проектирования пользовательских интерфейсов:

1. Естественность интерфейса. Естественный интерфейс не вынуждает пользователя значительно менять привычные для него пути решения задачи. Использование знакомых пользователю понятий и метафор, позволяет пользователю интуитивно ориентироваться в ПО при решении поставленных им задач.

2. Согласованность интерфейса. Согласованность позволяет пользователям использовать накопившийся опыт использования приложения для решения новых задач, быстрее осваивать новые аспекты программы, и благодаря этому большее внимание уделять решаемой задачи, нежели тратить время на изучение особенностей использования тех или иных элементов управления, команд и прочего. Согласованность важна для всех аспектов пользовательского интерфейса: имена команд, визуальное представление информации и поведение интерактивных элементов.

3. Дружественность интерфейса. Обычно пользователи изучают особенности работы с новым программным продуктом самостоятельного обучения методом проб и ошибок. Эффективный интерфейс должен позволять предотвращать ситуации, которые бы привели к появлению ошибок.

4. Принцип «Обратной связи». Каждое действие пользователя должно получать визуальное, в некоторых случаях, и звуковое подтверждение того, что программное обеспечение восприняло введенную команду. Вид реакции при этом, должен учитывать природу выполненного действия.

5. Простота интерфейса. При этом имеется в виду не упрощение интерфейса, а обеспечение легкости в его изучении и в использовании. Кроме того, пользовательский интерфейс должен предоставлять доступ ко всем функциональным возможностям, предусмотренных данным программным продуктом.

6. Гибкость интерфейса. Гибкость интерфейса — это его способность учитывать уровень подготовки и производительность труда пользователя. То есть, пользователь, который использует данное программное обеспечение впервые, должен получать различные подсказки и подтверждения его действий. Опытный же пользователь, наоборот, для повышения производительности труда, должен получать минимальное количество оповещений.

7. Эстетическая привлекательность. Проектирование визуальных компонентов является важной частью разработки пользовательского интерфейса. Корректное визуальное представление используемых объектов должно обеспечивать не только понимание пользователем представленной информации, но и позволять сосредоточиться на наиболее важных ее аспектах.

1.3.2 Ключевые компоненты современных графических пользовательских интерфейсов

Среди различных видов визуальных компонентов пользовательского интерфейса выделяются две главных группы – окна и оконные элементы управления, и дополнительные – панели управления и другие вспомогательные средства.

Окна являются основными объектами пользовательского интерфейса. Они образуют структуру пользовательского интерфейса. Ввод и вывод данных для каждой отдельной задачи выполняется в соответствующем окне. Все остальные объекты интерфейса существуют только в качестве составной части того или иного окна.

Оконные элементы управления располагаются внутри окна. Они позволяют пользователю осуществлять непосредственное взаимодействие с программным обеспечением. Примеры подобных оконных элементов: поля ввода, списки, кнопки, и т.д.

Панели управления представляют собой области, содержащие интерфейсные элементы. На панелях управления обычно размещаются средства доступа к наиболее часто исполняемым операциям.

Вспомогательными средствами интерфейса называются объекты, не вошедшие в перечисленные группы, такие как разделители, всплывающие подсказки и другие.

Разделители – это элементы пользовательского интерфейса, которые располагаются внутри окна и служат для его разбиения на самостоятельные части, с каждой из которых пользователь может взаимодействовать как с самостоятельным окном. Разделитель разбивает окно на несколько частей, организованных на экране в форме таблицы. Каждая ячейка подобной таблицы может содержать свои собственные оконные элементы управления.

Всплывающей подсказкой называется окно, появляющееся при наведении указателя мыши на соответствующий интерфейсный элемент и содержащее дополнительную справочную информацию об этом элементе.

1.3.3 Критерии качества пользовательского интерфейса

Существует три основных критерия качества пользовательского интерфейса:

1. Скорость работы пользователя. Согласно Дональду Норману, взаимодействие пользователя с системой состоит из семи шагов:

- Формирование цели действий
- Определение общей направленности действий
- Определение конкретных действий
- Выполнение действий
- Восприятие нового состояния системы
- Интерпретация состояния системы
- Оценка результата

Таким образом, мыслительный процесс пользователя занимает большую часть времени работы с компьютером, поскольку шесть из семи этапов полностью заняты умственной деятельностью. Следовательно, увеличение скорости размышлений пользователя приводит к существенному улучшению скорости его работы.

Существенно повысить скорость мышления пользователей невозможно, но качественный пользовательский интерфейс должен максимально уменьшить влияние факторов, усложняющих процесс мышления.

2. Количество человеческих ошибок. Пользовательский интерфейс должен содержать элементы, которые позволят уменьшить количество допускаемых пользователем ошибок. К подобным элементам относятся:

- плавное обучение пользователя в процессе работы
- снижение требований к бдительности пользователя
- повышение заметности и разборчивости элементов индикации

Кроме того, пользовательский интерфейс должен содержать средства, позволяющие снизить чувствительность системы к ошибкам. К ним относятся:

- блокировка потенциально опасных действий пользователя до получения подтверждения правильности действия
- проверка системой действий пользователя перед их принятием
- самостоятельный выбор системой необходимых команд или параметров, когда от пользователя требуется только проверка

3. Скорость обучения. Пользовательский интерфейс должен содержать в себе средства, позволяющие пользователю за кратчайшие сроки научиться работать с программным продуктом. К таким средствам относятся справочные материалы, информационные сообщения, различные подсказки и т. д.

1.4 Этапы проектирования пользовательского интерфейса

Проектирование интерфейса - сложный и многоэтапный процесс, каждый этап которого состоит в свою очередь из отдельных ступеней. В общем случае процесс разработки можно представить в виде четырех этапов, направленных на обеспечение оптимального взаимодействия пользователя с программным продуктом:

- Начало работы над проектом
- Постановка задачи
- Высокоуровневое проектирование

- Низкоуровневое проектирование

При проектировании пользовательского интерфейса нельзя рассматривать составляющие его части отдельно друг от друга. В системе «человек-машина» главным элементом является человек, а система - подчиненное звено. Поэтому программное обеспечение должно реагировать на действия человека, а не наоборот. Следовательно, возникает задача проектирования опыта взаимодействия.

Проектирование опыта взаимодействия – область научно-практической деятельности, сосредоточенная на проектировании поведения пользователя продукта.

Проектирование взаимодействия – это описание возможного поведения пользователя и определение того, как система будет реагировать на его поведение, и приспосабливаться к нему.

Проектирование взаимодействия касается не столько эстетических аспектов, сколько понимания потребностей пользователей и принципов их познавательной деятельности.

Именно опыт взаимодействия с программным продуктом формирует у пользователя впечатление о том, будет ли пользователь в дальнейшем работать с данным продуктом.

Положительный опыт взаимодействия является результатом множества проектных решений о том, как продукт себя ведет, как выглядит и какие действия может совершать пользователь. Эти решения, опираясь друг на друга, формируют опыт взаимодействия.

Этапы проектирования взаимодействия можно разделить на следующие уровни:

- стратегия
- набор возможностей
- структура
- компоновка
- поверхность

Каждый уровень зависит от уровней, расположенных ниже: поверхность зависит от компоновки, компоновка – от структуры, структура – от набора возможностей, а набор возможностей зависит от выбранной стратегии проектирования.

1.5 Актуальность выбранной темы

Исходя из анализа уже имеющихся решений, можно наблюдать, что сейчас существует не так много систем, предназначенных для учета пользователей и компьютеров в организации. Рассмотренные выше приложения являются как коммерческим, так и некоммерческим программным обеспечением.

Существующие на рынке системы мониторинга будет тяжело внедрить в среду образовательного учреждения, они требуют обширной доработки, и имеют избыточный функционал, в связи с чем большая часть приложения использоваться не будет. Также, стоимость некоторых вышеперечисленных коммерческих приложений достаточно высока, что еще больше подвергает сомнению использование существующих решений.

Исходя из данных позиций, было сделано решение разработать приложение учета пользователей ЛВС, которое будет обладать только необходимый функционал в рамках образовательного учреждения. Стоимость разработки при этом будет значительно ниже за счет упрощения системы.

2 ТРЕБОВАНИЯ К РАЗРАБОТКЕ ПРИЛОЖЕНИЯ

Приложение для учета пользователей ЛВС ПГУ представляет собой приложение, предназначенное для использования внутри университета его сотрудниками. Оно использует базу данных с доступом через веб-интерфейс.

Пользователями данного приложения являются:

- администраторы
- операторы

2.1 Выбор и обоснование средств разработки

В ходе разработки были проанализированы различные средства и технологии. Программные средства, использованные в данной работе:

- локальный сервер MAMP для запуска веб-интерфейса;
- СУБД MySQL;
- Веб-интерфейс PhpMyAdmin для администрирования СУБД MySql.

Использованный язык программирования:

- SQL для работы с базой данных
- PHP для связи веб-интерфейса и базы данных

Для создания веб-интерфейса приложения использовался язык HTML.

С помощью перечисленных функциональных средств можно создавать приложения использующие базы данных и веб-интерфейс, для доступа к содержимому базы данных. Выбор средств разработки исходил из:

- функциональных возможностей;
- высокой производительности;
- бесплатного использования.

2.2 MAMP

MAMP - это бесплатная локальная серверная среда, которая может быть установлена под macOS и Windows очень просто. MAMP предоставляет все инструменты, необходимые для запуска WordPress на настольном ПК,

например, для тестирования или разработки. Можно также легко протестировать свои проекты на мобильных устройствах.

Среди других популярных локальных веб-серверов можно выделить: AppServ, Open Server, Denwer. Их главный недостаток – отсутствие удобного преднастроенного программного комплекса.

AppServ это отличный сервер от тайского производителя, концепцией которого является легкая установка и настройка всех дистрибутивов за 1 минуту. AppServ очень прост в установке, он работает стабильно. Apache и MySQL устанавливаются как сервисы. Но так как нет никакой управляющей программы, то своей сути AppServ является графическим установщиком Apache, PHP, MySQL. Здесь компоненты просто устанавливаются в качестве сервисов и управляются самой Windows.

Open Server – это портативная серверная платформа и программная среда, созданная специально для веб-разработчиков с учётом их рекомендаций и пожеланий. Она имеет полный набор серверного программного обеспечения, удобный и многофункциональный интерфейс, обладает мощными возможностями по администрированию и настройке компонентов. Платформа широко используется с целью разработки, отладки и тестирования веб-проектов, а также для предоставления веб-сервисов в локальных сетях.

Denwer – один из самых популярных веб-серверов. Его комплектация: Apache, PHP и СУБД MySQL настраивается самостоятельно пользователем. Можно после установки этого веб-сервера изменить конфигурации и версии сервисов, которые входят в комплект.

MAMP включает различные версии PHP и систем управления базами данных, а так же есть возможность быстрого переключения между ними. Включенные СУБД модули: MySQL, PostgreSQL, MariaDB, MongoDB. В состав программного комплекса входят и различные веб-интерфейсы администрирования: PHPMyAdmin, Adminer, PHPRedisAdmin, PHPPgAdmin, PHPMemcachedAdmin и другие компоненты.

Все необходимые компоненты можно выбрать самостоятельно на усмотрение пользователя. Преимуществом использования MAMP является:

- дружественный интерфейс
- многофункциональность;
- быстрый переход между базами и веб-интерфейсом;
- мощные возможности проектирования;
- гибкие настройки;
- поддержка;
- постоянное развитие;
- выпуск новых версий программы.

2.3 СУБД MySQL

SQL – язык структурированных запросов. Он применяется для создания, модификации и управления информацией в базе данных, которая в свою очередь управляется соответствующей СУБД.

MySQL – это одна из самых популярных систем управления базами данных.

MySQL является популярной платформой баз данных для бизнеса, потому что она чрезвычайно проста в использовании. Обычно используется в сочетании с PHP. Существует множество инструментов, использующих преимущества MySQL, особенно платформы электронной коммерции с открытым исходным кодом, такие как OsCommerce, Magento, Joomla и многие другие.

Некоторые из преимуществ работы с MySQL:

1. MySQL безопасен: он использует надежные уровни защиты данных для защиты конфиденциальных данных от внешнего доступа. Индивидуальные права могут быть установлены для обеспечения нескольких уровней пользователя, что позволяет нескольким людям получать доступ к хранимым данным. Все пароли, используемые для доступа к базе данных MySQL, зашифрованы.

2. MySQL является недорогим:

поскольку программное обеспечение базы данных MySQL можно бесплатно загрузить с официального портала, и оно поставляется предварительно установленным на большинстве хост-серверов, его использование и обслуживание не требует больших затрат.

3. MySQL управляет памятью:

программное обеспечение базы данных отлично управляет памятью и фактически предотвращает утечки памяти, которые могут возникнуть на сервере, где он не развернут.

4. MySQL активно поддерживается:

можно запускать программное обеспечение на различных платформах и операционных системах, включая Windows, Linux, Novell NetWare, UNIX и другие.

5. MySQL быстр:

по сравнению с другими программами баз данных, такими как Sybase и Oracle, MySQL предлагает меньше возможностей, но это хорошо. Это означает, что MySQL намного быстрее с точки зрения производительности, и пользователи по-прежнему имеют доступ ко всем необходимым функциям, которые они получают через другие платформы баз данных.

6. Хранилище данных MySQL является масштабируемым:

по умолчанию MySQL может обрабатывать ограничение общего размера файла около 4 ГБ, однако его можно увеличить в соответствии с потребностями - до 8 ТБ данных.

7. MySQL поддерживает Novell Cluster Services:

Novell Cluster Services от NetWare предлагает расширение базы данных на стороне сервера для защиты ваших данных в случае аппаратного сбоя. Если контент MySQL хранится в кластере Novell, в случае сбоя одного сервера программное обеспечение перейдет на альтернативный сервер - это обеспечит бесперебойную работу приложений, без каких-либо сбоев. Клиенты никогда даже не заметят, что сервер вышел из строя.

Руководствуясь всеми перечисленными достоинствами, был сделан вывод, что СУБД MySQL отлично подходит для работы с объемами данных, использованных в данном приложении.

2.4 PHPMyAdmin

PHPMyAdmin – это приложение, представляющее собой веб-интерфейс для администрирования СУБД MySQL. Оно позволяет с использованием веб-браузера осуществлять администрирование базы данных, выполнять SQL-запросы, просматривать содержимое таблиц и баз данных. Это приложение имеет интуитивно понятный и удобный интерфейс, поддерживает различные функции MySQL, а также является одним из самых популярных веб-интерфейсов. PHPMyAdmin обеспечивает полноценную, в том числе удаленную, работу с базой данных MySQL посредством браузера. PHPMyAdmin служит для разработки структуры БД и отладки запросов в веб-приложении.

Самые популярные действия с базами данных – это их импорт и экспорт, а также выполнение различных SQL – запросов; создание, удаление и редактирование записей – все это позволяет осуществлять работу с данным веб-интерфейсом.

Таким образом, с помощью PHPMyAdmin возможно:

- Создание и корректирование БД, таблиц и записей;
- создание пользователей;
- исполнение SQL – команд;
- поиск по БД.

Исходя из вышесказанного был выбран для разработки данный инструмент, так как является самым удобным для создания таблиц и написания запросов.

2.5 Язык PHP

Язык PHP –это один из самых часто используемых языков для разработки веб-интерфейса. Он является удобным, функциональным и конечно же бесплатным. PHP имеет множество различных версий, которые постоянно обновляются. Этот язык имеет понятный синтаксис и является своеобразной надстройкой над системным языком.

Для написания программы на данном языке можно использовать обычный текстовый редактор. Но для разработки данного приложения использовался VS Code, который является удобным инструментом для написания программ.

Весь код, используемый в приложении в момент загрузки страницы, выгружается в память сервера, обрабатывается. Если код написан без ошибок и не произошло никаких системных аварий, то он выполняется. Результат выполнения выводится пользователю.

Он хорошо работает с базой данных MySQL.Исходя из выше сказанного PHP был выбран для разработки приложения, т.к. полностью подходит для реализации поставленной задачи.

2.6 Язык HTML

HyperText Markup Language (HTML) - предназначен для написания гипертекстовых документов.

Гипертекстовый документ – это текстовый файл, имеющий специальные метки, называемые тегами, которые впоследствии опознаются браузером и используются им для отображения содержимого файла на экране компьютера. С помощью этих меток можно выделять заголовки документа, изменять цвет, размер и начертание букв, вставлять графические изображения и таблицы. Но основным преимуществом гипертекста перед обычным текстом является возможность добавления к содержимому документа гиперссылок - специальных конструкций языка HTML, которые позволяют щелчком мыши перейти к просмотру другого документа.

Существует два способа создания гипертекстовых документов. Можно воспользоваться одним из WYSIWYG (What You See Is What You Get) HTML-редакторов (например, Netscape Composer, Microsoft FrontPage, HotDog) для работы с которыми не требуется специальных знаний о внутренней структуре создаваемого документа. Этот способ позволяет создавать документы для WWW без знания языка HTML. HTML-редакторы автоматизируют создание гипертекстовых документов, избавляют от рутинной работы. Однако их возможности ограничены, они сильно увеличивают размер получаемого файла и не всегда полученный с их помощью результат соответствует ожиданиям разработчика.

Второй вариант - создание и разметка документа при помощи текстового редактора (например, NotePad, VS Code). При этом способе, код страницы пишется непосредственно разработчиком, используя язык HTML напрямую. Создавая документы таким способом, разработчик имеет полный контроль над выходным продуктом.

3 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ

3.1 Алгоритм работы

Приложение состоит из двух функциональных частей:

- клиентской части
- базы данных

Авторизованный пользователь осуществляет какое-либо действие, в соответствии с этим выполняется требуемая задача. В случае успеха он обращается к СУБД и выполняет SQL – запрос. В случае успешного прохождения запроса выполняется запрашиваемый SQL – запрос, и пользователь видит результат запроса.

3.2 Формирование ролей

Структура проекта включает в себя следующие роли:

- Администратор
- Оператор

Рассмотрим функциональные возможности каждой роли:

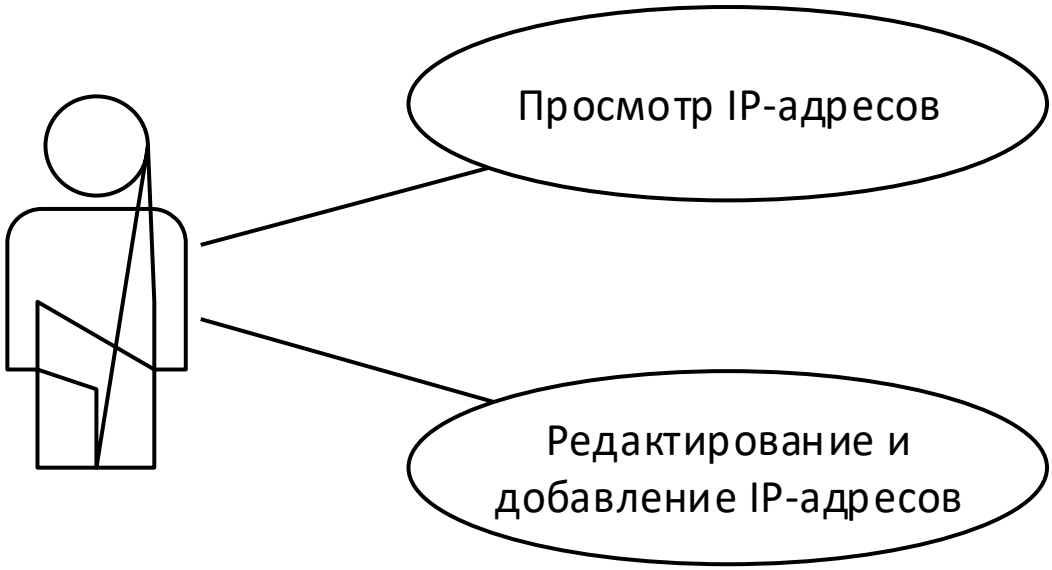


Рисунок 2 - UML-диаграмма прав оператора

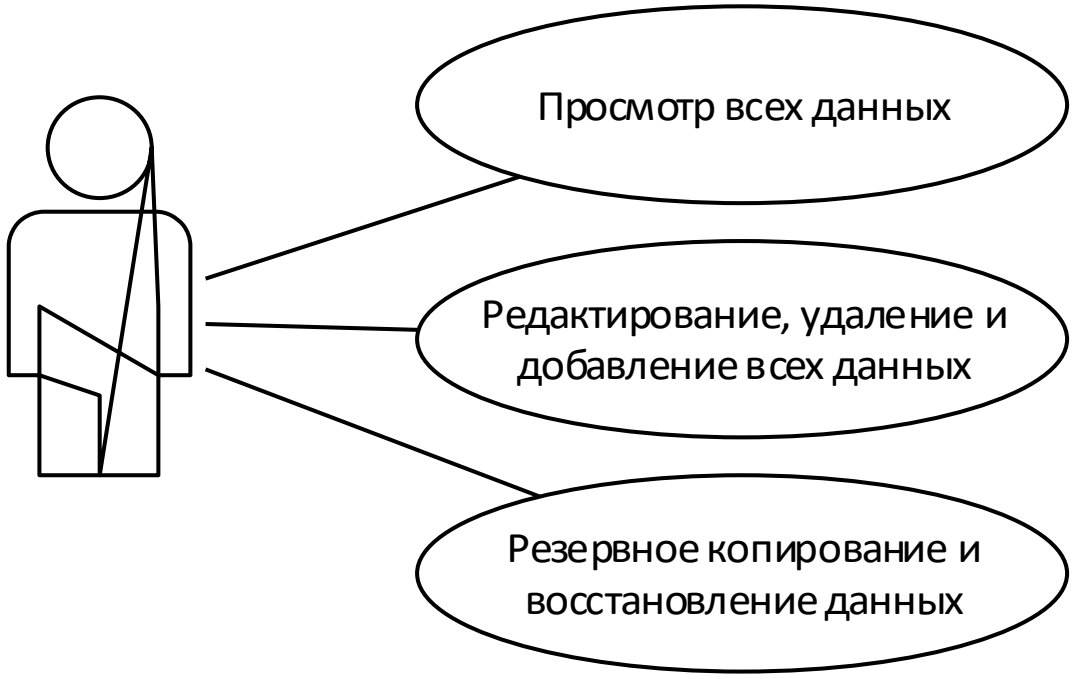


Рисунок 3 - UML-диаграмма прав администратора

Таким образом администратор имеет полный функционал и может совершать любые манипуляции с данными. Оператор же имеет ограниченный функционал. Ему доступны только добавление, редактирование и удаление IP-адресов аудиторий.

3.3 Проектирование веб-интерфейса

Определив функционал приложения, можно приступить к проектированию веб-интерфейса приложения. Структура веб-приложения состоит из множества страниц, каждая из которых отвечает за свой набор функций. На каждой странице (кроме страницы авторизации) присутствует заголовок с меню, с помощью которого можно перемещаться по приложению и набор элементов интерфейса в соответствии с функционалом данной страницы. В зависимости от роли пользователя будет отличаться и веб-интерфейс, в соответствии с его правами.

3.3.1 Определение структуры страниц

Определив набор функциональных возможностей и ролей пользователей, необходимо определить, какие страницы доступны из определенных секций веб-приложения. Для этого создадим иерархию страниц для каждой роли:

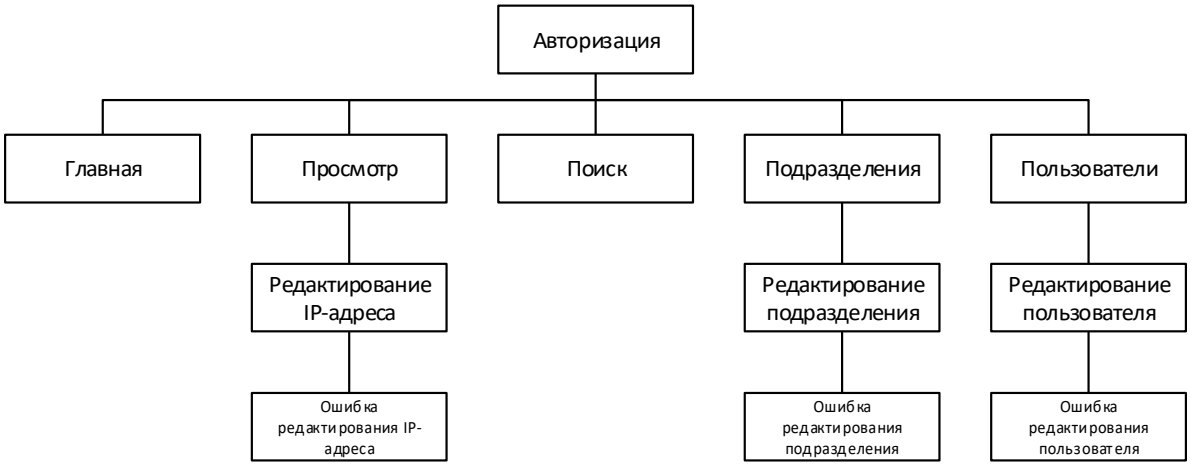


Рисунок 4 – Иерархия страниц веб-интерфейса для администратора

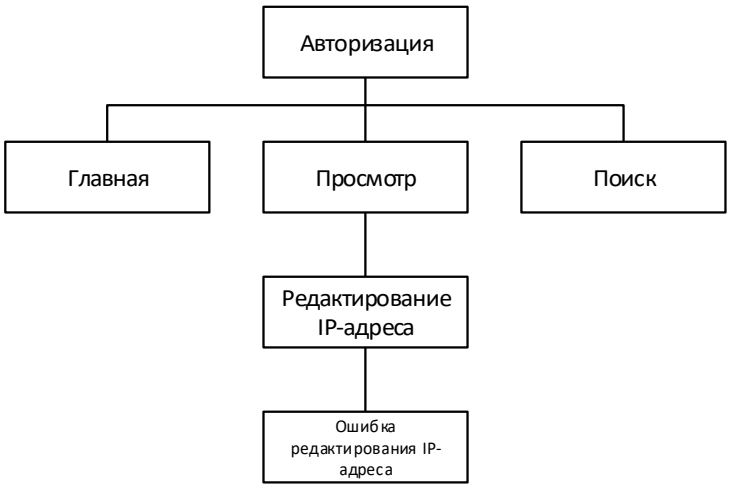


Рисунок 5 – Иерархия страниц веб-интерфейса для оператора

Как видно из рисунков 4 и 5 первое, что видит любой пользователь при запуске приложения – страница авторизации. Он не может продолжить работу с приложением, до тех пор, пока не войдет в систему под своим логином и паролем. В зависимости от роли авторизированного пользователя он может переходить к работе с приложением. Администратор может работать со всеми функциями приложения, оператор, только с ее частью.

3.3.2 Определение функциональных возможностей страниц

На каждой странице приложения доступен только тот функционал, под который эта страница выделена. Рассмотрим функциональные возможности каждой страницы:

- Страница «Авторизация». На странице авторизации пользователь, может ввести свои данные для входа. В случае совпадения введенных данных с данными, расположенными на сервере, пользователя перенаправляет на страницу «Главная». В случае несовпадения данных, пользователя информирует об ошибке.

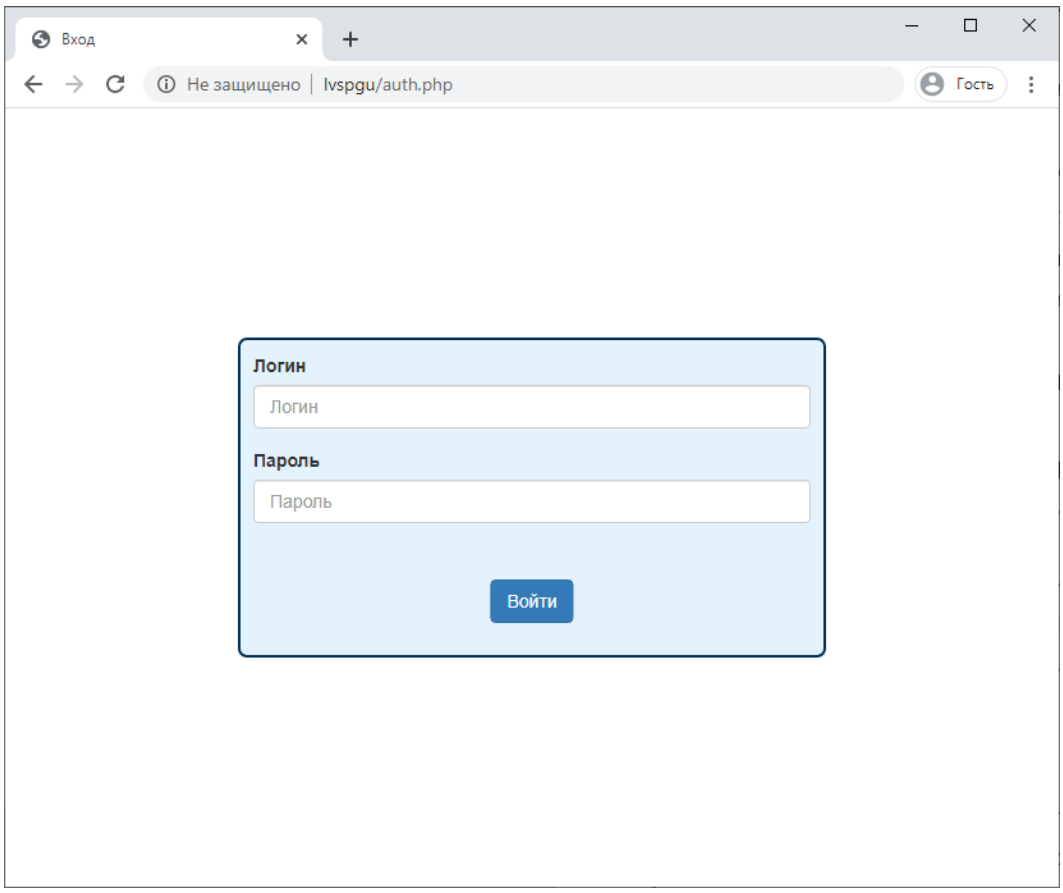


Рисунок 6 – Страница «Авторизация»

- Страница «Главная». На главной странице администратор может выполнить резервное копирование текущей базы данных, а также произвести ее восстановление из созданной копии. Оператор видит сообщение о недоступности данных функций для его роли.

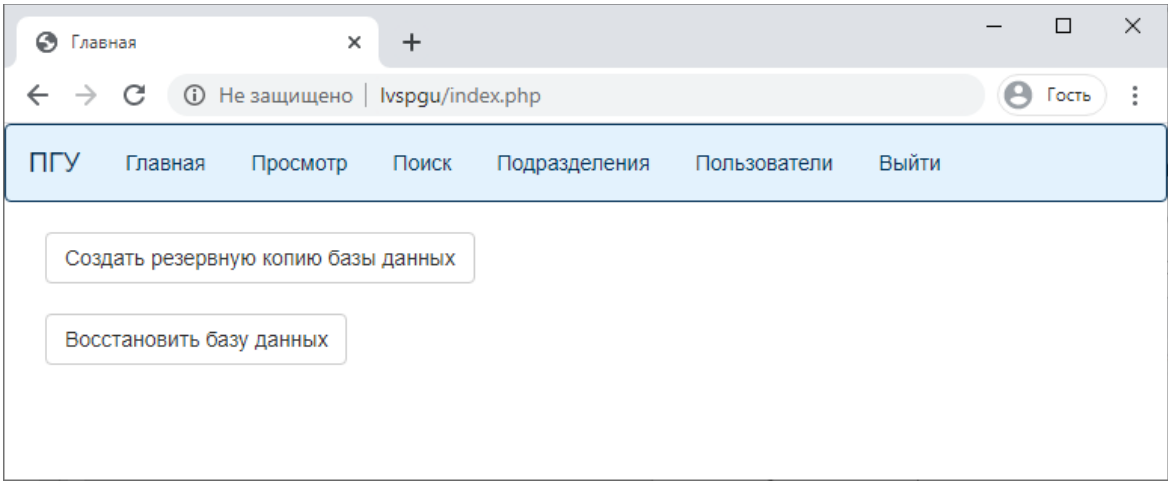


Рисунок 7 – Страница «Главная» для администратора

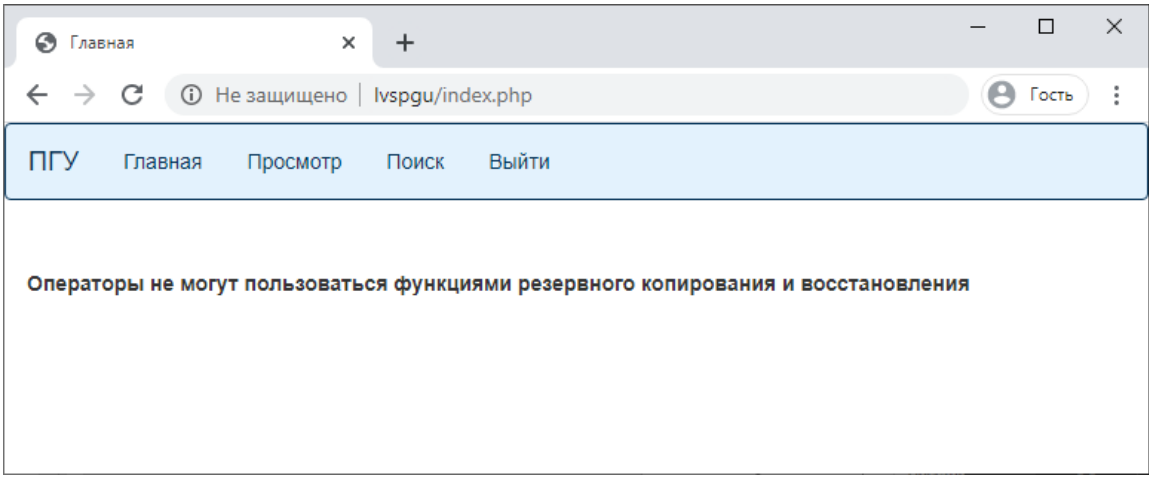


Рисунок 8 – Страница «Главная» для оператора

- Страница «Просмотр». На данной странице любой авторизированный пользователь может просмотреть данные о IP-адресах аудиторий, добавить новые записи и изменить существующие. Администратор дополнительно имеет возможность удалять записи из базы данных.

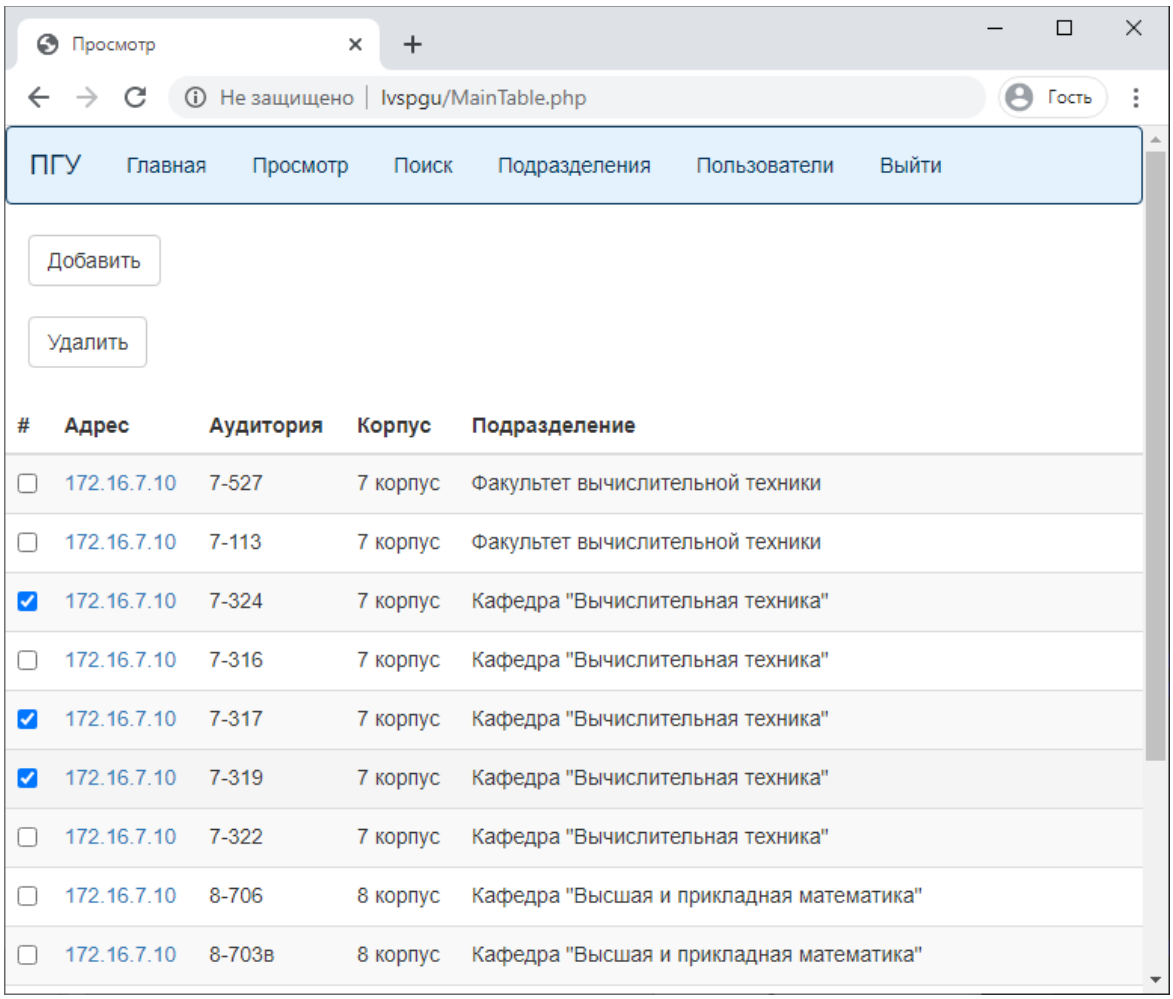


Рисунок 9 – Страница «Просмотр»

- Страница «Поиск». В поиске любой авторизированный пользователь может произвести поиск данных по заданным параметрам. Поиск делится на две части: «Поиск по подразделению» и «Поиск по IP-адресу». Заполнив соответствующие параметры, пользователь получает таблицу результатов, соответствующих требованиям.

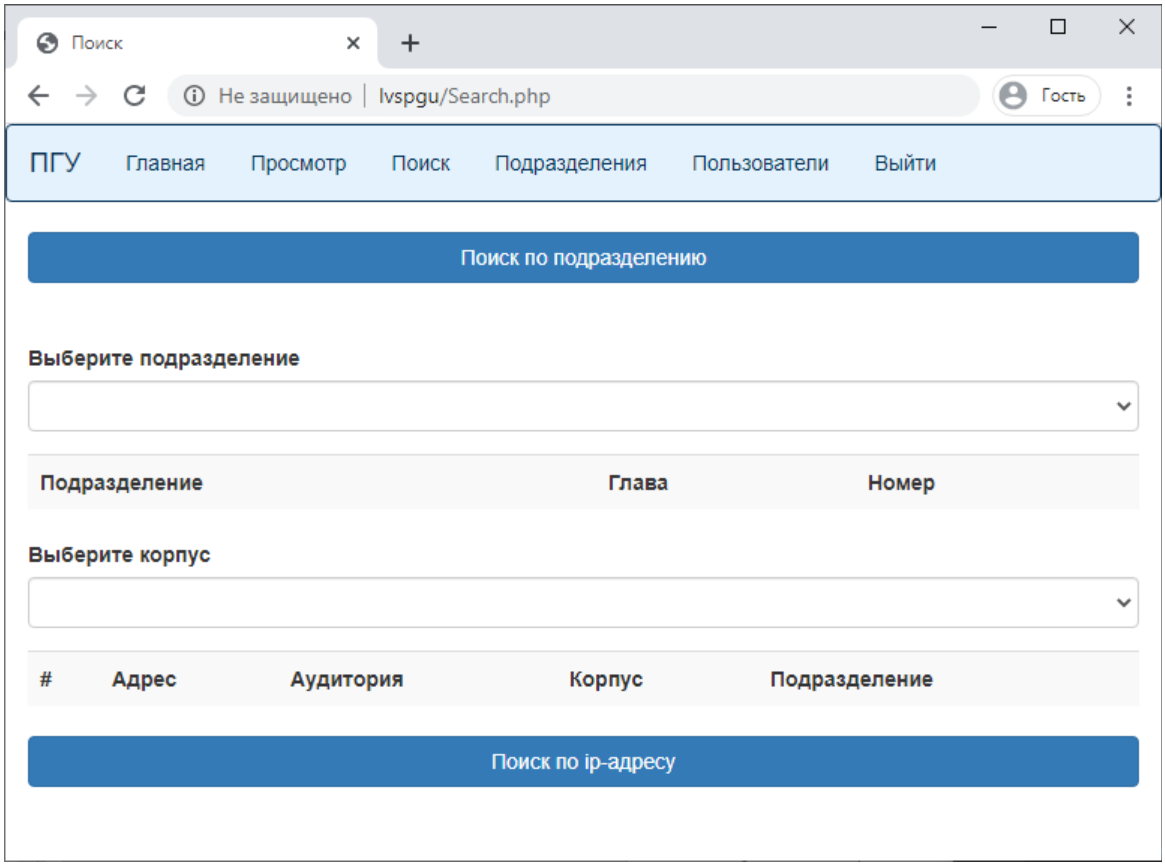


Рисунок 10 – Страница «Поиск»

- Страница «Подразделения». Доступ к данной странице имеет только администратор. Здесь пользователь может увидеть информацию о подразделениях, а также добавить новые записи, изменить или удалить уже существующие.

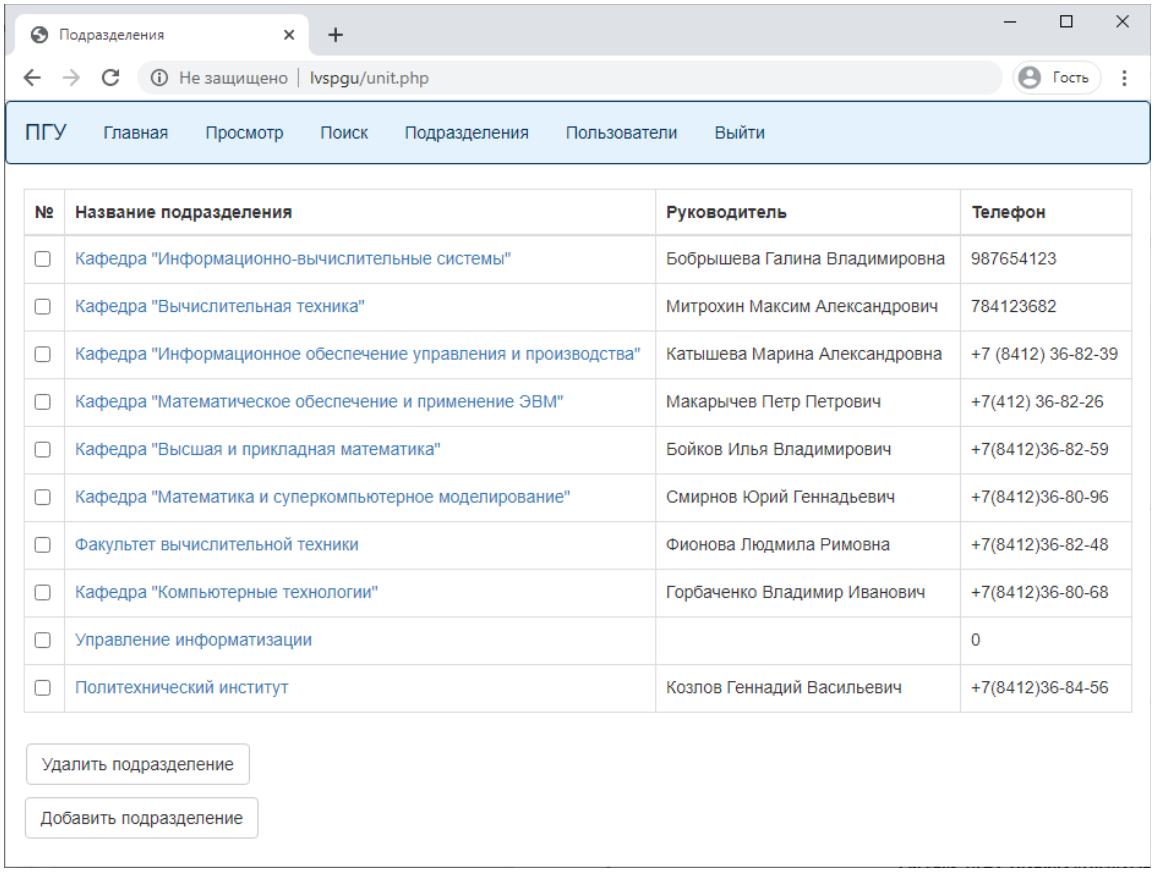


Рисунок 11 – Страница «Подразделения»

- Страница «Пользователи». Доступ к данной странице также имеет только администратор. Здесь пользователь может просмотреть информацию о всех пользователях системы, однако пароль, в целях безопасности зашифрован. Также здесь он может создать нового пользователя или изменить уже существующего.

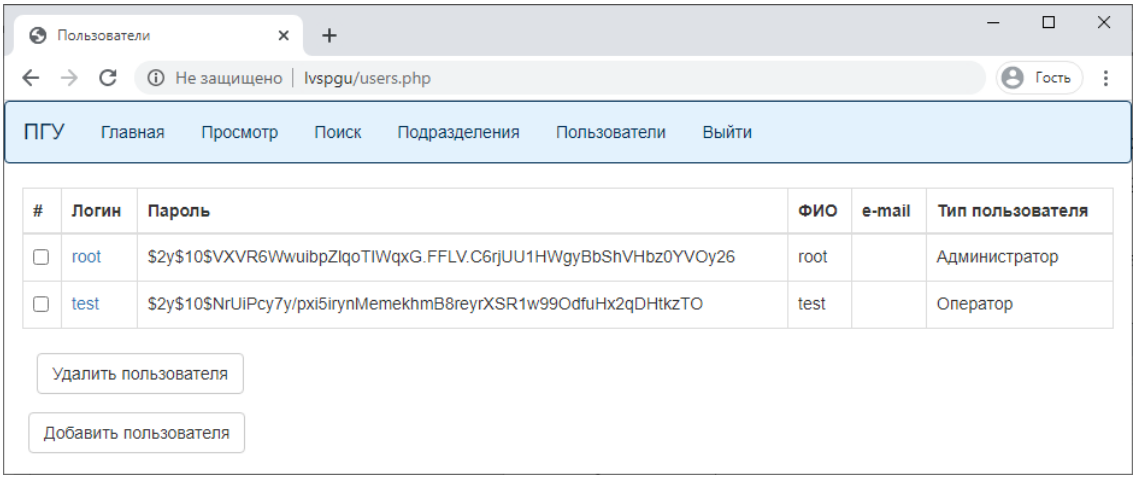


Рисунок 12 – Страница «Пользователи»

- Страницы редактирования. При изменении какой-либо записи, пользователя перенаправляет на страницу редактирования соответствующего типа записи. Здесь он может поменять значения записи, изменив значение соответствующих полей и нажав кнопку сохранить.

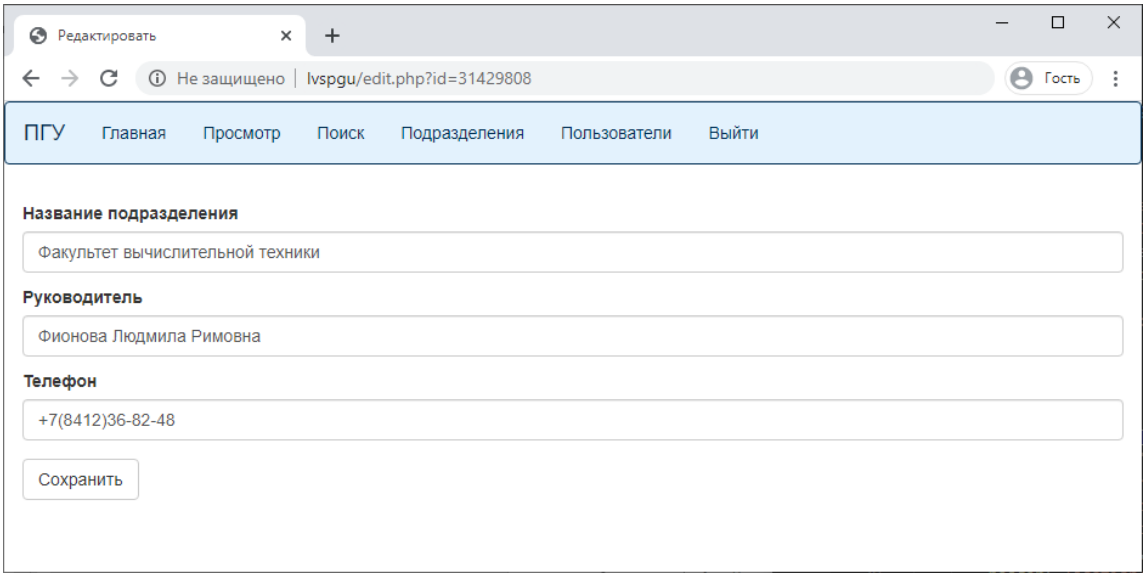


Рисунок 13 – Страница «Редактирование»

- Страницы ошибок сохранения. Если при сохранении каких-либо значений произошла ошибка, пользователя перенаправляет на соответствующую страницу ошибки.

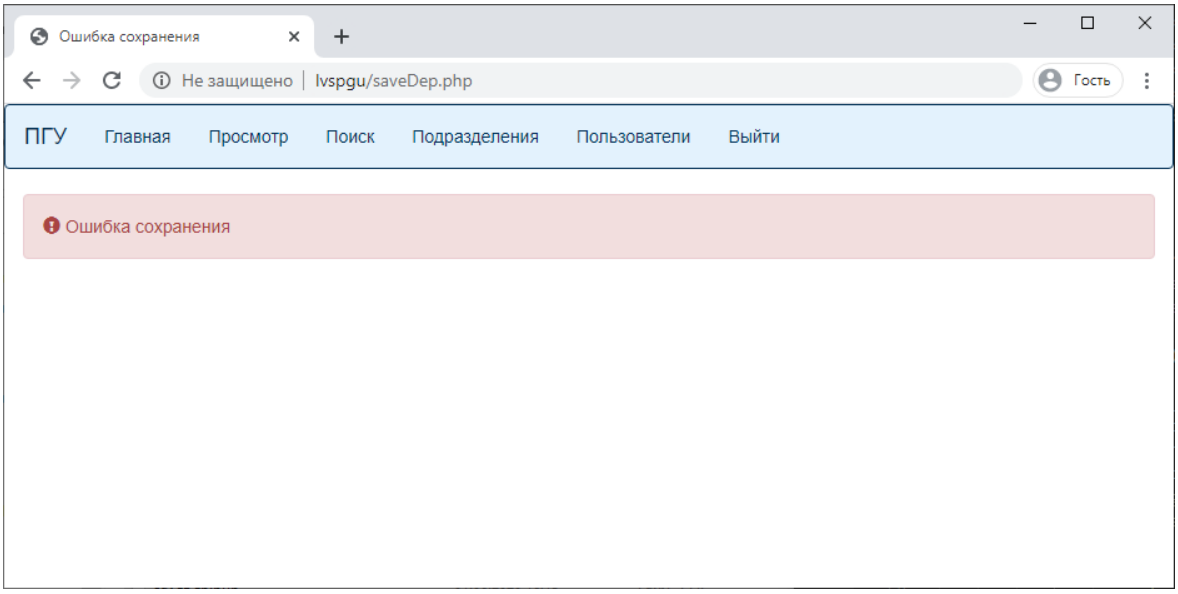


Рисунок 14 – Страница «Ошибка сохранения»

3.3.3 Создание страниц веб-интерфейса

Используя язык гипертекстовой разметки HTML были созданы базовые страницы, без какого-либо содержимого, которые затем постепенно дополнялись различными элементами интерфейса, в зависимости от функционала, описанного в пункте 3.3.2.

Для создания меню для навигации было решено использовать тег <nav> вложенный в тег <header>.

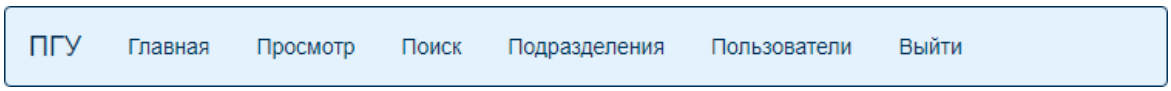


Рисунок 15 – Меню навигации приложения

Тег <nav> (навигация) используется для обозначения содержимого в виде основных навигационных ссылок.

Тег <header> (верхний колонтитул) является признаком верхнего колонтитула или баннера верхней части страницы, которая обычно содержит логотип, навигацию, относящуюся ко всему сайту, форму поиска по сайту, заголовок страницы с подзаголовком, авторскую информацию и прочее.

С помощью CSS-шаблонов фреймворка Bootsrap была организована некоторая адаптивность меню навигации под размер окна. При уменьшении окна до размеров, когда все пункты меню не могут вместиться в одну строку, они преобразуются в кнопку, при нажатии которой, появляется выпадающее меню, содержащее все пункты меню.

Рисунок 16 – Меню навигации приложения при малой ширене окна

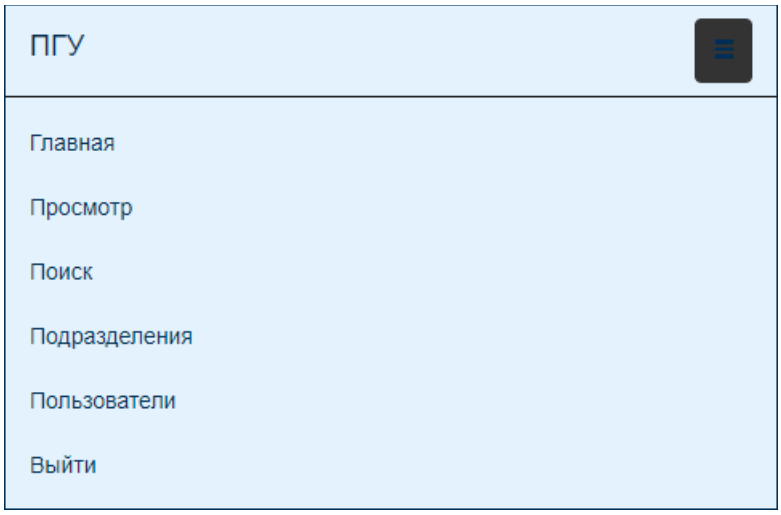


Рисунок 17 – Раскрытое меню навигации приложения при малой ширене окна

Для создания полей ввода необходимых для ввода информации использовался тег <input> типа «text», когда не требуется точного соответствия вводимых данных.

Рисунок 18 – Поле ввода типа text

Для выбора нескольких пунктов таблицы (например, для их удаления) так же используется тег <input> типа checkbox.

Рисунок 19 – Поля ввода типа checkbox

Элемент <input> является основным элементом формы и определяет пользовательское поле для ввода информации. Поле ввода принимает различный вид, в зависимости от значения атрибута type.

Когда от пользователя требуется точное соответствие вводимых данных, можно использовать тег <select> и выдавать пользователю на выбор один из нескольких вариантов, перечисленных во вложенных тегах <option>.

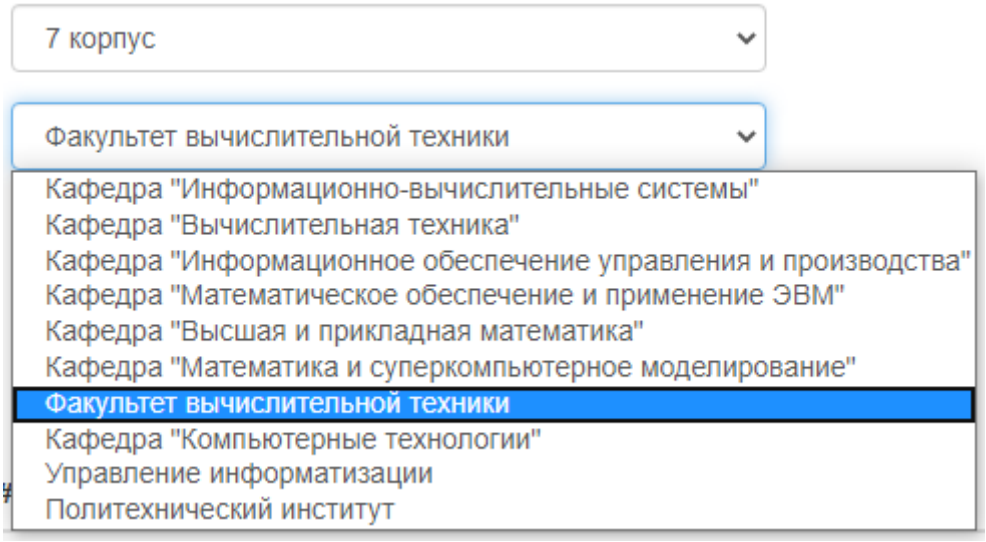


Рисунок 20 – Раскрывающийся список

Тег <select> представляет собой элемент управления, который позволяет создать раскрывающийся список. Тег <option> определяет пункты раскрывающегося списка, он применяется как вложенный элемент тега <select>.

Для вызова различных функций или раскрытия\сокрытия подробных полей ввода, используются кнопки, созданные при помощи тега <button>.

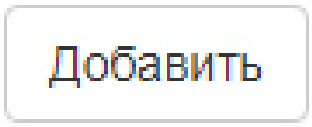


Рисунок 21 – Кнопка

Для вывода данных в формате таблицы используется тег <table> и вложенные теги <tr> и <th> для задания строк и столбцов таблицы соответственно.

№	Название подразделения	Руководитель	Телефон
☐	Кафедра "Информационно-вычислительные системы"	Бобрышева Галина Владимировна	987654123

Рисунок 22 – Таблица

Теги <select>, <button> и <input> вкладываются в так называемую форму, задающуюся тегом <form>. Форма — это часть документа, которая позволяет пользователю ввести информацию, которую затем можно принять и обработать на стороне сервера.

3.3.4 Функциональная связь страниц с базой данных

Заполнив все страницы необходимыми элементами интерфейса, можно приступить к наделению страниц соответствующим функционалом. Для этого использовались скрипты, написанные на языке PHP.

Большинство PHP скриптов были обернуты в if-конструкцию для того, чтобы они не выполнялись при каждом обновлении страницы, а только при поступлении необходимых входных данных. Как было описано в пункте 3.3.3 на страницах все поля, предназначенные для ввода, были обернуты в тег <form>. После нажатия кнопки, отвечающую за выполнение функции на данной странице, все введенные данные помещаются в тело POST-запроса и отправляются на сервер. Если в этом POST-запросе есть поля, удовлетворяющие условию, в который обернут скрипт – скрипт выполняется. В теле этого скрипта вызывается соответствующая функция, описанная в файле api.php. В этой функции формируется SQL-запрос к подключенной базе

данных, затем данные форматируются и выводятся на страницу, если это необходимо.

Вывод содержимого базы данных на страницу так же основан на PHP скриптах. Так как база данных постоянно будет изменяться, в нее будут добавляться новые записи, изменятся и удаляться уже существующие, вывод этих записей на страницу, должен также производиться динамически. Для этого был создан скрипт, который выводит в таблицу каждую запись, полученную соответствующей функцией, как новую строку таблицы. Таким образом не важно, сколько записей в базе данных, каждая из них, будет выведена на страницу.

3.3.5 Разделение функционала страниц по ролям

Для того, чтобы пользователь, не обладающий соответственными правами, не мог получить доступ к определенным данным и функциям, необходимо: во-первых, авторизовать пользователя, и во-вторых, сохранить данные об этом пользователе, в частности его права доступа, которые были бы доступны на всех страницах приложения. Для этого была организованна так называемая сессия. При успешной авторизации пользователем в систему, факт его входа и его права доступа записываются в суперглобальный массив $_SESSION, который хранится в cookie на стороне пользователя.

Теперь, создали на страницах скрипт, которые не позволят странице загрузиться, если пользователь не авторизован:

```
<?php
    session_start();
    if(!isset($_SESSION['logged_user'])){
        header("Location: auth.php");
        exit;
    }
?>
```

Если поле logged_user в массиве $_SESSION не заполнено, это означает, что пользователь не авторизован и доступ к этой странице ему запрещен, поэтому его сразу перенаправляет обратно на страницу авторизации. Если пользователь авторизован успешно, то продолжается загрузка всей страницы.

Теперь для разделения функционала между ролями оператора и администратора, достаточно поместить все зависимые элементы интерфейса в PHP скрипт, который будет проверять права доступа пользователя и выводить их только в случае совпадения этих прав. Таким образом, если у пользователя нет права пользоваться определенными функциями, например функциями резервного копирования и восстановления, на странице не будет соответствующих кнопок.

4 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

4.1 Введение

Трудно представить себе жизнь современного человека без компьютера. Но вместе с этим, встаёт вопрос о безопасности пользователя при работе за компьютером, ведь при достаточно длительном использовании компьютер может стать источником вредного воздействия на организм человека и, соответственно, источником профессиональных заболеваний. Каждый пользователь должен знать о вредном воздействии машин на организм и необходимых мерах защиты, чтобы сохранить свое здоровье и успешно работать за компьютером.

4.2 Требования при работе с компьютером

Основными требованиями при работе с компьютерами являются:

- Требования к параметрам воздушной среды
- Требования к освещению помещений и рабочих мест
- Требования к уровню шума и вибрации
- Требования к производственному оборудованию
- Требования к размещению производственного оборудования и организации рабочих мест пользователей компьютеров
- Требования к применению средств защиты
- Режимы труда и отдыха при работе с компьютером

4.2.1 Требования к параметрам воздушной среды

- Воздух, поступающий в рабочие помещения операторов ЭВМ, должен быть очищен от пыли и микроорганизмов.
- Кондиционирование воздуха должно поддерживать параметры микроклимата в необходимых пределах в течение всех сезонов года. Также обеспечивать очистку воздуха от пыли и вредных веществ, создание избыточного давления в чистых помещениях для исключения поступления

неочищенного воздуха. Температура подаваемого воздуха должна быть не ниже 19 градусов.

- Температуру в помещении следует регулировать с учетом тепловых потоков от оборудования. Оборудование надо устанавливать так, чтобы тепловые потоки от него не были направлены на пользователей. Следует также ограничивать количество вычислительной техники в помещении и избегать напольных отопительных систем.
- Поверхность пола в помещениях должна быть ровной, нескользкой, удобной для очистки и влажной уборки. Покрытия рабочих столов и пола должны обладать антистатическими свойствами и сохранять их в процессе эксплуатации.
- В помещениях должна проводиться ежедневная влажная уборка.

4.2.2 Требования к освещению помещений и рабочих мест

Важное место по созданию условий труда пользователей, работающих с ЭВМ, занимает создание оптимальной световой среды, то есть правильная организация естественного и искусственного освещения помещения и рабочих мест.

- Помещения должны иметь естественное и искусственное освещение.
- Естественное освещение 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 в помещениях необходимо оборудовать устройствами регулирования интенсивности естественного освещения: жалюзи, занавесы, внешние козырьки и прочее.
- Искусственное освещение в помещениях должно осуществляться системой общего равномерного освещения.

- Для использования в качестве источников местного освещения пригодны светильники, позволяющие избежать возникновения ослепления и бликов.
- Местное освещение на рабочих местах пользователей персональных компьютеров не должно создавать бликов на поверхности экрана и увеличивать освещенность экрана.

4.2.3 Требования к уровню шума и вибрации

- Для снижения шума и вибрации в помещениях вычислительных центров оборудование необходимо устанавливать на специальные фундаменты и амортизирующие прокладки.
- Уровень шума на рабочих местах не должен превышать 50 дБА. Нормируемые уровни шума обеспечиваются путем использования малошумного оборудования, применением звукопоглощающих материалов.
- Шумящее оборудование, уровни шума которого превышают нормированные, должно находиться вне помещения с компьютерами.
- Производственные помещения, в которых для работы используются преимущественно персональные не должны граничить с помещениями, в которых уровни шума и вибрации превышают нормируемые значения.

4.2.4 Требования к производственному оборудованию

- Для обеспечения электропитания компьютеров должна быть смонтирована отдельная сеть электроснабжения.
- Периферийное оборудование компьютеров должно быть подключено только к линиям электроснабжения компьютерной сети.
- Не допускается включение электропотребителей, не относящихся к компьютерам в линии электроснабжения компьютерной сети.
- Не допускается использование оборудования без подключения к защитному заземлению, если такое подключение предусмотрено конструкцией оборудования.

4.2.5 Требования к размещению производственного оборудования и организации рабочих мест пользователей компьютеров

- Рабочее место пользователя компьютера должно быть расположено по отношению к световым проемам таким образом, чтобы естественный свет падал на него сбоку. Не допускается располагать рабочие места так, чтобы естественный свет падал на них со стороны спины или лица пользователя.
- При размещении рабочих мест с компьютерами необходимо учитывать расстояния между рабочими столами с видеомониторами.
- Проходы между рабочими местами должны иметь ширину, обеспечивающую беспрепятственное перемещение персонала без прикосновения к оборудованию или материалам, расположенным на рабочем месте.
- Конструкция рабочего кресла должна обеспечивать поддержание рациональной рабочей позы при работе на компьютере, 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. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой.
- Экран видеомонитора должен находиться от глаз пользователя на оптимальном расстоянии 60-70 см, но не ближе 50 см с учетом размеров алфавитно-цифровых знаков и символов.
- Клавиатуру следует располагать на поверхности стола на расстоянии 10-30 см от края.

4.2.6 Требования к применению средств защиты

- Для обеспечения защиты от поражения электрическим током все оборудование на рабочем месте должно быть подключено к защитному

заземлению, если такое подключение предусмотрено конструкцией оборудования.

- В случае выявления несоответствия параметров микроклимата оптимальным значениям следует применять технические средства кондиционирования воздуха рабочего помещения. Для повышения влажности воздуха в помещениях следует применять увлажнители воздуха.
- Для обеспечения чистоты воздуха необходимо применять аппаратные воздушные фильтры.
- Снизить уровень шума в помещениях с персональными компьютерами можно использованием для отделки помещений звукопоглощающих материалов.
- Использование антистатических напольных покрытий в сочетании со специальной антистатической обувью и рабочей одеждой, обеспечивает полную защиту от образования и накопления электростатического заряда на теле пользователя.
- Для предотвращения накопления заряда на корпусах и элементах оборудования следует, в случае необходимости, использовать специальные химические средства.

4.2.7 Режимы труда и отдыха при работе с компьютером

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

- группа А - работа по считыванию информации с ПЭВМ с предварительным запросом

- группа Б - работа по вводу информации

- 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 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.

ЗАКЛЮЧЕНИЕ

Результатом данной выпускной квалификационной работы является веб-интерфейс системы учета пользователей ЛВС ПГУ. Была разработан веб-интерфейс, оформлена документация на приложение. Проведенное тестирование показало, что часть приложения, созданная в ходе выпускной квалификационной работы, работает правильно.

В дальнейшем приложение может быть улучшено за счет расширения функциональных возможностей приложения, оптимизации пользовательского интерфейса и увеличении его адаптивности.

В ходе выполнения выпускной квалификационной работы был значительно повышен уровень практических навыков по созданию веб-приложений, освоен процесс разработки веб-интерфейса 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.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. К. Янк. PHP и MySQL. От новичка к профессионалу, 5-е изд. Москва, 2013. 585 с.

2. Мандел Т. Разработка пользовательского интерфейса: Пер. с англ. М.: ДМК Пресс, 2001.

3. Гультяев А.К., Машин В.А. Проектирование и дизайн пользовательского интерфейса. СПб.: КОРОНА-принт, 2000

4. Л. Томсон, Л. Веллинг. Разработка web-приложений на PHP и MySQL, 5-е изд. Москва, Санкт-Петербург, Киев, 2017. 652 с.

5. Р. Никсон. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5, 4-е изд. 2016. 766 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. К. Шмитт. CSS. Рецепты программирования, 3-е изд. Санкт – Петербург, 2011. 672 стр.

9. Н. Закас. JavaScript для профессиональных веб-разработчиков. Санкт – Петербург, 2015. 960 с.

10. MySQL [Электронный ресурс]. – Электрон.дан. URL: http://www.mysql.com/

11. Организация труда на предприятии [Электронный ресурс] // Организация труда на предприятии Режим доступа URL http://www.up-pro.ru/encyclopedia/organizaciya-truda.html [Дата обращения 10.05.20 20 г.].

12. 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.]

ПРИЛОЖЕНИЯ

Приложение А Коды страниц веб-интерфейса

Файл auth.php:

```
<?php include 'db.php'; ?>
<?php include 'api.php'; ?>

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Вход</title>
        <link href="https://cdnjs.cloudflare.com/ajax/libs/twitter-
bootstrap/3.3.7/css/bootstrap.css" rel="stylesheet" crossorigin="anonymou
s" >
        <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/
3.1.0/jquery.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.
7/js/bootstrap.min.js" crossorigin="anonymous"></script>
    </head>
    <body>
      <div class="container-fluid" style=" position: absolute; top:
50%; left: 50%; padding: 10px; border: 2px solid #002f55; border-radius:
7px; width: 460px; height: 250px; transform: translate(-50%, -50%); backg
round-color: #e3f2fd">
        <form id="form" action="" method="POST">
          <div class="form-group">
            <label for="exampleDropdownFormEmail1">Логин</label>
            <input name="login" class="form-control" id="login" pla
ceholder="Логин"></input>
          </div>
          <div class="form-group">
            <label for="exampleDropdownFormPassword1">Пароль</label
>
            <input type="password" name="password" class="form-
control" id="password" placeholder="Пароль"></input>
            <?php
              if(isset($_POST['login']) && $_POST['password'] != ""
&& $_POST['login'] != "") {
                $login = $_POST['login'];
                $password = $_POST['password'];
                $users = getAllUsers($db);
                foreach ($users as $user) {
                  if ($user['login'] == $login && password_verify($
password, $user['password']) == TRUE) {
                    session_start();
                    $_SESSION['logged_user'] = $_POST['login'];
                    $_SESSION['status_id'] = $user['status_id'];
                    header('location: http://'.
$_SERVER['HTTP_HOST'].'/index.php');
                  }
                }
                echo('<p style="position: relative; font-weight: bo
ld; margin-top: 15px">Пользователя с указанным логином и паролем не сущес
твует</p>');
```

```
                    }
                  ?>
                </div>
                <button id="submit" class="btn btn-primary" style="positi
on: absolute; left: 50%; bottom: 0; margin-bottom: 25px; transform: trans
late(-50%, 0%)">Войти</button>
              </form>
            </div>
          </body>
      </html>
```

Файл index.php:

```
      <?php
        session_start();
        if(!isset($_SESSION['logged_user'])){
            header("Location: auth.php");
            exit;
        }
      ?>

      <!DOCTYPE html>
      <html>
          <head>
              <meta charset="utf-8">
              <title>Главная</title>
              <link href="https://cdnjs.cloudflare.com/ajax/libs/twitter-
bootstrap/3.3.7/css/bootstrap.css" rel="stylesheet" crossorigin="anonymou
s" >
              <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/
3.1.0/jquery.min.js"></script>
              <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.
7/js/bootstrap.min.js" crossorigin="anonymous"></script>
          </head>
          <body>
            <header>
              <nav class="navbar navbar-inverse bg-inverse" role="navigat
ion" style="background-color: #e3f2fd; border-color: #002f55" >
                <div class="container-fluid" background="#022d46">
                  <div class="navbar-header">
                    <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target=".navbar-ex1-collapse">
                      <span class="sr-only">Toggle navigation</span>
                      <span class="glyphicon glyphicon-menu-hamburger" st
yle="color: #002f55"></span>

                    </button>
                    <a class="navbar-brand" href="index.php" style="color
: #002f55">ПГУ</a>
                  </div>

                  <div class="collapse navbar-collapse navbar-ex1-
collapse">
                    <ul class="nav navbar-nav">
                      <li><a href="index.php" style="color: #002f55">Глав
ная</a></li>
                      <li><a href="MainTable.php" style="color: #002f55">
Просмотр</a></li>
```

```
                    <li><a href="Search.php" style="color: #002f55">Пои
ск</a></li>
                    <?php
                      if($_SESSION['status_id'] == 2) {
                        echo('<li><a href="unit.php" style="color: #002
f55">Подразделения</a></li>');
                        echo('<li><a href="users.php" style="color: #00
2f55">Пользователи</a></li>');
                      }
                    ?>
                    <li><a href="exit.php" style="color: #002f55">Выйти
</a></li>
                  </ul>
                </div>
              </div>
            </nav>
          </header>

          <div class="container-fluid">
            <?php include 'db.php'; ?>
            <?php include 'api.php'; ?>

            <div>
              <form action="" method="POST">
                <?php
                      if(isset($_POST['backup'])) {
                        backupDataBase($db);
                      }
                ?>
                <input style="visibility: hidden; width: 0" name="backu
p"></input>
                <?php
                  if($_SESSION['status_id'] == 2){
                    echo ('<button id="addButton1" class="btn btn-
default" style="margin-bottom: 20px">Создать резервную копию базы данных<
/button>');
                  }
                  else {
                    echo ('<p style="font-weight: bold">Операторы не мо
гут пользоваться функциями резервного копирования и восстановления</p>');
                  }
                ?>
              </form>

              <form action="" method="POST">
                <?php
                  if(isset($_POST['restore'])) {
                    recoveryDataBase($db);
                  }
                ?>
                <input style="visibility: hidden; width: 0" name="resto
re"></input>
                <?php
                  if($_SESSION['status_id'] == 2)
                    echo ('<button id="addButton1" class="btn btn-
default" style="margin-bottom: 20px">Восстановить базу данных</button>');
                ?>
              </form>
```

```
            </div>
          </div>
        </body>
    </html>
```

Файл MainTable.php:

```
    <?php
        session_start();
        if(!isset($_SESSION['logged_user'])){
            header("Location: auth.php");
            exit;
        }
    ?>

    <?php include 'db.php'; ?>
    <?php include 'api.php'; ?>

    <?php
        if(isset($_POST['delete']) && $_POST['Check'] != NULL) {
            $name = $_POST['Check'];
            foreach ($name as $r_id){
                deleteMainTableById($db, $r_id);
            }
            header('location: http://'.
$_SERVER['HTTP_HOST'].'/MainTable.php');
        }
    ?>

    <?php
        if(isset($_POST['ip']) && $_POST['ip'] != '') {
            $ip = $_POST['ip'];
            $room = $_POST['room'];
            $b_id = $_POST['build'];
            $d_id = $_POST['dep'];
            addMainTable($db, $ip, $room, $b_id, $d_id);
            header('location: http://'.
$_SERVER['HTTP_HOST'].'/MainTable.php');
            //exit();
        }
    ?>

    <!DOCTYPE html>
    <html lang="ru">
    <head>
        <meta charset="UTF-8">
        <title>Просмотр</title>
        <link href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/boots
trap.min.css" rel="stylesheet">
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.
0/jquery.min.js"></script>
        <script src="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/boots
trap.min.js"></script>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/tablesort/5
.0.2/tablesort.min.js"></script>

    </head>
    <body>
```

```
        <header>
            <nav class="navbar navbar-inverse bg-inverse" role="navigat
ion" style="background-color: #e3f2fd; border-color: #002f55" >
                <div class="container-fluid" background="#022d46">
                    <div class="navbar-header">
                        <button type="button" class="navbar-toggle" dat
a-toggle="collapse" data-target=".navbar-ex1-collapse">
                            <span class="sr-only">Toggle navigation</sp
an>
                            <span class="glyphicon glyphicon-menu-
hamburger" style="color: #002f55"></span>

                        </button>
                        <a class="navbar-brand" href="index.php" style=
"color: #002f55">ПГУ</a>
                    </div>

                    <div class="collapse navbar-collapse navbar-ex1-
collapse">
                        <ul class="nav navbar-nav">
                            <li><a href="index.php" style="color: #002f
55">Главная</a></li>
                            <li><a href="MainTable.php" style="color: #
002f55">Просмотр</a></li>
                            <li><a href="Search.php" style="color: #002
f55">Поиск</a></li>
                            <?php
                                if($_SESSION['status_id'] == 2) {
                                    echo('<li><a href="unit.php" style=
"color: #002f55">Подразделения</a></li>');
                                    echo('<li><a href="users.php" style
="color: #002f55">Пользователи</a></li>');
                                }
                            ?>
                            <li><a href="exit.php" style="color: #002f5
5">Выйти</a></li>
                        </ul>
                    </div>
                </div>
            </nav>
        </header>

        <div id="content">
            <div class="container-fluid">
                <button id="addButton" class="btn btn-
default">Добавить</button>
                <form id="form1" action="" method="POST" role="form" st
yle="display: none; margin-top: 20px; margin-right: 400px;">
                    <div class="form-group">
                        <label for="">Адрес</label>
                        <input type="text" class="form-control" id="ip"
name="ip" placeholder="ip" size="10px">
                    </div>

                    <div class="form-group">
                        <label for="">Аудитория</label>
                        <input type="text" class="form-control" id="roo
m" name="room" placeholder="room" size="10px">
```

```
                    </div>

                    <div class="form-group">
                        <select name="build" class="form-control" id="b
uild">
                        <?php
                            $getAllBbuildings = getAllBbuildings($db);
                            foreach ($getAllBbuildings as $key => $valu
e) {
                                echo "<option value=".
$value['b_id'].">".$value['build_name']."</option>";
                            }
                        ?>
                        </select>
                    </div>

                    <div class="form-group">
                        <select name="dep" class="form-control" id="dep
">
                        <?php
                            $getAllDepartment= getAllDepartment($db);
                            foreach ($getAllDepartment as $key => $valu
e) {
                                echo "<option value=".
$value['d_id'].">".$value['dep_name']."</option>";
                            }
                        ?>
                        </select>
                    </div>

                    <button type="submit" id="add"class="btn btn-
default">Добавить</button><br><br>
                </form>
            </div>

            <form id="form2" action="" method="POST">
                <?php
                    $rooms = getAllRooms($db);
                ?>
                <?php
                    if($_SESSION['status_id'] == 2)
                    echo('<button id="deleteButton" class="btn btn-
default" style="margin: 20px; margin-left: 15px; margin-top: 20px">Удалит
ь</button>');
                ?>
                <br><table id="table-id" class="table table-striped tab
le-hover table-responsive">
                <thead>
                    <tr>
                        <th>#</th>
                        <th>Адрес</th>
                        <th>Аудитория</th>
                        <th>Корпус</th>
                        <th>Подразделение</th>
                    </tr>
                    </thead>
                    <tbody>
                    <?php foreach ($rooms as $room) { ?>
```

```
                        <tr>
                            <td><input class="form-check-input" type="c
heckbox" value="<?php echo $room['r_id'];?>" name="Check[]" id="defaultCh
eck1"></td>
                            <td><a href="editMain.php?r_id=<?php echo $
room['r_id'];?>"><?php echo $room['ip_address']; ?></td>
                            <td><?php echo $room['room_name']; ?></td>
                            <td><?php echo $room['build_name']; ?></td>

                            <td><?php echo $room['dep_name']; ?></td>
                        </tr>
                    <?php } ?>
                    </tbody>
                </table>
                <input style="visibility: hidden; width: 0" name="delet
e"></input>
            </form>
        </div>

        <footer>
        </footer>

        <script>
            new Tablesort(document.getElementById('table-id'));
        </script>

        <script>
            $("#addButton").click(function(){
                $("#form1").slideToggle();
            });
        </script>
    </body>
    </html>
```

Файл Search.php:

```
    <?php
        session_start();
        if(!isset($_SESSION['logged_user'])){
            header("Location: auth.php");
            exit;
        }
    ?>

    <!DOCTYPE html>
    <html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>Поиск</title>
        <link href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/boots
trap.min.css" rel="stylesheet">
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.
0/jquery.min.js"></script>
        <script src="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/boots
trap.min.js"></script>
    </head>
    <body>
        <header>
```

```
            <nav class="navbar navbar-inverse bg-inverse" role="navigat
ion" style="background-color: #e3f2fd; border-color: #002f55" >
                <div class="container-fluid" background="#022d46">
                    <div class="navbar-header">
                        <button type="button" class="navbar-toggle" dat
a-toggle="collapse" data-target=".navbar-ex1-collapse">
                            <span class="sr-only">Toggle navigation</sp
an>
                            <span class="glyphicon glyphicon-menu-
hamburger" style="color: #002f55"></span>

                        </button>
                        <a class="navbar-brand" href="index.php" style=
"color: #002f55">ПГУ</a>
                    </div>

                    <div class="collapse navbar-collapse navbar-ex1-
collapse">
                        <ul class="nav navbar-nav">
                            <li><a href="index.php" style="color: #002f
55">Главная</a></li>
                            <li><a href="MainTable.php" style="color: #
002f55">Просмотр</a></li>
                            <li><a href="Search.php" style="color: #002
f55">Поиск</a></li>
                            <?php
                                if($_SESSION['status_id'] == 2) {
                                    echo('<li><a href="unit.php" style=
"color: #002f55">Подразделения</a></li>');
                                    echo('<li><a href="users.php" style
="color: #002f55">Пользователи</a></li>');
                                }
                            ?>
                            <li><a href="exit.php" style="color: #002f5
5">Выйти</a></li>
                        </ul>
                    </div>
                </div>
            </nav>
        </header>

        <div id="content">
            <div class="container-fluid">
                <?php include 'db.php'; ?>
                <?php include 'api.php'; ?>

                <button id="findButton1" type="button" class="btn btn-
primary btn btn-block">Поиск по подразделению</button><br>
                <form id="form1" action="" method="POST" role="form" st
yle="margin-top: 20px;" >
                    <?php

                    ?>
                    <label for="">Выберите подразделение</label>

                    <div class="form-group">
                        <select name="unit" class="form-control" id="un
it" onchange="submit()">
```

```
                                <option></option>
                                <?php
                                    $departments = getAllDepartment($db);
                                    foreach ($departments as $key => $value
) {
                                        echo "<option value=".
$value['d_id'].">".$value['dep_name']."</option>";
                                    }
                                ?>
                            </select>
                        </div>

                        <table class="table table-striped table-hover table
-responsive ">
                            <tr>
                                <th>Подразделение</th>
                                <th>Глава</th>
                                <th>Номер</th>
                            </tr>

                            <?php
                                session_start();
                                $unit = $_POST['unit'];
                                if(isset($_POST['unit']) && $_POST['unit']
!= '')
                                    $_SESSION['unit']=$_POST['unit'];
                                $findDepartment=findDepartment($db, $unit);
                                foreach ($findDepartment as $findDep) { ?>
                                    <tr>
                                        <td><?php echo $findDep['dep_name']
; ?></td>
                                        <td><?php echo $findDep['head']; ?
></td>
                                        <td><?php echo $findDep['phone']; ?
></td>
                                    </tr>
                            <?php } ?>
                        </table>

                        <label for="">Выберите корпус</label>
                        <div class="form-group">
                            <select name="build" class="form-control" id="b
uild" onchange="submit()" style="margin: 15px; margin-left: 0px; margin-
top: 0px">
                                <option></option>
                                <?php
                                    $findbuildings = findBuildings($db, $un
it);
                                    foreach ($findbuildings as $key => $val
ue) {
                                        echo "<option value=".
$value['b_id'].">".$value['build_name']."</option>";
                                    }?>
                            </select>

                            <?php
                                $build = $_POST['build'];
                                $unit = $_SESSION['unit'];
```

```
                            $rooms = getAllPCInBuilding($db, $build, $u
nit);
                        ?>

                        <table class="table table-striped table-hover t
able-responsive ">
                            <tr>
                                <th>#</th>
                                <th>Адрес</th>
                                <th>Аудитория</th>
                                <th>Корпус</th>
                                <th>Подразделение</th>

                            </tr>
                            <?php foreach ($rooms as $room) { ?>
                                <tr>
                                    <td><input class="form-check-input"
type="checkbox" value="" id="defaultCheck2"></td>
                                    <td><?php echo $room['ip_address'];
?></td>
                                    <td><?php echo $room['room_name'];
?></td>
                                    <td><?php echo $room['build_name'];
?></td>
                                    <td><?php echo $room['dep_name']; ?
></td>
                                </tr>
                            <?php } ?>
                        </table>
                    </div>
                </form>

                <button id="findButton2" type="button" class="btn btn-
primary  btn btn-block">Поиск по ip-адресу </button><br>
                <form1 id="form2" action="" method="G" role="form" styl
e="margin-top: 20px;">
                    <label for="">Введите IP</label>

                    <div class="form-group">
                        <form action="" method="POST" role="form">
                            <input type="text" class="form-control" id=
"ipaddress" name="ipaddress" placeholder="Ipaddress" size="10px">
                            <button id="findButton3" type="submit" clas
s="btn btn-default" style="margin: 15px; margin-left: 0px">Найти</button>
                        </form>

                        <table class="table table-striped table-hover t
able-responsive ">
                            <tr>
                                <th>#</th>
                                <th>Адрес</th>
                                <th>Аудитория</th>
                                <th>Корпус</th>
                                <th>Подразделение</th>
                            </tr>

                            <?php
```

```
                                    if(isset($_POST['ipaddress']) && $_POST
['ipaddress'] != '') {
                                        $ipaddress = $_POST['ipaddress'];
                                    }
                                    $rooms_ip = getPCByIP($db, $ipaddress);
                                ?>

                                <?php foreach ($rooms_ip as $room) { ?>
                                    <tr>
                                        <td><input class="form-check-input"
type="checkbox" value="" id="defaultCheck2"></td>
                                        <td><?php echo $room['ip_address'];
?></td>
                                        <td><?php echo $room['room_name'];
?></td>
                                        <td><?php echo $room['build_name'];
?></td>
                                        <td><?php echo $room['dep_name']; ?
></td>
                                    </tr>
                                <?php } ?>
                            </table>
                        </div>
                    </form1>
                </div>
            </div>

            <footer>
            </footer>

            <script>
                function selected()
                {
                    return $("#unit option:selected").value();
                };
            </script>

            <script>
                $("#findButton1").click(function(){
                    $("#form1").slideToggle();
                });
            </script>

            <script>
                $("#findButton2").click(function(){
                    $("#form2").slideToggle();
                });
            </script>

        </body>
        </html>
```

Файл unit.php:

```
        <?php
            session_start();
            if(!isset($_SESSION['logged_user'])){
                header("Location: auth.php");
```

```
            exit;
        }
        if($_SESSION['status_id'] != 2){
            header("Location: auth.php");
            exit;
        }
    ?>

    <?php include 'db.php'; ?>
    <?php include 'api.php'; ?>

    <?php
        if(isset($_POST['delete']) && $_POST['Check'] != NULL) {
            $name = $_POST['Check'];
            foreach ($name as $unit){
                deleteDepartment($db, $unit);
            }
            header('location: http://'.
$_SERVER['HTTP_HOST'].'/unit.php');
        }
    ?>

    <!DOCTYPE html>
    <? ob_start();?>
    <html lang="ru">
    <head>
        <meta charset="UTF-8">
        <title>Подразделения</title>
        <link href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/boots
trap.min.css" rel="stylesheet">
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.
0/jquery.min.js"></script>
        <script src="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/boots
trap.min.js"></script>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/tablesort/5
.0.2/tablesort.min.js"></script>
    </head>
    <body>
        <header>
            <nav class="navbar navbar-inverse bg-inverse" role="navigat
ion" style="background-color: #e3f2fd; border-color: #002f55" >
                <div class="container-fluid" background="#022d46">
                    <div class="navbar-header">
                        <button type="button" class="navbar-toggle" dat
a-toggle="collapse" data-target=".navbar-ex1-collapse">
                            <span class="sr-only">Toggle navigation</sp
an>
                            <span class="glyphicon glyphicon-menu-
hamburger" style="color: #002f55"></span>

                        </button>
                        <a class="navbar-brand" href="index.php" style=
"color: #002f55">ПГУ</a>
                    </div>

                    <div class="collapse navbar-collapse navbar-ex1-
collapse">
                        <ul class="nav navbar-nav">
```

```
                            <li><a href="index.php" style="color: #002f
55">Главная</a></li>
                            <li><a href="MainTable.php" style="color: #
002f55">Просмотр</a></li>
                            <li><a href="Search.php" style="color: #002
f55">Поиск</a></li>
                            <li><a href="unit.php" style="color: #002f5
5">Подразделения</a></li>
                            <li><a href="users.php" style="color: #002f
55">Пользователи</a></li>
                            <li><a href="exit.php" style="color: #002f5
5">Выйти</a></li>
                        </ul>
                    </div>
                </div>
            </nav>
        </header>

        <div id="content">
            <div class="container-fluid">
                <?php
                if(isset($_POST['dep']) && $_POST['dep'] != '' ) {
                    $ancestor = $_POST['dep'];
                    $dep_name = $_POST['dep_name'];
                    $head = $_POST['head'];
                    $phone = $_POST['phone'];
                    addDepartment($db, $ancestor, $dep_name, $head, $ph
one);
                    header ("Location: unit.php");
                    ob_end_flush();
                    exit();
                } ?>

                <form action="" method="POST">
                    <?php
                        $departments = getAllDepartment($db);
                    ?>

                    <table id="tablesortdep" class="table table-
bordered">
                    <thead>
                        <tr>
                            <th>№</th>
                            <th>Название подразделения</th>
                            <th>Руководитель</th>
                            <th>Телефон</th>
                        </tr>
                        </thead>
                        <tbody>
                        <?php
                        foreach ($departments as $department) { ?>
                            <tr>
                                <td><input class="form-check-input" typ
e="checkbox" value="<?php echo $department['d_id'];?>" name="Check[]" id=
"defaultCheck1"></td>
                                <td><a href="edit.php?id=<?php echo $de
partment['d_id'];?>"><?php echo $department['dep_name']; ?></a></td>
```

```
                                    <td><?php echo $department['head']; ?
></td>
                                    <td><?php echo $department['phone']; ?
></td>
                                </tr>
                            <?php } ?>
                            </tbody>
                        </table>

                        <input style="visibility: hidden; width: 0" name="d
elete"></input>
                        <button id="deleteButton" class="btn btn-default" s
tyle="margin: 5px; margin-left: -10px">Удалить подразделение</button>
                    </form>

                    <button id="addButton" class="btn btn-default" style="m
argin: 5px; margin-left: 1px">Добавить подразделение</button>
                    <form id="form1" action="" method="POST" role="form" st
yle="display: none; margin-top: 20px;margin-right: 400px;">
                        <label for="">Выберите над подразделение</label>
                        <div class="form-group">
                            <select name="dep" class="form-control" id="dep
">
                                <?php
                                    $getAllDepartment= getAllDepartment($db
);
                                    foreach ($getAllDepartment as $key => $
value) {
                                        echo "<option value=".
$value['d_id'].">".$value['dep_name']."</option>";
                                    }
                                ?>
                            </select>
                        </div>

                        <div class="form-group">
                            <label for="">Название подразделения</label>
                            <input type="text" class="form-control" id="dep
_name" name="dep_name" placeholder="Руководитель подразделения" size="10p
x">
                        </div>

                        <div class="form-group">
                            <label for="">Руководитель подразделения</label
>
                            <input type="text" class="form-control" id="hea
d" name="head" placeholder="Руководитель подразделения" size="10px">
                        </div>

                        <div class="form-group">
                            <label for="">Телефон</label>
                            <input type="text" class="form-control" id="pho
ne" name="phone" placeholder="Телефон" size="10px">
                        </div>

                        <button type="submit" class="btn btn-
default">Добавить</button>
                    </form>
```

```
                <form id="form2" action="" method="POST" role="form" st
yle="display: none; margin-top: 20px;margin-right: 400px;">
                    <?php
                        if(isset($_POST['dep_name'])){
                            $id = $_GET['id'];
                            if($id) {
                                $department = getDepartmentById($db, $i
d);
                            }
                            else {
                                echo '<div class="alert alert-danger" r
ole="alert">
                                        <span class="glyphicon glyphico
n-exclamation-sign" aria-hidden="true"></span>
                                        <span class="sr-
only">Error:</span>
                                        Подразделение не найдено
                                        </div>';
                            }
                        }
                    ?>
                    <div class="form-group">
                        <label for="name">Название подразделения</label
>
                        <input type="text" class="form-control" id="dep
_name" name="dep_name" value="<?php echo $department[$id]; ?>">
                    </div>
                      <button type="submit" class="btn btn-
default">Сохранить</button>
                </form>
            </div>
        </div>

        <script>
            $("#addButton").click(function(){
                $("#form1").slideToggle();
            });
        </script>

        <script>
                new Tablesort(document.getElementById('tablesortdep'));
        </script>

        <script>
            $("#changeButton").click(function(){
                $("#form2").slideToggle();
            });
        </script>
    </body>
    </html>
```

Файл users.php:

```
    <?php
        session_start();
        if(!isset($_SESSION['logged_user'])){
            header("Location: auth.php");
```

```
            exit;
        }
        if($_SESSION['status_id'] != 2){
            header("Location: auth.php");
            exit;
        }
    ?>

    <?php include 'db.php'; ?>
    <?php include 'api.php'; ?>

    <?php
        if(isset($_POST['login']) && $_POST['login'] != '') {
            $login = $_POST['login'];
            $password = $_POST['password'];
            $name = $_POST['name'];
            $email = $_POST['email'];
            $status_id = $_POST['status'];
            addUser($db, $login, $password, $name, $email, $status_id);
            header ("Location: users.php");
            ob_end_flush();
        }
    ?>

    <?php
        if(isset($_POST['delete']) && $_POST['Check'] != NULL) {
            $name = $_POST['Check'];
            foreach ($name as $user){
                deleteUser($db, $user);
            }
            header('location: http://'.
$_SERVER['HTTP_HOST'].'/users.php');
        }
    ?>

    <!DOCTYPE html>
    <? ob_start();?>
    <html lang="ru">
    <head>
        <meta charset="UTF-8">
        <title>Пользователи</title>
        <link href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/boots
trap.min.css" rel="stylesheet">
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.
0/jquery.min.js"></script>
        <script src="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/boots
trap.min.js"></script>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/tablesort/5
.0.2/tablesort.min.js"></script>
    </head>
    <body>
        <header>
            <nav class="navbar navbar-inverse bg-inverse" role="navigat
ion" style="background-color: #e3f2fd; border-color: #002f55" >
                <div class="container-fluid" background="#022d46">
                    <div class="navbar-header">
                        <button type="button" class="navbar-toggle" dat
a-toggle="collapse" data-target=".navbar-ex1-collapse">
```

```
                            <span class="sr-only">Toggle navigation</sp
an>
                            <span class="glyphicon glyphicon-menu-
hamburger" style="color: #002f55"></span>

                        </button>
                        <a class="navbar-brand" href="index.php" style=
"color: #002f55">ПГУ</a>
                    </div>

                    <div class="collapse navbar-collapse navbar-ex1-
collapse">
                        <ul class="nav navbar-nav">
                            <li><a href="index.php" style="color: #002f
55">Главная</a></li>
                            <li><a href="MainTable.php" style="color: #
002f55">Просмотр</a></li>
                            <li><a href="Search.php" style="color: #002
f55">Поиск</a></li>
                            <?php
                                if($_SESSION['status_id'] == 2) {
                                    echo('<li><a href="unit.php" style=
"color: #002f55">Подразделения</a></li>');
                                    echo('<li><a href="users.php" style
="color: #002f55">Пользователи</a></li>');
                                }
                            ?>
                            <li><a href="exit.php" style="color: #002f5
5">Выйти</a></li>
                        </ul>
                    </div>
                </div>
            </nav>
        </header>

        <div id="content">
            <div class="container-fluid">
                <?php
                    $users = getAllUsers($db);
                ?>
                <form action="" method="POST" role="form" style="">
                    <table id="tablesortuser" class="table table-
bordered">
                    <thead>
                        <tr>
                            <th>#</th>
                            <th>Логин</th>
                            <th>Пароль</th>
                            <th>ФИО</th>
                            <th>e-mail</th>
                            <th>Тип пользователя</th>

                        </tr>
                        </thead>
                        <tbody>
                        <?php foreach ($users as $user) { ?>
                            <tr>
```

```
                                    <td><input class="form-check-input" typ
e="checkbox" value="<?php echo $user['user_id'];?>" name="Check[]" id="de
faultCheck1"></td>
                                    <td><a href="edituser.php?user_id=<?php
echo $user['user_id']; ?>"><?php echo $user['login']; ?></a></td>
                                    <td><?php echo $user['password']; ?
></td>
                                    <td><?php echo $user['name']; ?></td>
                                    <td><?php echo $user['email']; ?></td>
                                    <td><?php echo $user['status_name']; ?
></td>

                                </tr>
                            <?php } ?>
                            </tbody>
                        </table>

                        <input style="visibility: hidden; width: 0" name="d
elete"></input>
                        <button id="deleteButton" class="btn btn-
default">Удалить пользователя</button>
                    </form>

                    <button id="addButton" class="btn btn-default" style="m
argin-top: 15px; margin-left: 5px">Добавить пользователя</button>
                    <form id="form1" action="" method="POST" role="form" st
yle="display: none; margin-top: 20px; margin-right: 400px;">
                        <div class="form-group">
                            <label for="">Логин</label>
                            <input type="text" class="form-control" id="log
in" name="login" placeholder="Login" size="10px">
                        </div>

                        <div class="form-group">
                            <label for="">Пароль</label>
                            <input type="text" class="form-control" id="pas
sword" name="password" placeholder="Password" size="10px">
                        </div>

                        <div class="form-group">
                            <label for="">ФИО</label>
                            <input type="text" class="form-control" id="nam
e" name="name" placeholder="Name" size="10px">
                        </div>

                        <div class="form-group">
                            <label for="">E-mail</label>
                            <input type="text" class="form-control" id="ema
il" name="email" placeholder="Email" size="10px">
                        </div>

                        <label for="">Выберите тип пользователя</label>
                        <div class="form-group">
                            <select name="status" class="form-control" id="
status">
                                <?php
                                    $statuses = getAllStatus($db);
                                    foreach ($statuses as $key => $value) {
```

```
                                        echo "<option value=".
$value['status_id'].">".$value['status_name']."</option>";
                                    }
                                ?>
                            </select>
                        </div>

                        <button type="submit" id="add"class="btn btn-
default">Добавить</button><br><br><br>
                    </form>
                </div>
            </div>

            <script>
                new Tablesort(document.getElementById('tablesortuser'));
            </script>

            <script>
                $("#addButton").click(function(){
                    $("#form1").slideToggle();
                });
            </script>
        </body>
        </html>
```

Файл saveMain.php:

```
<?php
    session_start();
    if(!isset($_SESSION['logged_user'])){
        header("Location: auth.php");
        exit;
    }
?>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <title>Ошибка сохранения</title>
    <link href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/boots
trap.min.css" rel="stylesheet">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.
0/jquery.min.js"></script>
    <script src="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/boots
trap.min.js"></script>
</head>
<body>
    <header>
        <nav class="navbar navbar-inverse bg-inverse" role="navigat
ion" style="background-color: #e3f2fd; border-color: #002f55" >
            <div class="container-fluid" background="#022d46">
                <div class="navbar-header">
                    <button type="button" class="navbar-toggle" dat
a-toggle="collapse" data-target=".navbar-ex1-collapse">
                        <span class="sr-only">Toggle navigation</sp
an>
                        <span class="glyphicon glyphicon-menu-
hamburger" style="color: #002f55"></span>

                    </button>
                    <a class="navbar-brand" href="index.php" style=
"color: #002f55">ПГУ</a>
                </div>

                <div class="collapse navbar-collapse navbar-ex1-
collapse">
                    <ul class="nav navbar-nav">
                        <li><a href="index.php" style="color: #002f
55">Главная</a></li>
                        <li><a href="MainTable.php" style="color: #
002f55">Просмотр</a></li>
                        <li><a href="Search.php" style="color: #002
f55">Поиск</a></li>
                        <?php
                            if($_SESSION['status_id'] == 2) {
                                echo('<li><a href="unit.php" style=
"color: #002f55">Подразделения</a></li>');
                                echo('<li><a href="users.php" style
="color: #002f55">Пользователи</a></li>');
                            }
                        ?>
```

```
                            <li><a href="exit.php" style="color: #002f5
5">Выйти</a></li>
                        </ul>
                    </div>
                </div>
            </nav>
        </header>

        <div class="container-fluid">
                <?php include 'db.php'; ?>
                <?php include 'api.php'; ?>
                <?php
                    if(!empty($_POST['r_id'])){
                        $r_id = $_POST['r_id'];
                        $ip_address = $_POST['ip_address'];
                        $room_name = $_POST['room_name'];
                        $build_name = $_POST['build_name'];
                        $dep_name = $_POST['dep_name'];
                        changeMainTable($db, $ip_address, $room_name, $
build_name, $dep_name, $r_id);
                        header("Location: MainTable.php");
                        exit;
                    }
                    else{
                        echo '<div class="alert alert-danger" role="ale
rt" style="margin-left: 45px; margin-top: 30px; width: 250px">
                                <span class="glyphicon glyphicon-
exclamation-sign" aria-hidden="true"></span>
                                    <span class="sr-only">Error:</span>
                                    Ошибка сохранения
                                </div>';
                    }
                ?>
            </div>
        </body>
    </html>
```

Файл saveDep.php:

```
    <?php
        session_start();
        if(!isset($_SESSION['logged_user']) && $_SESSION['status_id'] !
= 2){
            header("Location: auth.php");
            exit;
        }
    ?>

    <!DOCTYPE html>
    <html lang="ru">
    <head>
        <meta charset="UTF-8">
        <title>Ошибка сохранения</title>
        <link href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/boots
trap.min.css" rel="stylesheet">
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.
0/jquery.min.js"></script>
```

```
        <script src="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/boots
trap.min.js"></script>
    </head>
    <body>
        <header>
            <nav class="navbar navbar-inverse bg-inverse" role="navigat
ion" style="background-color: #e3f2fd; border-color: #002f55" >
                <div class="container-fluid" background="#022d46">
                    <div class="navbar-header">
                        <button type="button" class="navbar-toggle" dat
a-toggle="collapse" data-target=".navbar-ex1-collapse">
                            <span class="sr-only">Toggle navigation</sp
an>
                            <span class="glyphicon glyphicon-menu-
hamburger" style="color: #002f55"></span>
                        </button>
                        <a class="navbar-brand" href="index.php" style=
"color: #002f55">ПГУ</a>
                    </div>

                    <div class="collapse navbar-collapse navbar-ex1-
collapse">
                        <ul class="nav navbar-nav">
                            <li><a href="index.php" style="color: #002f
55">Главная</a></li>
                            <li><a href="MainTable.php" style="color: #
002f55">Просмотр</a></li>
                            <li><a href="Search.php" style="color: #002
f55">Поиск</a></li>
                            <li><a href="unit.php" style="color: #002f5
5">Подразделения</a></li>
                            <li><a href="users.php" style="color: #002f
55">Пользователи</a></li>
                            <li><a href="exit.php" style="color: #002f5
5">Выйти</a></li>
                        </ul>
                    </div>
                </div>
            </nav>
        </header>

        <div class="container-fluid">
                <?php include 'db.php'; ?>
                <?php include 'api.php'; ?>
                <?php
                    if(empty($_POST['d_id']))
                    {
                        $dep_name = $_POST['dep_name'];
                        $head = $_POST['head'];
                        $phone = $_POST['phone'];
                        $d_id = $_POST['d_id'];
                        changeDepartment($db, $dep_name, $phone, $head,
$d_id);
                        header('location: http://'.
$_SERVER['HTTP_HOST'].'/unit.php');
                    }
                    else {
```

```
                        echo '<div class="alert alert-danger" role="ale
rt">
                                <span class="glyphicon glyphicon-
exclamation-sign" aria-hidden="true"></span>
                                    <span class="sr-only">Error:</span>
                                    Ошибка сохранения
                                </div>';
                    }
                ?>
        </div>
    </body>
    </html>
```

Файл save.php:

```
    <?php
        session_start();
        if(!isset($_SESSION['logged_user']) && $_SESSION['status_id'] !
= 2){
            header("Location: auth.php");
            exit;
        }
    ?>

    <!DOCTYPE html>
    <html lang="ru">
    <head>
        <meta charset="UTF-8">
        <title>Ошибка сохранения</title>
        <link href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/boots
trap.min.css" rel="stylesheet">
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.
0/jquery.min.js"></script>
        <script src="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/boots
trap.min.js"></script>
    </head>
    <body>
        <header>
            <nav class="navbar navbar-inverse bg-inverse" role="navigat
ion" style="background-color: #e3f2fd; border-color: #002f55" >
                <div class="container-fluid" background="#022d46">
                    <div class="navbar-header">
                        <button type="button" class="navbar-toggle" dat
a-toggle="collapse" data-target=".navbar-ex1-collapse">
                            <span class="sr-only">Toggle navigation</sp
an>
                            <span class="glyphicon glyphicon-menu-
hamburger" style="color: #002f55"></span>
                        </button>
                        <a class="navbar-brand" href="index.php" style=
"color: #002f55">ПГУ</a>
                    </div>

                    <div class="collapse navbar-collapse navbar-ex1-
collapse">
                        <ul class="nav navbar-nav">
                            <li><a href="index.php" style="color: #002f
55">Главная</a></li>
```

```
                                <li><a href="MainTable.php" style="color: #
002f55">Просмотр</a></li>
                                <li><a href="Search.php" style="color: #002
f55">Поиск</a></li>
                                <li><a href="unit.php" style="color: #002f5
5">Подразделения</a></li>
                                <li><a href="users.php" style="color: #002f
55">Пользователи</a></li>
                                <li><a href="exit.php" style="color: #002f5
5">Выйти</a></li>
                            </ul>
                        </div>
                    </div>
                </nav>
            </header>

            <div class="container-fluid">
                    <?php include 'db.php'; ?>
                    <?php include 'api.php'; ?>
                    <?php
                    var_dump($_POST);
                    var_dump($_POST['login']);
                         if(!empty($_POST['user_id']))
                          {

                            $login = $_POST['login'];
                            $password = $_POST['password'];
                            $name = $_POST['name'];
                            $email = $_POST['email'];
                            $status_id = $_POST['status_id'];
                            $user_id = $_POST['user_id'];
                            changeUser($db, $user_id, $login, $password, $n
ame, $email, $status_id);
                            header('location: http://'.
$_SERVER['HTTP_HOST'].'/users.php');
                            }
                            else {
                                echo '<div class="alert alert-danger" role=
"alert">
                                        <span class="glyphicon glyphicon-
exclamation-sign" aria-hidden="true"></span>
                                          <span class="sr-
only">Error:</span>
                                          Ошибка сохранения
                                        </div>';
                            }
                    ?>
            </div>
        </body>
        </html>
```

Файл editMain.php:

```
<?php
    session_start();
    if(!isset($_SESSION['logged_user'])){
        header("Location: auth.php");
        exit;
    }
?>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <title>Редактировать</title>
    <link href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/boots
trap.min.css" rel="stylesheet">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.
0/jquery.min.js"></script>
    <script src="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/boots
trap.min.js"></script>
</head>
<body>
    <header>
        <nav class="navbar navbar-inverse bg-inverse" role="navigat
ion" style="background-color: #e3f2fd; border-color: #002f55" >
            <div class="container-fluid" background="#022d46">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target=".navbar-ex1-collapse">
                <span class="sr-only">Toggle navigation</span>
                <span class="glyphicon glyphicon-menu-hamburger" st
yle="color: #002f55"></span>

                </button>
                <a class="navbar-brand" href="index.php" style="col
or: #002f55">ПГУ</a>
            </div>

            <div class="collapse navbar-collapse navbar-ex1-
collapse">
                <ul class="nav navbar-nav">
                <li><a href="index.php" style="color: #002f55">Глав
ная</a></li>
                <li><a href="MainTable.php" style="color: #002f55">
Просмотр</a></li>
                <li><a href="Search.php" style="color: #002f55">Пои
ск</a></li>
                <?php
                    if($_SESSION['status_id'] == 2) {
                        echo('<li><a href="unit.php" style="color:
#002f55">Подразделения</a></li>');
                        echo('<li><a href="users.php" style="color:
#002f55">Пользователи</a></li>');
                    }
                ?>
                <li><a href="exit.php" style="color: #002f55">Выйти
</a></li>
```

```
                    </ul>
                </div>
                </div>
            </nav>
        </header>

        <div class="container-fluid">
            <?php include 'db.php'; ?>
            <?php include 'api.php'; ?>
            <?php
                    $r_id = $_GET['r_id'];
                    if($r_id) {
                        $room = getMainTableById($db, $r_id);
                    }
                    else {
                        echo '<div class="alert alert-danger" role="ale
rt">
                                    <span class="glyphicon glyphicon-
exclamation-sign" aria-hidden="true"></span>
                                        <span class="sr-only">Error:</span>
                                        Пользователь не найден
                                    </div>';
                    }

                ?>
            <form action="saveMain.php" method="post">

                <input type="hidden" name="r_id" value="<?php echo $roo
m['r_id']; ?>">

                    <label for="" style="margin-top: 10px">Адрес</label
>
                    <input type="text" class="form-control" id="ip_addr
ess" name="ip_address" value="<?php echo $room['ip_address']; ?>">

                    <label for="" style="margin-top: 10px">Аудитория</l
abel>
                    <input type="text" class="form-control" id="room_na
me" name="room_name" value="<?php echo $room['room_name']; ?>">

                    <label for="" style="margin-top: 10px">Подразделени
e</label>
                    <select name="dep_name" class="form-control" id="de
p_name">
                        <option><?php echo $room['dep_name'];?
></option>
                        <?php
                            $departments = getAllDepartment($db);
                            foreach ($departments as $key => $value) {
                                echo "<option>".
$value['dep_name']."</option>";
                            }
                        ?>
                    </select>

                    <label for="" style="margin-top: 10px">Корпус</labe
l>
```

```
                    <select name="build_name" class="form-control" id="
build_name">
                        <option><?php echo $room['build_name'];?
></option>
                        <?php
                            $departments = getAllBbuildings($db);
                            foreach ($departments as $key => $value) {
                                echo "<option>".
$value['build_name']."</option>";
                            }
                        ?>
                    </select>

                </div>

                <button type="submit" class="btn btn-default" style="ma
rgin: 15px">Сохранить</button>
            </form>
        </div>
    </body>
```

Файл edit.php:

```
    <?php
        session_start();
        if(!isset($_SESSION['logged_user'])){
            header("Location: auth.php");
            exit;
        }
        if($_SESSION['status_id'] != 2){
            header("Location: unit.php");
        }
    ?>

    <!DOCTYPE html>
    <html lang="ru">
    <head>
        <meta charset="UTF-8">
        <title>Редактировать</title>
        <link href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/boots
trap.min.css" rel="stylesheet">
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.
0/jquery.min.js"></script>
        <script src="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/boots
trap.min.js"></script>
    </head>
    <body>
        <header>
            <nav class="navbar navbar-inverse bg-inverse" role="navigat
ion" style="background-color: #e3f2fd; border-color: #002f55" >
                <div class="container-fluid" background="#022d46">
                <div class="navbar-header">
                    <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target=".navbar-ex1-collapse">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="glyphicon glyphicon-menu-hamburger" st
yle="color: #002f55"></span>
```

```
                    </button>
                    <a class="navbar-brand" href="index.php" style="col
or: #002f55">ПГУ</a>
                </div>

                <div class="collapse navbar-collapse navbar-ex1-
collapse">
                    <ul class="nav navbar-nav">
                    <li><a href="index.php" style="color: #002f55">Глав
ная</a></li>
                    <li><a href="MainTable.php" style="color: #002f55">
Просмотр</a></li>
                    <li><a href="Search.php" style="color: #002f55">Пои
ск</a></li>
                    <li><a href="unit.php" style="color: #002f55">Подра
зделения</a></li>
                    <li><a href="users.php" style="color: #002f55">Поль
зователи</a></li>
                    <li><a href="exit.php" style="color: #002f55">Выйти
</a></li>
                    </ul>
                </div>
                </div>
            </nav>
        </header>

        <div class="container-fluid">
            <?php include 'db.php'; ?>
            <?php include 'api.php'; ?>
            <?php
                $id = $_GET['id'];
                if($id) {
                    $department = getDepartmentById($db, $id);
                }
                else {
                    echo '<div class="alert alert-danger" role="alert">
                            <span class="glyphicon glyphicon-
exclamation-sign" aria-hidden="true"></span>
                                <span class="sr-only">Error:</span>
                                Подразделение не найдено
                            </div>';
                }
            ?>
            <form action="saveDep.php" method="post">
                <input type="hidden" name="d_id" value="<?php echo $dep
artment['d_id']; ?>">
                <div class="form-group">
                    <label for="name" style="margin-top: 10px">Название
подразделения</label>
                    <input type="text" class="form-control" id="dep_nam
e" name="dep_name" value="<?php echo $department['dep_name']; ?>">
                    <label for="name" style="margin-top: 10px">Руководи
тель</label>
                    <input type="text" class="form-control" id="head" n
ame="head" value="<?php echo $department['head']; ?>">
                    <label for="name" style="margin-top: 10px">Телефон<
/label>
```

```
                    <input type="text" class="form-control" id="phone"
name="phone" value="<?php echo $department['phone']; ?>">
                </div>
                    <button type="submit" class="btn btn-
default">Сохранить</button>
            </form>
        </div>
    </body>
```

Файл edituser.php:

```
    <?php
        session_start();
        if(!isset($_SESSION['logged_user'])){
            header("Location: auth.php");
            exit;
        }
        if($_SESSION['status_id'] != 2){
            header("Location: users.php");
        }
    ?>

    <!DOCTYPE html>
    <html lang="ru">
    <head>
        <meta charset="UTF-8">
        <title>Редактировать</title>
        <link href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/boots
trap.min.css" rel="stylesheet">
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.
0/jquery.min.js"></script>
        <script src="//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/boots
trap.min.js"></script>
    </head>
    <body>
        <header>
            <nav class="navbar navbar-inverse bg-inverse" role="navigat
ion" style="background-color: #e3f2fd; border-color: #002f55" >
                <div class="container-fluid" background="#022d46">
                <div class="navbar-header">
                    <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target=".navbar-ex1-collapse">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="glyphicon glyphicon-menu-hamburger" st
yle="color: #002f55"></span>

                    </button>
                    <a class="navbar-brand" href="index.php" style="col
or: #002f55">ПГУ</a>
                </div>

                <div class="collapse navbar-collapse navbar-ex1-
collapse">
                    <ul class="nav navbar-nav">
                    <li><a href="index.php" style="color: #002f55">Глав
ная</a></li>
                    <li><a href="MainTable.php" style="color: #002f55">
Просмотр</a></li>
```

```
                    <li><a href="Search.php" style="color: #002f55">Пои
ск</a></li>
                    <li><a href="unit.php" style="color: #002f55">Подра
зделения</a></li>
                    <li><a href="users.php" style="color: #002f55">Поль
зователи</a></li>
                    <li><a href="exit.php" style="color: #002f55">Выйти
</a></li>
                    </ul>
                </div>
                </div>
            </nav>
        </header>

        <div class="container-fluid">
            <?php include 'db.php'; ?>
            <?php include 'api.php'; ?>
            <?php
                $user_id = $_GET['user_id'];
                if($user_id) {
                    $user = getUserById($db, $user_id);
                }
                else {
                    echo '<div class="alert alert-danger" role="alert">
                            <span class="glyphicon glyphicon-
exclamation-sign" aria-hidden="true"></span>
                                <span class="sr-only">Error:</span>
                                Пользователь не найден
                            </div>';
                }
            ?>
            <form action="save.php" method="post">
                <input type="hidden" name="user_id" value="<?php echo $
user['user_id']; ?>">
                <div class="form-group">
                    <label for="name" style="margin-top: 10px">Логин</l
abel>
                    <input type="text" class="form-control" id="login"
name="login" value="<?php echo $user['login']; ?>">
                </div>
                <div class="form-group">
                    <label for="" style="margin-top: 10px">Пароль</labe
l>
                    <input type="text" class="form-control" id="passwor
d" name="password" value="<?php echo $user['password'];?>">
                </div>
                <div class="form-group">
                    <label for="" style="margin-top: 10px">ФИО</label>
                    <input type="text" class="form-control" id="name" n
ame="name" value="<?php echo $user['name'];?>">
                </div>
                <div class="form-group">
                    <label for="" style="margin-top: 10px">E-
mail</label>
                    <input type="text" class="form-control" id="email"
name="email" value="<?php echo $user['email'];?>">
                </div>
```

```
                <label for="" style="margin-top: 10px">Выберите тип пол
ьзователя</label>
                <div class="form-group">
                    <select name="status_id" class="form-control" id="s
tatus_id">
                        <?php
                            $statuses = getAllStatus($db);
                            foreach ($statuses as $key => $value) {
                                echo "<option value=".
$value['status_id'].">".$value['status_name']."</option>";
                            }
                        ?>
                    </select>
                </div>
                <button type="submit" class="btn btn-
default">Сохранить</button>
            </form>
        </div>
    </body>
```

Приложение В

Презентация

Система учета пользователей ЛВС ПГУ. Графическое представление расположения

Выполнил: студент гр. 16ВВ2
Воронков Е.В.
Руководитель: к.т.н., доцент
Попов К.В.

Пенза, 2020

Постановка задачи

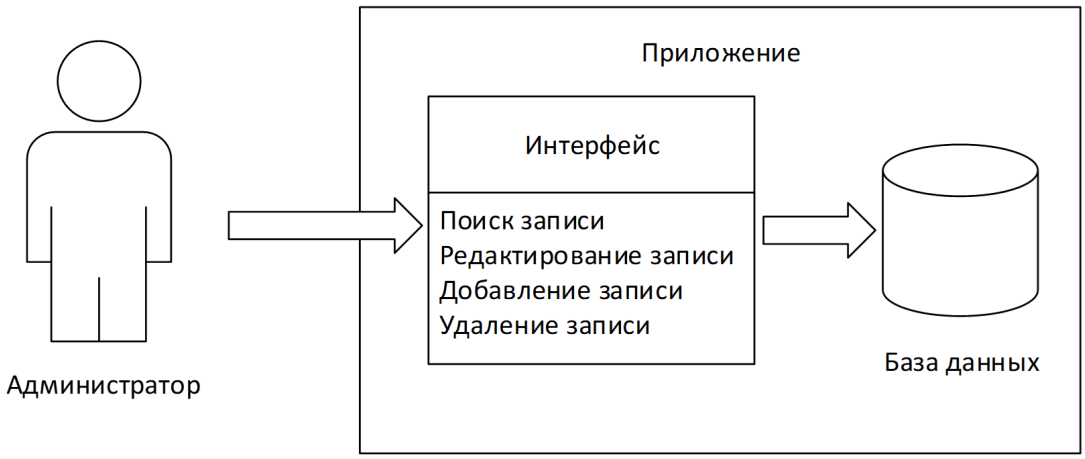
• Цель выпускной квалификационной работы:

Разработка и внедрение полученной системы учета пользователей ЛВС ПГУ, и реализация ее веб-интерфейса

• Задачи выпускной квалификационной работы:

Разработка веб-интерфейса системы учета пользователей ЛВС ПГУ

Структура взаимодействия компонентов



Выбор средств и технологий разработки

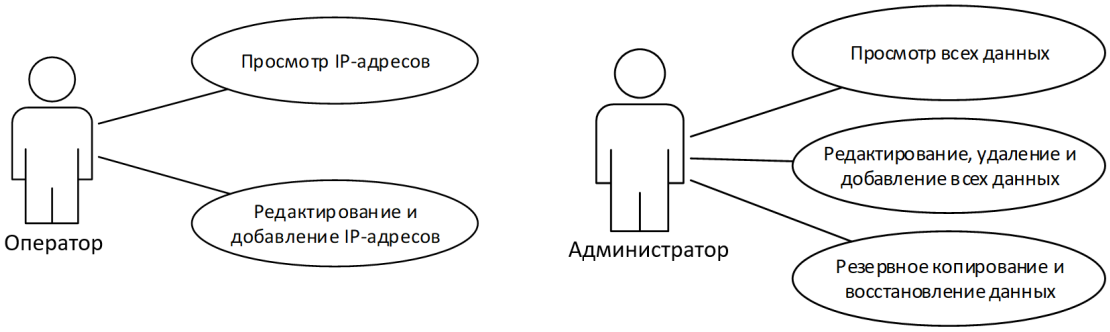
Средства разработки:

- Язык программирования PHP
- СУБД MYSQL
- Язык гипертекстовой разметки HTML
- CSS-фреймворк Bootsrap

Программные средства:

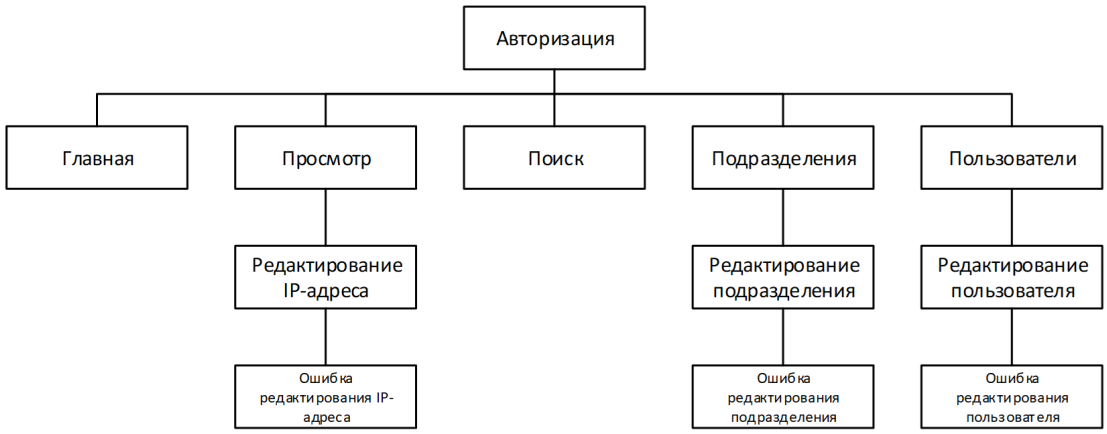
- Локальный сервер MAMP
- PhpMyAdmin

Формирование ролей



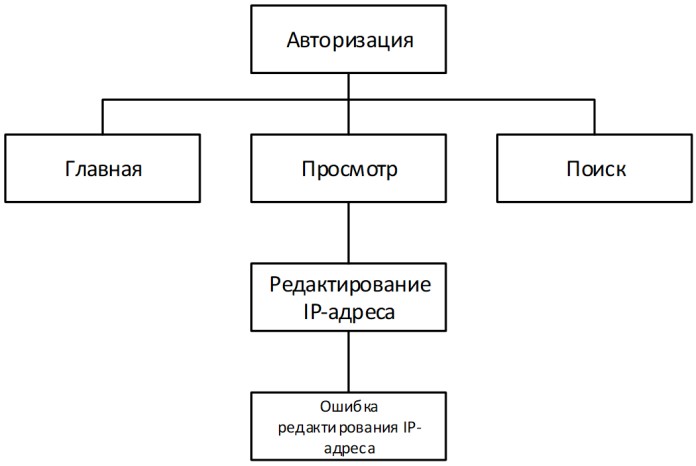
Определение структуры страниц

Иерархия страниц веб-интерфейса для администратора

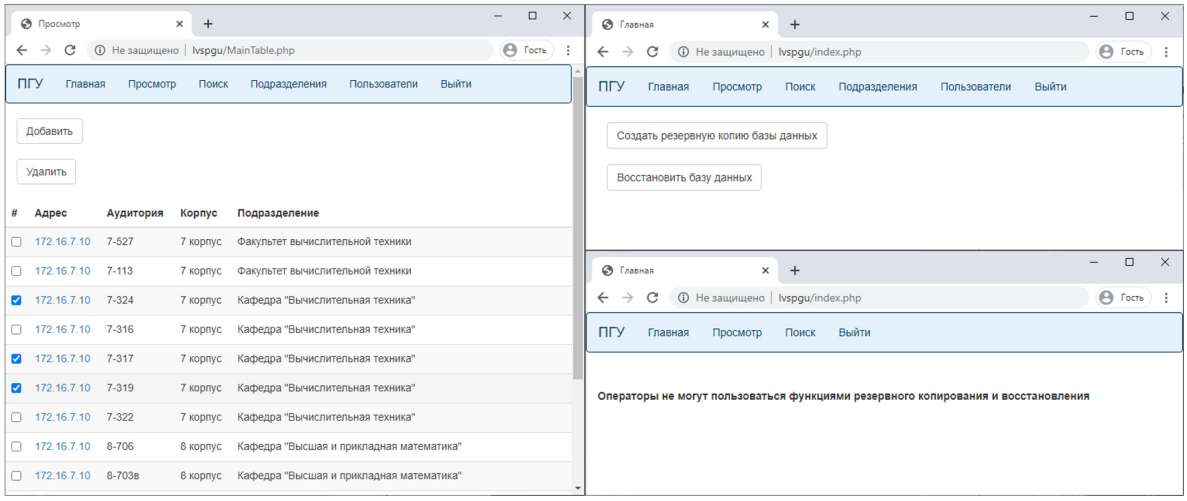


Определение структуры страниц

Иерархия страниц веб-интерфейса для оператора



Интерфейс приложения



Заключение

В результате выполнения выпускной квалификационной работы был разработан веб-интерфейс системы учета пользователей ЛВС ПГУ

Спасибо за внимание