**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 4](#_Toc11323611)

[ГЛАВА 1. АНАЛИТИЧЕСКАЯ ЧАСТЬ 5](#_Toc11323612)

[1.1. Характеристика организации 5](#_Toc11323613)

[1.2. Цель разработки 6](#_Toc11323614)

[1.3. Средства разработки 7](#_Toc11323615)

[1.4. Анализ существующих разработок и обоснование выбора технологии проектирования 9](#_Toc11323616)

[1.4.1. Программный продукт– “schedule.rea-yar.ru” 9](#_Toc11323617)

[1.4.2. Программный продукт – “ruz.spbstu.ru” 10](#_Toc11323618)

[1.4.3. Сравнение программ автоматизации работы 11](#_Toc11323619)

[1.5. Постановка задачи 12](#_Toc11323620)

[1.5.1. Входные и выходные данные предметной области 13](#_Toc11323621)

[1.5.2. Требования к проекту 14](#_Toc11323622)

[1.6. Внешняя спецификация задачи 14](#_Toc11323629)

[1.6.1. Описание задачи 14](#_Toc11323630)

[1.6.2. Методы 19](#_Toc11323631)

[ГЛАВА 2. ПРОЕКТНАЯ ЧАСТЬ 22](#_Toc11323632)

[2.1. Схема архитектуры программы 22](#_Toc11323633)

[2.2. Программное обеспечение задачи 22](#_Toc11323634)

[2.3. Структурная схема программы 25](#_Toc11323635)

[2.4. Схема данных 28](#_Toc11323636)

[2.5. Бизнес-процессы 31](#_Toc11323637)

[2.6. Схема пользовательского интерфейса 33](#_Toc11323638)

[2.7. Инструментальные средства разработки 46](#_Toc11323639)

[2.8. Характеристики программы 47](#_Toc11323640)

[ЗАКЛЮЧЕНИЕ 52](#_Toc11323641)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 54](#_Toc11323642)

[ПРИЛОЖЕНИЯ](#_Toc11323643)

[ПРИЛОЖЕНИЕ 1. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЮ 56](#_Toc11323644)

[ПРИЛОЖЕНИЕ 2. ТЕКСТ ПРОГРАММЫ 67](#_Toc11323645)

# **ВВЕДЕНИЕ**

В Ярославском филиале РЭУ имени Г.В. Плеханова в настоящий момент нет программного обеспечения (ПО), которое визуализирует занятость аудиторий при реализации учебного процесса. Процесс планирования учебных занятий происходит следующим образом:

Преподавателям нужно согласовывать свои занятия с начальником учебно-методического отдела, который в свою очередь отвечает за составлением расписания. Он оперирует лишь таблично-текстовой информацией из-за чего ему приходится рассматривать и анализировать имеющиеся данные, чтобы выбрать безошибочно нужную аудиторию, в которой можно провести занятие. Без визуального представления этот процесс более сложен, занимает больше времени и усилий.

Темой выпускной квалификационной работы является: «Разработка дистанционной системы мониторинга занятости аудитории при реализации учебного процесса в Ярославском филиале РЭУ им Г.В. Плеханова».

Тема выпускной квалификационной работы является актуальной, поскольку наглядное представление занятости аудитории позволит улучшить процесс планирования учебных занятий начальником учебно-методического отдела (УМО) или заведующим кафедрой, предоставляя наглядную информацию о занятости аудиторного фонда, и улучшить обслуживание баз данных расписания ведущим специалистом, то есть администратором.

# **ГЛАВА 1. АНАЛИТИЧЕСКАЯ ЧАСТЬ**

## **Характеристика организации**

Ярославский филиал РЭУ им. Плеханова ведет образовательную деятельность по программам высшего, среднего профессионального и дополнительного образования и занимает достойное место среди вузов г. Ярославля по подготовке высококвалифицированных специалистов в области экономики, менеджмента, маркетинга, торгового дела, информационных технологий.

Учебное заведение было основано в 1996 году как Ярославский филиал Московского государственного университета экономики, статистики и информатики (МЭСИ); 2015 (филиал был присоединен к РЭУ им. Г.В. Плеханова).

Полное наименование – Ярославский филиал федерального государственного бюджетного образовательного учреждения высшего образования «Российский экономический университет имени Г.В. Плеханова».

Краткое наименование – Ярославский филиал РЭУ им. Г.В. Плеханова.

Тип учреждения – Федеральное государственное бюджетное образовательное учреждение высшего образования.

Миссия – Удовлетворенность потребностей личности, государства, общества в доступном и качественном образовании на протяжении жизни на основе интеграции передовых образовательных технологий, научных исследований, инноваций и бизнеса в условиях информационного общества.

Основные виды деятельности – образовательная деятельность.

Расположение – Россия, г. Ярославль, ул. Большие Полянки, д. 3

Сайт – [www.rea-yar.ru](https://clck.yandex.ru/redir/nWO_r1F33ck?data=NnBZTWRhdFZKOHQxUjhzSWFYVGhXY1h2UVp3ZlZaOTJ4OHFfLXdCT05ETmVJQjJUQkJmSy1NSnJNOHJ3ZkQyRkJ3MjRlV0ppR2VlSnNMQWtNM3hrRnJZU1l0SG0wSXBudUxCMDhRM21ROG8&b64e=2&sign=d91314c2930e23648f8474b4d4618a33&keyno=17)

На рисунке 1 представлена организационная структура учебно-методического отдела в Ярославском филиале РЭУ имени Г.В. Плеханова.

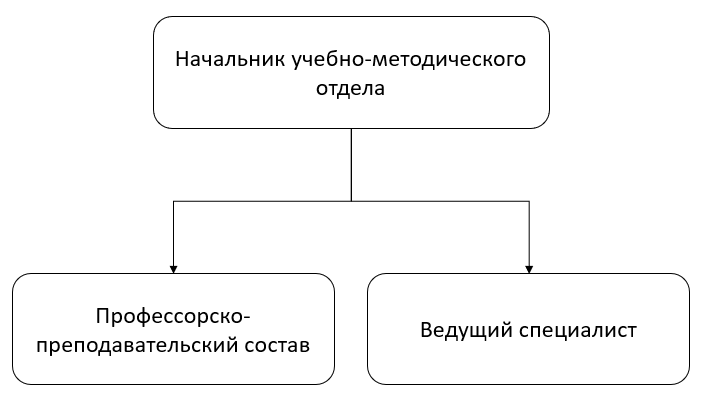


Рисунок 1 – Структура учебно-методического отдела

## **Цель разработки**

Целью разработки, выпускной квалификационной работы является создание дистанционной информационной системы, позволяющей в наглядном виде осуществлять планирование и организацию учебного процесса в соответствии с имеющимся аудиторным фондом.

## **Средства разработки**

В качестве технических средств использовался персональный компьютер со следующими характеристиками, представленными в таблице 1.

Таблица 1 – Технические средства

|  |  |
| --- | --- |
| Наименование | Характеристика |
| Ноутбук ASUS X 550 LNV | |
| Процессор | Intel Core i7 |
| Видео адаптер | Nvidia GeForce 1050Ti |
| ОЗУ | DDR4 8GB 2133 МГц, 2800 МГц |
| HDD | 1 ТБ (1128 ГБ) |
| Дисплей | 15.6 дюймов, 1920x1080, широкоформатный |

Программные средства, используемые для разработки представлены в таблице 2.

Таблица 2 – Программные средства

|  |  |
| --- | --- |
| Наименование программного средства | Характеристика |
| Windows 10 | Операционная система |
| Denwer | Локальный веб-сервер |
| Visual Studio 2013 | Интегрированная среда разработки программного обеспечения |
| Notepad++ | Текстовый редактор |
| Ramus educational 1.2.8.1 | Проектирование бизнес-процессов |
| phpMyAdmin | Приложение, представляющее собой веб-интерфейс для администрирования СУБД MySQL |

При анализе существующих операционных систем выбор пал на Microsoft Windows 10. Во-первых, потому что дипломный проект создавался на компьютере, на котором была установлена данная операционная система. Во-вторых, Windows 10 является новой и передовой. Она поддерживается своим разработчиком и часто совершенствуется, закрывая уязвимости, оптимизируя свою работоспособность и улучшая собственный функционал.

Позже встал вопрос о выборе языка программирования и программной среды разработки. Так как дистанционная система состоит из двух приложений, а именно web-приложение и exe приложение, то в качестве языка программирования были выбраны PHP и C# (Си Шарп).

* PHP - это язык программирования, специально разработанный для написания web-приложений, которые исполняются на web-сервере.
* C# - это объектно-ориентированный язык программирования для разработки приложений для платформы Microsoft .NET Framework.

Также для разработки был выбран локальный веб сервер – Denwer, так как он предоставляет удобство при удаленной работе сразу над несколькими независимыми проектами и возможность размещения их на Flash-накопителе.

Впоследствии, после выбора операционной системы и среды разработки программного обеспечения, нужно было выбрать наиболее подходящий инструментарии для первоначальной работы над базами данных. Выбор пал на phpMyAdmin из-за простоты установки и удобностью пользования. С его помощью была настроена база данных, таблицы и связи между ними.

## **Анализ существующих разработок и обоснование выбора технологии проектирования**

## **Программный продукт– “schedule.rea-yar.ru”**

Schedule.rea-yar.ru – веб-приложение, обеспечивающее доступ пользователя к актуальному расписанию Ярославского филиала РЭУ им. Г.В. Плеханова, рисунок 2.

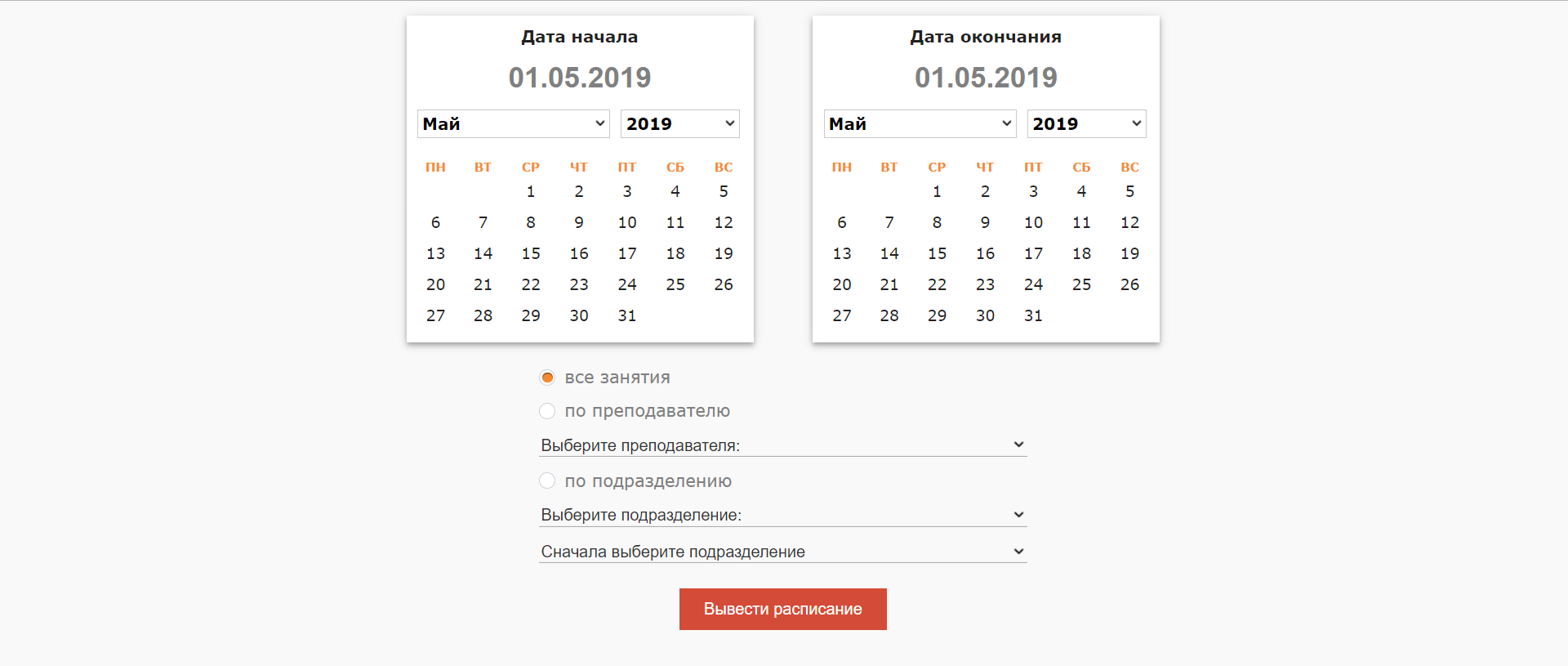


Рисунок 2 – Программный продукт “schedule.rea-yar.ru”

Schedule.rea-yar.ru представляет из себя веб-сайт, доступ к которому имеет любой пользователь. Наличие веб-браузера и интернет-соединения достаточно чтобы было возможно пользоваться сайтом, что безусловно является преимуществом. Используя данный сайт, пользователь может выбрать интересующую дату и применить три фильтра на выбор. “Все занятия”, “По преподавателю”, “По подразделению”. При применении фильтра и нажатии на кнопку “Вывести расписание”, пользователь увидит расписание. После выведения расписания пользователь может воспользоваться возможностью распечатать его или применить поиск по словам или числам.

Использование schedule.rea-yar.ru дает ряд преимуществ:

1. Позволяет быстро и доступно получить информации об актуальном расписании;

2. Минималистичность, конкретность и понятность интерфейса;

3. Возможно выбрать дату или диапазон дат, выступающие фильтром к выводимому расписанию;

4. Возможность применять фильтры к выводимому расписанию;

5. Возможность применить поиск к выведенному расписанию;

6. Возможность распечатать выведенное расписание;

## **Программный продукт – “ruz.spbstu.ru”**

Ruz.spbstu.ru является веб-приложением, обеспечивающим доступ пользователя к актуальному расписанию Политехнического университета Петра Великого, рисунок 3.

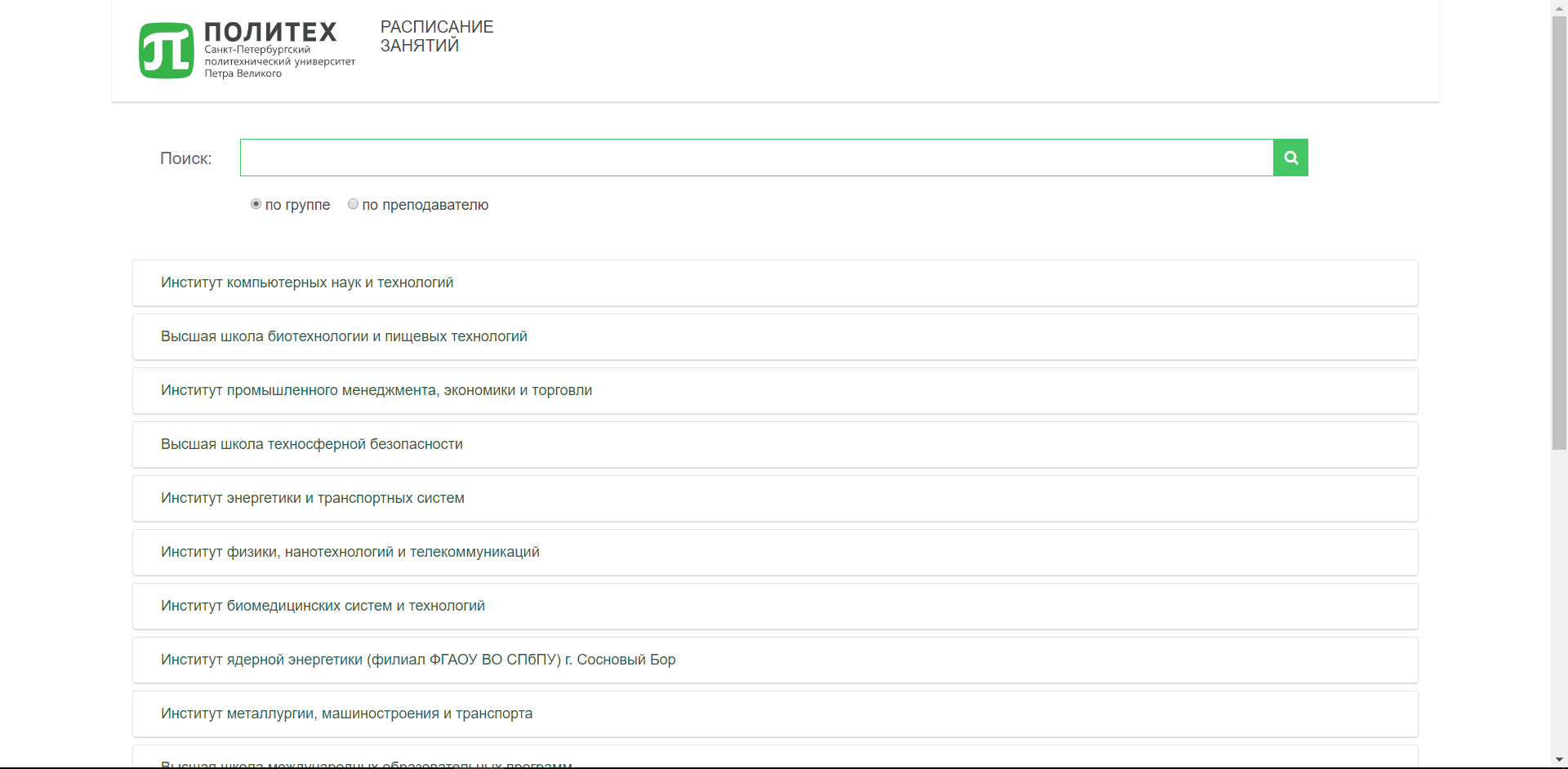


Рисунок 3 – Программный продукт “ruz.spbstu.ru”

Ruz.spbstu.ru представляет из себя веб-сайт, аналогично предыдущему анализируемому продукту, доступ к которому есть у любого пользователя.

Наличие веб-браузера и интернет-соединения достаточно чтобы было возможно пользоваться сайтом, что является преимуществом.

Используя данный сайт, пользователь может сразу воспользоваться поиском, применив соответствующие фильтры, предоставленные ниже, а именно фильтр “по группе” или “по преподавателю”. Используя поиск, пользователь сможет увидеть все найденные запросы с учетом выбранного фильтра. Если “по группе”, то покажет найденные группы, аналогичным образом работает и для фильтра “по преподавателям”. Ещё пользователь может выбрать интересующую кафедру и группу, без применения средств поиска. Выбрав интересующую группу пользователю предоставится расписание, к сожалению, без возможностей применить к расписанию интересующее время.

Преимущества ruz.spbstu.ru:

1. Легкодоступность сайта;

2. Минималистичный и конкретизированный интерфейс;

3. Возможность использовать поиск и применить к нему фильтры;

4. Возможность распечатать выведенное расписание;

5. Возможность изменить представление расписания (список или сетка)

## Сравнение программ автоматизации работы

Приведем сравнительную характеристику раннее представленных программных продуктов в таблице 3.

Таблица 3 – Сравнительный анализ существующих разработок

|  |  |  |
| --- | --- | --- |
| Критерий сравнения | Программный продукт “schedule.rea-yar.ru” | Программный продукт “ruz.spbstu.ru” |
| Наличие системы авторизации и регистрации | - | - |
| Возможность выбрать интересующую дату или диапазон дат к расписанию | + | - |
| Возможность вывести расписание с помощью фильтров “по группе” или “по преподавателю” | + | + |
| Наличие поиска | + | + |
| Возможность распечатывания выведенного расписания | + | + |
| Наглядное представление занятости аудиторий | - | - |
| Цена | Бесплатно | Бесплатно |

## **Постановка задачи**

1. Разработать веб-сайт, обладающее следующими требованиями:

* Наличие визуального отображение занятости аудитории;
* Отображение расписания в зависимости от выбранных фильтров
* Интуитивно понятный и минималистичный интерфейс
* Механизм аутентификации пользователей;
* Обеспечение информационной безопасности при работе в веб приложении;

1. Разработать программу, обладающая следующими требованиями:

* Информационная безопасность при работе с программой
* Наличие механизма аутентификации пользователей
* Создание, редактирование и удаление записей в базе данных
* Наличие визуального редактора для таблицы данных “Аудитории”
* Сортировка информации
* Сохранение базы данных для возможности последующего восстановления

Основные функции:

* Просмотр занятости аудитории с учетом расписания
* Просмотр актуального расписания;
* Возможности администрирования базы данных;

## **Входные и выходные данные предметной области**

Входными данными предметной области будут служить данные, которые пользователь может выбрать:

* Выбор даты начала и окончания;
* ФИО преподавателя (руководителя);
* Наименование группы;
* Подразделение;
* Тип аудитории;

Выходными данными предметной области будут служить данные, которые пользователь получает при работе с рабочей областью:

* Отфильтрованное расписание
* Визуальное отображение занятости аудитории в зависимости от расписания и группы пользователя, если он имеет учетную запись на сайте

## **Требования к проекту**

Разработать веб-приложение и программу, которые должны строго выполнять свои роли. Веб-приложение отвечает за отображение расписания, занятости аудитории в зависимости от того, вошёл ли пользователь в существующий профиль и какими правами он обладает. Программа отвечает за управление базой данных. В ней должны быть возможности делать резервные копии и визуально редактировать данные аудиторного фонда.



## **Внешняя спецификация задачи**

## **Описание задачи**

Дистанционная система поделена на две программы. Одна представляет собой веб-сайт, который выступает в качестве пользовательского функционала, и исполняемую exe программу, которая выступает в качестве администраторского функционала.

Сайт должен выводить расписание в зависимости от выбранных фильтров, а также обеспечивать возможность наглядно увидеть занятые и свободные аудитории, если пользователь авторизуется на сайте.

Программа должна предоставлять возможности в зависимости от прав учетной записи администратора.

Ведущий специалист, в лице главного администратор имеет все права, а именно управлять базой данных, редактировать таблично или визуально данные аудиторного фонда, регистрировать пользователей сайта и назначать им группу, делать и использовать резервные копии всей базы данных.

Заведующий кафедрой имеет ограниченные администраторские права – в его распоряжении лишь администрирования таблицей данных о расписании и возможность сделать или использовать резервную копию лишь для таблицы расписания. На рисунках 4-9 изображены схема возможности каждой роли на веб-сайте и в программе.

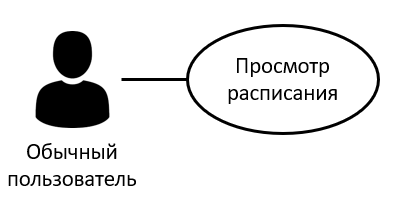


Рисунок 4 – Схема возможностей обычного пользователя сайта

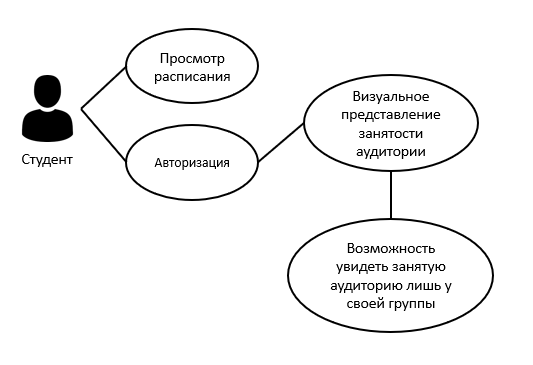


Рисунок 5 – Схема возможностей студента на сайте



Рисунок 6 – Схема возможностей преподавателя на сайте



Рисунок 7 – Схема возможностей главного администратора (ведущего специалиста) на сайте

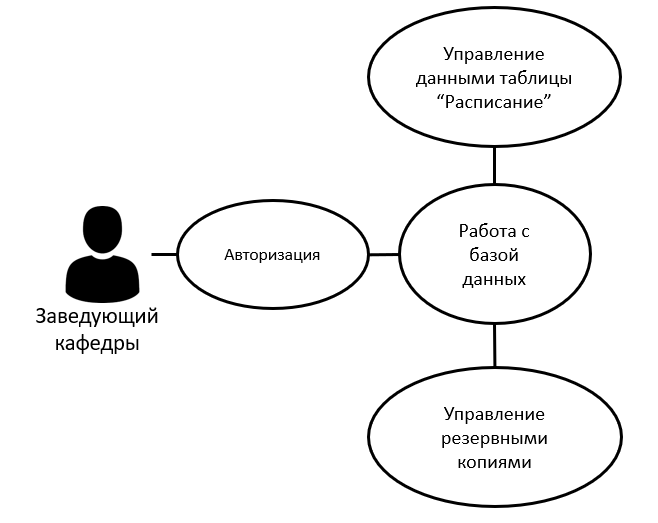


Рисунок 8 – Схема возможностей заведующего кафедры в программе

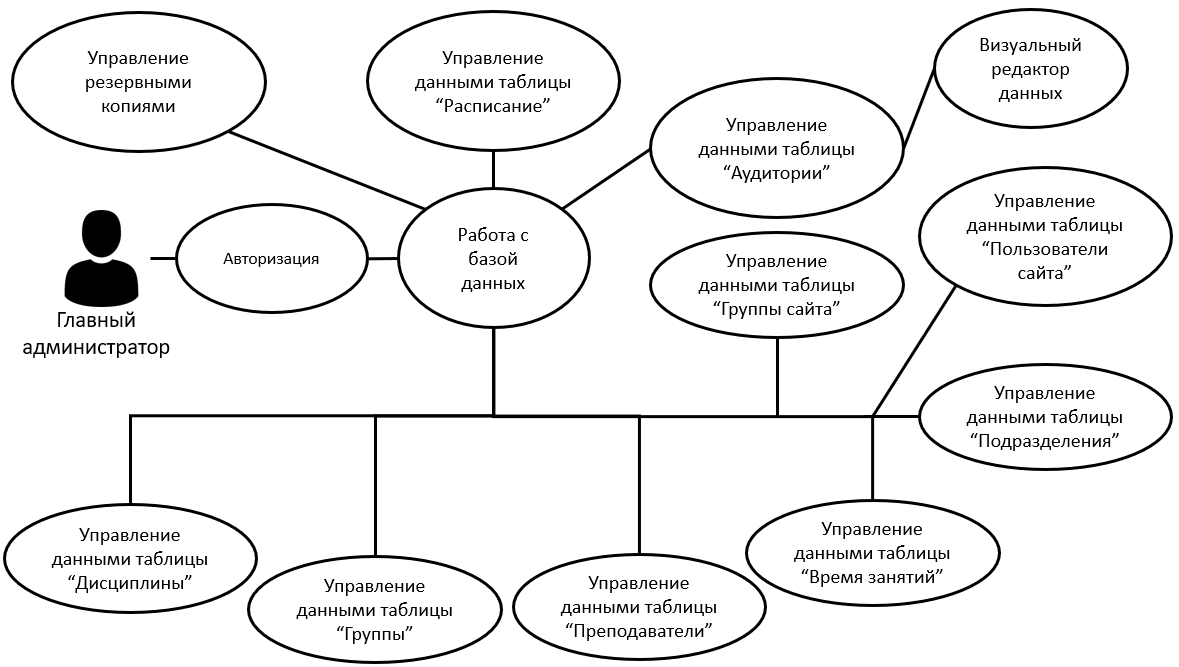


Рисунок 9 – Схема возможностей главного администратора (ведущего специалиста) в программе

## **Методы**

В данном веб-приложении при разработке применялись следующие методы:

* Архитектура

При разработке информационной системы использовалась централизованная архитектура базы данных, такая как клиент – сервер. База данных находится на сервере, то есть в сети, а веб-приложение и программа для управления базой данных располагается у клиента на компьютере.

* Объектно-ориентированное программирование

Используется функция или процедура, принадлежащая определенному классу или объекту. Как и процедура в процедурном программировании, метод состоит из некоторого количества операторов для выполнения действия и имеет набор входных аргументов.

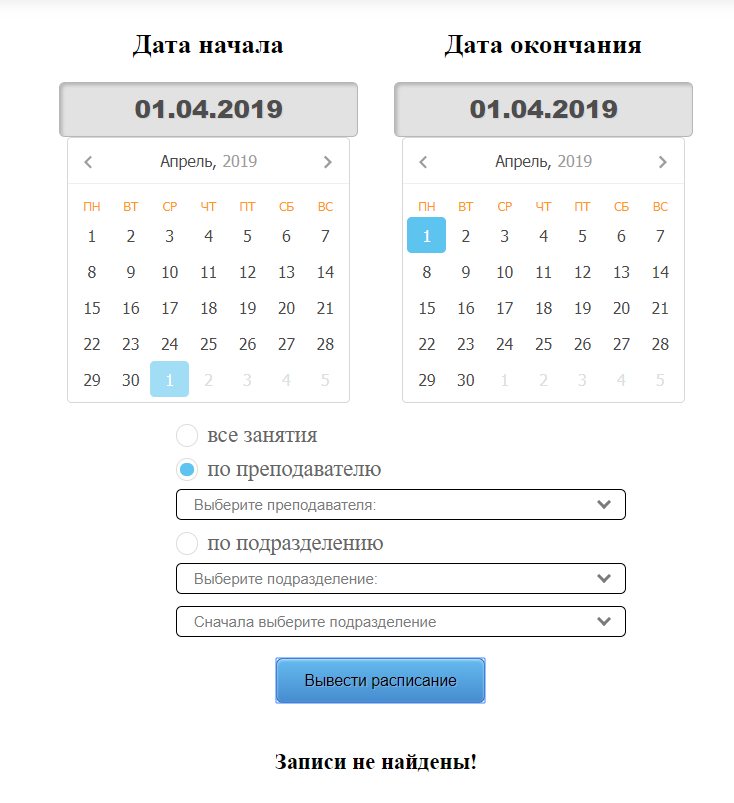


Рисунок 10 – Записи не найдены

Если нужно увидеть занятость аудитории, то пользователю нужно войти в учетную запись. При вводе неверных данных пользователь получит сообщение об ошибке, рисунок 11.

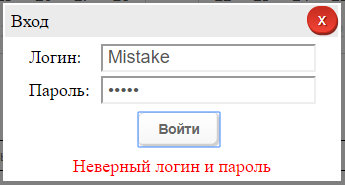


Рисунок 11 – Ошибка

При работе с программой если пользователь введет неверные данные, а именно адрес сервера, логин и пароль, то пользователь также натолкнется на ошибку входа, рисунок 12

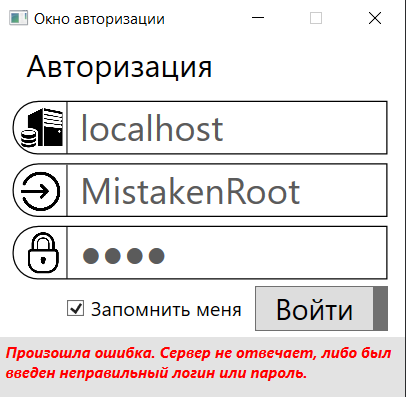


Рисунок 12 – Ошибка 2

# **ГЛАВА 2. ПРОЕКТНАЯ ЧАСТЬ**

## **Схема архитектуры программы**

Для разработки веб-приложения и программы была использована архитектура клиент-сервер, которая представлена на рисунке 13

Данная архитектура основана на двухзвенной архитектуре. Сервер отвечает на клиентские запросы напрямую и в полном объеме, при этом используя только собственные ресурсы. То есть сервер не вызывает сторонние сетевые приложения и не обращается к сторонним ресурсам для выполнения какой – либо части запроса.

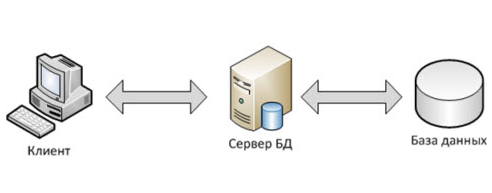


Рисунок 13 – Клиент сервер

## **Программное обеспечение задачи**

На рисунках 14-18 изображены функциональные схемы веб-приложения и программы, поясняющие распределение определённых процессов по ролям, происходящие в приложении и где показано какие действия может выполнять каждая роль.

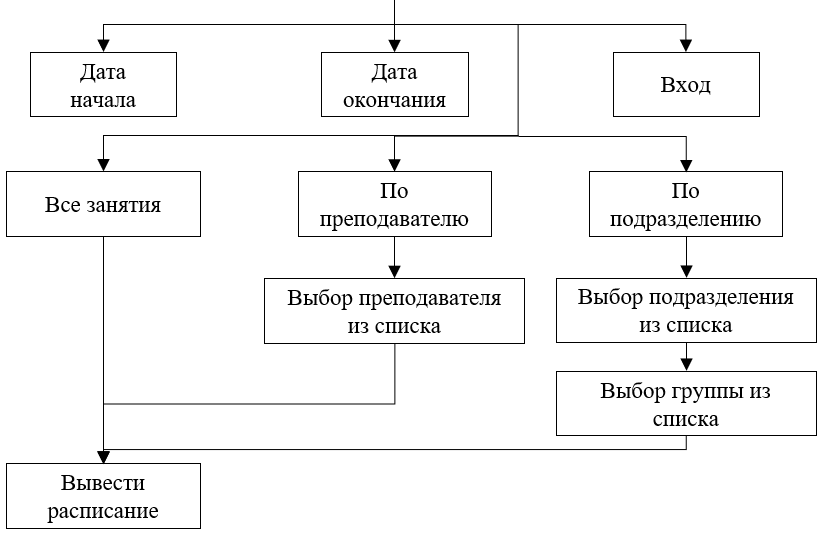


Рисунок 14 – Функциональная схема для веб-сайта, роль – любой пользователь

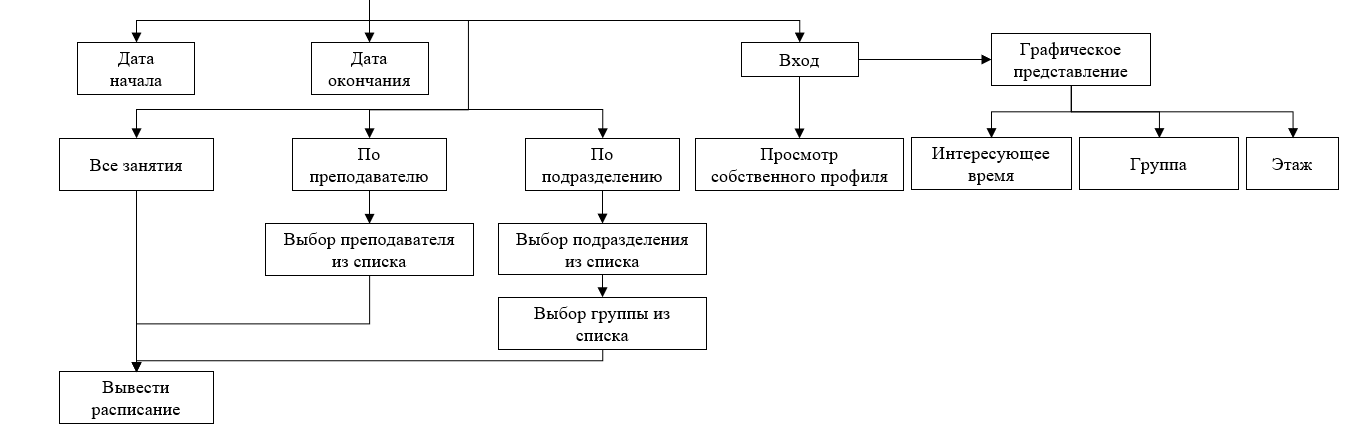


Рисунок 15 – Функциональная схема для веб-сайта, роль – студент

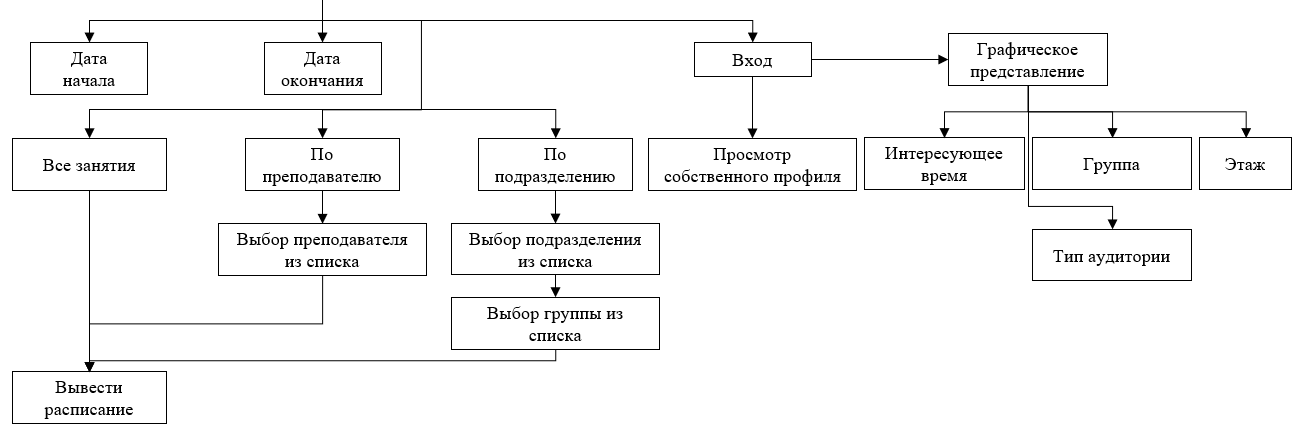


Рисунок 16 – Функциональная схема для веб-сайта, роль – преподаватель и администратор

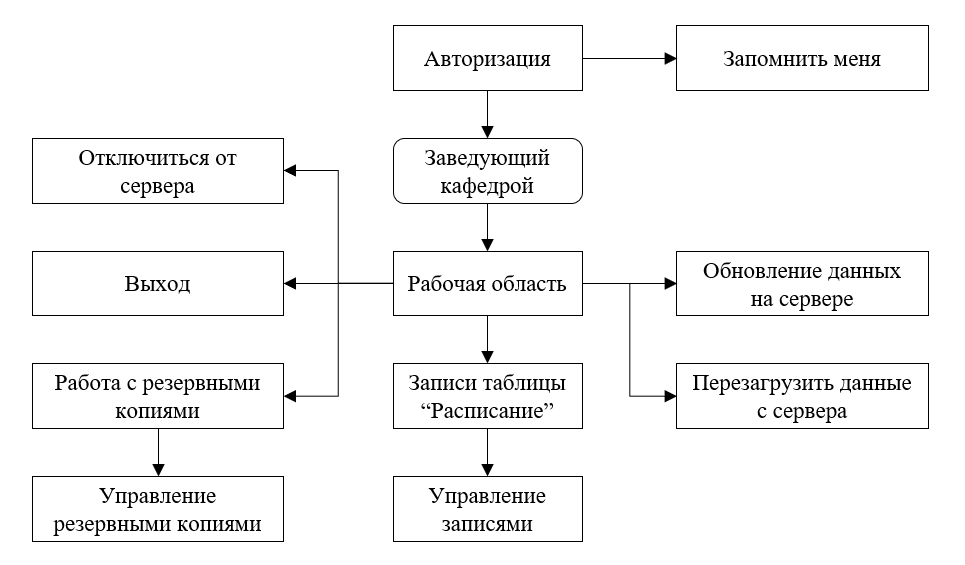


Рисунок 17 – Функциональная схема для программы, роль – заведующий кафедрой

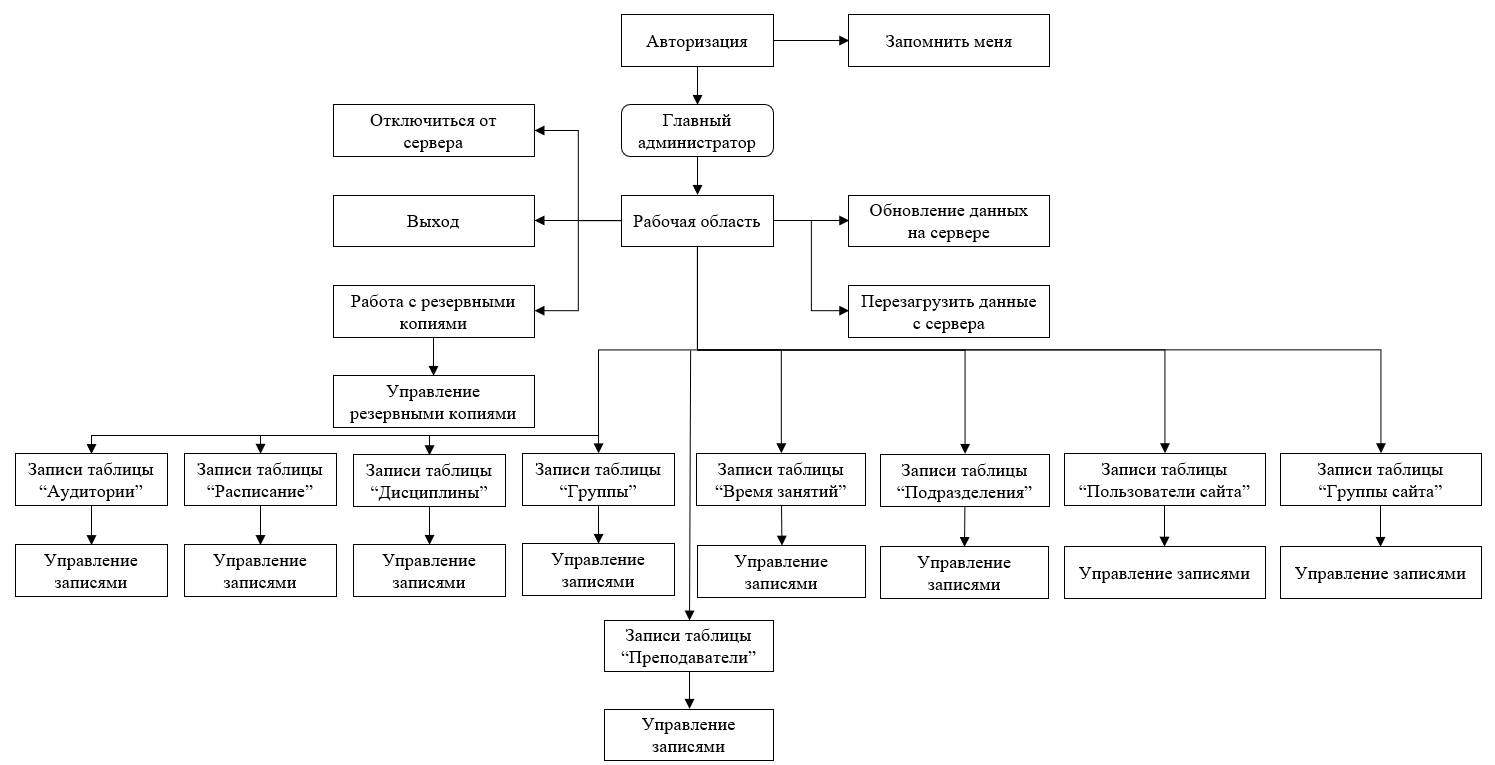


Рисунок 18 – Функциональная схема для программы, роль – главный администратор

## **Структурная схема программы**

В данном проекте использовалась схема разделения данных приложения Model-View-Controller (MVC, «Модель-Вид-Контроллер»).

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

Рассмотрим концептуальную схему шаблона MVC на рисунке 19.

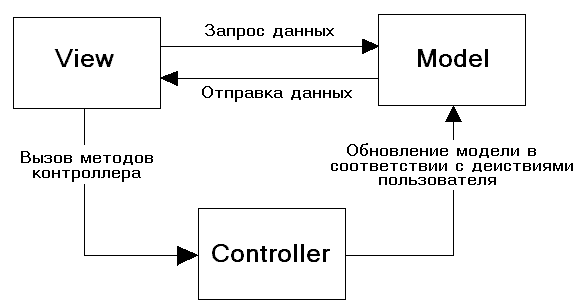


Рисунок 19 – Клиент сервер

В архитектуре MVC модель предоставляет данные и правила бизнес-логики, представление отвечает за пользовательский интерфейс, а контроллер обеспечивает взаимодействие между моделью и представлением.  
Типичную последовательность работы MVC-приложения можно описать следующим образом:

1. При заходе пользователя на веб-ресурс, скрипт инициализации создает экземпляр приложения и запускает его на выполнение.  
   При этом отображается вид, скажем главной страницы сайта.
2. Приложение получает запрос от пользователя и определяет запрошенные контроллер и действие. В случае главной страницы выполняется действие по умолчанию (*index*).
3. Приложение создает экземпляр контроллера и запускает метод действия,  
   в котором, к примеру, содержаться вызовы модели, считывающие информацию из базы данных.
4. После этого действие формирует представление с данными, полученными из модели, и выводит результат пользователю.

На рисунке 20 предоставлена структурная схема веб-приложения, показывающая совокупность звеньев и связей между ними, которая представлена в таблице 4.

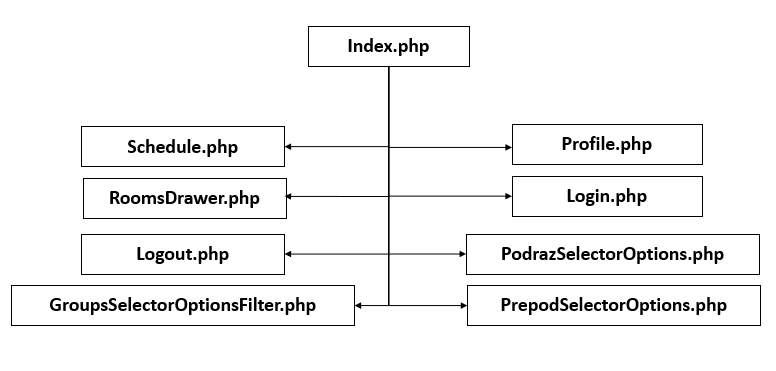


Рисунок 20 – Структурная схема

Таблица 4 – Описание файлов

|  |  |  |
| --- | --- | --- |
| № | Название | Описание |
| 1 | index.php | точка в хода в приложение и главная страница |
| 2 | Schedule.php | Вывод расписания |
| 3 | Profile.php | Представление пользовательского профиля |
| 4 | RoomsDrawer.php | Формирование картинки занятости аудитории |
| 5 | Login.php | Используется для авторизации пользователя |
| 6 | Logout.php | Используется для выхода из учетной записи |
| 7 | PodrazSelectorOptions | Генерирует список подразделении из базы данных для спискового элемента |
| 8 | GroupsSelectorOptionsFilter | Генерирует список групп из базы данных для спискового элемента |
| 9 | PrepodSelectorOptions | Генерирует список преподаватель из базы данных для спискового элемента |

## **2.4. Схема данных**

Инфологическая модель отображает логические взаимосвязи между элементами данных, их содержанием и физической организацией. Логическая модель разрабатывается с учетом спецификации конкретной предметной области. Эта модель представляет данные, подлежащие хранению в базе данных (Рисунок 21).

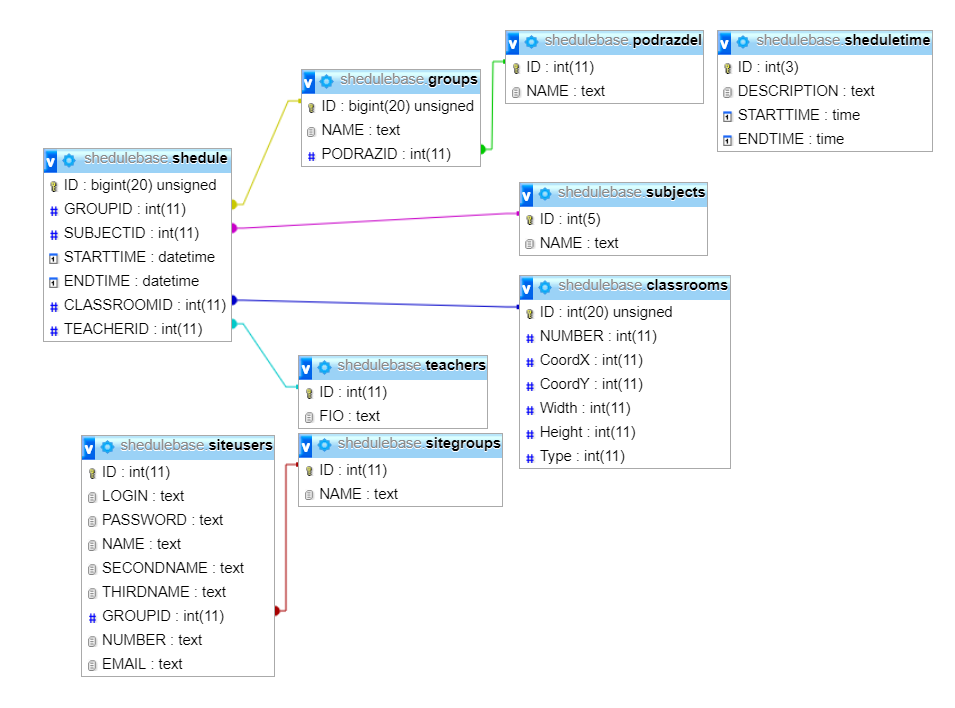


Рисунок 21 – Инфологическая модель

Таблица 5 – Назначение объектов базы данных

|  |  |
| --- | --- |
| Описание объекта | Значение объекта |
| Таблица «shedule» | |
| ID | ID занятия |
| GROUPID | ID группы |
| SUBJECTID | ID дисциплины |
| STARTTIME | Дата и время начала занятия |
| ENDTIME | Дата и время окончания занятия |
| CLASSROOMID | ID аудитории |
| TEACHERID | ID преподавателя |
| Таблица «groups» | |
| ID | ID группы |
| NAME | Наименование группы |
| PODRAZID | ID подразделения |
| Таблица «podrazdel» | |
| ID | ID подразделения |
| NAME | Наименование подразделения |
| Таблица «subjects» | |
| ID | ID дисциплины |
| NAME | Наименование дисциплины |
| Таблица «classrooms» | |
| ID | ID аудитории |
| NUMBER | Номер аудитории |
| CoordX | Начальная координата X аудитории |
| CoordY | Начальная координата Y аудитории |
| Width | Ширина аудитории |
| Height | Высота аудитории |
| Type | Разновидность аудитории |
| Таблица «teachers» | |
| ID | ID преподавателя |
| FIO | Фамилия Имя Отчество преподавателя |
| Таблица «siteusers» | |
| ID | ID пользователя сайта |
| LOGIN | Логин пользователя сайта |
| PASSWORD | Пароль пользователя сайта |
| NAME | Имя пользователя сайта |
| SECONDNAME | Фамилия пользователя сайта |
| THIRDNAME | Отчество пользователя сайта |
| GROUPID | ID группы пользователя сайта |
| NUMBER | Номер телефона пользователя сайта |
| EMAIL | E-mail пользователя сайта |
| Таблица «sitegroups» | |
| ID | ID группы пользователя сайта |
| NAME | Наименование группы пользователя сайта |
| Таблица «sheduletime» | |
| ID | ID времени занятия |
| DESCRIPTION | Описание времени занятия |
| STARTTIME | Время начала занятия |
| ENDTIME | Время окончания занятия |

## 2.5. Бизнес-процессы

На сегодняшний день в ЯФ РЭУ им. Г.В. Плеханова отсутствует дистанционная система наглядного представления занятости аудитории и потому начальнику учебно-методического отдела или заведующему кафедры приходится вести внимательный учет по занятости аудиторного фонда, учитывать всё по имеющемуся расписанию, как собственно и преподавателям, которые желали бы провести занятие, к примеру, в другой аудитории, в которой есть проектор.

Рассмотрим модель «Как есть» (AS-IS), то есть модель существующей на кафедре информатики и математики. Основные бизнес-процессы работы отдела по организации учебного процесса:

1. Составление расписания учебных занятий, зачетов и экзаменов. Заполнение важных элементов для составления расписания;
2. Контроль занятости аудиторного фонда;
3. Контроль соответствия нормативным документам;
4. Консультирование преподавателей
5. Внесение изменений.

На рисунках 22-23 отображены схемы описания процессов в предметной области.

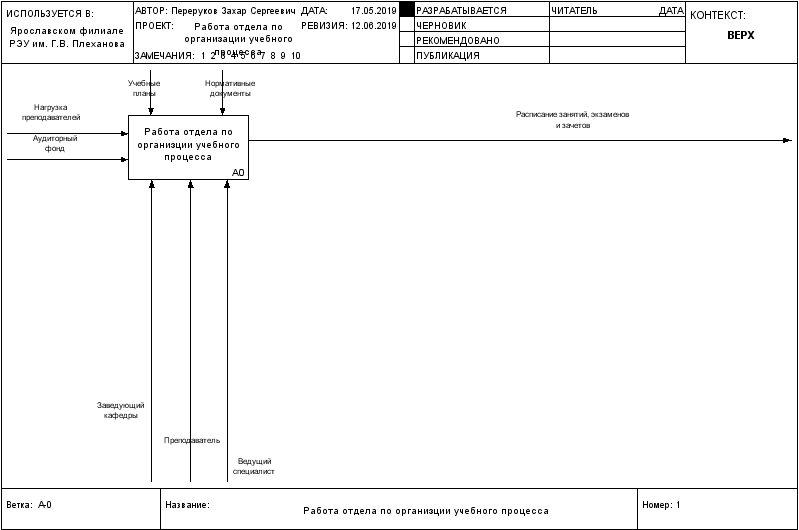


Рисунок 22 – контекстная диаграмма A-0 «Работа отдела по организации учебного процесса “как есть”»

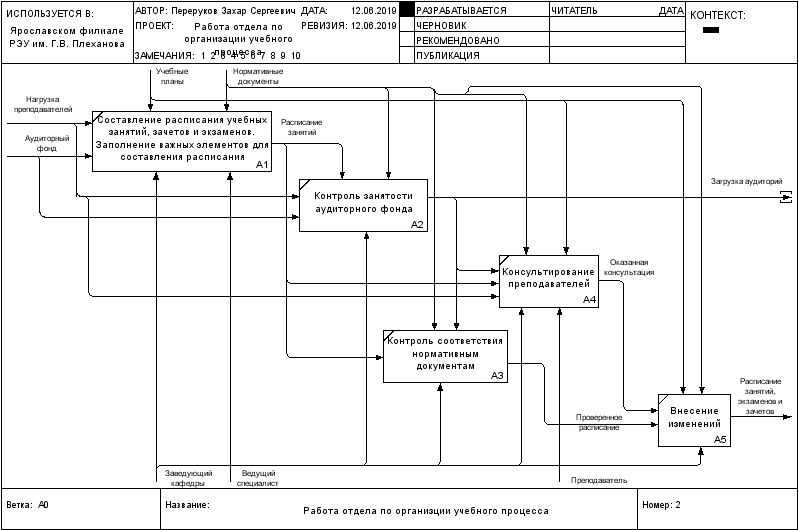


Рисунок 23 – Диаграмма А0 – «Работа отдела по организации учебного процесса»

## 2.6. Схема пользовательского интерфейса

Схема пользовательского интерфейса нужна для наглядного поднимая компонентов в веб-приложении и программе, и их назначения. На рисунке 24 изображена схема главной страницы сайта. При попадании на главную страницу пользователь может авторизоваться, чтобы включить возможности визуального отображения аудитории и просмотреть свой профиль. Для этого необходимо найти текстовую кнопку “Вход” в элементе 1. При нажатии на кнопку окно главной страницы немного затемняется и перед пользователем появляется окно входа, схема которого изображена на рисунке 25. В этом окне пользователь вводит логин и пароль от учетной записи (в полях 2 и 3, рисунок 25). Если пользователю хочется закрыть окно и отказаться от авторизации, то он может нажать на красную кнопку с крестиком (элемент 1, рисунок 25). При нажатии на кнопку “Вход”, отмеченная как элемент 4, рисунок 25, происходит процесс на сервере, в котором идет проверка логина и пароля в базе данных на их правильность и достоверность. Если логин и пароль совпадает с значениями в базе данных, то происходит переадресация на главную страницу. Если логин или пароль оказались введены неверно, то в самой нижней частью окошка входа пользователя предупредят об ошибке красным текстом “Неверный логин или пароль”.

После авторизации в элементе 1, рисунок 24, происходят изменения и теперь можно будет включить отображение графической карты в элементе 5 с помощью ползунка, посмотреть собственный профиль, нажав на имя вашего профиля и выйти из учетной записи, нажав на текстовую кнопку “Выход”.

Если пользователь нажмет на текстовую кнопку с именем профиля, то откроется уменьшенное окошко, в котором указана информация о профиле. Схема окна профиля пользователя изображена на рисунке 26. В данном окне, в элементе 1, располагается текстовая подпись к окну профиля. В элементе 2 располагается иконка профиля пользователя, а в элементе 3 находится информация о профиле (Имя, фамилия, группа, телефон и т.д.).

Также пользователь может, без надобности в авторизации, просмотреть расписание. Для того, чтобы начать просмотр расписания, пользователю необходимо

1. Указать дату начала и окончания занятий в отображенных календарях в области 2, рисунок 24
2. Выбрать среди трёх вариантов-фильтров нужный. Они располагаются в области 3, рисунок 24

Указав все нужные фильтры, пользователю нужно нажать на кнопку “Вывести расписание”, которая указана как элемент 4.

Если занятия присутствуют в указанный промежуток, а в фильтрах “по преподавателю” или “по подразделению” были выбраны соответственно преподаватель или группа, то расписание выводится ниже, в элементе 5. Выведенное расписание представляет собой таблицу, которую можно отсортировать по любому нужному столбцу и в которой присутствует возможность поиска по таблице, наряду с кнопками навигации по таблице.

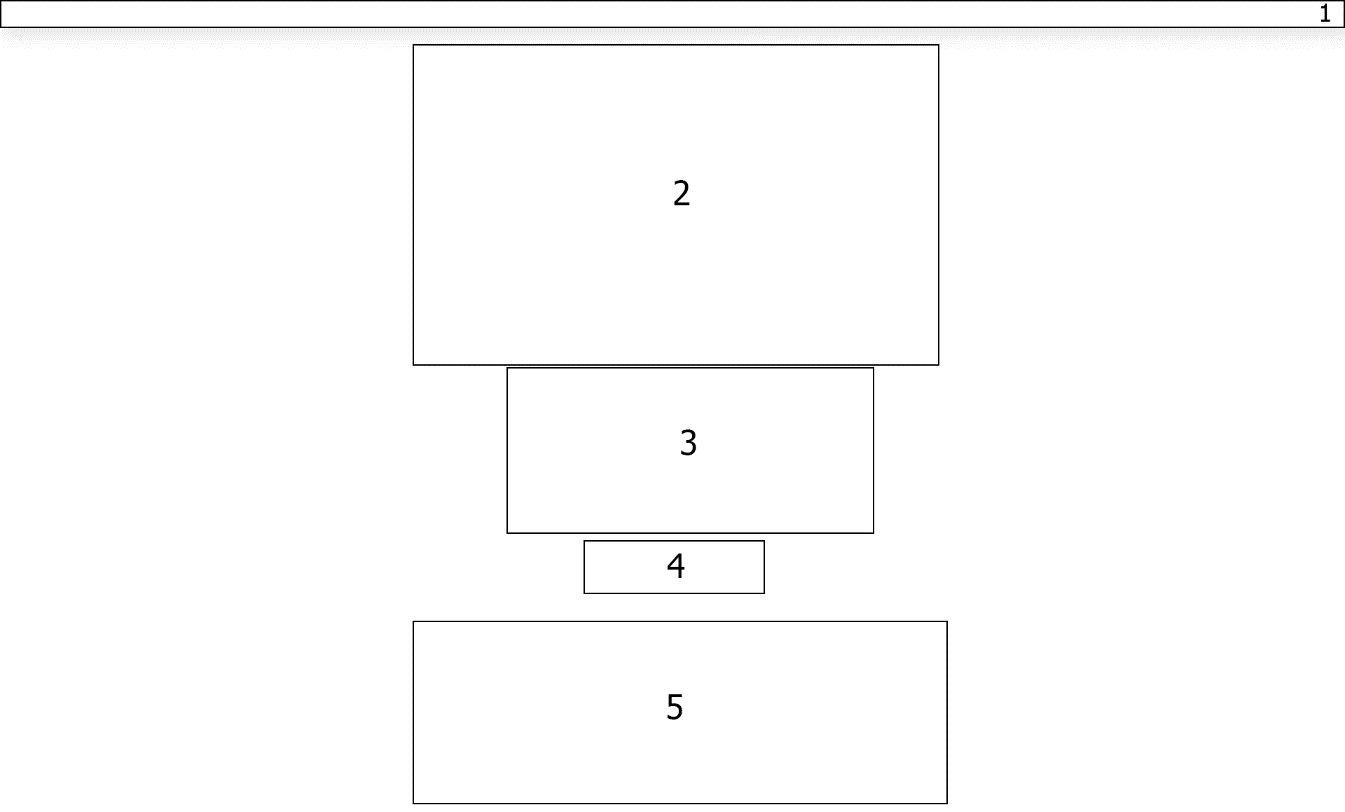


Рисунок 24 – Окно главной страницы

Таблица 6 – Интерфейс страницы «Главная страница»

|  |  |
| --- | --- |
| № | Описание элемента |
| 1 | Меню сайта, в котором изначально располагается кнопка входа в учетную запись. При успешном входе на данном меню появляются кнопки для включения графического отображения занятости аудитории, кнопка входа в собственный профиль и выход из учетной записи |
| 2 | Область, в которой располагается два календаря, в котором можно выбрать диапазон даты начала и окончания занятий |
| 3 | Область, в которой располагаются три фильтра на выбор:   1. Все занятия 2. По преподавателю. Ниже расположен список в котором можно выбрать интересующего преподавателя 3. По подразделению. Ниже расположено два списка в которых можно выбрать интересующее подразделение и группу, в зависимости от выбора в списке подразделений |
| 4 | Кнопка для вывода расписания с учетом выбранного диапазона дат и фильтров |
| 5 | Область, в которой выводится графическая карта занятости аудитории и само расписание в виде таблицы. Также в таблице доступно текстовое поле поиска и кнопки навигации по ней |

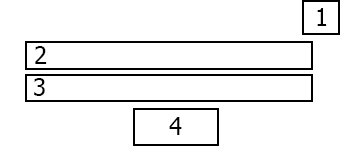


Рисунок 25 – Окно входа

Таблица 7 – Интерфейс окна «Вход»

|  |  |
| --- | --- |
| № | Описание элемента |
| 1 | Кнопка для закрытия окошка входа |
| 2 | Поле для ввода логина пользователя |
| 3 | Поле для ввода пароля пользователя |
| 4 | Кнопка для запуска процесса авторизации пользователя по введенному логину и паролю |

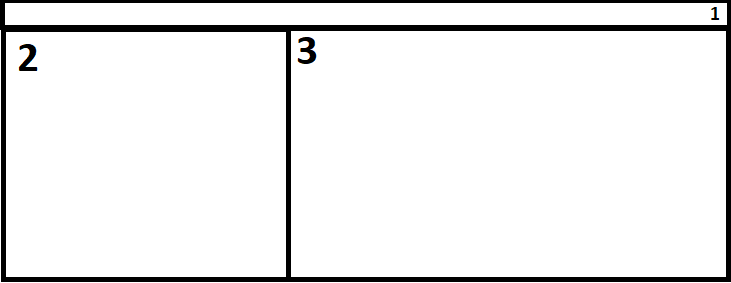


Рисунок 26 – Окно профиля пользователя

Таблица 8 – Интерфейс окна «Профиль пользователя»

|  |  |
| --- | --- |
| № | Описание элемента |
| 1 | Подпись к окошку формата “ Информация о профиле <Имя профидя> [ID: <Идентификационный номер пользователя>]” |
| 2 | Иконка профиля пользователя |
| 3 | Область, в которой расположена информация о пользователе. |

Теперь рассмотрим интерфейс программы. На рисунке 27 изображена схема окна авторизации пользователя в программе. Оно появляется при её запуске.

Здесь пользователь вводит в поле 1 адрес сервера, в поле 2 и 3 – логин и пароль от учетной записи администратора или заведующего кафедры. После ввода данных в поля 1,2,3 пользователь может поставить галочку для запоминания данных перед следующим входом в программу в элементе 4. При нажатии на кнопку “Войти”, расположенную в поле 5, в окне размывается интерфейс, поверх которого прорисовывается анимация загрузки. В этот момент происходит сверка логина и пароля, на сервере, адрес которого был указан в поле 1. Если адрес сервера был введен неверно или вовсе не отвечает, а также введенный логин или пароль был введен некорректно, то интерфейс возвращается в норму, загрузка исчезает, а в нижней части окна пользователь получает сообщение об ошибке:

“Произошла ошибка. Сервер не отвечает, либо был введен неправильный логин или пароль.”

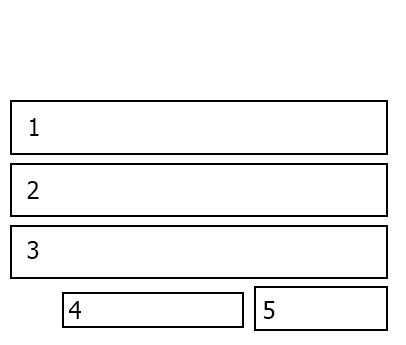


Рисунок 27 – Окно авторизации пользователя в программе

Таблица 9 – Интерфейс окна «Авторизация»

|  |  |
| --- | --- |
| № | Описание элемента |
| 1 | Поле для ввода адреса сервера |
| 2 | Поле для ввода логина |
| 3 | Поле для ввода пароля |
| 4 | Элемент, отвечающий за запоминания вводимых данных |
| 5 | Кнопка для запуска процесса авторизации |

В случае, если адрес сервера, логин и пароль заведующего кафедрой или главного администратора оказались корректными, то окно авторизации закрывается и перед глазами возникает главное окно программы, схемы которого изображены на рисунке 28-29.

Сначала рассмотрим интерфейс программы от лица заведующего кафедры (рисунок 28). При появлении главного окна, заведующий кафедры видит перед собой меню (элементы 1,2,3), записи расписания из базы данных (БД) в формате таблицы (элемент 4) и области для работы с выбранной в таблице записью (область 5 и 6). В области 5 располагаются поля или списки для редактирования отдельных атрибутов выбранной записи, а в области 6 расположены две кнопки – назначить занятие и удалить занятие.

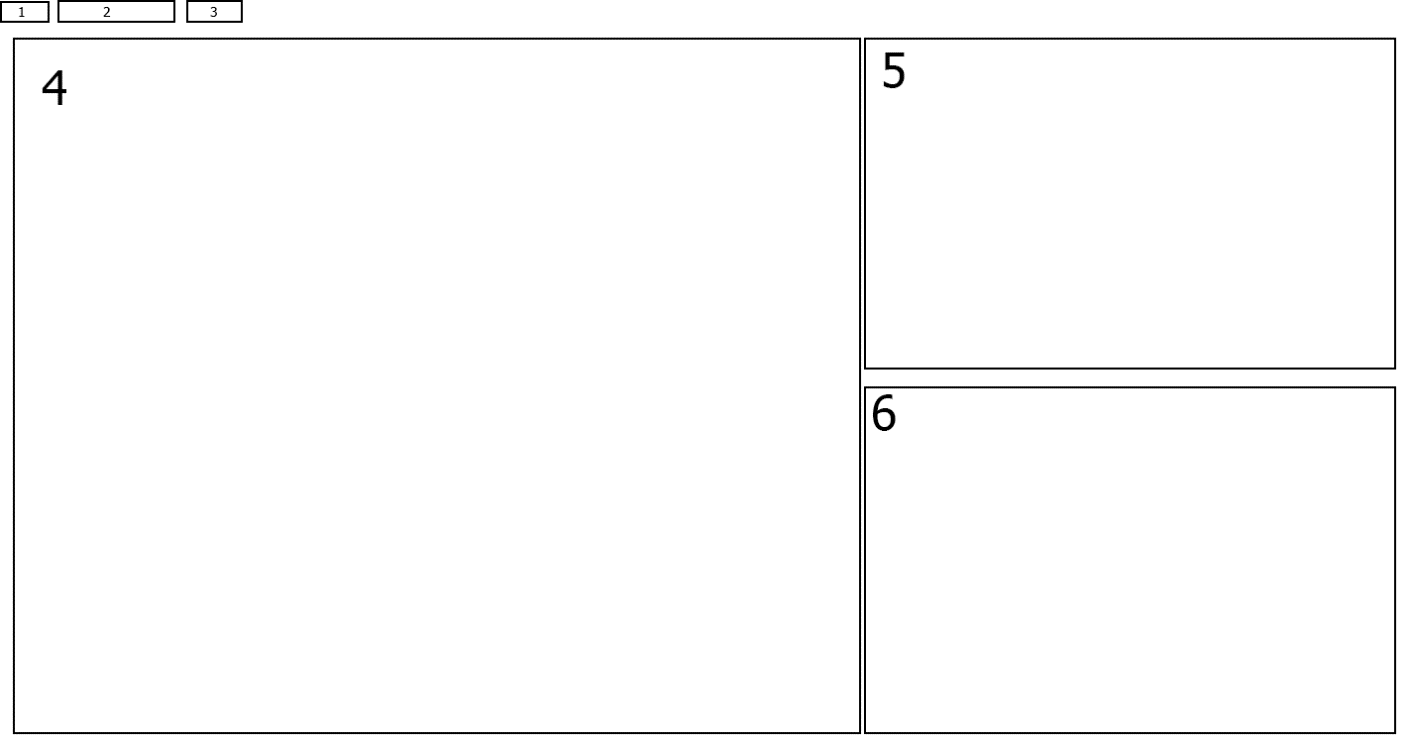


Рисунок 28 – Главное окно программы от лица заведующего кафедрой

Таблица 10 – Интерфейс главного окна от лица заведующего кафедрой

|  |  |
| --- | --- |
| № | Описание элемента |
| 1 | Элемент меню “Файл”. В нем можно открыть окно для работы с резервными копиями, настроить приложение, отключиться от сервера и выйти из программы |
| 2 | Элемент меню “Редактирование”. Содержит множественные операции редактирования: от копирования, до отмены и повторения действий |
| 3 | Элемент меню “Сервер”. В нем можно отослать и применить все изменения к БД на сервере, а также перезагрузить данные внутри программы, сделав запрос на повторное получение данных |
| 4 | Элемент, представляющий собой таблицу, в которой можно видеть и работать с записями таблицы “Расписание” |
| 5 | Область, в которой располагаются параметры редактирования записи |
| 6 | Область, в которой располагаются кнопки управления записями, а именно создание и удаление |

Теперь рассмотрим интерфейс программы от лица главного администратора (рисунок 29). При появлении главного окна, администратор видит перед собой меню (элементы 1,2,3), вкладки, названия которых отражают определенную таблицу в БД и при нажатий на них изменяется элемент 5.

В области 6 располагаются поля или списки для редактирования отдельных атрибутов выбранной записи, а в области 7 расположены две кнопки: одна отвечает за создание записи, вторая – за удаление записи. Если выбрана вкладка “Аудитории”, то в области 7 появляется третья кнопка, а именно кнопка позволяющая переключиться на визуальный редактор записей, что позволяет наглядно редактировать атрибуты координат расположения и размеров аудитории, что крайне удобно.

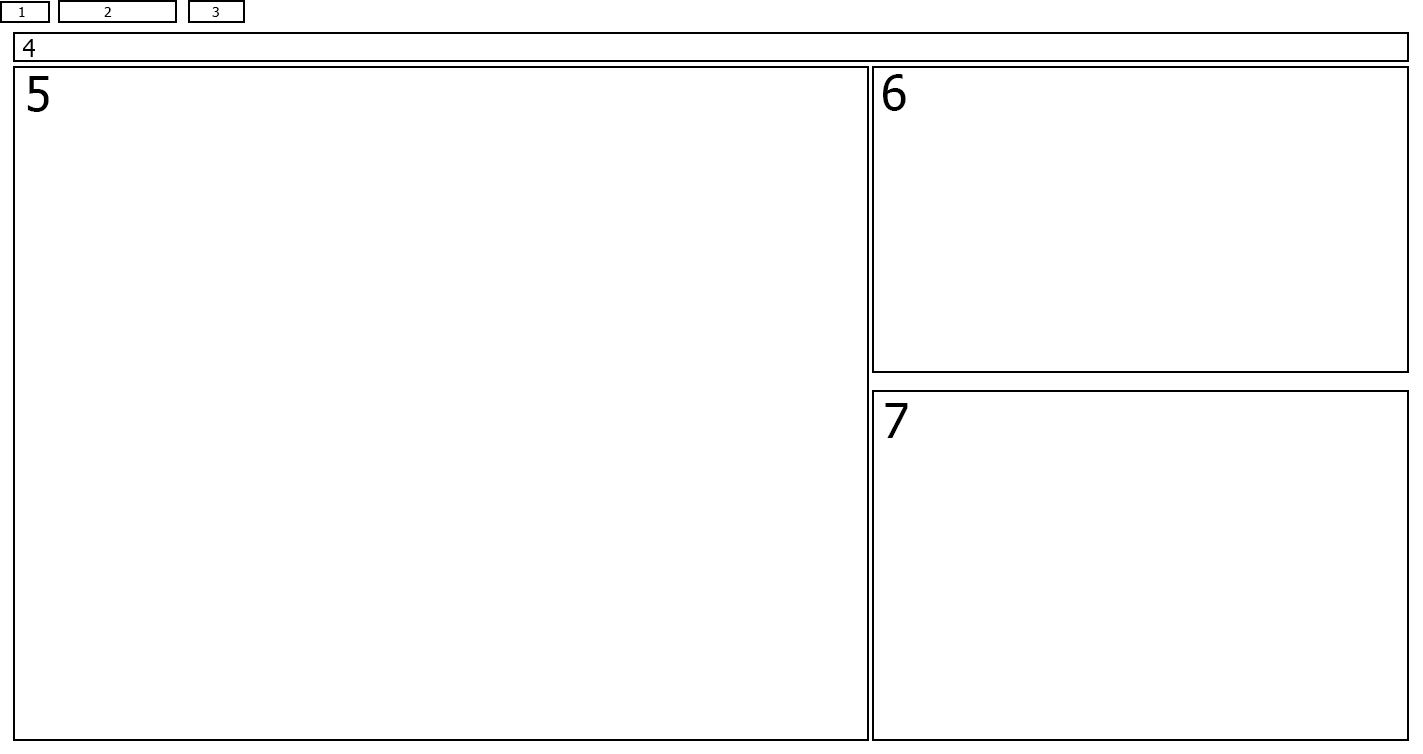


Рисунок 29 – Главное окно программы от лица главного администратора

Таблица 11 – Интерфейс главного окна от лица главного администратора

|  |  |
| --- | --- |
| № | Описание элемента |
| 1 | Элемент меню “Файл”. В нем можно открыть окно для работы с резервными копиями, настроить приложение, отключиться от сервера и выйти из программы |
| 2 | Элемент меню “Редактирование”. Содержит множественные операции редактирования: от копирования, до отмены и повторения действий |
| 3 | Элемент меню “Сервер”. В нем можно отослать и применить все изменения к БД на сервере, а также перезагрузить данные внутри программы, сделав запрос на повторное получение данных |
| 4 | Вкладки переключения на определенные таблицы из базы данных |
| 5 | Элемент, представляющий собой таблицу, в которой можно видеть и работать с записями таблицы в выбранной вкладке. В случае редактирования аудитории и включенного визуального редактора данный элемент меняется на графический элемент, в котором можно перемещать, удалять аудитории. В нижней части графического элемента появляется возможность переключаться между этажами с помощью кнопок навигации “<” и “>” |
| 6 | Область, в которой располагаются параметры редактирования записи |
| 7 | Область, в которой располагаются кнопки управления записями, а именно создание и удаление. В случае редактирования аудиторий доступна третья кнопка, которая дает возможность переключиться на визуальный редактор и обратно на табличный |

Затем можно рассмотреть общие элементы, а именно элементы 1, 2, 3. Данные элементы представляют собой кнопки меню, при активации которых происходят различные действия. Элемент 1, то есть кнопка “Файл”, показывает список возможных действий пользователю, которые он может выбрать. Список действии указан в таблице 12.

Таблица 12 – Описание кнопок меню “Файл”

|  |  |
| --- | --- |
| Название кнопки элемента меню | Функциональное предназначение элемента |
| Работа с резервными копиями | Открывает окно для работы с резервными копиями базы данных |
| Настройки приложения | Позволяет настраивать определенные аспекты приложения (На данный момент отключено) |
| Отключиться от сервера | Обеспечивает выход пользователя из учетной записи, возвращая пользователя в окно авторизации |
| Выход | Закрывает программу |

Кнопка “Редактирование” (элемент 2) показывает список возможных действий по редактированию информации пользователю. Список действий указан в таблице 13.

Таблица 13 – Описание кнопок меню “Редактирование”

|  |  |
| --- | --- |
| Название кнопки элемента меню | Функциональное предназначение элемента |
| Отмена | Отменяет предыдущее действие пользователя |
| Повторить | Возвращает отмененное действие |
| Вырезать | Позволяет вырезать информацию в буфер обмена |
| Копировать | Позволяет копировать информацию в буфер обмена |
| Вставить | Позволяет вставлять информацию из буфер обмена |
| Удалить | Позволяет удалять выбранную запись в таблице |

Список действий для кнопки “Сервер” (элемент 3) указан в таблице 14.

Таблица 14 – Описание кнопок меню “Сервер”

|  |  |
| --- | --- |
| Название кнопки элемента меню | Функциональное предназначение элемента |
| Обновить БД на сервере | Отправляет все изменения в базу данных на сервере, а значит всё что вы сделали в программе применится окончательно |
| Перезагрузить БД с сервера | Перезагрузить все полученные данные из БД с сервера |

В конечном счете осталось рассмотреть интерфейс окна работы с резервными копиями, схема которого изображена на рисунке 30

В данном окне пользователь может ознакомиться со списком уже существующих резервных копии (элемент 1) и выбрать их.

* Если пользователю нужно создать резервную копию, то ему нужно ввести уникальное имя (в элемент 2) для этой копии и затем нажать на кнопку “создать резервную копию” (элемент 3).
* Если пользователю нужно загрузить и применить уже существующую резервную копию, то ему нужно выбрать в элементе 1 интересующую резервную копию и затем нажать на кнопку “Загрузить резервную копию”. При загрузке резервной копии может произойти три события:

1. Резервная копия загружена и применена успешно, так как она не повреждена
2. Резервная копия частично повреждена. В данном случае пользователь получит оповещение о том, что в данной копии содержатся не все данные и ему предлагается сделать выбор – загрузить оставшиеся данные, но ценой того что они возможно могут иметь противоречия из-за неполноты или же отменить загрузку резервной копии
3. Резервная копия повреждена полностью. В данном случае пользователь получает оповещение о том, что данная копия повреждена и предложит ему либо удалить её из списка, либо оставить всё как есть.

* Если пользователю требуется выйти из окна для работы с резервными копиями, то ему нужно нажать на кнопку “Закрыть” (элемент 6).

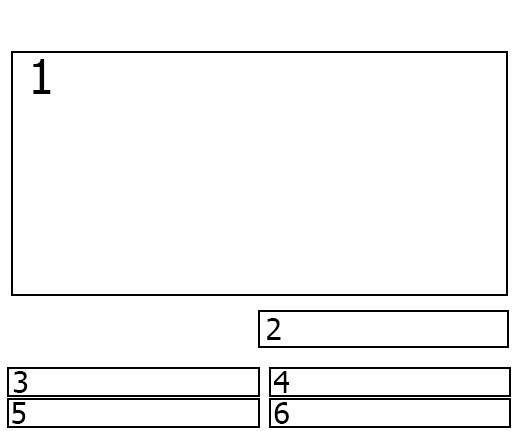


Рисунок 30 – Окно для работы с резервными копиями БД

Таблица 15 – Интерфейс окна для работы с резервными копиями БД

|  |  |
| --- | --- |
| № | Описание элемента |
| 1 | Элемент, в котором отображается список уже существующих и доступных резервных копии |
| 2 | Область, в которой назначается название резервной копии |
| 3 | Кнопка, позволяющая создать резервную копию или же переименовать её |
| 4 | Кнопка, позволяющая удалить выбранную резервную копию |
| 5 | Кнопка, позволяющая загрузить и применить выбранную резервную копию |
| 6 | Кнопка, закрывающая данное окно |

## **Инструментальные средства разработки**

Во время разработки программного продукта использовались следующие программные средства, представленные в таблице 16.

Таблица 20 – Программные средства

|  |  |
| --- | --- |
| Наименование программного средства | Характеристика |
| Windows 10 | Операционная система |
| Visual Studio 2013 Ultimate (WPF, C#) | Среда разработки программного обеспечения |
| phpMyAdmin | Интерфейс создания баз данных |
| Notepad++ | Написание php кода, написание каскадных стилей сайта и написание js файлов |
| Denwer | Локальный веб-сервер |

## **Характеристики программы**

Веб приложение состоит из следующих модулей, файлов и библиотек, показанных в таблицах 21-24.

Таблица 21 – Модули, файлы и библиотеки веб-приложения

|  |  |
| --- | --- |
| Исходники | Описание |
| RoomsDrawer.php | Формирования картинки занятости аудитории |
| Login.php | Модуль, отвечающий за авторизацию пользователя на сайте |
| index.php | Главная страница сайта |
| Logout.php | Модуль, отвечающий за выход пользователя из учетной записи на сайте |
| Schedule.php | Формирование таблицы расписания, с помощью отправления MySQL запросов к БД |
| GroupsSelectorOptionsFilter.php | Формирование элементов списка групп, с помощью отправления MySQL запросов к БД |
| GroupsSelectorOptions.php | Формирование элементов списка групп, с помощью отправления MySQL запросов к БД |
| PodrazSelectorOptions.php | Формирование элементов списка подразделений, с помощью отправления MySQL запросов к БД |
| PrepodSelectorOptions.php | Формирование элементов списка преподавателей, с помощью отправления MySQL запросов к БД |
| MainScript.js | Главный скрипт веб-приложения |
| indexStyle.css | Файл каскадных стилей веб-приложения |
| Prettycheckbox.css | Сторонний файл каскадных стилей веб-приложения |
| В папке Tools/ | Находятся файлы сторонних приложений: phpMyAdmin и dnsearch |
| В папке DataTables/ | Находятся файлы сторонней библиотеки DataTables |
| В папке dist/ | Находятся файлы сторонней библиотеки Air Datepicker |
| В папке errors/ | Находятся файлы ошибок при присоединении к веб-приложению |
| В папке images/ | Находятся файлы, содержащий картинки |

Таблица 22 – Сторонние библиотеки веб-приложения

|  |  |  |
| --- | --- | --- |
| Название библиотеки | Описание | Заимствование |
| jQuery | Библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML | Заимствованная библиотека |
| Air Datepicker | Плагин-библиотека для веб-сайта, позволяющий легко внедрить календарь, написанный на языке jQuery  GitHub разработчика: <https://github.com/t1m0n/air-datepicker> | Используется в формировании календаря, в котором можно выбирать дату |
| Pretty checkbox | Плагин-библиотека, позволяющий создавать красивые вариации для элементов с галочками  GitHub разработчика:  <https://lokesh-coder.github.io/pretty-checkbox/> | Применено для ползунка включения графического отображения |
| DataTables | Плагин, позволяющий создавать удобную таблицу на веб-странице.  Сайт разработчика: <https://datatables.net/> | Использовано в формировании таблицы расписания |

Таблица 23 – Модули, файлы и библиотеки скомпилированной программы

|  |  |
| --- | --- |
| Файлы | Описание |
| MWLData.rmf | Генерируемый программой файл, отвечающий за хранение данных авторизации (запомнить меня) |
| SheduleAuditorVisualisator.exe | Исполняемый файл программы |
| Xceed.Wpf.AvalonDock.dll, Xceed.Wpf.AvalonDock.Themes.Aero.dll, Xceed.Wpf.AvalonDock.Themes.Metro.dll, Xceed.Wpf.AvalonDock.Themes.VS2010.dll, Xceed.Wpf.Toolkit.dll,  EntityFramework.dll, EntityFramework.SqlServer.dll, EntityFramework.dll, WpfAnimatedGif.dll, MySql.Data.dll | Файлы сторонних библиотеки |
| Папки de, es, fr, hu, it, pt-BR, ro, sv, zh-Hans | Находятся файлы локализации одной из сторонних библиотек |
| В папке Backups/ | Содержит резервные копии базы данных |

Таблица 24 – Сторонние библиотеки, задействованные в создании программы

|  |  |  |
| --- | --- | --- |
| Название библиотеки | Описание | Заимствование |
| Dragablz | Библиотека, позволяющая создавать стилизованные перетаскиваемые вкладки  GitHub разработчика:  <https://github.com/ButchersBoy/Dragablz> | Перетаскиваемые вкладки таблиц БД |
| MySQL | Позволяет программе работать с MySQL | Заимствованная библиотека |
| Extended.Wpf.Toolkit v3.5.0 | Расширяет возможности в WPF.  NuGet ссылка:  <https://www.nuget.org/packages/Extended.Wpf.Toolkit/> | С его помощью были задействованы элементы, отвечающие за выбор времени |
| WpfAnimatedGif v1.4.18 | Позволяет программе понимать и воспроизводить анимированные .gif картинки  Сайт разработчика: <https://www.nuget.org/packages/WpfAnimatedGif/> | Заимствованная библиотека |

# **ЗАКЛЮЧЕНИЕ**

В процессе выполнения выпускной квалификационной работы был проведен анализ предметной области. В результате проведенного сравнительного анализа аналогичных продуктов («schedule.rea-yar.ru» и «ruz.spbstu.ru») были сформулированы требования к разрабатываемому ПО, на основе которых были реализованы реляционная база данных и соответствующее веб-приложение и исполнительный exe файл, которые вместе создают дистанционную систему мониторинга занятости аудиторий при реализации учебного процесса в Ярославском филиале РЭУ им. Г.В. Плеханова и решающая поставленные задачи.

Входной информацией для приложения являются:

* Выбор даты начала и окончания;
* ФИО преподавателя (руководителя);
* Наименование группы;
* Подразделение;
* Тип аудитории;

К основным функциям системы относятся:

* Просмотр занятости аудитории с учетом расписания
* Просмотр актуального расписания;
* Возможности администрирования базы данных;

Выходная информация:

* Отфильтрованное расписание
* Визуальное отображение занятости аудитории в зависимости от расписания и группы пользователя, если он имеет учетную запись на сайте

Произведено обоснование выбора программных средств реализации проекта.

Перспективы развития программы направлены на расширение имеющегося функционала. Например, планируется добавить систему заявок для проведения занятия в определенной аудитории у преподавателей, более продвинутое отображение занятости аудиторного фонда с добавлением куда более явных отличительных особенностей между выводимыми аудиториями и добавление расположение лестниц и других не менее значимых объектов на графической карте, чтобы, например, студенту было ещё удобнее и проще ориентироваться в графической карте. Имеющаяся база с авторизацией пользователей даёт возможности расширение возможностей ПО, например добавление дневника явки студента или создание самой настоящей системы управления курсами, типа Moodle.

# **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Веб-стандарты [электронный ресурс]: <http://site.nic.ru/node/478>
2. Реализация MVC паттерна на примере создания сайта-визитки на PHP [электронный ресурс]: <https://habr.com/post/150267/>
3. Обязанности менеджера [электронный ресурс]: <http://www.rb-edu.ru/library/articles/articles_9387.html>
4. Руководство по WPF [электронный ресурс]: <https://metanit.com/sharp/wpf/>
5. Требования к Сайту [электронный ресурс]: <http://www.studfiles.ru/preview/5609211/page:2>
6. Словарь терминов [электронный ресурс]: <http://www.setup.ru/wiki?letter=%D0%9B>
7. Создание веб-сайтов на основе стандартов [электронный ресурс]: <https://msdn.microsoft.com/ru-ru/library/hh273394(v=vs.85).aspx>
8. Общие сведения требований к Web-сайту в целом [электронный ресурс]: <http://lektsii.org/6-87564.html>
9. Подключение к базе данных MySQL с использованием C# [электронный ресурс]: <https://o7planning.org/ru/10513/connecting-to-mysql-database-using-csharp>
10. Формирование требований к веб-сайту [электронный ресурс]: <https://cyberleninka.ru/article/n/formirovanie-trebovaniy-k-veb-saytu-kak-osnovnomu-rezultatu-proekta-sozdaniya-internet-soobschestv>
11. Гончаров А.Ю. Web-дизайн: HTML, JavaScript и CSS. Карманный справочник / А.Ю. Гончаров, 2012 —320с.
12. Формулировка задач проектирования [электронный ресурс]: <http://studbooks.net/2023283/informatika/formulirovka_zadach_proektirovaniya>
13. Руководство пользователя [электронный ресурс]: <http://php.net/>
14. Подробное руководство по установке и настройке Denwer [электронный ресурс]: <https://www.internet-technologies.ru/articles/podrobnoe-rukovodstvo-po-ustanovke-i-nastroyke-denwer.html>
15. MySQL Connector/NET Developer Guide [электронный ресурс]: <https://dev.mysql.com/doc/connector-net/en/connector-net-tutorials-sql-command.html>
16. Современный учебник JavaScript [электронный ресурс]: [https://learn.javascript.ru](https://learn.javascript.ru/)
17. jQuery для начинающих [электронный ресурс]: <https://habr.com/ru/post/38208/>
18. Стандарты шифрования данных [электронный ресурс]: <https://www.internet-technologies.ru/articles/standart-shifrovaniya-dannyh.html>
19. Знакомство с нотацией IDEF0 и пример использования [электронный ресурс]: <https://habr.com/ru/company/trinion/blog/322832/>
20. Пространство имен - System.IO (Работа с файлами, директориями, дисками, памятью) [электронный ресурс]: <http://www.cyberforum.ru/csharp-beginners/thread163620.html>
21. Базовая сериализация [электронный ресурс]: <https://docs.microsoft.com/ru-ru/dotnet/standard/serialization/basic-serialization>

# **ПРИЛОЖЕНИЯ**

# **ПРИЛОЖЕНИЕ 1. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЮ**

1. Требования к техническому обеспечению и их характеристикам, представлены в таблице 1. В данной таблице предъявлены минимальные и рекомендованные характеристики для использования программного продукта.

Таблица 1 – Системные требования

|  |  |  |  |
| --- | --- | --- | --- |
| № | Наименование | Минимальные характеристика | Рекомендованная характеристика |
| 1 | Процессор | 1,3 ГГц | 2,3 Ггц |
| 2 | Оперативная память | 2 гб | 4 гб |
| 3 | Свободное место на серевер | 500 мб | 2 гб |

1. Требования к системному программному обеспечению представлены в таблице 2.

Таблица 2 – Программное обеспечение для запуска

|  |  |  |
| --- | --- | --- |
| № | Наименование | Минимальные характеристика |
| 1 | Windows 10 | Операционная система |
| 2 | Denwer | Локальный сервер (Apache, PHP, MySQL, Perl и т.д.) |
| 3 | Net Framework 4.5 и выше | Программная платформа |

Для того, чтобы начать работу нужно открыть директорию …Denwer3\Denwer3\denwer\ и запустить файл Run.exe. Затем нужно дождаться, когда исчезнут командные окна.

После, пользователю следует открыть веб-браузер и прописать в адресной строке <http://localhost/> или просто localhost. При вводе данного адреса пользователь попадает на главную страницу расписания (рисунок 1)

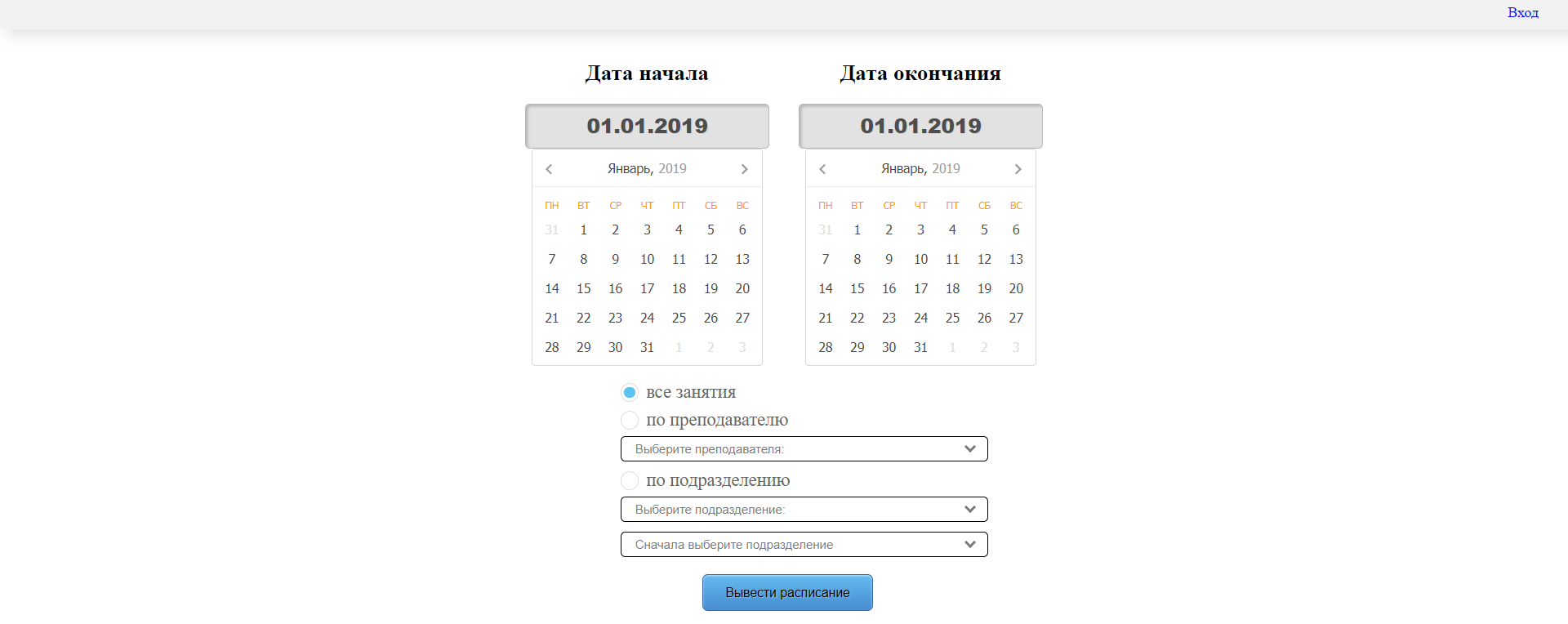


Рисунок 1 – Главная страница сайта

На главной странице пользователь может просматривать актуальное расписание. В проекте уже присутствует расписание на месяц январь от 10 до 31 числа, потому можно выбрать, например, в дате начала дату 10.01.2019 и в дате окончания дату 12.01.2019, а потом применить 3 фильтра на выбор “Все занятия”, “по преподавателю”, “по подразделению”. Чтобы вывести расписание с учетом выбранного промежутка даты и фильтров пользователю нужно нажать на кнопку “Вывести расписание”. При нажатий появляется таблица с актуальным расписанием на момент от 10.01.2019 по 12.01.2019 (рисунок 2)

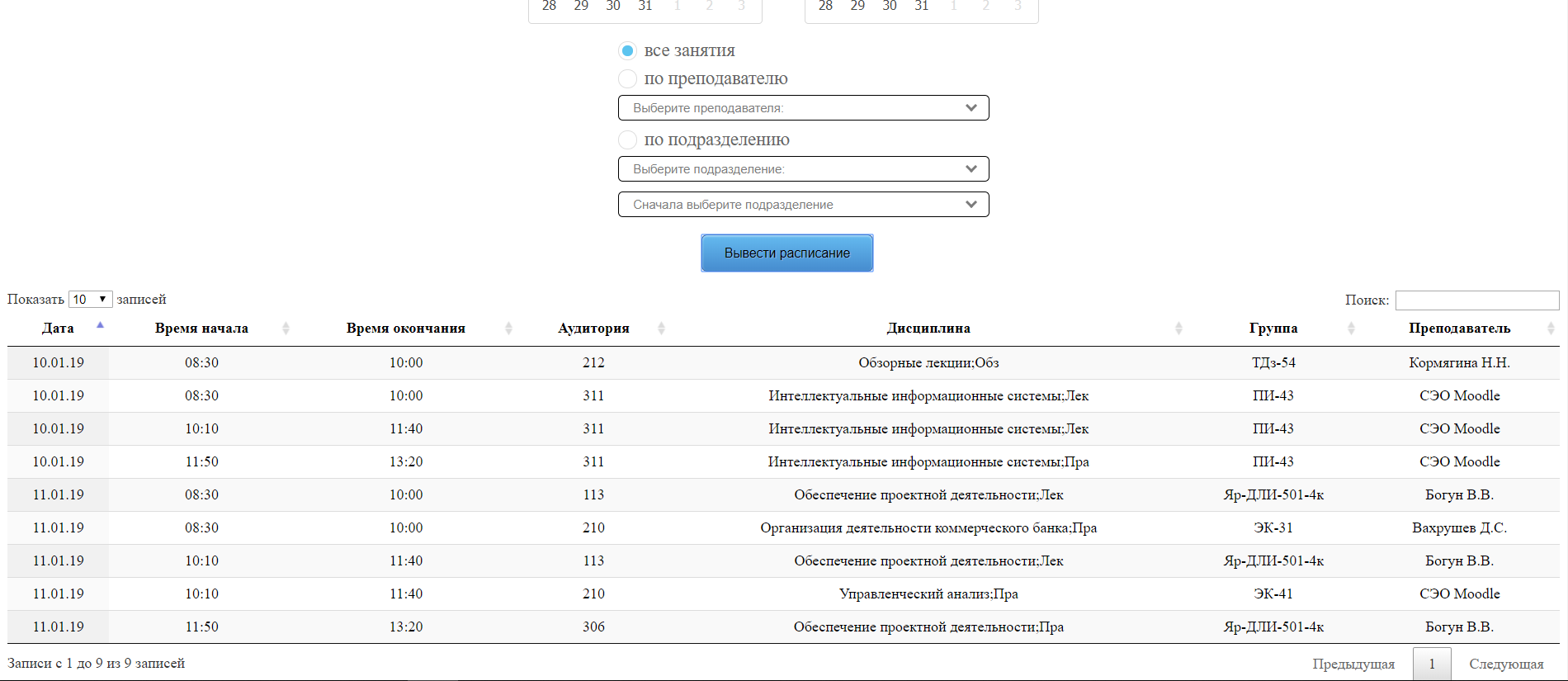


Рисунок 2 – Выведенная таблица с расписанием на период от 10.01.19 по 12.01.19

Но почему не видна визуальная занятость аудитории? Всё просто. Пользователю нужно иметь профиль на сайте, который регистрируется строго ведущим специалистом. Это сделано для того, чтобы чужие люди не могли пользоваться данным сайтом в полном мере. Чтобы войти в профиль, нужно нажать на текстовую кнопку “Вход”, расположенную в шапке сайта (рисунок 3)



Рисунок 3 – Месторасположение кнопки “Вход”

При нажатии на данную кнопку страница сайта немного затемняется и перед пользователем появляется окошко авторизации (рисунок 4). В появившемся окне нужно указать логин и пароль от существующей учетной записи. В проекте уже существует три учетные записи (Student, Teacher, Admin). Пароль к учетным записям соответствуют логину с учетом регистра.

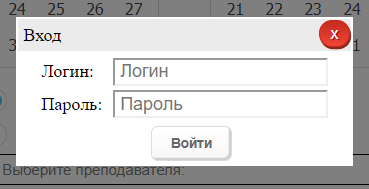


Рисунок 4 – Окно авторизации

Если введенные логин и пароль оказались корректными, то страница пользователя перезагрузится, и шапка сайта изменится (рисунок 5)

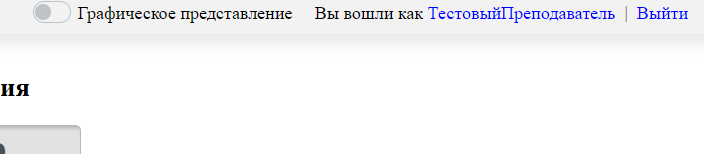


Рисунок 5 – Изменения в шапке сайта после авторизации

После авторизации у пользователя появляется возможность включить графическое представление занятости аудитории, посмотреть свой профиль и возможность выйти из собственной учетной записи. При нажатии на кнопку “Выйти” страница пользователя перезагружается, отключая его от учетной записи. При нажатии на имя, расположенное после надписи “Вы вошли как” перед пользователем открывается небольшое окошко, в котором предоставлена небольшая сводка о его профиле (рисунок 6).

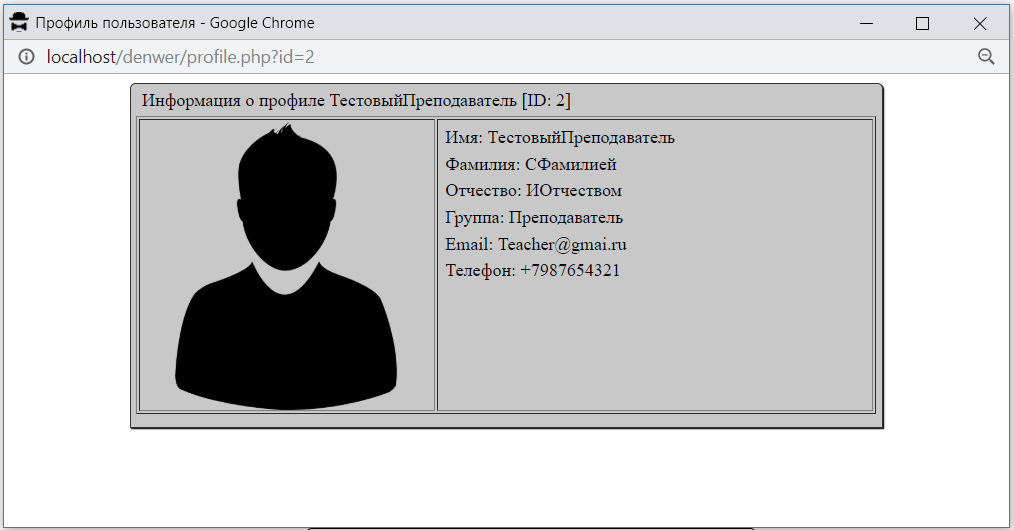


Рисунок 6 – Окно профиля пользователя

Если нажать на ползунок “графическое представление”, то снизу, перед таблицей с расписанием появится графическая карта, которая как раз и рисует свободные и занятые аудитории с учетом назначенного момента времени и дополнительных входных фильтров (рисунок 7).

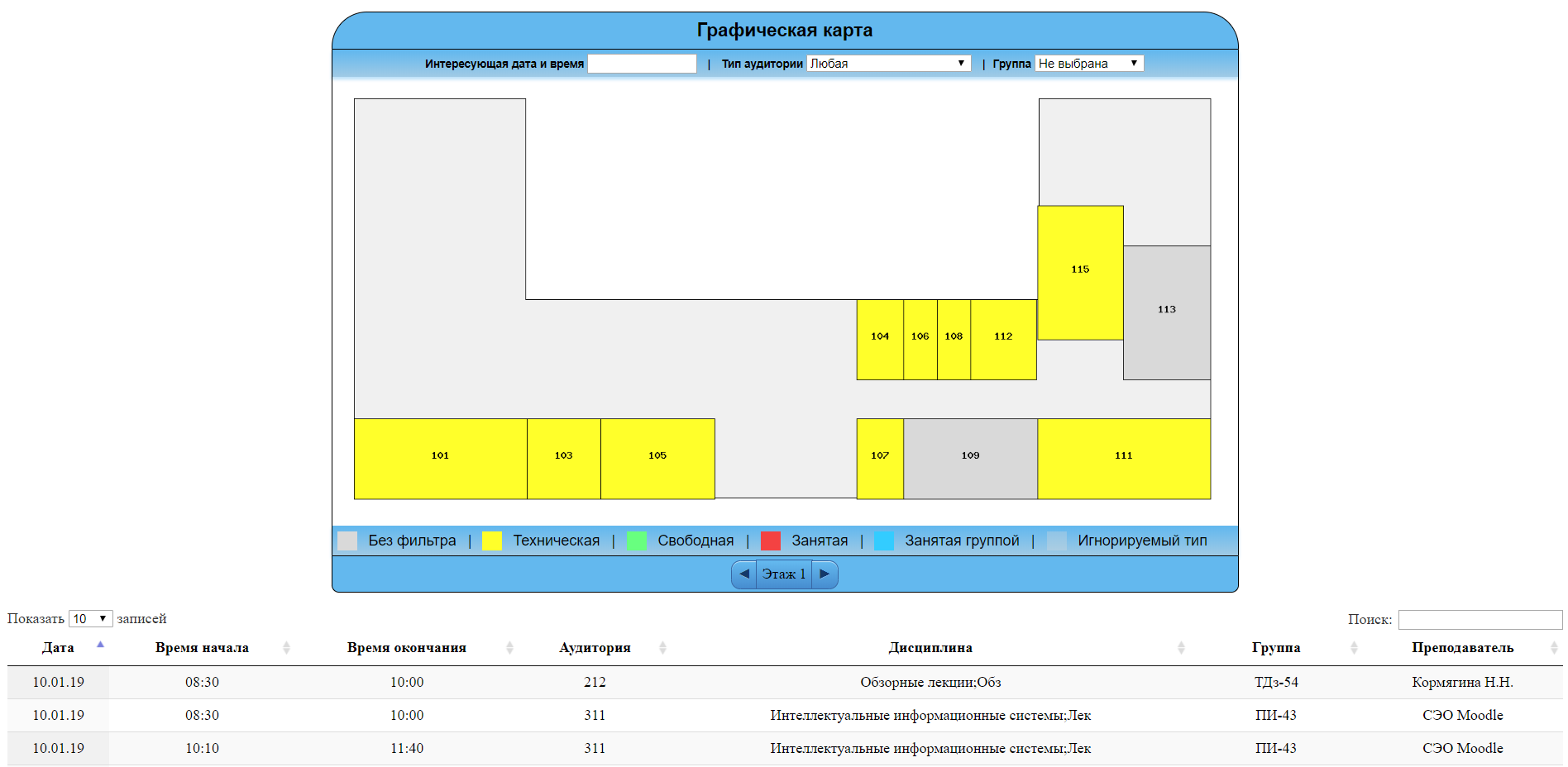


Рисунок 7 – Графическая карта, представляющая наглядно свободно и занятые аудитории

Для примера можно ввести 11 число и выбрать время 10:10. Изображение карты динамически изменится и, например, на 2 этаже можно увидеть, как аудитория 210 занята в этот временной момент (рисунок 8)

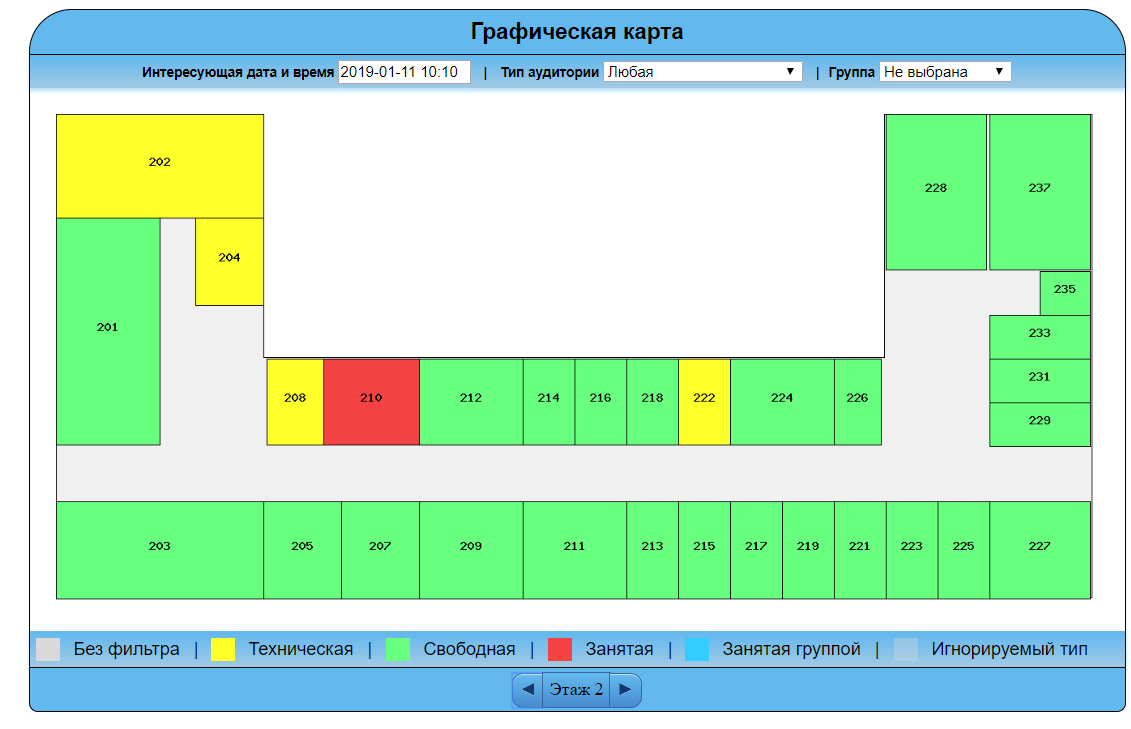


Рисунок 8 – Графическая карта, к которой применен фильтр интересующей даты и времени. Вид 2 этажа.

Доступные фильтры отличаются у студента и преподавателя. Например, студенты могут лишь выбрать группу и посмотреть на графической карте только занятую аудиторию выбранной группой, без возможности увидеть занятые аудитории другими группами и свободные.

Теперь можно приступить к рассмотрению программы для администрирования БД. Для этого нужно открыть файл ScheduleAuditorVisualisator.exe, расположенный в директории …\SheduleAuditorVisualisator\SheduleAuditorVisualisator\bin\Release\

При запуске программы перед пользователем появится окно авторизации (рисунок 9)

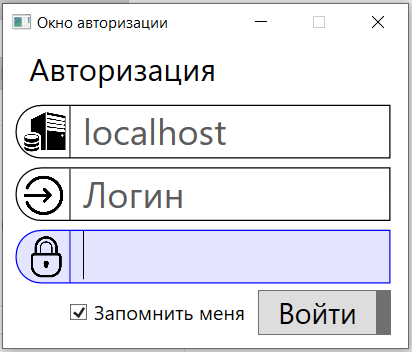


Рисунок 9 – Окно авторизации в программе.

Для наглядного примера были созданы две учетные записи. Учетная запись заведующего кафедрой имеет логин ZavKafedroi и пароль Password123, учетная запись ведущего специалиста имеет логин root и пароль root. Также пользователь может поставить галочку в “Запомнить меня”, чтобы при следующем запуске программы данные входа были уже введены в соответствующие поля и всё что требуется сделать, это нажать на кнопку “Войти”. Рассмотрим первый случай, когда пользователь входит под учетной записью заведующего кафедрой. При вводе соответствующего логина и пароля и нажатии кнопки “Войти” окошко становится неактивным и появится анимация загрузки. После загрузки данных с сервера, перед заведующим кафедрой появляется главное окно (рисунок 10)

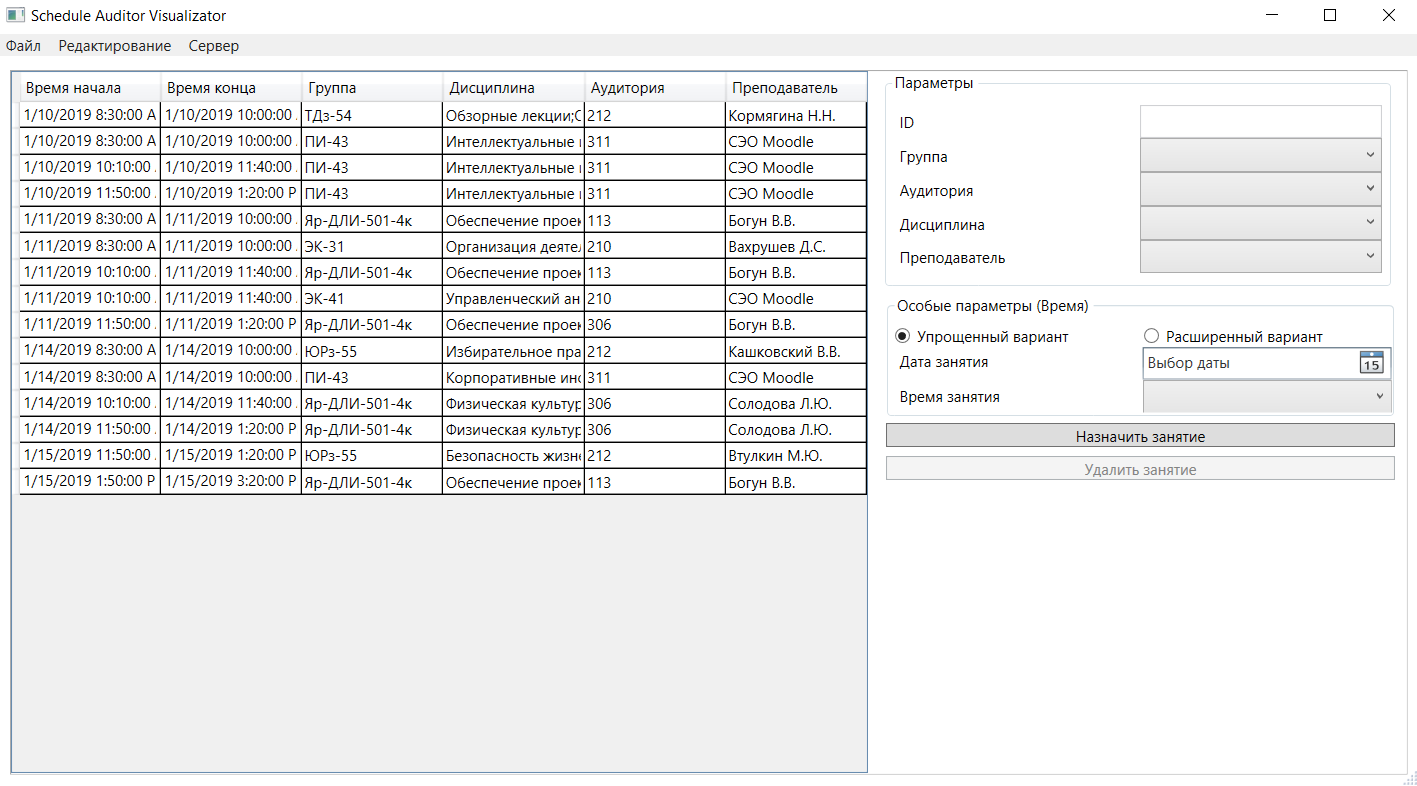


Рисунок 10 – Главное окно программы от лица заведующего кафедры

Теперь заведующий кафедры может видеть таблицу, в которой находятся записи актуального расписания, а также имеет возможность манипулировать данными расписания (создавать, удалять и изменять их). Для того, чтобы применить все изменения к БД на сервере, нужно нажать в верхнем меню кнопку “Сервер” и выбрать из выпадающего списка пункт “Обновить БД на сервере”. При выполнении данной операции все записи отправляются на сервер и там они обрабатываются. После завершения операции пользователь будет уведомлен, а данные на сервере будут обновлены. В возможностях также присутствует функция перезагрузки локальной БД с помощью запроса к серверу. Для этого нужно нажать в верхнем меню кнопку “Сервер” и выбрать из выпадающего списка пункт “Перезагрузить БД с сервера”. В этом случае заведующий кафедры перезагрузит базу данных на компьютере, обновив её актуальными данными с БД, распложенной на сервере.

Ещё заведующий кафедры может воспользоваться созданием резервных копии. Для этого нужно нажать на верхнее меню, на кнопку “Файл”. Затем в выпадающем меню выбрать пункт “Работа с резервными копиями”. Перед лицом заведующего кафедры появляется окошко, в котором можно управлять резервными копиями (рисунок 11)

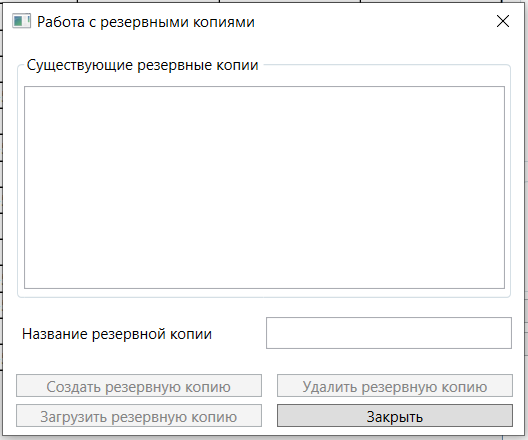


Рисунок 11 – Окно для работы с резервными копиями БД

В данном окне заведующий кафедры может ознакомиться с уже существующим списком резервных копии и может создавать, изменять и загружать резервные копии. Резервные копии нужны для того, чтобы можно было произвести откат БД к определенному моменту времени, в случае повреждении или утери данных.

Теперь рассмотрим случай, когда пользователь входит под логином и паролем ведущего специалиста. При вводе соответствующего логина и пароля и нажатии кнопки “Войти” окошко становится неактивным и появится анимация загрузки. После загрузки данных с сервера, перед ведущим специалистом появляется главное окно. Оно практически не отличается от окна заведующего кафедрой и обладает большим функционалом и возможностями (рисунок 12)

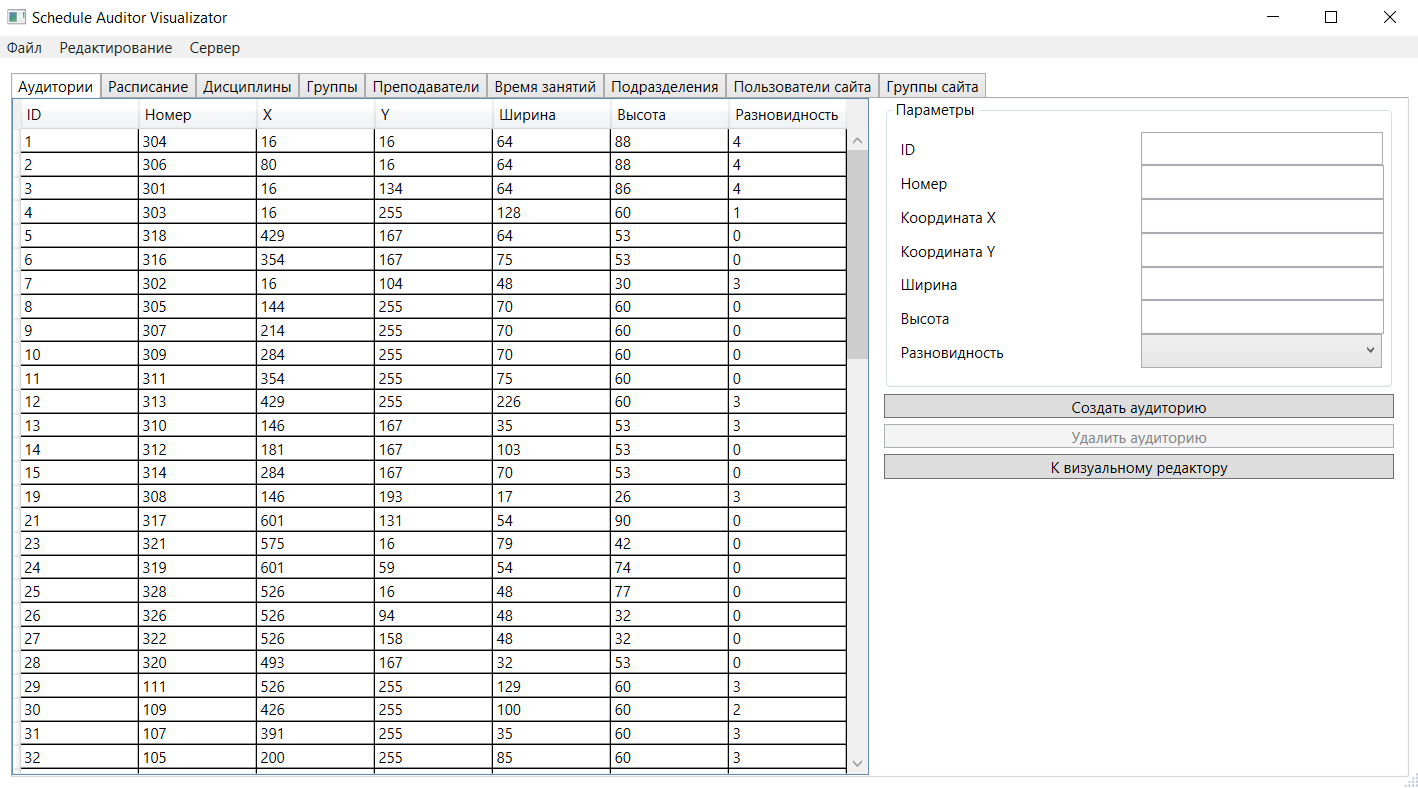


Рисунок 12 – Главное окно программы от лица ведущего специалиста

У ведущего специалиста гораздо больше возможностей. Он имеет доступ ко всем таблицам БД и может переключаться между ними с помощью вкладок. В каждой вкладке можно производить манипуляции с данными в каждой таблице, а в самой вкладке “Аудитории”, ведущий специалист может перейти на визуальный редактор. Для этого нужно нажать на кнопку “К визуальному редактору”. Таблица исчезнет и появится графический редактор аудитории (рисунок 13).

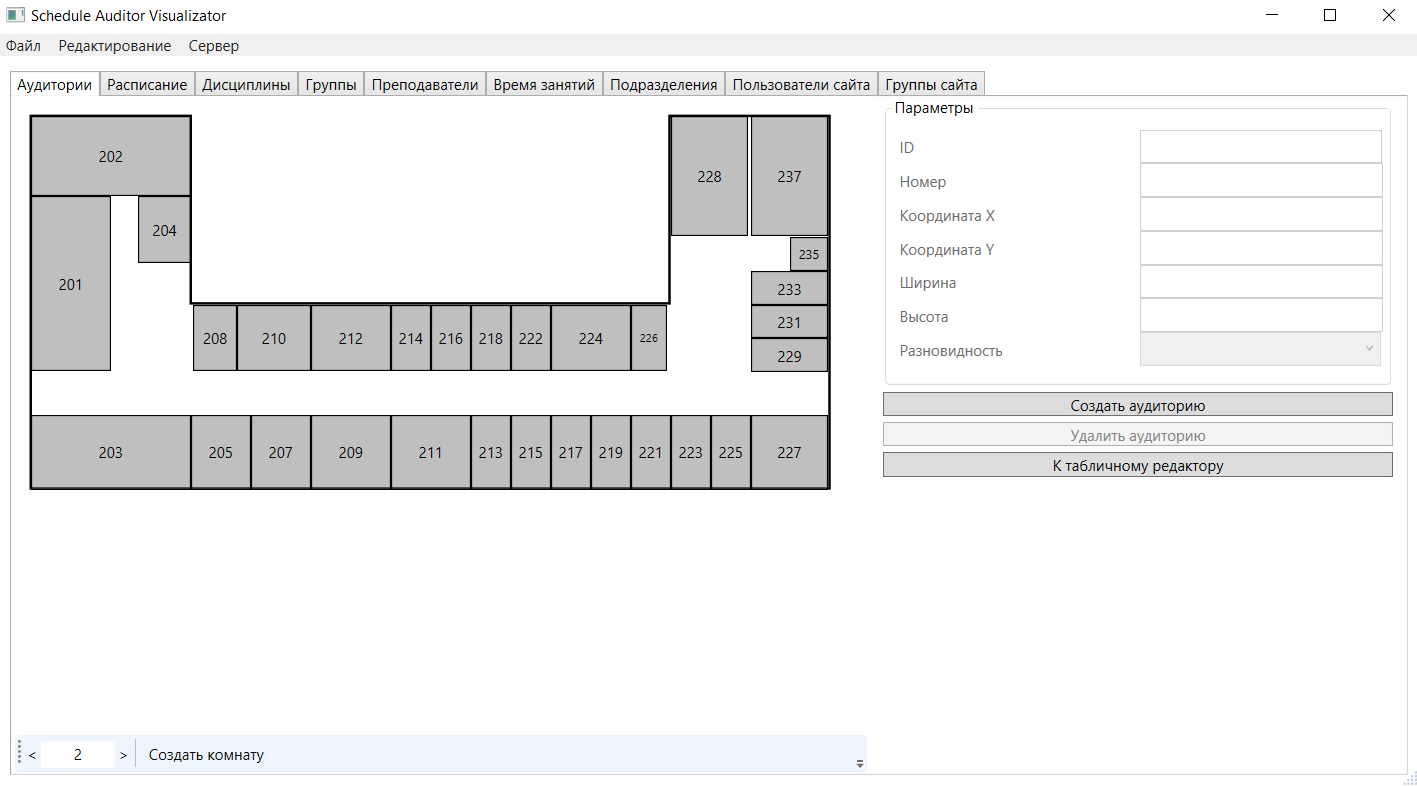


Рисунок 13 – Графический редактор аудиторий

В графическом редакторе удобнее работать с координатами и размерами аудитории, чем с табличным вариантом, так как всё имеет наглядность. Остальная часть программы не отличается по функционалу от функционала заведующего кафедры.

# **ПРИЛОЖЕНИЕ 2. ТЕКСТ ПРОГРАММЫ**

Текст программы веб-приложения:

1. RoomsDrawer.php

Модуль, который отвечает за создание визуальной картинки занятости аудитории

<?php

session\_start();

if ($\_SESSION["Username"]==null || $\_SESSION["Username"]=="guest")

{

return;

}

if($\_GET['size']==null || $\_GET['lvl']==null) {

return;

}

$iSizeModifier = intval($\_GET['size']);

$cLevel = intval($\_GET['lvl']);

$dSelDate = $\_GET['date'];

$iRoomTypeFilter = $\_GET['room'];

$iGroup = $\_GET['grp'];

// задание массива точек для многоугольника

$values = array(

16\*$iSizeModifier, 16\*$iSizeModifier,

144\*$iSizeModifier, 16\*$iSizeModifier,

144\*$iSizeModifier, 166\*$iSizeModifier,

527\*$iSizeModifier, 166\*$iSizeModifier,

527\*$iSizeModifier, 16\*$iSizeModifier,

655\*$iSizeModifier, 16\*$iSizeModifier,

655\*$iSizeModifier, 314\*$iSizeModifier,

16\*$iSizeModifier, 314\*$iSizeModifier

);

// создание изображения

$image = imagecreatetruecolor(675\*$iSizeModifier, 335\*$iSizeModifier);

$black = imagecolorallocate($image, 0, 0, 0);

$hbcg = imagecolorallocate($image, 240, 240, 240);

$bbl = imagecolorallocate($image, 1, 1, 1);

imagecolortransparent($image, $black);

// рисование многоугольника

imagefilledpolygon($image, $values, 8, $hbcg);

imagepolygon($image, $values, 8, $bbl);

$con=mysqli\_connect("localhost","root","root","shedulebase");

if (mysqli\_connect\_errno())

{

echo "Failed to connect to MySQL: " . mysqli\_connect\_error();

return;

}

$sqlClassRooms="SELECT ID,NUMBER,CoordX,CoordY,Width,Height,Type FROM classrooms ORDER BY ID";

$sqlSchedule="SELECT ID,GROUPID,STARTTIME,ENDTIME,CLASSROOMID FROM shedule ORDER BY ID";

$FreeRoomColor = imagecolorallocatealpha($image, 104, 255, 127, 0);

$OccupiedRoomColor = imagecolorallocatealpha($image, 244, 66, 66, 0);

$RoomColor = imagecolorallocatealpha($image, 217, 217, 217, 0);

$ServiceRoomColor = imagecolorallocatealpha($image, 255, 255, 42, 0);

$RoomBorder = imagecolorallocate($image, 1, 1, 1);

$OccupiedByGroupRoomColor = imagecolorallocatealpha($image, 51, 204, 255, 0);

$RoomBorderUnactive = imagecolorallocatealpha($image, 1, 1, 1, 99);

$FreeRoomColorUnactive = imagecolorallocatealpha($image, 104, 255, 127, 99);

$OccupiedRoomColorUnactive = imagecolorallocatealpha($image, 244, 66, 66, 99);

$RoomColorUnactive = imagecolorallocatealpha($image, 217, 217, 217, 99);

$ServiceRoomColorUnactive = imagecolorallocatealpha($image, 255, 255, 42, 99);

if ($result=mysqli\_query($con,$sqlClassRooms))

{

while ($row=mysqli\_fetch\_row($result))

{

$CoordX = intval($row[2])\*$iSizeModifier;

$CoordY = intval($row[3])\*$iSizeModifier;

$Width = $CoordX + intval($row[4])\*$iSizeModifier;

$Height = $CoordY + intval($row[5])\*$iSizeModifier;

$RoomNum = intval($row[1]);

$Level = $cLevel;

$txtX = $CoordX+($Width-$CoordX)/2-13;

$txtY = $CoordY+($Height-$CoordY)/2-13;

//немного хардкода

if($RoomNum==310) {

$txtY = $CoordY+($Height-$CoordY)/2-32;

}

$iType = intval($row[6]);

$cCol = $RoomColor;

$cColBorder = $RoomBorder;

if($iType==3) {

$cCol = $ServiceRoomColor;

}

if($RoomNum >= 100 \* $Level && $RoomNum <= (100\*$Level)+99) {

if($dSelDate!=null) {

if($\_SESSION["GroupID"]==1) {

$cCol = $RoomColor;

$cColBorder = $RoomBorder;

if ($result2=mysqli\_query($con,$sqlSchedule))

{

while ($row2=mysqli\_fetch\_row($result2))

{

$sTimeR = new DateTime(strval($row2[2]));

$sTimeR = $sTimeR->getTimestamp();

$eTimeR = new DateTime(strval($row2[3]));

$eTimeR = $eTimeR->getTimestamp();

if($row[0]==$row2[4] && (intval($sTimeR)<=intval($dSelDate) && intval($eTimeR)>=intval($dSelDate))) {

$cCol = $RoomColor;

$cColBorder = $RoomBorder;

if($iGroup!=null) {

if(intval($iGroup)==intval($row2[1])) {

$cCol = $OccupiedByGroupRoomColor;

$cColBorder = $RoomBorder;

}

}

break;

}

}

mysqli\_free\_result($result2);

}

} else {

if($iType==3) {

$cCol = $ServiceRoomColor;

} else {

$cCol = $FreeRoomColor;

}

if ($result2=mysqli\_query($con,$sqlSchedule))

{

while ($row2=mysqli\_fetch\_row($result2))

{

$sTimeR = new DateTime(strval($row2[2]));

$sTimeR = $sTimeR->getTimestamp();

$eTimeR = new DateTime(strval($row2[3]));

$eTimeR = $eTimeR->getTimestamp();

if($row[0]==$row2[4] && (intval($sTimeR)<=intval($dSelDate) && intval($eTimeR)>=intval($dSelDate))) {

$cCol = $OccupiedRoomColor;

if($iGroup!=null) {

if(intval($iGroup)==intval($row2[1])) {

$cCol = $OccupiedByGroupRoomColor;

}

}

break;

}

}

mysqli\_free\_result($result2);

}

}

}

if($iRoomTypeFilter!=null) {

if($iType!=$iRoomTypeFilter) {

$cCol=$RoomColorUnactive;

$cColBorder = $RoomBorderUnactive;

}

}

imagefilledrectangle($image, $CoordX, $CoordY, $Width, $Height, $cCol);

imagerectangle ($image, $CoordX , $CoordY, $Width, $Height, $cColBorder);

imagestring($image, 4\*$iSizeModifier, $txtX, $txtY, $RoomNum, $cColBorder);

}

}

mysqli\_free\_result($result);

}

// вывод изображения

header('Content-type: image/png');

imagepng($image);

imagedestroy($image);

?>

Текст программы exe программы:

1. MainWindow.xaml.cs

C# код для окна MainWindow

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using MySql.Data;

using MySql.Data.MySqlClient;

using System.Data.SqlClient;

using System.ComponentModel;

using System.Security.Cryptography;

using System.IO;

using SheduleAuditorVisualisator.FileSystemClasses;

namespace SheduleAuditorVisualisator

{

public partial class MainWindow : Window

{

public bool ServerBoxEmpty = true;

public bool LoginBoxEmpty = true;

public bool PasswordBoxEmpty = true;

public MySqlConnection connection;

public string KeepConnectionString;

public bool sheduleExist = false;

private readonly BackgroundWorker worker = new BackgroundWorker();

private const string initEncrKey = "pemgail9uzpgzl88";

private const int keysize = 256;

public RememberMeClass currentData;

public string StartingFilePath = @"" + System.AppDomain.CurrentDomain.BaseDirectory + "MWLData.rmf";

public MainWindow()

{

InitializeComponent();

worker.DoWork += worker\_DoWork;

worker.RunWorkerCompleted += worker\_RunWorkerCompleted;

}

private void worker\_DoWork(object sender, DoWorkEventArgs e)

{

List<object> argumentList = e.Argument as List<object>; //acquires the passed in arguments

string argument1 = (string) argumentList [0];

string argument2 = (string) argumentList [1];

string argument3 = (string) argumentList [2];

string connectionString = "SERVER=" + argument1 + ";UID=" + argument2 + ";" + "PASSWORD=" + argument3 + ";";

connection = new MySqlConnection(connectionString);

try

{

connection.Open();

string connectionString2 = "SELECT \* FROM information\_schema.tables WHERE table\_schema = 'shedulebase2'";

MySqlCommand cmd = new MySqlCommand(connectionString2, connection);

using (MySqlDataReader rdr = cmd.ExecuteReader())

{

if (rdr.Read().Equals("True"))

{

sheduleExist = true;

}

else

{

sheduleExist = false;

}

e.Result = "Open";

}

}

catch (MySqlException ex)

{

e.Result = "Closed";

}

}

private void worker\_RunWorkerCompleted(object sender,

RunWorkerCompletedEventArgs e)

{

if (e.Result.ToString().Equals("Open"))

{

int iMode = -1;

bool bAdmMode = false;

string connectionString2 = "SHOW GRANTS FOR '"+LoginBox.Text+"'@'"+ServerBox.Text+"';";

MySqlCommand cmd = new MySqlCommand(connectionString2, connection);

using (MySqlDataReader rdr = cmd.ExecuteReader())

{

while (rdr.Read())

{

if (rdr.GetValue(0).ToString().StartsWith("GRANT ALL PRIVILEGES ON \*.\* TO"))

{

iMode = 1;

break;

}

if (rdr.GetValue(0).ToString().StartsWith("GRANT SELECT, INSERT, UPDATE, DELETE, DROP ON `shedulebase`.`shedule`"))

{

iMode = 0;

break;

}

}

}

if (iMode == -1)

{

AuthButton.IsEnabled = true;

MainGrid.Effect = null;

MainGrid.IsEnabled = true;

LoadingIcon.Visibility = Visibility.Hidden;

LoginWindow.Height = 325.522f;

ErrorLabel.Visibility = Visibility.Visible;

ErrorLabel.Text = "Произошла ошибка. Введенный пользователь не имеет достаточных прав для пользования программой.";

return;

}

else if (iMode == 1)

{

bAdmMode = true;

}

Console.Out.WriteLine("AdminMode=" + bAdmMode);

this.Hide();

AfterLoginWindow aflWindow = new AfterLoginWindow(KeepConnectionString, sheduleExist, bAdmMode);

aflWindow.Show();

this.Close();

}

else

{

AuthButton.IsEnabled = true;

MainGrid.Effect = null;

MainGrid.IsEnabled = true;

LoadingIcon.Visibility = Visibility.Hidden;

LoginWindow.Height = 325.522f;

ErrorLabel.Visibility = Visibility.Visible;

ErrorLabel.Text = "Произошла ошибка. Сервер не отвечает, либо был введен неправильный логин или пароль.";

}

}

private void LoginTextBoxFocus(object sender, RoutedEventArgs e)

{

TextBox txtBox = (TextBox)sender;

Grid eGrid = (Grid)VisualTreeHelper.GetParent(txtBox);

Border eBorder = (Border)VisualTreeHelper.GetParent(eGrid);

eBorder.Background = new SolidColorBrush(Colors.Blue) { Opacity = 0.1 };

((GridSplitter)VisualTreeHelper.GetChild(eGrid, 1)).Background = System.Windows.Media.Brushes.Blue;

eBorder.BorderBrush = System.Windows.Media.Brushes.Blue;

if (txtBox.Name.Equals("ServerBox"))

{

if (ServerBoxEmpty)

{

txtBox.Text = "";

}

} else if (txtBox.Name.Equals("LoginBox")) {

if (LoginBoxEmpty)

{

txtBox.Text = "";

}

}

}

private void LoginTextBoxFocusLoose(object sender, RoutedEventArgs e)

{

TextBox txtBox = (TextBox)sender;

Grid eGrid = (Grid)VisualTreeHelper.GetParent(txtBox);

Border eBorder = (Border)VisualTreeHelper.GetParent(eGrid);

eBorder.Background = new SolidColorBrush(Colors.White) { Opacity = 0 };

((GridSplitter)VisualTreeHelper.GetChild(eGrid, 1)).Background = System.Windows.Media.Brushes.Black;

eBorder.BorderBrush = System.Windows.Media.Brushes.Black;

if(txtBox.Text == "") {

if (txtBox.Name.Equals("ServerBox")) {

ServerBoxEmpty = true;

} else if (txtBox.Name.Equals("LoginBox")) {

LoginBoxEmpty = true;

}

}

else

{

if (txtBox.Name.Equals("ServerBox"))

{

ServerBoxEmpty = false;

}

else if (txtBox.Name.Equals("LoginBox"))

{

LoginBoxEmpty = false;

}

}

}

private void LoginPasswordBoxFocus(object sender, RoutedEventArgs e)

{

PasswordBox pswBox = (PasswordBox)sender;

Grid eGrid = (Grid)VisualTreeHelper.GetParent(pswBox);

Border eBorder = (Border)VisualTreeHelper.GetParent(eGrid);

eBorder.Background = new SolidColorBrush(Colors.Blue) { Opacity = 0.1 };

((GridSplitter)VisualTreeHelper.GetChild(eGrid, 1)).Background = System.Windows.Media.Brushes.Blue;

eBorder.BorderBrush = System.Windows.Media.Brushes.Blue;

}

private void LoginPasswordBoxFocusLoose(object sender, RoutedEventArgs e)

{

PasswordBox pswBox = (PasswordBox)sender;

Grid eGrid = (Grid)VisualTreeHelper.GetParent(pswBox);

Border eBorder = (Border)VisualTreeHelper.GetParent(eGrid);

eBorder.Background = new SolidColorBrush(Colors.White) { Opacity = 0 };

((GridSplitter)VisualTreeHelper.GetChild(eGrid, 1)).Background = System.Windows.Media.Brushes.Black;

eBorder.BorderBrush = System.Windows.Media.Brushes.Black;

}

private void StartAuth(object sender, RoutedEventArgs e)

{

if (RememberMeCheckbox.IsChecked.Value)

{

currentData = new RememberMeClass() { strUserLogin = EncryptString(LoginBox.Text, "PZSISFB97"), strUserPassword = EncryptString(PassBox.Password, "PZSISFB97"), strUserServer = ServerBox.Text, bRememberMeChecked = RememberMeCheckbox.IsChecked.Value };

}

else

{

currentData = new RememberMeClass() { strUserLogin = "", strUserPassword = "", strUserServer = ServerBox.Text, bRememberMeChecked = RememberMeCheckbox.IsChecked.Value };

}

using (Stream stream = File.Open(StartingFilePath, FileMode.Create))

{

var bformatter = new System.Runtime.Serialization.Formatters.Binary.BinaryFormatter();

bformatter.Serialize(stream, currentData);

}

((Button)sender).IsEnabled = false;

MainGrid.Effect = new System.Windows.Media.Effects.BlurEffect

{

Radius = 5

};

MainGrid.IsEnabled = false;

LoadingIcon.Visibility = Visibility.Visible;

LoginWindow.Height = 283.122f;

ErrorLabel.Visibility = Visibility.Hidden;

List<object> warguments = new List<object>();

warguments.Add(ServerBox.Text);

warguments.Add(LoginBox.Text);

warguments.Add(PassBox.Password);

KeepConnectionString = "SERVER=" + ServerBox.Text + ";UID=" + LoginBox.Text + ";" + "PASSWORD=" + PassBox.Password + ";";

if (worker.IsBusy != true)

{

worker.RunWorkerAsync(warguments);

}

}

//Шифровка

public static string EncryptString(string plainText, string passPhrase)

{

byte[] initVectorBytes = Encoding.UTF8.GetBytes(initEncrKey);

byte[] plainTextBytes = Encoding.UTF8.GetBytes(plainText);

PasswordDeriveBytes password = new PasswordDeriveBytes(passPhrase, null);

byte[] keyBytes = password.GetBytes(keysize / 8);

RijndaelManaged symmetricKey = new RijndaelManaged();

symmetricKey.Mode = CipherMode.CBC;

ICryptoTransform encryptor = symmetricKey.CreateEncryptor(keyBytes, initVectorBytes);

MemoryStream memoryStream = new MemoryStream();

CryptoStream cryptoStream = new CryptoStream(memoryStream, encryptor, CryptoStreamMode.Write);

cryptoStream.Write(plainTextBytes, 0, plainTextBytes.Length);

cryptoStream.FlushFinalBlock();

byte[] cipherTextBytes = memoryStream.ToArray();

memoryStream.Close();

cryptoStream.Close();

return Convert.ToBase64String(cipherTextBytes);

}

//Расшифровка

public static string DecryptString(string cipherText, string passPhrase)

{

byte[] initVectorBytes = Encoding.UTF8.GetBytes(initEncrKey);

byte[] cipherTextBytes = Convert.FromBase64String(cipherText);

PasswordDeriveBytes password = new PasswordDeriveBytes(passPhrase, null);

byte[] keyBytes = password.GetBytes(keysize / 8);

RijndaelManaged symmetricKey = new RijndaelManaged();

symmetricKey.Mode = CipherMode.CBC;

ICryptoTransform decryptor = symmetricKey.CreateDecryptor(keyBytes, initVectorBytes);

MemoryStream memoryStream = new MemoryStream(cipherTextBytes);

CryptoStream cryptoStream = new CryptoStream(memoryStream, decryptor, CryptoStreamMode.Read);

byte[] plainTextBytes = new byte[cipherTextBytes.Length];

int decryptedByteCount = cryptoStream.Read(plainTextBytes, 0, plainTextBytes.Length);

memoryStream.Close();

cryptoStream.Close();

return Encoding.UTF8.GetString(plainTextBytes, 0, decryptedByteCount);

}

private void LoginWindow\_Loaded(object sender, RoutedEventArgs e)

{

if (File.Exists(StartingFilePath))

{

using (Stream stream = File.Open(StartingFilePath, FileMode.Open))

{

var bformatter = new System.Runtime.Serialization.Formatters.Binary.BinaryFormatter();

currentData = (RememberMeClass)bformatter.Deserialize(stream);

}

ServerBox.Text = currentData.strUserServer;

ServerBoxEmpty = false;

if (currentData.strUserLogin.Length > 0)

{

LoginBox.Text = DecryptString(currentData.strUserLogin, "PZSISFB97");

LoginBoxEmpty = false;

}

if (currentData.strUserPassword.Length > 0)

{

PassBox.Password = DecryptString(currentData.strUserPassword, "PZSISFB97");

PasswordBoxEmpty = false;

}

RememberMeCheckbox.IsChecked = currentData.bRememberMeChecked;

}

else

{

currentData = new RememberMeClass() { strUserLogin = "", strUserPassword = "", strUserServer = "" };

using (Stream stream = File.Open(StartingFilePath, FileMode.Create))

{

var bformatter = new System.Runtime.Serialization.Formatters.Binary.BinaryFormatter();

bformatter.Serialize(stream, currentData);

}

}

}

}

}

1. ScheduleBaseClass.cs

Код, содержащий класс ScheduleBaseClass, предназначенный для хранения данных таблицы Расписание

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SheduleAuditorVisualisator.DBClasses

{

[Serializable]

public class ScheduleBaseClass

{

public int iID { get; set; }

public int iGroupID { get; set; }

public int iSubjectID { get; set; }

public DateTime dStartTime { get; set; }

public DateTime dEndTime { get; set; }

public int iClassRoomID { get; set; }

public int iTeacherID { get; set; }

}

}

Остальная большая часть программного кода находится на диске с исходным кодом дипломной работы.