**РОСЖЕЛДОР**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ПУТЕЙ СООБЩЕНИЯ (СГУПС)**

|  |  |  |  |
| --- | --- | --- | --- |
| **К защите:** |  |  | |
| **Заведующий кафедрой** | **Информационные** | |
| **технологии транспорта** | | |
|  | д-р техн. наук, проф. | |
|  |  | В. И. Хабаров | |
| *подпись* |  | *инициалы, фамилия* | |
|  |  |  | |
| *дата* |  |  | |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Тема:** | Разработка веб-приложения «Личный кабинет преподавателей онлайн-школы» | | | | | | |
|  | |  | | | | | |
|  | | |  | БР.БПИ.07.2022 |  |  |
|  | | |  | *шифр документа* |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Выполнил** |  |  |  | **Руководитель** |
|  |  | М. А. Кирьянов |  |  |  | канд. техн. наук, доц.  А. О. Коломеец |
| *подпись* |  | *инициалы, фамилия* |  | *подпись* |  | *инициалы, фамилия* |
|  |  |  |  |  |  |  |
| *дата* |  |  |  | *дата* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Консультанты по разделам** |  |  |  |  |
| 1 |  |  |  |  |
|  |  |  |  |  |
| 2 |  |  |  |  |
|  |  |  |  |  |
| 3 |  |  |  |  |
|  |  |  |  |  |
| Нормоконтролер работы |  |  |  | ст. преп.  Т. А. Распопина |
|  |  | *подпись* |  |  |
|  |  |  |  |  |
|  |  | *дата* |  |  |

**2022 г.**

**СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ПУТЕЙ СООБЩЕНИЯ (СГУПС)**

Факультет: Бизнес-информатики

Кафедра: Информационные технологии транспорта

Направление: 09.03.03 «Прикладная информатика»

Профиль: «Корпоративные информационные системы на транспорте»

|  |  |
| --- | --- |
|  | ***УТВЕРЖДАЮ****: зав. кафедрой «Информационные технологии транспорта»*  д-р техн. наук, проф.  В. И. Хабаров |
|  | *«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_ 2022 г.* |

**З А Д А Н И Е**

**на выполнение выпускной квалификационной работы**

|  |  |  |  |
| --- | --- | --- | --- |
| студенту | Кирьянову Михаилу Александровичу | | |
|  |  | | |
| 1. Тема Разработка веб-приложения «Личный кабинет преподавателей онлайн-школы» утверждена приказом № 203/с от «30» мая 2022 г. | | | |
| 2. Задание выдано «12» мая 2022 г. | | | |
| 3. Срок сдачи законченной работы на кафедру «20» июня 2022 г. | | | |
| 4. Исходные данные: данные, полученные в ходе прохождения преддипломной практики | | | |
| 5. Содержание расчетно-пояснительной записки | | | |
| Наименование разделов и вопросов | | Примерное количество страниц | График (сроки) выполнения |
| Введение | | 2 | 13.05.2022 |
| Анализ предметной области | | 11 | 13.05.2022 |
| Проектирование информационной системы | | 21 | 28.05.2022 |
| Руководство пользователя | | 5 | 01.06.2022 |
| Заключение | | 1 | 02.06.2022 |

6. Содержание и объемы графической части

|  |  |  |
| --- | --- | --- |
| Наименование графического документа | Количество  слайдов | График  (сроки)  выполнения |
| Презентация PowerPoint | 15 | 07.06.2022 |

7. Консультанты по разделам

|  |  |  |
| --- | --- | --- |
| Наименование  раздела | Фамилия, И. О.  консультанта | Подпись консультанта,  дата выдачи задания |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Руководитель |  | А. О. Коломеец |
|  | *(подпись руководителя)* |  |
| Задание к использованию принял |  | М. А. Кирьянов |
|  | *(подпись студента)* |  |

УДК 004.41

**АННОТАЦИЯ**

В работе 49 страниц, 29 рисунков, 8 таблиц, 11 источников.

Ключевые слова: *веб-приложение, автоматизация, онлайн-школы*.

Предметная область – учебное заведение, которое использует информационные технологии для проведения занятий и использует принцип дистанционного контроля за проведением уроков. Система работает с базой данных, откуда берет данные о назначенном преподавателе, группах и времени занятий. Веб-приложение работает на любых платформах.

**ABSTRACT**

The work contains 49 pages, 29 figures, 8 tables, 11 sources.

Keywords: *web application, automation, online schools.*

Subject area – an educational institution that uses information technology to conduct classes and uses the principle of remote control of the lessons. The system works with a database, from where it takes data about the assigned teacher, groups and class times. The web application works on any platform.

**ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

ИС – информационная система.

БД – база данных.

MS – Microsoft

SQL (Structured Query Language) – декларативный язык программирования, применяемый для управления данными в реляционной базе данных.

MSSQL – это программная библиотека, которая реализует автономный, транзакционный механизм базы данных SQL с нулевой конфигурацией.

HTTP – протокол прикладного уровня передачи данных.

API (Application Programming Interface или интерфейс программирования приложений) – это совокупность инструментов и функций в виде интерфейса для создания новых приложений, благодаря которому одна программа будет взаимодействовать с другой.

Система удаленного контроля работы сотрудников, Система управления расписанием, Онлайн Расписание – службы составления графиков работы.

Сайты-сервисы и интернет-сервисы – это синонимы веб-сервисов.

Онлайн сервисы – это сайты, которые предоставляют всевозможные услуги, что значительно облегчает работу и позволяет существенно сэкономить время.

Базы данных – это организованная структура, предназначенная для хранения, изменения и обработки взаимосвязанной информации, преимущественно больших объемов.

СОДЕРЖАНИЕ

[Введение 6](#_Toc106698219)

[1 Анализ предметной области 8](#_Toc106698220)

[1.1 Общее положение 8](#_Toc106698221)

[1.2 Описание проблемы и постановка задачи 10](#_Toc106698222)

[1.3 Обзор существующих систем составления расписания 11](#_Toc106698223)

[1.4 Обзор существующих систем отображения информации 16](#_Toc106698224)

[1.5 Выводы к разделу 17](#_Toc106698225)

[2 Проектирование информационной системы 20](#_Toc106698226)

[2.1 Описание бизнес-процессов 20](#_Toc106698227)

[2.2 Общая архитектура системы 26](#_Toc106698228)

[2.4 Проектирование базы данных 33](#_Toc106698229)

[2.5 Администрирование базы 40](#_Toc106698230)

[2.6 Выводы к разделу 42](#_Toc106698231)

[3 Руководство пользователя 43](#_Toc106698232)

[Заключение 48](#_Toc106698233)

[Список использованных источников 49](#_Toc106698234)

# **ВВЕДЕНИЕ**

Актуальность выполнения работы заключается в том, что в эпоху мировых ограничений в связи с эпидемической обстановкой дистанционное обучение является одним из перспективных направлений в образовании. Оно позволяет осуществлять подготовку кадров без отрыва от производства, получать образование за пределами своего города или региона. Для организации дистанционного обучения многие образовательные учреждения организуют учебные порталы. Они представляют собой ресурсы, которые содержат учебные материалы, задания для самостоятельного выполнения, тесты для самопроверки и множество других материалов. Этот факт сильно увеличивает спрос на преподавателей, работающих из дома. Многие компании в этом случае прибегают к почасовому найму сотрудников как самозанятых. Но исходя из этого компаниям необходим инструмент, который в автоматизированном формате сможет хранить информацию о свободных и занятых часах преподавателей.

Целью автоматизации является: снижение большого количество затрат по времени, трудовых и материальных ресурсов, ведение и контроль отчетов, поддержание данных в надежном состоянии.

Так такая тенденция только сейчас начинает активно развиваться, сфера образования не так сильно пополняется необходимыми сервисами, в отличии от сферы менеджмента, где уже создано очень много приложений для отслеживания работы сотрудников. Сейчас в сфере образования прибегают к «костыльным» методам учета работы преподавателей по типу использования таблиц в сервисе Google Sheets, являющейся онлайн таблицей (базой данных).

Целью данной ВКР является разработка информационной системы, которая должна посодействовать в упрощении отслеживания расписания и актуальной информации о текущих группах преподавателей, с помощью хранения и изменения данных для всех видов занятий, которые проводятся в онлайн-школах с помощью веб-сервиса, на основании представленной информации о времени проведении занятий. Эта система будет работать в рамках определенной компании и выполнять функции площадки для отслеживания занятий и просмотра актуального свободного времени преподавателей.

Для достижения поставленной цели необходимо выполнить ряд основных задач:

* изучить предметную область;
* сделать анализ рынка по схожей тематике;
* построить модели бизнес процессов предметной области;
* выбрать и обосновать инструменты разработки приложения;
* разработать информационную систему;
* настроить и протестировать информационную систему;
* разработать руководство пользователя.

Практическая значимость работы состоит в возможности внедрения результатов исследования в деятельность онлайн-школ в целях подготовки и реализации мер, направленных на совершенствование информационной системы.

**1 Анализ предметной области**

**1.1 Общее положение**

Объектом автоматизации является просмотр списка актуальной информации по образовательным группам и свободного времени преподавателей. Основной целью онлайн-школ является обеспечение проведение образовательной деятельности для учеников любого возраста.

Учебное заведение имеет отдел продаж, которое открывает набор в группы на определенное время исходя из предпочтений клиентов. После того, как группа полностью наберет учеников на обучение, группа будет распределяться на преподавателя. Информация о группе передается в учебную часть онлайн-школы, которая обязана подобрать для группы преподавателя и обеспечить проведение качественных. Количество свободного времени преподавателей контролирует учебная часть и предоставляет в отдел продаж во избежание случаев нехватки преподавателей на группы.

Обучение в школе проходит в соответствии с выбранным учеником курсов. Количество уроков в курсе определяется на этапе проектирования курсов, данные о продолжительности обучение могут менять в случае отмены уроков или переносов занятий.

Обучение одним преподавателем группы на протяжении всего курса не гарантировано, школа оставляет за собой право сменять его по определенным ей причинам.

У школы имеет свой набор форматов обучения. Обучение может проходить в индивидуальном формате, то есть занятия 1 на 1 с учеником. Также могут быть разные вариации групповых занятий с разным количеством учеников. Время проведения занятий также может меняться от формата к формату зависимости от установленный правил проведения занятий. Возможность ведение занятий преподавателей по определенному формату определяет школа и действующие в ней регламенты работы преподавателей.

Такое разнообразие форматов обучение обусловлено коммерческой выгодой.

Все устанавливается исходя из желаний клиентов. Чем шире будут возможности для клиента, тем дольше у него будет желание обучаться в данной онлайн-школе.

Основными задачами онлайн-школы являются:

* обеспечение высокой конкурентоспособной позиции в отечественной образовательной и научной среде в области развития навыков по различным;
* ориентация на текущие потребности целевой аудитории с учетом мировых тенденций;
* повышение качества учебного процесса на основе внедрения инновационных методов обучения и современных дистанционных, телекоммуникационных и компьютерных технологий в образовательный процесс;
* обеспечение высокого уровня востребованности компетенций на рынке труда выпускников онлайн-школы.

Для обеспечения поставленных задач, факультет осуществляет следующие функции:

* разработать и реализовать планы по привлечению учеников на образовательные курсы;
* провести анализ перечня направлений и курсов обучения, разработанных школой;
* разработать и реализовать основные образовательные программы, направленные на получение прикладных навыков;
* использовать индивидуальный адресный подход в организации работы с учениками.

**1.2 Описание проблемы и постановка задачи**

Данная выпускная квалификационная работа направлена на создание прототипа информационной системы для онлайн-школы, для того чтобы упростить процесс просмотра расписания преподавателей и их доступного времени работы. Актуальность данной проблемы поясняется тем, что в настоящее время, сфера образования не так сильно пополняется необходимыми сервисами, в отличие от сферы менеджмента, где уже создано очень много приложений для отслеживания работы сотрудников. Сейчас используются сервисы онлайн-таблиц, которые представляют собой записи в базе данных, то есть процесс просмотра по ним полностью ручным трудом. Пример отображения листа с расписанием представлен на рисунке 1.1.

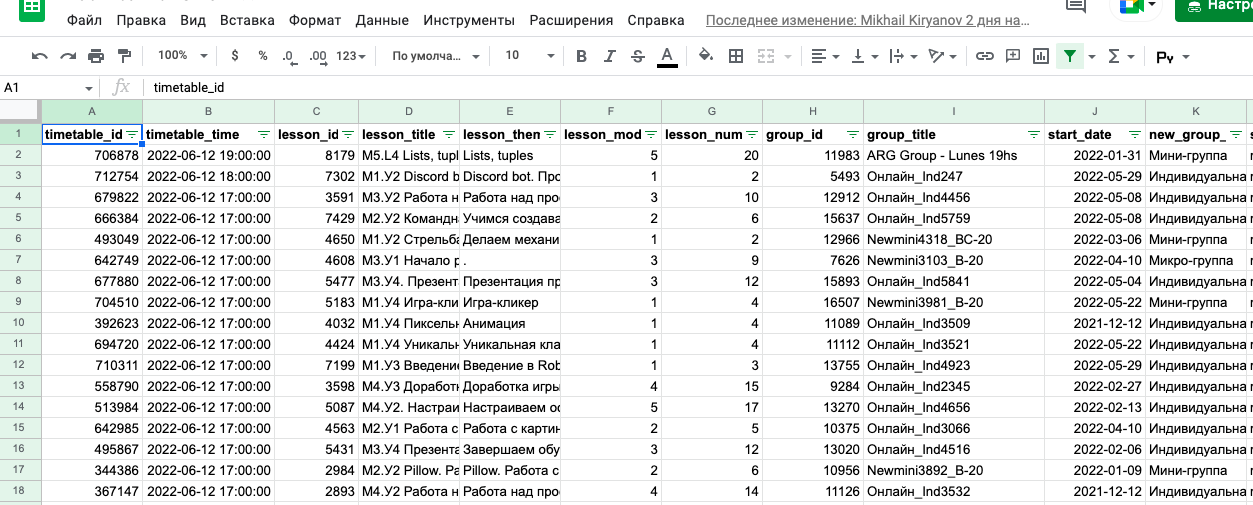


Рисунок 1.1 – Пример расписания

Если рассматривать с точки зрения затраченного времени, то этот способ является не рациональным, так как каждый раз, когда нужно узнать расписание или уточнить, в какой группе должно идти занятие, необходимо искать группу или предмет в огромной таблице, что может занять продолжительное количество времени.

Рассмотрим алгоритм просмотра расписания администратором или преподавателем:

1. Администратору или преподавателю, следует потратить время, для того чтобы зайти на сайт, найти нужную группу и там найти пункт с временем.
2. После чего необходимо подобрать преподавателя, данные по которому хранятся в формате таких же таблиц.

Данный алгоритм не комфортен для постоянного использования в качестве просмотра занятий и информации о группах, также сам файл может потеряться в огромном количестве таблиц, либо в само расписание могут внести правки, и если их не запомнить, то администратор или преподаватель может путаться.

Следовательно, можно сделать вывод о необходимости приложения, в котором будут выполнены следующие фрагменты:

* хранения информации о закрытых и открытых слотах;
* хранение информации о текущих группах;
* подбор преподавателя исходя из его открытых слотов и изученных курсах;
* изменение доступных слотов преподавателей.

**1.3 Обзор существующих систем составления расписания**

Сегодня проблема хранения информации о группах и преподавателях в онлайн-школах актуальна как никогда. Эта потребность появилась в связи с эпидемиологической обстановкой – дистанционное обучение стало одним из перспективных направлений в образовании. Справиться с такой нагрузкой администраторам становится всё сложнее. Естественны попытки переложить решение этой проблемы на плечи компьютера, что привело к созданию специальных программ для составления расписания занятий. Был проведен обзор основных программ, используемых для составления хранилища расписания, рассмотрена спецификация каждой из них.

Эти программы можно разделить по их назначению. Одни предназначены для составления только школьного расписания, другие – для создания расписания в вузе, третьи совмещают в себе обе спецификации, предоставляя пользователю возможность составить расписание с учётом особенностей конкретного учебного заведения.

Популярными сейчас программами для составления расписания в онлайн-школах в настоящее время являются: «МойКласс», «Rubitime», «Арника».

В ходе написания работы был произведен анализ основных программ для составления расписания, который производился по следующим критериям:

* выполнение задач хранение информации о расписании;
* необходимость закупки дополнительных программ;
* платформа;
* стоимость лицензии на один компьютер;
* удобство и простота интерфейса.

По результатам проведённого анализа можно сделать следующие выводы. Программа «Rubitime» не отвечает всем критериям, так как жестко ориентирование на пользование только со стороны администратора и не предоставляет возможность пользования преподавателем.

Система «МойКласс» не является достаточно подходящей, так как её основная область применения – это расписание кружков в школах и детских садах, она создана просто как общая доска со временем проведения мероприятий. К недостаткам программы «Арника» можно отнести проблематичность настройки и достаточно высокую стоимость. Выбирая из представленного списка программ, выделим наиболее подходящую с учетом недостатков.

Программа «Rubitime» для составления расписания занятий по результатам анализа занимает лидирующую позицию.

Эта программа обладает следующими преимуществами:

* удобный и понятный даже неопытному пользователю интерфейс;
* планирование распределения нагрузок по неделям (причём расписание по разным неделям может быть различным);
* расписание занятий составляется без «окон»;
* в графике работы преподавателя можно запретить какое-либо время;
* возможность интеграции с различными дополнительными сервисами, включая системы CRM;
* автоматический и ручной режимы составления расписания занятий.

К сожалению, программа не очень удобна в использовании для преподавателей, так как не имеет доступов только на просмотр информации. Это делает небезопасным выдачу доступов для необученных с ней работать сотрудникам и несет в себе риски поломки, утери информации.

Остановимся на работе программы «Rubitime» более подробно. Она состоит из множества разделов. Некоторые из них: «Журнал записей», «Клиенты», «Статистика» и «Расчет зарплат». Раздел «Журнал записей» служит для просмотра всех занятий на каждое время, как показано на рисунке 1.2.

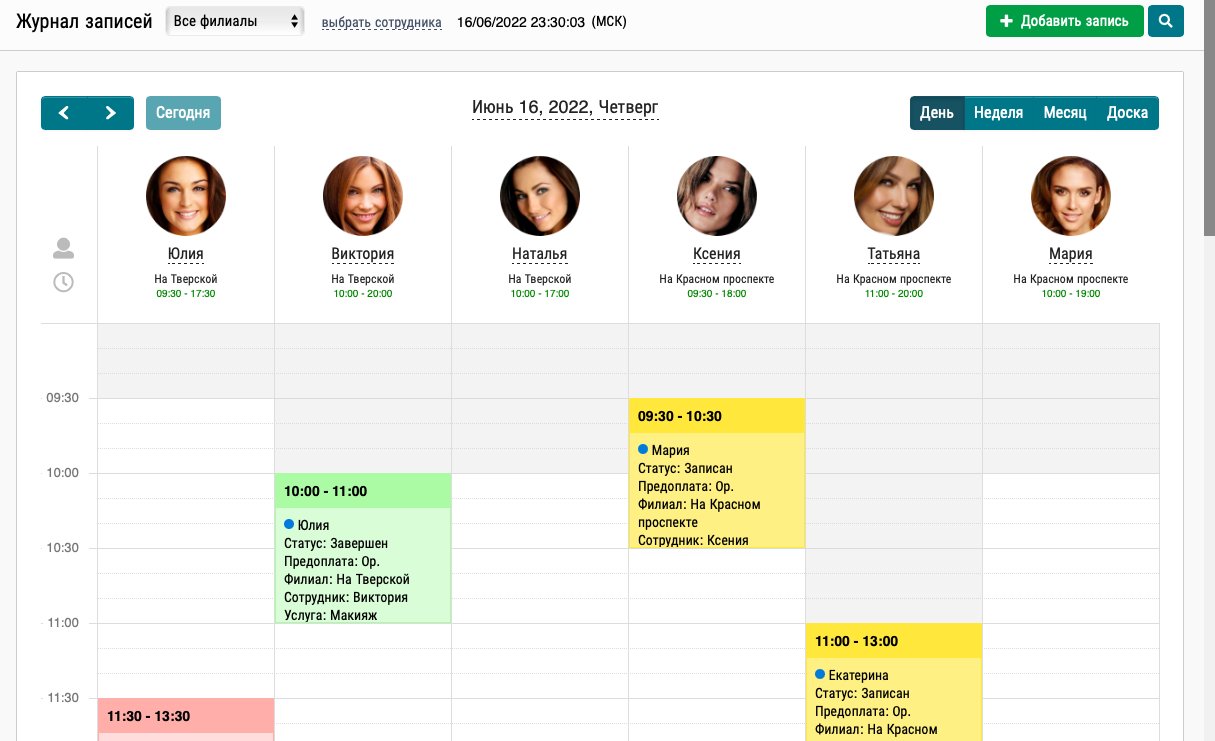


Рисунок 1.2 – Интерфейс программы

Также в пользование клиентам предоставлен раздел «Статистика». Он используется для вывода статистики по все проводимым занятиям, записям клиентов, продажам и по аналогии далее. Страница предлагает возможность проводить анализ по заданным параметрам, поэтому будет полезна для разбора бизнес-процессов и предиктивной аналитики.

Страница представлена на рисунке 1.3.

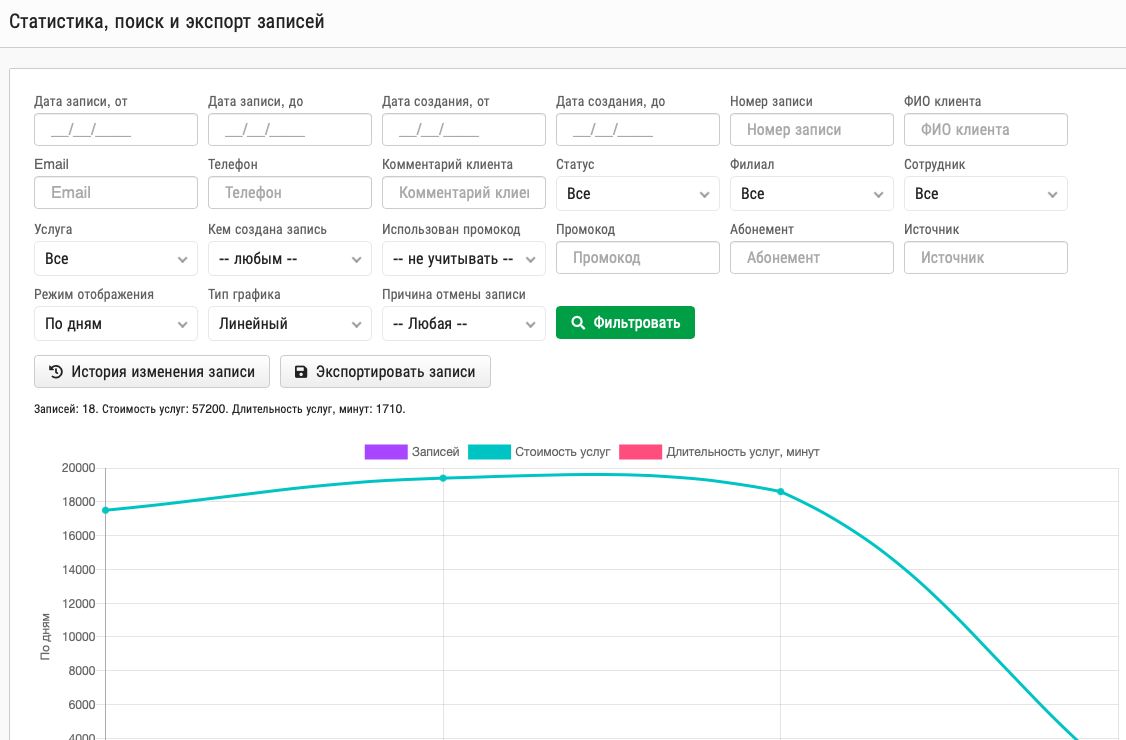


Рисунок 1.3 – Интерфейс раздела «Статистика»

Раздел «Клиенты» предназначен для отображения списка всех учеников и информации по ним. Там можно найти детальную информацию по каждому. Это будет крайне полезно для преподавателей, таким образом они, например могу добавить их групповые чаты и налаживать эффективную коммуникацию через чат.

Интерфейс страницы представлен на рисунке 1.4.

Расписание занятий можно распределять в автоматизированном, ручном или смешанном режимах. При составлении расписания в автоматизированном режиме программа учитывает все сформулированные требования к расписанию. При составлении расписания в ручном режиме программа подсказывает возможные варианты расстановки уроков выбранной группы, возможные варианты заполнения пустых клеток в расписании преподавателей, следит за количеством мест в расписании по курсам.

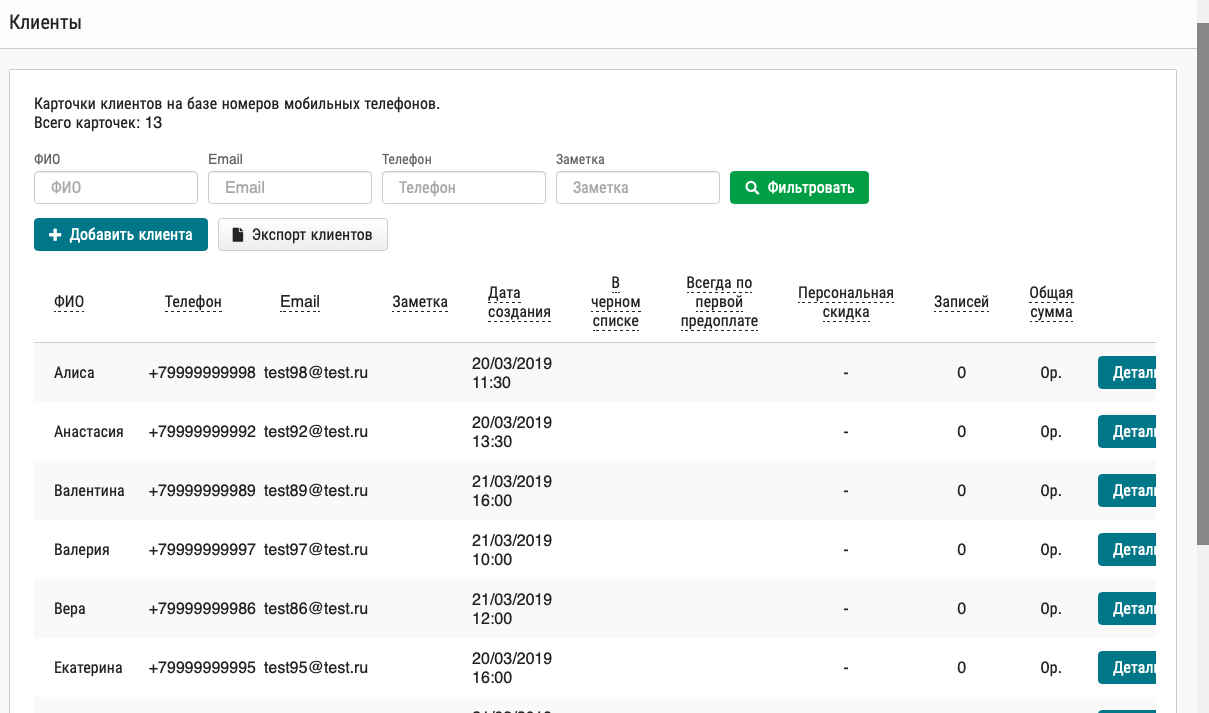


Рисунок 1.4 – Интерфейс раздела «Клиенты»

Готовое расписание занятий одной, всех или нескольких групп и преподавателей можно сохранить во всех популярных форматах по необходимости. Программа «Rubitime» работает на любом компьютере с установленным браузером с поддержкой современных протоколов. Распространение программы платное, однако демоверсия программы «Rubitime», предоставляя возможность посмотреть, как будет выглядеть распределение клиентов в ней. Лицензия стоит на сегодняшний день от 336 рублей в месяц до 1715 рублей в месяц в зависимости от количества сотрудников – в планах расписано разделение только до 50 сотрудников.

Безусловно, программа «Rubitime» не может полностью самостоятельно составить расписание в онлайн-школе, ведь необходимо учитывать огромное количество различных нештатных ситуаций. Однако можно сказать с уверенностью, что программа «Rubitime» существенно облегчает процесс составления расписания, экономит время, а также следит за корректным составлением расписания, не допуская ошибок, по причине человеческого фактора (усталость).

**1.4 Обзор существующих систем отображения информации**

Примерами решений в области упрощения просмотра расписания является программное обеспечение компании «Google».

«Google sheets» – является приложением, которое направленно в первую очередь отображение большого количества информации. Она используется как своеобразная база данных с готовым интерфейсом и возможностями использования встроенных функций.

Главной функцией этого приложения – это отображение информации в исходном виде. Его основной проблемой является экспоненциальный рост нагрузки на компьютер пользователя. С увеличением групп пользованием этим приложением является затруднительной задачей.

Интерфейс приложения представлен на рисунках 1.5-1.6.

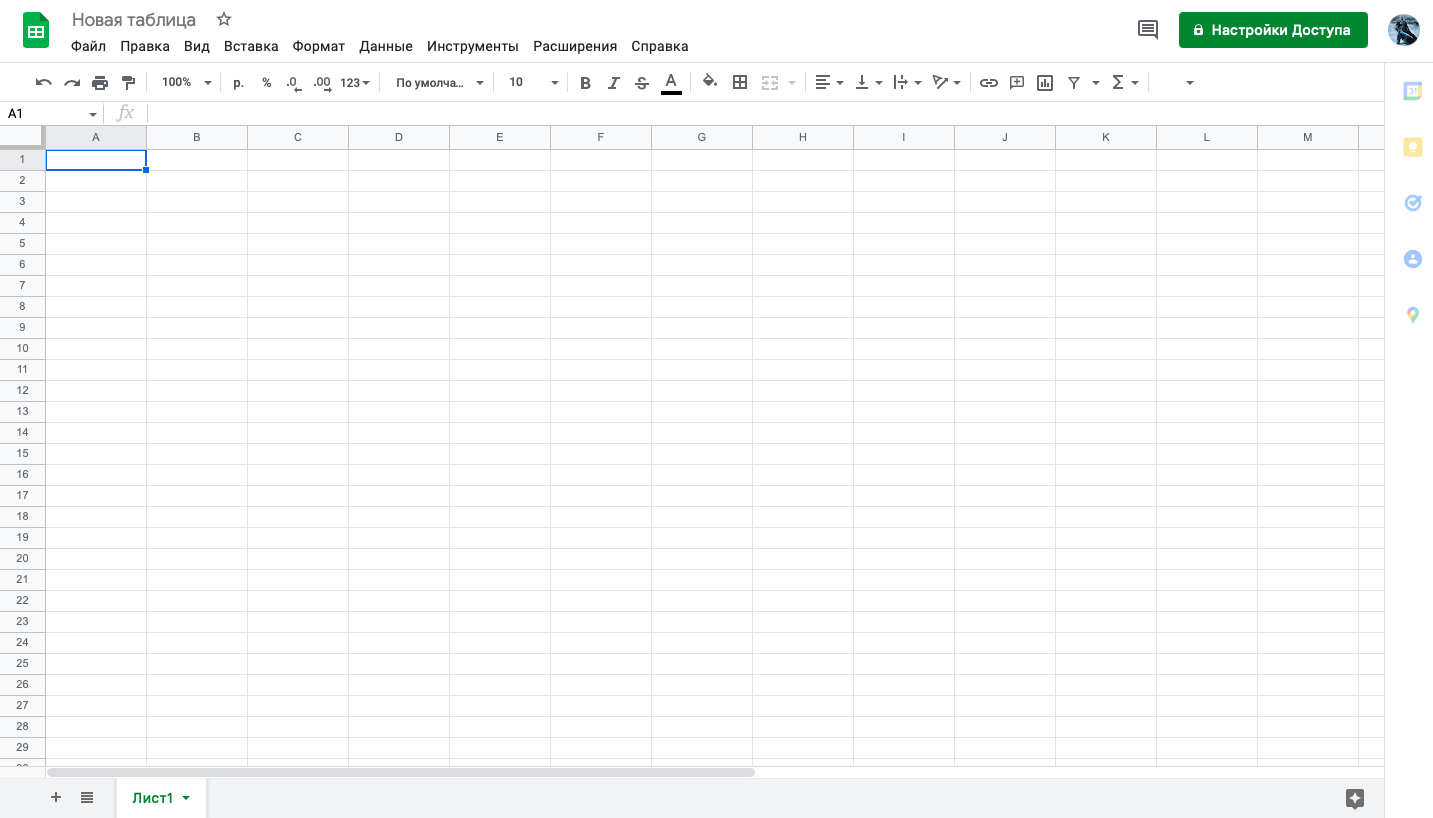


Рисунок 1.5 – Интерфейс онлайн версии

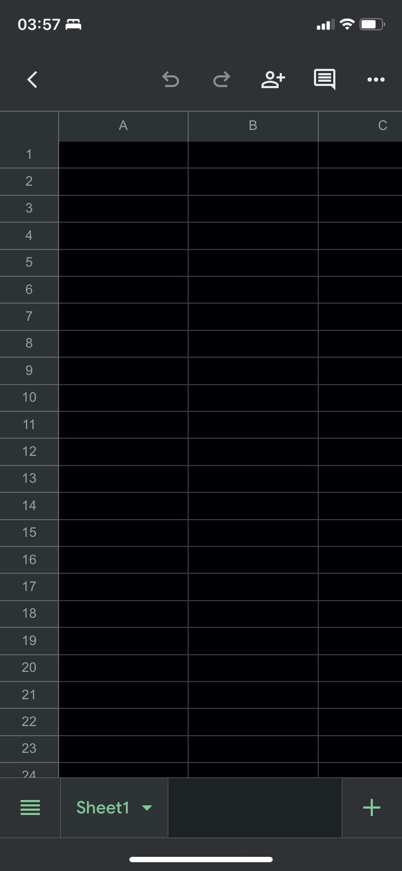


Рисунок 1.6 – Интерфейс мобильного приложения

**1.5 Выводы к разделу**

В первом разделе данной выпускной квалификационной работе:

* проведен анализ предметной области;
* выявлены функции, требующие автоматизации посредством реализации веб-приложения «Личный кабинет преподавателей онлайн-школы»;
* проведен краткий обзор существующих систем хранения информации.

В результате анализа аналогов, было принято решение разработать собственный программный продукт, так как закрытый исходный код большинства готовых продуктов не дает возможности самостоятельной модификации и расширения функционала системы, что может привести к дополнительным финансовым и временным затратам в процессе внедрения обновлений. Такой проект будет разделен на две части – это проектирование и разработка базы данных и разработка веб-сервиса. Наличие недостатков объясняет необходимость разработки собственного приложения.

К преимуществам создания собственной системы можно отнести:

* открытый исходный код позволит проводить модификацию, расширение и усложнение системы в зависимости от потребности заказчиков;
* кроссплатформенность, доступ через браузер с устройства на любой современной платформе;
* возможность добавление информации о преподавателе.

Веб-приложение с расписанием для администраторов и преподавателей – сервис для электронных устройств, предназначенный для просмотра и редактирования расписания занятий для администраторов и преподавателей, включает в себя следующие функциональные возможности:

– просмотр расписания;

– выбор группы;

– администрирование любой информации на сайте;

– редактирование расписания со стороны администратора;

– отображение информации по группам.

Основными целями разработки системы являются:

– упрощение взаимодействия с расписанием для администрации школы и преподавателей;

– повышение удобства информирования преподавателей о расписании.

Основная функция системы – обеспечение процесса дистанционного просмотра расписания для пользователя, с использованием приложения.

Система должна обеспечивать выполнение перечисленных ниже функций (функциональные характеристики):

* круглосуточная доступность и работоспособность системы;
* возможность хранения расписания и его отображения;
* возможность отображения информации для пользователей.

К нефункциональным требованиям относятся следующие пункты.

Требования к надежности:

* скорость восстановления работоспособности системы при сбое должна составлять не более 30 минут;
* должен быть разработан механизм и регламент полного резервного копирования системы;
* максимальное время остановки системы для проведения технологических операций не более 2 часа в сутки.

Требования к объему хранимой информации:

* система должна хранить всю информацию о расписании групп активных пользователей в течение актуальности расписания.

Требования к производительности:

* количество одновременно работающих пользователей: до 1000 человек;
* время подготовки к работе с момента запуска приложения до запуска отображения расписания не должно превышать 5 секунд;
* время отклика системы на действия пользователя не должно превышать 2 секунд;
* максимальная продолжительность вычислений должна быть не более 15 секунд.

Требования к численности и квалификации персонала.

Для эксплуатации системы определены следующие роли:

* пользователь;
* модератор;
* системный администратор.

Минимальные системные требования к клиентской части:

* браузер с поддержкой TLS 1.2;
* сеть: 5 Мбит/сек;
* оперативная память: 4 ГБ.

**2 Проектирование информационной системы**

Проектирование информационных систем – это упорядоченная совокупность методологий и средств создания или модернизации информационных систем.

Перед непосредственной разработкой программного продукта необходимо спроектировать каждый этап, предусмотреть требуемую функциональность системы и степень адаптации к изменяющимся условиям ее использования .

Данная информационная система будет представлять из себя связку двух систем: веб-приложения и связанной с ним базой данных. Таким образом рассматривать также нужно два программных продукта.

## 2.1 Описание бизнес-процессов

Моделирование бизнес-процессов является одним из методов повышения эффективности и прозрачности работы организации. В его основе лежит процессный подход к управлению: процессы описываются через присущие им элементы – действия, данные, события, материалы. Полученное описание позволяет глубоко разобраться в бизнес-процессах, увидеть потенциал их улучшения и эффективно организовать взаимодействие всех участников [1].

Моделирование бизнес-процессов позволяет разобрать рабочие алгоритмы и провести эффективный анализ организации в целом. Это становится достижимым за счет того, что модели могут быть декомпозированы к более подробным и наоборот. В больших компаниях контроль бизнес-процессов выполняется более подробно чем в малых, это связано с большим количеством кросс-функциональных связей.

Сейчас для моделирования бизнес-процессов всегда применяются различные компьютерные сервисы и программные средства. Это облегчает управление моделями, отслеживание в них изменений и позволяет уменьшить затраты на аналитику.

Моделирование бизнес-процессов предполагает конкретные цели:

* это описания процессов, которое можно проследить, от начала, до завершения;
* моделирование бизнес-процессов задает правила выполнения процессов, то есть то, каким образом они должны быть выполнены. Если следовать установленным в моделях правилам, руководящим указаниям или требованиям, то можно достичь желаемой производительности процессов;
* установление взаимосвязей в процессах для выстраивания связи между процессами и требованиями, которые они должны выполнять;
* моделирование бизнес-процессов, как правило, включает в себя выполнение нескольких последовательных стадий [2].

Состав стадий, которые включает в себя моделирование бизнес-процессов, следующий:

* выявление процессов и построение исходной модели «as is». Занимаясь улучшением процесса, всегда нужно, в первую очередь, понимать, как он работает в данный момент. На этой стадии определяются границы процесса, ключевые элементы, порядок работы процесса. В результате создается исходная модель процесса «как есть». Эта модель не всегда адекватно может отражать порядок процесса, поэтому модель этой стадии можно назвать исходной моделью «as is»;
* уточнение исходной модели. На этой стадии выявляются противоречия и повторения действий в процессе, выделяются ограничивающие факторы для процесса, взаимосвязи процесса, устанавливается необходимость изменения процесса. В результате формируется окончательный вариант модели «as is»;
* разработка модели «as must be». После анализа текущей ситуации, необходимо определиться с желаемым состоянием процесса. Это желаемое состояние представляется в модели «as must be». Такая модель показывает, как процесс должен выглядеть в будущем, включая все необходимые улучшения. В ходе этой стадии моделирования бизнес процессов и разрабатываются такие модели;
* тестирование и применение модели «as must be». Эта стадия моделирования связана с внедрением разработанной модели в практику деятельности организации. Модель бизнес-процесса проходит апробацию, и в нее вносятся необходимые изменения;
* улучшение модели «as must be». Моделирование бизнес-процессов не заканчивается на создании модели «as must be». Каждый из процессов в ходе работы продолжает преображаться и модифицироваться, поэтому модели процессов должны регулярно просматриваться и анализироваться. Эта стадия моделирования связана с постоянным улучшением процессов и улучшением модели бизнес-процессов.

Диаграмма вариантов использования.

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

Варианты использования обычно применяются для спецификации внешних требований к системе.

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

При моделировании системы с помощью диаграммы прецедентов системный аналитик стремится:

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

Диаграмма вариантов использования состоит из актеров, для которых система производит действие и собственно действия UseCase, которое описывает то, что актер хочет получить от системы. Актер обозначается значком человечка, а UseCase – овалом.

На рисунках 2.1-2.3 представлены диаграммы вариантов использования для администратора, пользователя и модератора, диаграмма функции пользователя.

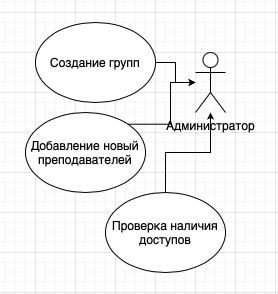


Рисунок 2.1 – Функции администратора

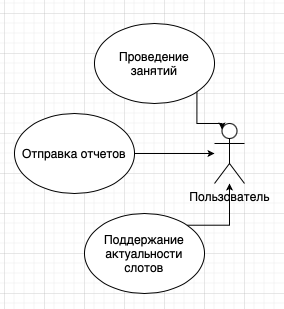


Рисунок 2.2 – Функции пользователя



Рисунок 2.3 – Функции модератора

Диаграмма деятельности.

Диаграмма деятельности – еще одна важная диаграмма в UML, описывающая динамические аспекты системы.

Диаграмма деятельность – это, по сути, блок-схема, представляющая поток от одного действия к другому. Деятельность может быть описана как работа системы [3].

Поток управления передается от одной операции к другой. Этот поток может быть последовательным, разветвленным или параллельным. Диаграммы действий касаются всех типов управления потоком с использованием различных элементов, таких как fork, join, и многие другие.

Деятельность – это особая операция системы. Диаграммы действий используются не только для визуализации динамической природы системы, но они также используются для построения исполняемой системы с использование прямого и обратного проектирования. Единственная недостающая вещь на диаграмме активности – это часть сообщения. Он не показывает поток сообщений от одного действия к другому. Диаграмма деятельности иногда рассматривается как блок-схема.

Хотя диаграммы выглядят как блок-схема, это не так. Он показывает разные потоки, такие как параллельный, разветвленный, параллельный и одиночный.

Диаграмма активности подходит для моделирования потока активности системы. Приложение может иметь несколько систем. Диаграмма деятельности также охватывает эти системы и описывает поток от одной системы к другой.

Это конкретное использование не доступно на других диаграммах. Этими системами могут быть базы данных, внешние очереди или любая другая система.

На рисунке 2.4 представлена диаграмма деятельности, на которой представлен процесс предоставления образовательной деятельности в онлайн-школе.

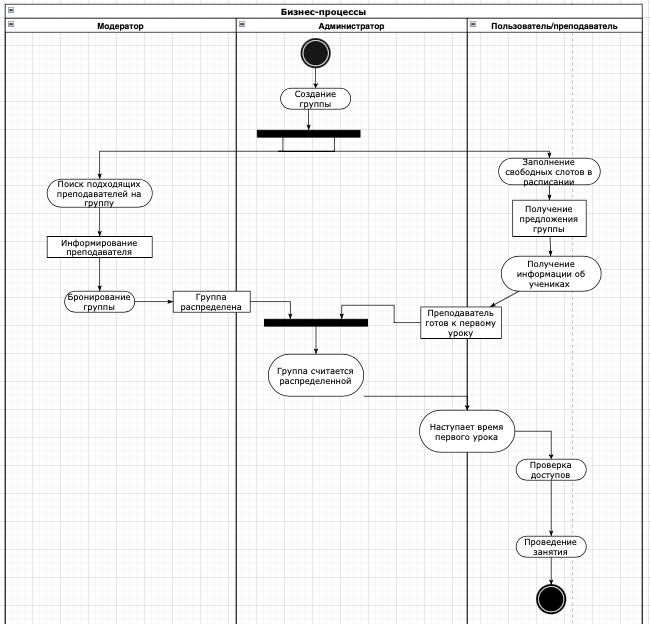


Рисунок 2.4 – Диаграмма деятельности «Предоставления образовательной деятельности»

В данном подразделе были описаны бизнес-процессы, которые будут учитываться при разработке прототипа веб-сервиса.

## 2.2 Общая архитектура системы

Система должна работать на основе технологии ASP.NET Core. ASP.NET это платформа, служащая для создания веб-сайтов и веб-приложений. Здесь вы можете писать на языках программирования, которые входят в комплект .NET Framework это: C#, Visual Basic NET, J# и другие. На основе этой технологии можно создавать сайты различной сложности и тематики.

ASP.NET намного быстрее, чем обычные скриптовые язык. Он выполняется быстрее, так как весь кол изначально компилируется при первом запросе пользователя и потом кешируется в памяти. Таким образом, все дальнейшие разы код просто выполняется из памяти и не требует повторной загрузки с сервера. Это сокращает время на парсинг, загрузку и оптимизацию файлов [3].

Современные веб-приложения строятся на основе микросервисной архитектуры – стиля, который структурирует приложение как набор сервисов. Каждый такой сервис максимально автономен, необходим для выполнения конкретной задачи и поддерживается конкретной командой. Эта архитектура позволяет применять модульный принцип построения приложений с учётом потребностей бизнеса. Другими словами, вы всегда можете поменять один или несколько сервисов, таких как веб-сервер или сервер обработки запросов, в случае изменения потребностей без ущерба для бизнеса. Графическое представление структуры работы веб-сервиса представлено на рисунке 2.6.

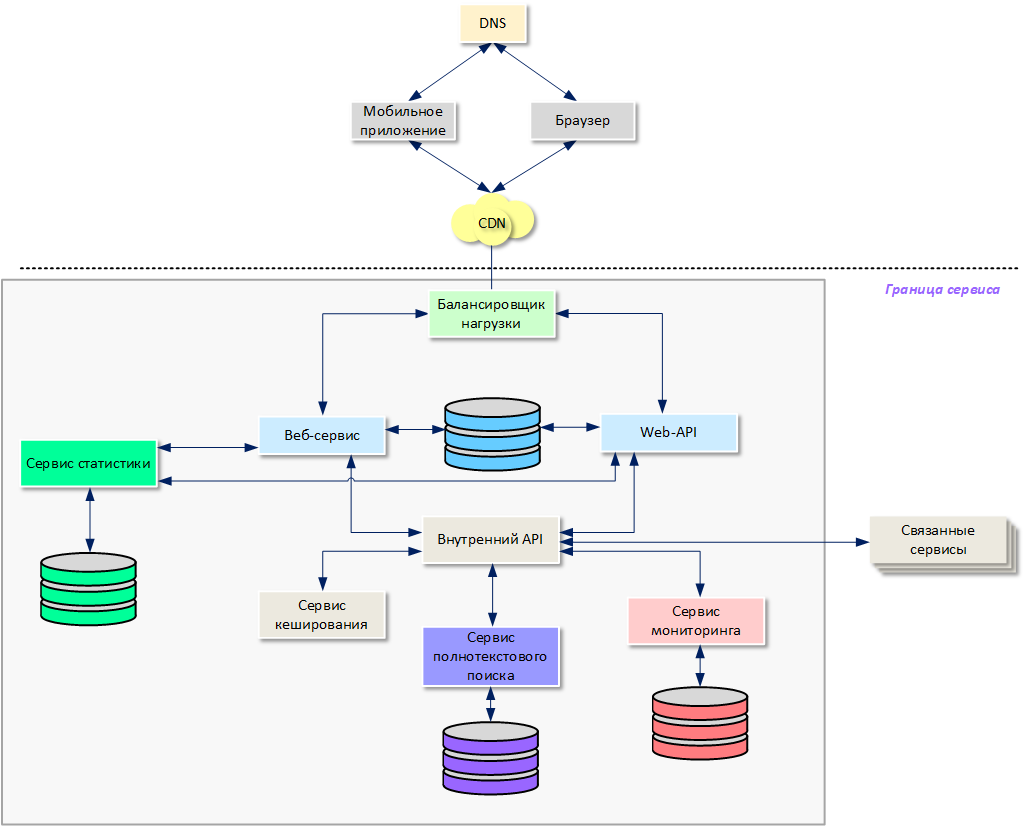


Рисунок 2.6 – Графическое представление архитектуры веб-сервисов

В первой версии протокола HTTP (HTTP/0.9), разработанной в 1991 году, был всего один метод взаимодействия клиента и сервера – метод «GET» с последующим путём к ресурсу, например «GET /mypage.html». Но по-настоящему всё это начало применяться с 1999 года, когда международной сетевой рабочей группой (International Network Working Group) была принята версия протокола HTTP/1.1, которая дорабатывается до сих пор. Стандарт устранял неясности и вносил значительные улучшения, которыми мы пользуемся и сейчас. Среди них – неоднократное использование соединений для сокращения времени на обработку новых запросов, конвейерная обработка, позволяющая отправлять второй запрос до того, как первый будет принят, дробление запросов. Разработаны дополнительные механизмы контроля кеша и согласования контента, добавлен заголовок «host», позволяющий размещать разные домены на одном IP-адресе.

Для того чтобы мобильное приложение функционировало ему нужно подключиться с указанием IP адресу сервера или доменного имени сервера. В прототипе нашего веб-приложения серверная часть реализована не будет по некоторым причинам. Были рассмотрены несколько вариантов бесплатных серверов для хостинга, такие как Hostinger и ONOS (1&1). Это одни из самых популярных хостинг-провайдеров, но из-за ситуации в мире они прекратили предоставлять услуги клиентам из России. Были рассмотрены Российские хостинг-провайдеры, но за их услуги нужно платить немалую сумму, так как для обслуживания серверов необходимы различные оборудования, аренды стоек для размещения оборудования в дата-центрах и прочие затраты. Конечно, у отечественных провайдеров есть бесплатные тарифы, но при поиске бесплатного хостинга важно понимать, что у таких тарифов нет стабильности в работе и есть определенные ограничения. Соответственно, для различных проектов настоятельно рекомендуют рассмотреть платные хостинги.

Архитектура прототипа веб-сервиса реализована как показано на рисунке 2.7.

Далее структура будет сильно расширяться.

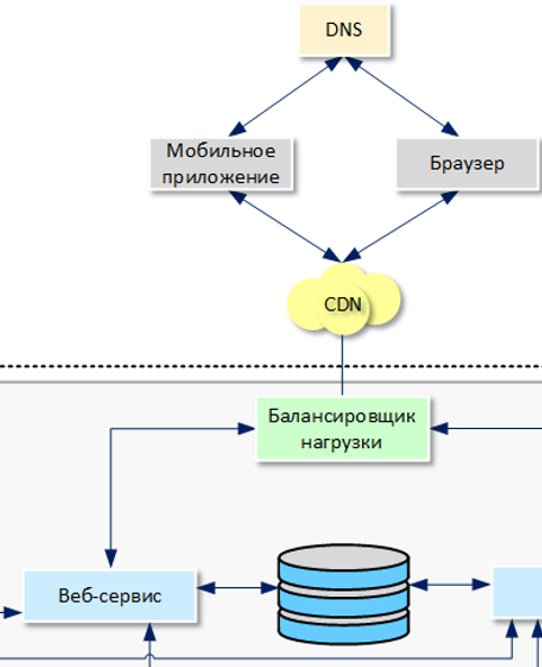


Рисунок 2.7 – Архитектура веб-приложения

Следующем важным шагом на пути к приложению, отвечающему всем современным требованиям, будет появление системы API. Она позволит нашему приложению быть расширением для существующих сервисов, а также может стать базой чей-то системы. Текущие тренды говорят от том, что данная система является для многих одним из решающих факторов при выборе систем подобного рода.

После внедрения данной системы структура разрабатываемого приложения приоритет вид, как показано на рисунке 2.8.

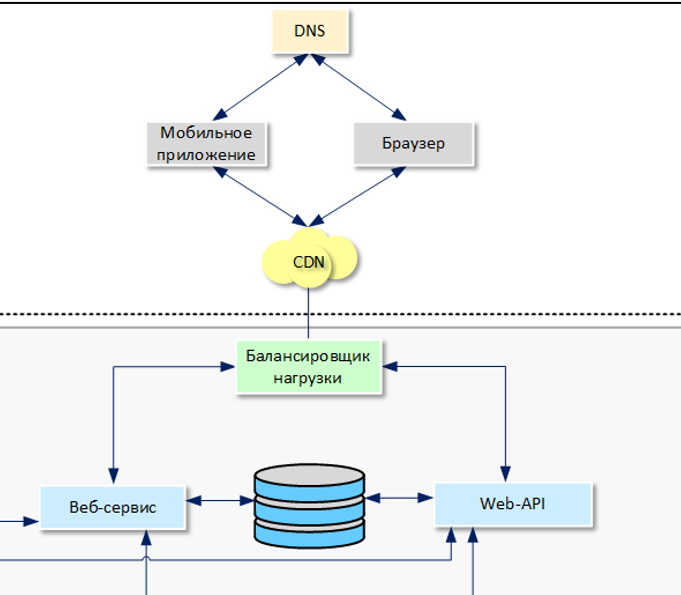


Рисунок 2.8 – Архитектура веб-приложения после внедрения API**2.3 Обоснование выбора базы данных**

Информацией, хранящейся в базе данных, может быть всё что угодно: каталог продукции, информация о клиентах, контент веб-сайта и другие категории. Для обеспечения доступа к информации, а также для управления ею, применяют систему управления базами данных.

Система управления базами данных – это комплекс языковых и программных средств, предназначенный для создания, ведения и совместного использования базы данных многими пользователями. Обычно системы управления базами данных различают по используемой модели данных.

Были выделены основные системы управления базами данных, использующиеся на предприятиях:

* MongoDB;
* SQLite;
* Microsoft SQL сервер.

Для сравнения и выбора нужной системы управления базами данных были выделены следующие критерии:

* стоимость;
* скорость работы;
* надежность;
* понятность документации;
* функционал.

MongoDB.

База данных, которая имеет коммерческую версию – MongoDB, она предназначена для приложений, которые используют как структурированные, так и неструктурированные данные. Ядро является очень гибким и работает при подключении базы данных к приложениям через драйверы MongoDB. Существует широкий выбор доступных драйверов, поэтому легко найти драйвер, который будет работать с требуемым языком программирования.

Поскольку изначально система MongoDB не была разработана для обработки моделей реляционных данных (хотя может это выполнять), могут возникнуть проблемы производительности, если вы попытаетесь использовать её таким образом. Однако, движок предназначен для обработки различных данных, которые нельзя отнести к реляционным, и может хорошо справляться там, где другие движки работают медленно или бессильны. Для хранения информации используется JSON подобный формат.

MongoDB 3.2 – это последняя версия, и она имеет новую подключаемую систему движков хранения. Документы могут быть проверены в процессе обновления или выполнения вставок, а функции текстового поиска были улучшены. Новая способность частичного индексирования может привести к более высокой производительности, уменьшая размер индексов.

SQLite.

Бесплатная компактная, легко встраиваемая в приложения база данных. Так как это система базируется на файлах, то она предоставляет довольно широкий набор инструментов для работы с ней, по сравнению с сетевыми СУБД. При работе с этой СУБД обращения происходят напрямую к файлам (в этих файлах хранятся данные), вместо портов и сокетов в сетевых СУБД. Именно поэтому SQLite очень быстрая, а также мощная благодаря технологиям обслуживающих библиотек [6].

Microsoft SQL сервер.

Ещё одной из популярных СУБД является программный продукт Microsoft SQL-сервер. Это система управления базами данных, движок которой работает на облачных серверах, а также локальных серверах, причем можно комбинировать типы применяемых серверов одновременно. Вскоре после выпуска Microsoft SQL сервер 2016, Microsoft адаптировала продукт для операционной системы Linux, а на Windows-платформе он работал изначально [4].

Последняя версия Microsoft SQL-сервер поддерживает dynamic data masking (динамическую маскировку данных), которая гарантирует, что только авторизованные пользователи будут видеть конфиденциальные данные.

Microsoft SQL отлично бы подходила к нашей информационной системе, но к сожалению она имеет высокую стоимость.

Обычно для работы СУБД, такой как MySQL, PostgreSQL и т. д., требуется отдельный серверный процесс. Приложения, которые хотят получить доступ к серверу базы данных, используют протокол TCP / IP для отправки и получения запросов. Это называется архитектурой клиент / сервер.

Рисунок 2.9 иллюстрирует архитектуру клиент / сервер СУБД.

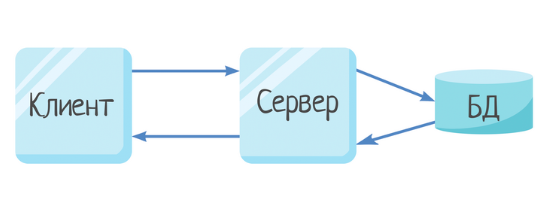


Рисунок 2.9 – Пример работы клиент-серверной архитектуры

Таким образом, для создания информационной системы была выбрана MSSQL Server, в связи с тем, что она легко интегрируется с технологией ASP.NET Core, так как разрабатывается одной компанией и имеет мощную поддержку от нее. При этом перед нами открывается возможность автоматического создана сущностей из кода. То есть применяется принцип «Code First». Сравнительный анализ выбранных систем представлен в таблице 2.1. Сравнение велось по таким параметрам как: интеграция, скорость работы, надежность, понятность документации и функционал.

Таблица 2.1 – Обзор базы данных по критериям

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Интеграция | Скорость работы | Надежность | Понятность документации | Функционал |
| MongoDB | сложна | высокая | высокая | Тяжела в освоении | обширный |
| SQLite | среднесложна | высокая | высокая | легка в освоении | обширный |
| Microsoft SQL | Адаптирована(легка) | высокая | высокая | легка в освоении | обширный |

## 2.4 Проектирование базы данных

База данных (БД) – это совокупность массивов и файлов данных, организованная по определённым правилам, предусматривающим стандартные принципы описания, хранения и обработки данных независимо от их вида.

SQL (англ. structured query language – «язык структурированных запросов») – декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных [7].

Изначально SQL был основным способом работы пользователя с базой данных и позволял выполнять следующий набор операций:

* создание в базе данных новой таблицы;
* добавление в таблицу новых записей;
* изменение записей;
* удаление записей;
* выборка записей из одной или нескольких таблиц (в соответствии с заданным условием);
* изменение структур таблиц.

Со временем SQL усложнился – обогатился новыми конструкциями, обеспечил возможность описания и управления новыми хранимыми объектами (например, индексы, представления, триггеры и хранимые процедуры) – и стал приобретать черты, свойственные языкам программирования.

При всех своих изменениях SQL остаётся самым распространённым лингвистическим средством для взаимодействия прикладного программного обеспечения с базами данных. В то же время современные СУБД, а также информационные системы, использующие СУБД, предоставляют пользователю развитые средства визуального построения запросов.

Проектирование базы данных – это процесс, который подразумевает использование определённой технологии при создании концептуальной схемы базы данных и определения ограничения целостности.

Основными задачами проектирования БД являются:

* обеспечение хранения в БД необходимой информации;
* обеспечение возможности получения данных по определенным запросам;
* сокращение избыточности и дублирования данных;
* обеспечение целостности БД.

*Сущность* – это объект, о котором в системе будут накапливаться данные. Для сущности указывается название и тип (сильная или слабая). Сильные сущности существуют сами по себе, а существование слабых сущностей зависит от существования сильных.

*Атрибут* – свойство сущности базы данных.

Ввод части данных в базу будет происходить через СУБД. Основная информация вводится через интерфейс сервиса, в будущем планируется настроить импорт данных из приложений-таблиц, так как самым частым решением является именно использование их. Наш сервис призван закрывать потребности компаний, которым необходимо хранить информацию о их образовательный процессах.

После ввода данных формируются таблицы в MSSQL Server. В общей сложности в БД будет восемь таблиц. Все, из которых это первичные таблицы, предназначенные для заполнения данных, а также таблица пользователей, которые будут заполняться администратором вручную и хранить информацию об уровне доступа каждого пользователя. Из-за того что у нас выбрано три роли на этапе проектирования бизнес-процессов, уровни доступа также распределяются на три вида. У роли администратора будет доступ ко всем, в том числе и распределении этих доступов, он будет только у единиц. Для администратора даже выделена отдельная функциональность в приложении, сервис выглядит иначе. У роли модератора будет дополнительные возможности по созданию групп и внесению в них изменений. Также будет роль обычного пользователя, которая выдается сразу после регистрации на платформе. К этому типу пользователей и будут относится преподаватели. Так как некоторые данные заполняются вручную, была предусмотрена проверка на корректность введенных данных. Эти таблицы необходимы для удобного первичного заполнения, а также редактирования. Далее будет представлена логическая модель предметной области.

В таблице 2.2 представлен набор сущностей и их атрибутов БД, которые были выделены в ходе анализа и моделирования бизнес-процессов предметной области.

Таблица 2.2 – Сущность и атрибуты сущности

|  |  |
| --- | --- |
| Сущность | Атрибут |
| User | Id |
| Surname |
| Name |
| Patronymic |
| RoleId |
| CompanyId |
| Group | Id |
| Group\_name |
| Group\_typeId |
| Teacher\_id |
| Time\_slot\_id |
| Course\_id |
| Current\_lesson\_id |

Продолжение таблицы 2.2

|  |  |
| --- | --- |
| Сущность | Атрибут |
| Roles | RoleId |
| Role\_name |
| Group\_type | Group\_typeId |
| Name |
| Timeslots | Time\_slot\_id |
| Timecode |
| Company | ID |
| Name |

Следующим шагом стало описание этих сущностей, которые представлены в таблицах 2.3–2.8.

Таблица 2.3 – Описание сущности «User»

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя полей | Тип данных | Характеристика |
| Id | Id | Int | Первичный ключ |
| Фамилия | Surname | Varchar | Обязательное поле |
| Содержание поля | Имя полей | Тип данных | Характеристика |
| Имя | Name | Varchar | Обязательное поле |
| Отчество | Patronymic | Varchar | Обязательное поле |
| Роль | RoleId | Int | Обязательное поле |
| Компания | CompanyId | Int | Обязательное поле |

Таблица 2.4 – Описание сущности «Group»

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя полей | Тип данных | Характеристика |
| Id | Id | Int | Первичный ключ |
| Название группы | Group\_name | Varchar | Обязательное поле |
| Тип группы | Group\_typeId | Varchar | Обязательное поле |
| Id преподавателя | Teacher\_id | Varchar | Обязательное поле |
| Временной слот | Time\_slot\_id | Int | Обязательное поле |

Таблица 2.5 – Описание сущности «Role»

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя полей | Тип данных | Характеристика |
| ID | RoleId | Int | Первичный ключ |
| Название роли | Role\_name | Varchar | Обязательное поле |

Таблица 2.6 – Описание сущности «Group\_type»

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя полей | Тип данных | Характеристика |
| ID | Group\_typeId | Int | Первичный ключ |
| Название типа группы | Name | Varchar | Обязательное поле |

Таблица 2.7 – Описание сущности «Timeslots»

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя полей | Тип данных | Характеристика |
| ID | Time\_slot\_id | Int | Первичный ключ |
| Время | Timecode | Varchar | Обязательное поле |

Таблица 2.8 – Описание сущности «Company»

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя полей | Тип данных | Характеристика |
| ID | ID | Int | Первичный ключ |
| Название компании | Name | Varchar | Обязательное поле |

Определение сходства.

Следующие правила в указанном порядке используются для определения типа столбца и сходства любого столбца:

* если объявленный тип столбца имеет строку «INT», то ему присваивается целочисленное сходство;
* если объявленный тип столбца содержит какие-либо строки типа «TEXT», «VARCHAR» или «string», то этот столбец имеет текстовое сходство;
* если объявленный тип столбца содержит строки типа «float» или «double», то столбец имеет реальное сходство.

В противном случае сходство является числовым [7].

Приведенный ниже рисунок 2.10 показывает, что общие типы данных из SQL преобразуются в группы по пяти правилам определения группами для небольшого набора типов данных, которые может принять MSSQL.

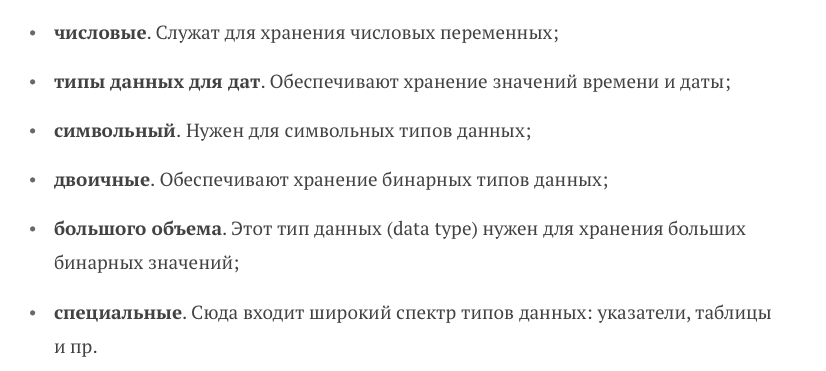


Рисунок 2.10 – Типы данных

ER – модель представляет собой высокоуровневую концептуальную модель данных, кото­рая была разработана с целью упрощения задачи проектирования структур баз данных.

В процессе разработки базы данных были исследованы различные способы реализации для упрощения ее заполнения, была предпринята попытка автоматизировать заполнение таблиц в MS SQL с помощью встроенных скриптов-генераторов случайных данных. Эти данные будут нужны для тестовых запусков и понимания адекватности разработанного программного обеспечения.

На рисунке 2.11 представлена ER-диаграмма разрабатываемого веб-сервиса.

Отношения между таблицами в данной версии не описаны четко по причине возможных изменений в структуре БД в будущем.

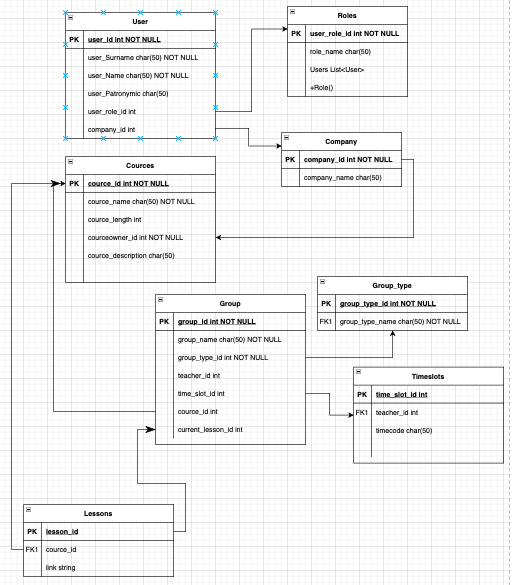


Рисунок 2.11 – ER-диаграмма

Наполним таблицы разработанной БД актуальной информацией, приведем в отчет пример наполнения. Для начала заполним таблицу пользователей. Первоначальную настройку проведем вручную. Далее таблица будет заполняться посредством регистрации пользователей на сайте-сервисе.

На рисунке 2.12 представлен пример заполнения таблицы «User».

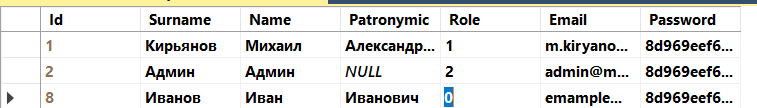


Рисунок 2.12 – Пример заполнения таблицы «User»

На рисунке 2.13 представлен пример заполнения таблицы «Group». Она отвечает за хранение информации о текущих группах.

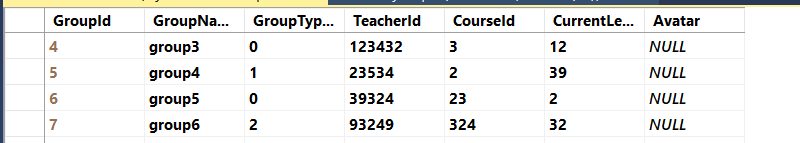


Рисунок 2.13 – Пример заполнения таблицы «Group»

На рисунке 2.14 представлен пример заполнения таблицы «Course». Она отвечает за хранение информации о текущих группах.

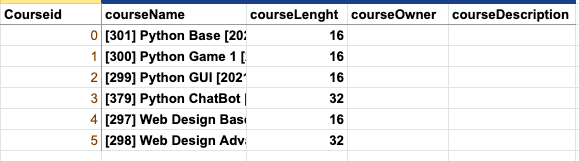


Рисунок 2.14 – Пример заполнения таблицы «Course»

## 2.5 Администрирование базы

База данных представляет хранилище данных, организованных определенным способом. Нередко физически база данных представляет файл на жестком диске, хотя такое соответствие необязательно. Для хранения и администрирования баз данных применяются системы управления базами данных (database management system) или СУБД (DBMS). И как раз MS SQL Server является одной из такой СУБД.

Для организации баз данных MS SQL Server использует реляционную модель. Эта модель баз данных была разработана еще в 1970 году Эдгаром Коддом. А на сегодняшний день она фактически является стандартом для организации баз данных.

SQL Server Management Studio (SSMS) – это интегрированная среда для управления любой инфраструктурой SQL. SSMS используется для доступа, настройки, администрирования и разработки всех компонентов SQL Server. Среда SSMS предоставляет единую комплексную служебную программу, которая сочетает в себе обширную группу графических инструментов с рядом многофункциональных редакторов скриптов для доступа к SQL Server для разработчиков и администраторов баз данных всех профессиональных уровней.

На рисунке 2.14 представлено окно программы.

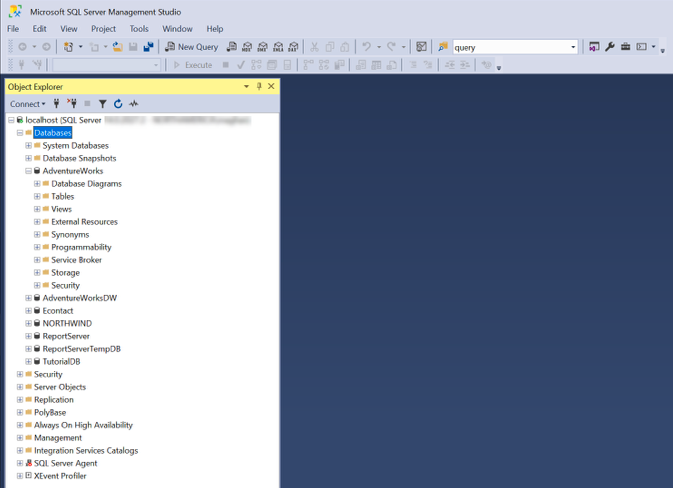


Рисунок 2.14 – Окно SQL Server Management Studio

Элементы управления, доступные для пользователей:

* создание и сжатие файлов базы данных;
* создание, определение, изменение и удаление таблиц;
* создание, определение и удаление индексов;
* просмотр, редактирование, добавление и удаление записей;
* поиск записей;
* импорт и экспорт записей в виде текста;
* импорт и экспорт баз данных из / в файлы SQL;
* выдавать SQL-запросы и проверять результаты.

## 2.6 Выводы к разделу

В данном разделе были подробно описаны обоснования выбора СУБД, аналитический обзор самых подходящих из них и сделан окончательный выбор системы. Также же было проведено проектирование базы данных. Для этого была составлена ER-диаграмма в соответствии со всеми требованиями. После это были ведены тестовые данные для БД приложения и определены основные инструменты внутри выбранной СУБД для дальнейшего администрирования БД по ходу разработки приложения.

**3 Руководство пользователя**

Конечный результат представляет собой веб-приложение «Личный кабинет для преподавателей онлайн-школ». Пользоваться данным сервисом сможет любой человек, хоть раз пользовавшийся браузером и умеющий переходить между ссылками на страницах. Интерфейс веб-сервиса приведен к современному и предполагает интуитивно понятное расположение разделов.

В верхней части экрана расположена панель навигации, которая содержит основные кнопки для взаимодействия.

Для редактирования данных на страницах необходим уровень доступа «Администратор» или «Модератор», такие роли будут назначаться на уполномоченное лицо от компании-пользователя.

Для просмотра, заполнение и редактирование информации, а также расписания, не требуется каких-либо сторонних программ. Все что касается изменения информации об уровне доступа, требует установки любой СУБД и подключение к действующей базе данных. Такая возможность будет у очень ограниченного числа людей и требует определенный знаний и пониманиям порядка внесения данной информации.

Для начала работы с сервисом необходимо пройти этап регистрации, после чего информация о вашей учетной записи попадет в БД и появится возможность для изменения уровня доступа. Страница входа загружается при переходе на сайте автоматически. В веб-сервисе предусмотрен также этап входа в аккаунт в соответствии с текущими современными технологиями. Реализовано это на основе файлов «куки», который сохраняют информацию о посещении сайта на компьютере пользователя, как показано на рисунке 3.1. В таком случае будет открыта главная страница.

Порядок регистрации аккаунта показан на рисунке 3.2.

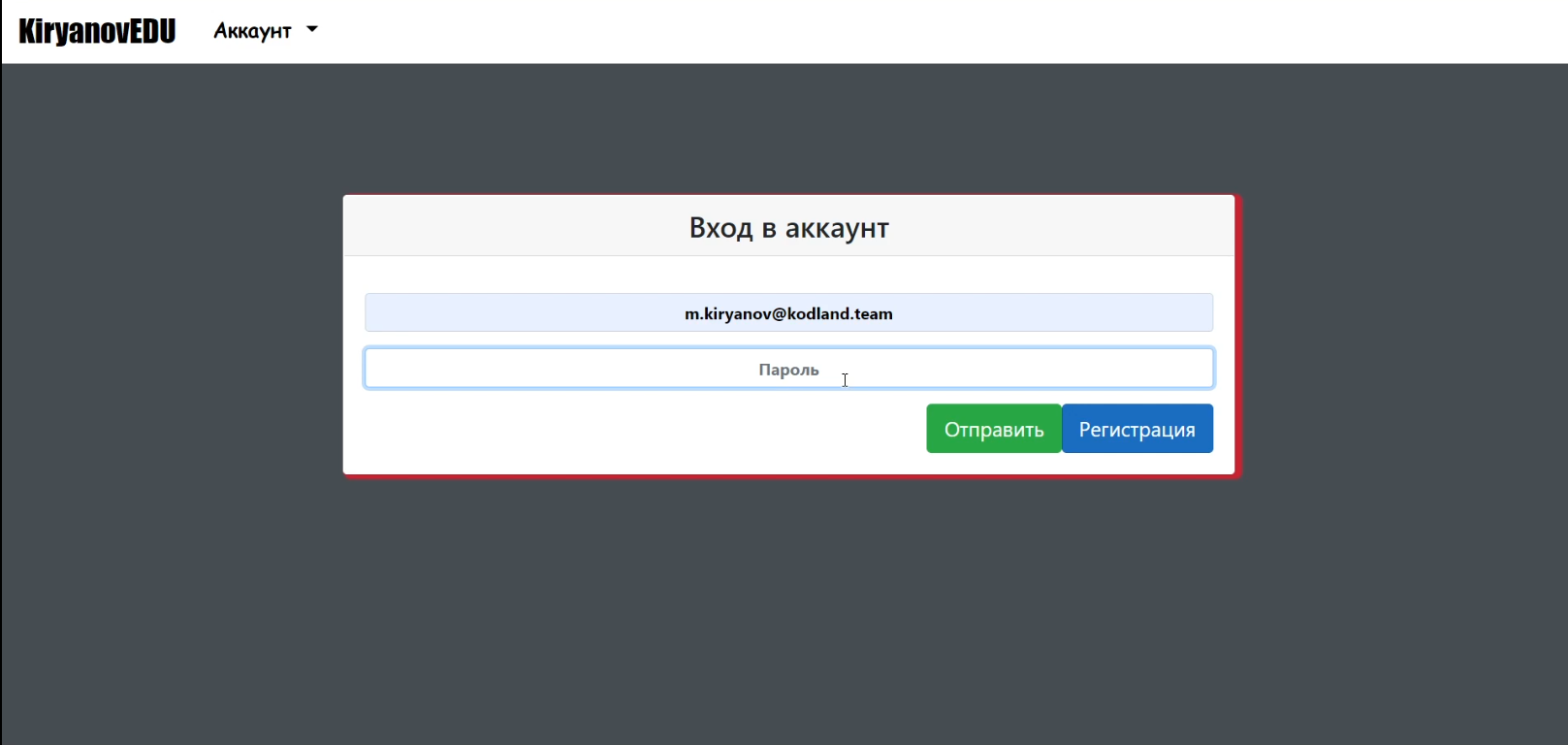


Рисунок 3.1 – Вход на сайт

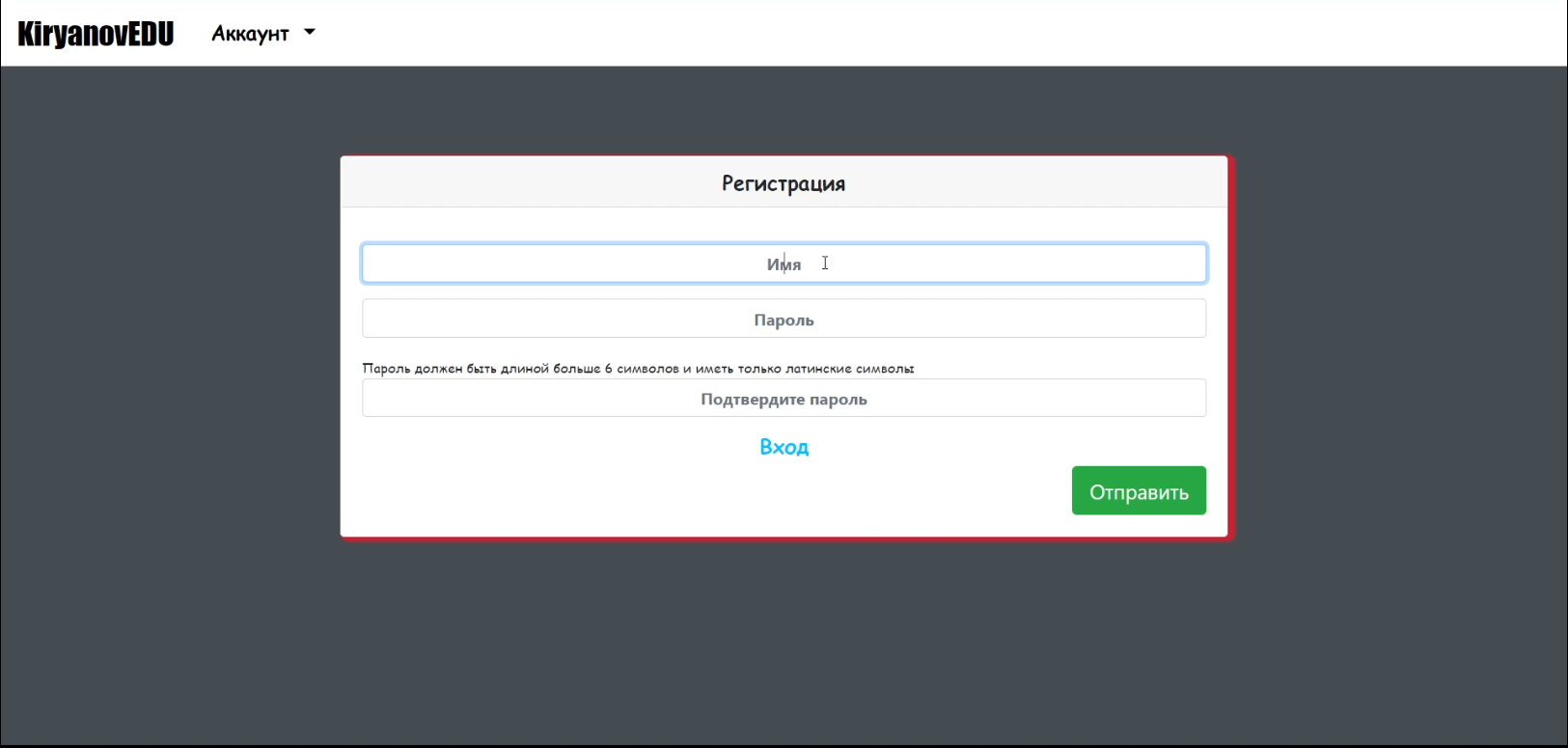


Рисунок 3.2 – Заполнение сущностей

После входа на платформу перед пользователем сразу открывается главная страница, которая на данном этапе разработки представляет информацию о разработанном сервисе и его возможностях.

В верхней части страницы также расположена панель навигации, которая позволяет переходить к разным разделам сервиса. Например: добавлению данных, вывода списка текущей информации.

На рисунке 3.3 - 3.4 представлен интерфейс главной страницы.

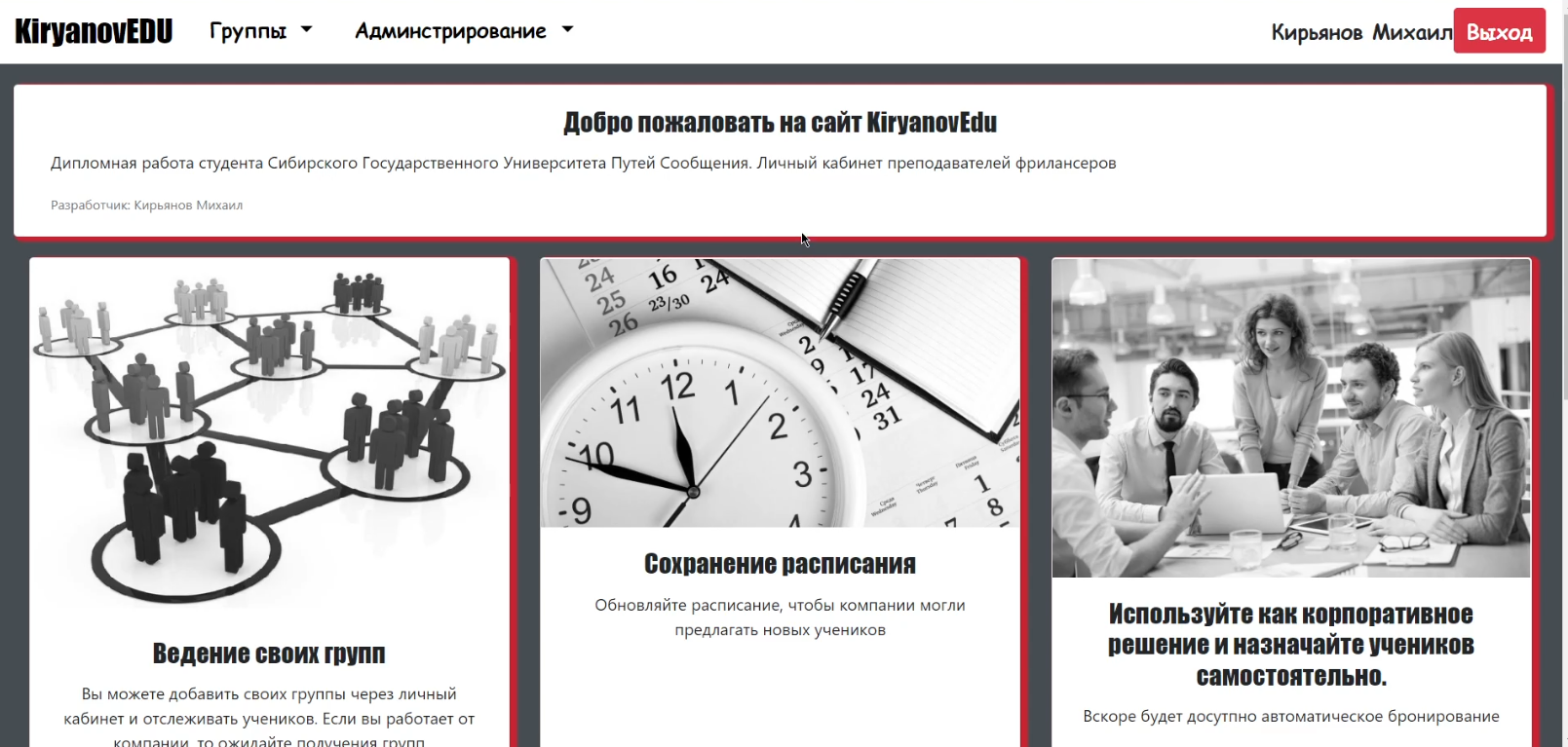


Рисунок 3.3 – Главная страница и панель навигации

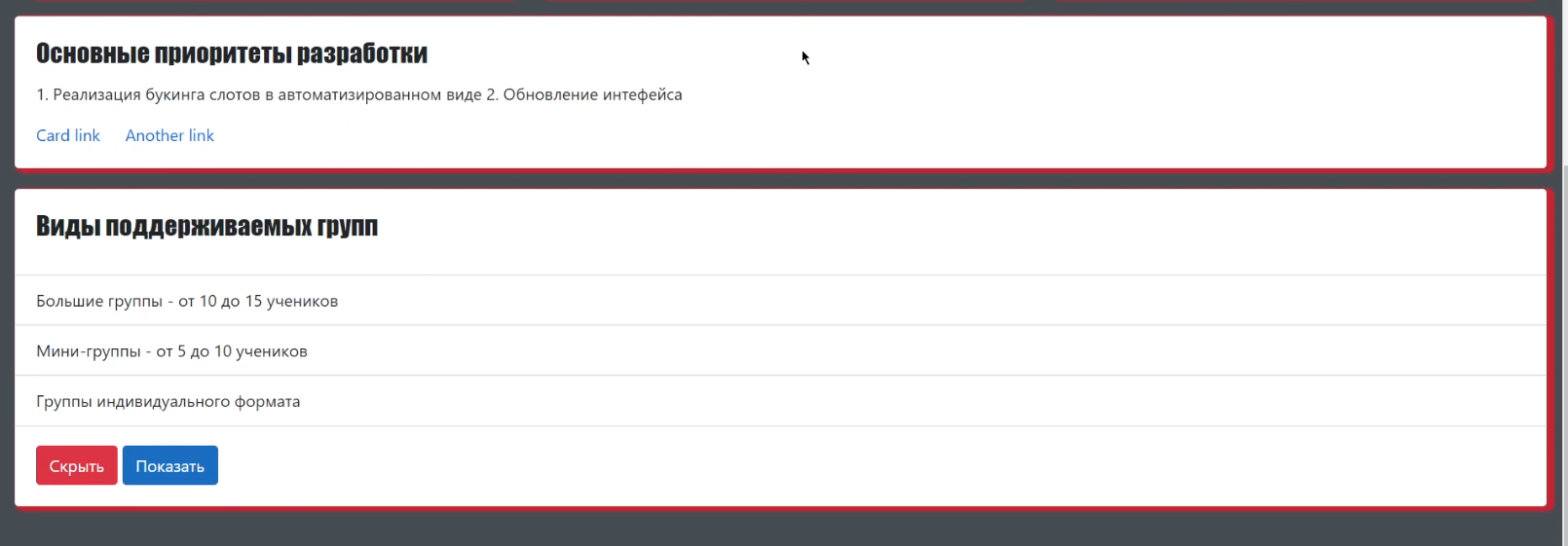


Рисунок 3.4 – Главная страница – далее

Для того чтобы попасть на страницу добавления группы, необходимо через панель навигации зайти в раздел «Администрирование» и выбрать пункт «Добавление». После чего откроется форма добавления со всеми необходимыми полями. Форма добавления новых групп представлена на рисунке 3.5.

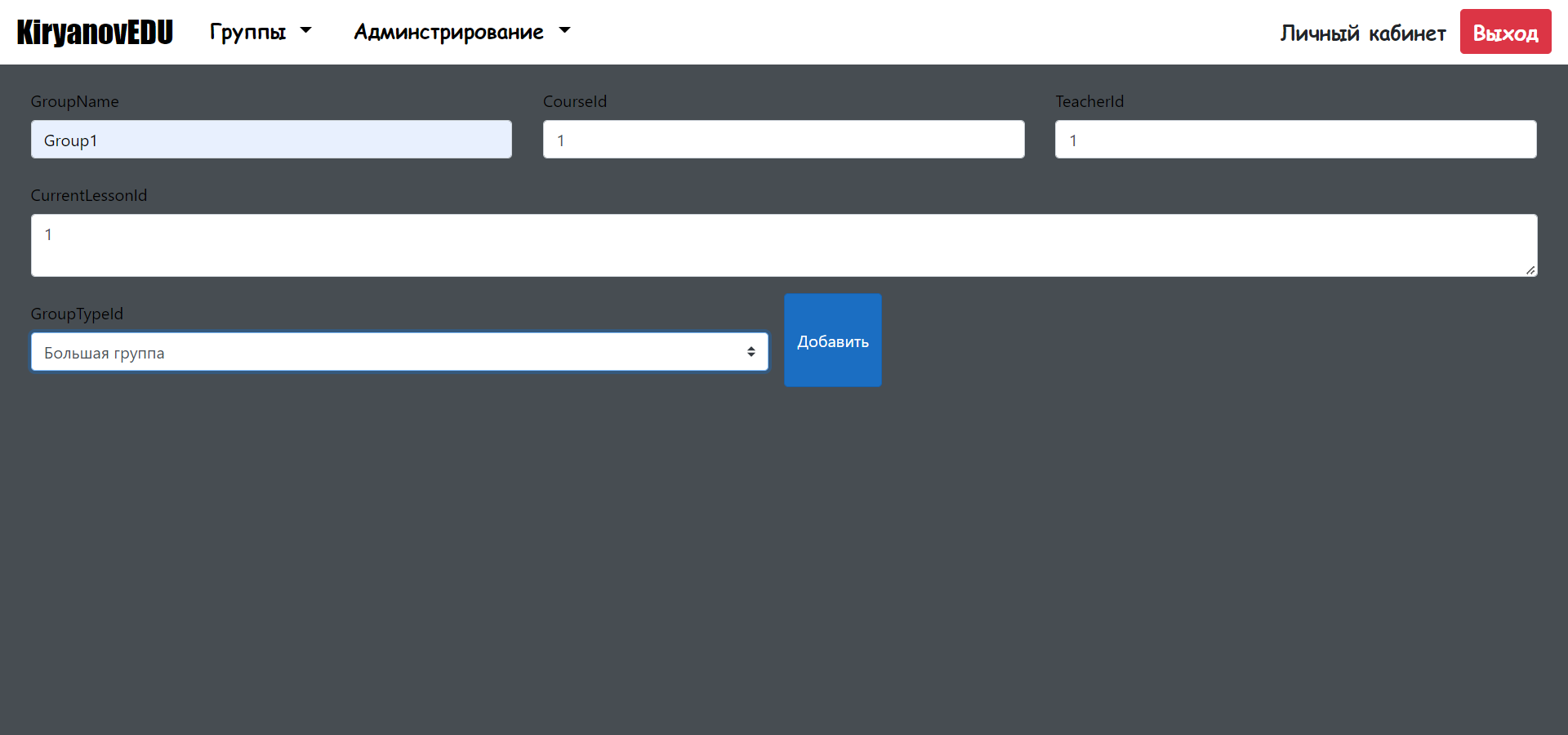


Рисунок 3.5 – Форма добавление групп

Так как формирование данных все равно производится путем ручного ввода части информации была реализована валидация данных. Например, если при заполнении формы, администратор или модератор введет в поле ID преподавателя буквенные строковый литералы, они выделятся цветом и будет выведено сообщение об ошибке, что и показано на рисунке 3.6.

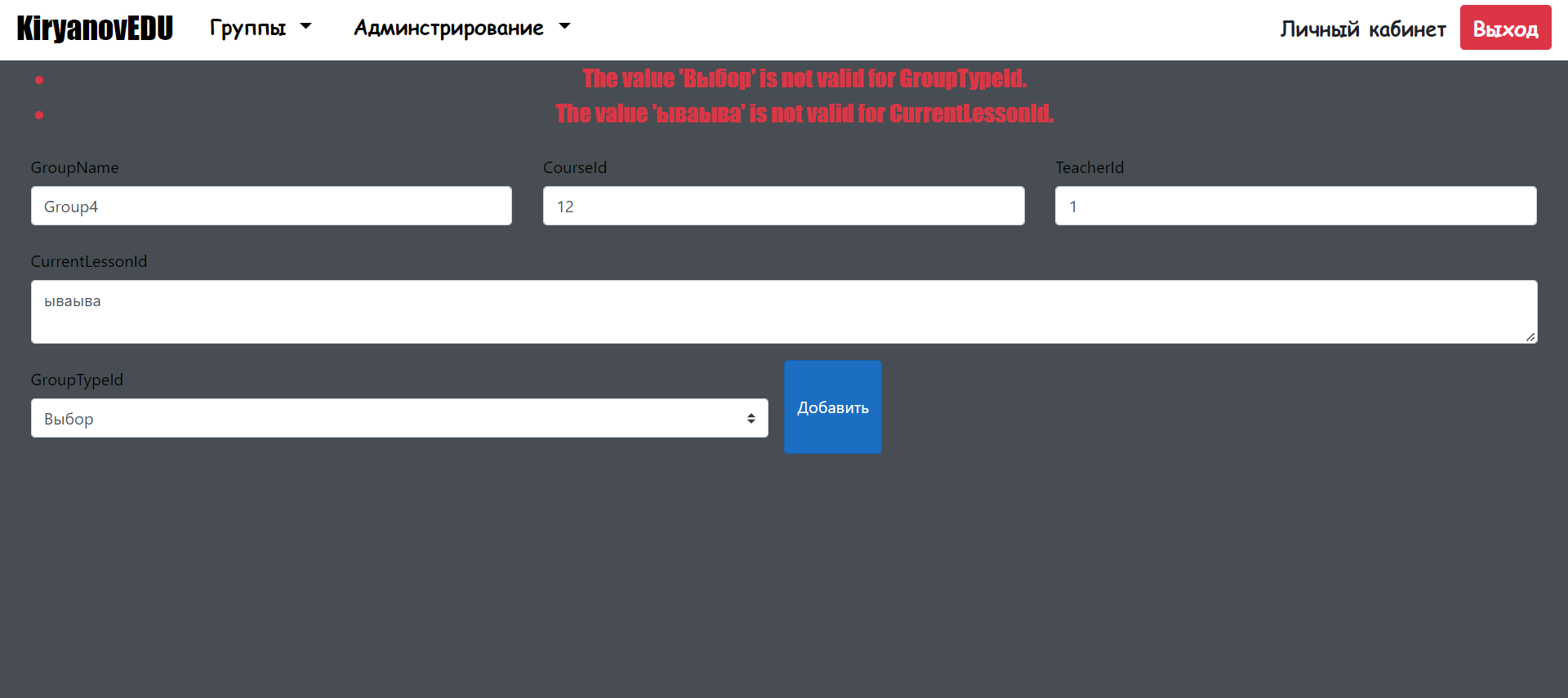


Рисунок 3.6 – Проверка корректности данных

После того как данные введены в систему они автоматически запишутся в базу данных, список данных мы сможем увидеть их на вкладке «Группы». Доступ к этому разделу осуществляется через панель навигации. Список записанных групп представлен на рисунке 3.7.

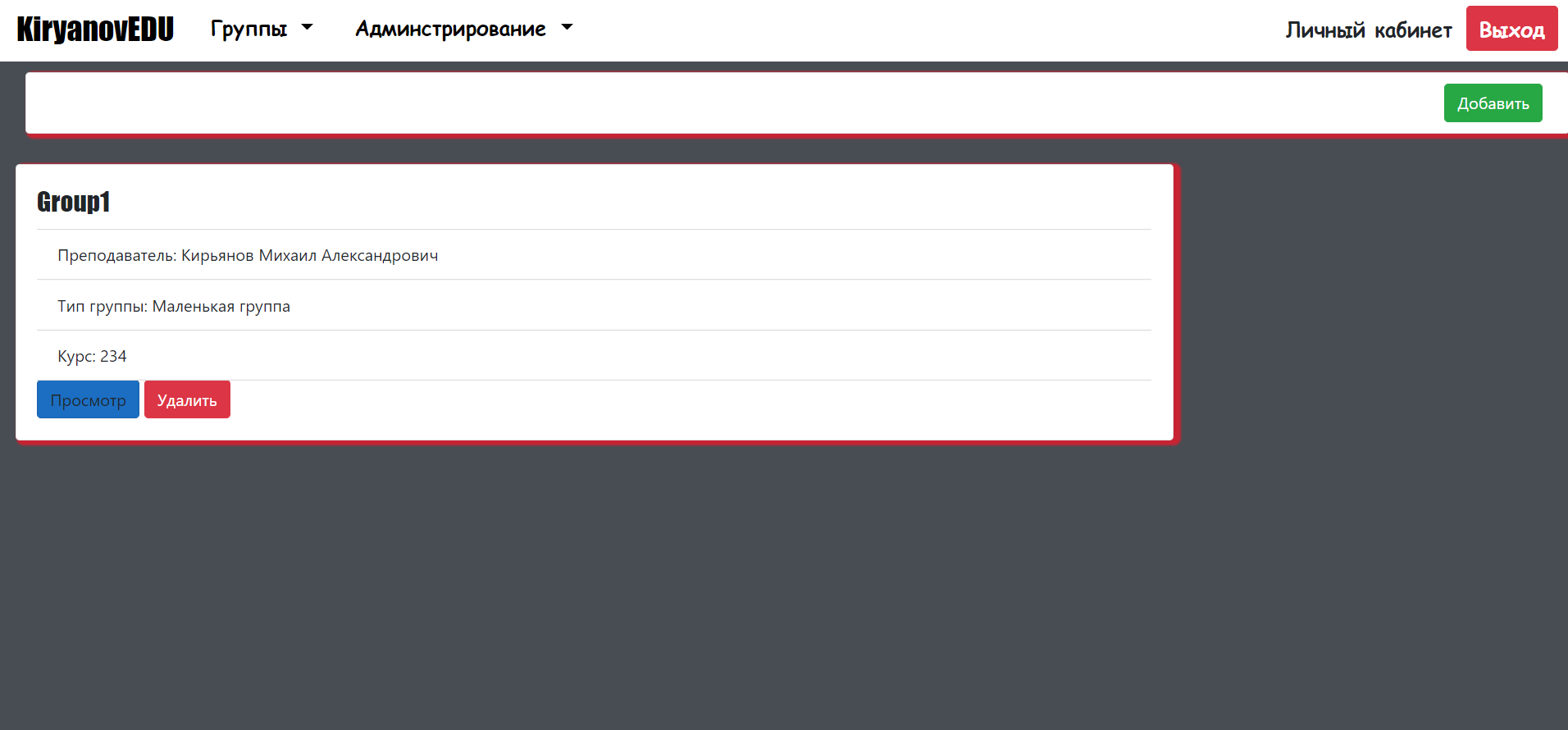


Рисунок 3.7 – Вывод списка групп

С помощью СУБД можно управлять уровнями доступа. Эта возможность предоставлена только администраторам и выдается со стороны владельца ПО. Процесс управления уровнями доступа показан на рисунке 3.8.

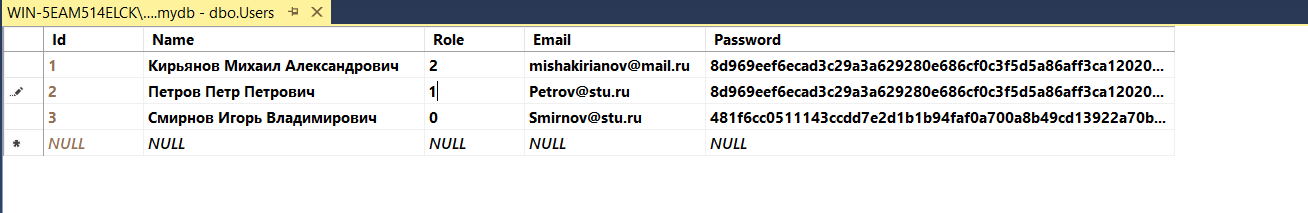


Рисунок 3.8 – Окно настройки доступов

Вход в систему осуществляется через адрес электронной почты. Для того чтобы избежать ситуаций создания аналогичных аккаунтов внедрена проверка на совпадение адресов электронных почт. Результат проверки представлен на рисунке 3.9.

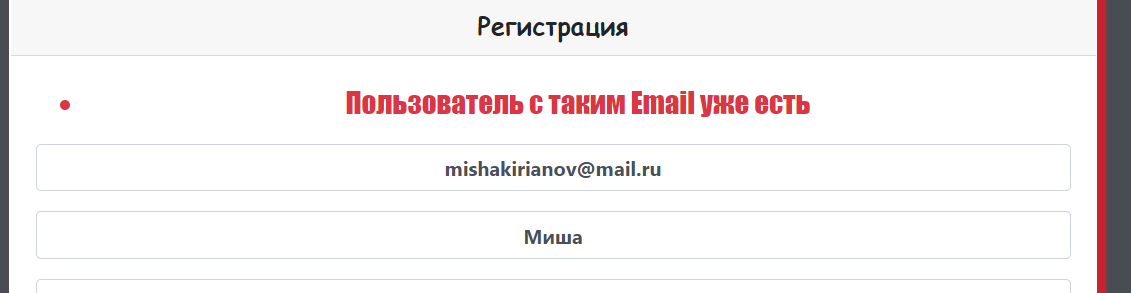


Рисунок 3.9 – Проверка на дубликаты почт пользователей

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения работы была разработана база данных для мобильного приложения согласно техническому заданию.

В процессе работы было сделано:

* проведено сравнение программных аналогов;
* проведен анализ деятельности онлайн-школ;
* исследованы образовательные процессы (как есть);
* смоделированы бизнес-процессы;
* разработано веб-приложение;
* описано руководство пользователя.

Все поставленные задачи были выполнены в полном объеме, а цель работы была достигнута.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Моделирование бизнес-процессов: цели, методы и результаты [Электронный ресурс] – URL: https://www.enterchain.ru/experience/ mbp/modelirovanie-biznes-protsessov-tseli-metody-i-rezultaty/ (дата обращения: 20.05.2022).
2. Моделирование бизнес процессов [Электронный ресурс] – URL: https://www.kpms.ru/Automatization/BPM.htm (дата обращения: 21.05.2022).
3. METANIT.COM С# [Электронный ресурс] – URL: https://metanit.com/ (дата обращения: 05.06.2022).
4. MSDN Microsoft [Электронный ресурс] – URL: https://docs.microsoft.com/ru-ru/ (дата обращения: 25.05.2022).
5. Кристиан Венц. Программирование в ASP.NET Ajax. [Текст] / Кристиан Венц. – Москва: Символ-Плюс, 2018. – 512 с.
6. Павел Агуров. ASP.NET. Сборник рецептов. [Текст] / Павел Агуров.– Санкт-Петербург: БХВ-Петербург, 2020. – 528 с.
7. Сущности (службы основных данных) [Электронный ресурс] – URL: https://docs.microsoft.com/ru-ru/sql/master-data-services/entities-master-data-services?view=sql-server-ver16 (дата обращения: 02.03.2022).
8. Учебном пособии по ER-диаграммам [Электронный ресурс] – URL: https://creately.com/blog/ru/uncategorized-ru/учебном-пособии-по-er-диаграммам/ (дата обращения: 07.05.2022).
9. C# и MongoDB [Электронный ресурс] – URL: https://metanit.com/nosql/mongodb/4.1.php (дата обращения: 20.05.2022).
10. Общие сведения о Microsoft.Data.Sqlite [Электронный ресурс] – URL: https://docs.microsoft.com/ru-ru/dotnet/standard/data/sqlite/?tabs=netcore-cli (дата обращения: 20.05.2022).
11. Сергей Тепляков. Паттерны проектирования на платформе.NET. [Текст] / Сергей Тепляков. – Санкт-Петербург: ПИТЕР, 2018. – 566 с.