Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

|  |
| --- |
| Институт информационных технологий и анализа данных |
| наименование института |

|  |  |
| --- | --- |
| Допускаю к защите |  |
| Руководитель | подпись |
|  | Т.А. Баталова |
|  | И.О. Фамилия |

|  |
| --- |
| Разработка программного обеспечения |
| для учёта и анализа частотно-временной информации |

наименование темы

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовому проекту по дисциплине

|  |  |  |
| --- | --- | --- |
| Технологии разработки программных комплексов | | |
|  | 1.7.00.00 - ПЗ |  |

обозначение документа

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент |  | АСУб-22-2 |  |  |  | Д.И. Гончаренко |
|  |  | шифр группы |  | подпись |  | И.О. Фамилия |
| Нормоконтроль |  |  |  |  |  | Т.А. Баталова |
|  |  |  |  | подпись |  | И.О. Фамилия |

Курсовой проект защищен с оценкой

|  |
| --- |
|  |

Иркутск 2025 г.

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

ЗАДАНИЕ

НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| По курсу | Технологии разработки программных комплексов | | | |
| Студенту | Гончаренко Д.И. | | | |
|  | (фамилия, инициалы) | | | |
| Тема проекта | | Разработка информационной системы для студии | | |
| по обучению игре на духовых музыкальных инструментах | | | | |
| Исходные данные: | | |  | |
| Разработка программного обеспечения для автоматизации записи на занятия для студии (вариант №7) | | | | |
|  | | | | |
| |  |  |  | | --- | --- | --- | | Рекомендуемая литература: |  | | | 1. Гутгарц Р.Д Проектирование автоматизированных систем обработки информации и управления: учебное пособие для академического бакалавриата. – М.: Издательство Юрайт, 2019. | | | 1. Проектирование АСОИУ [Электронный ресурс] : методические указания по выполнению курсового проекта: 09.03.01 «Информатика и вычислительная техника» для бакалавров по специальности «Автоматизированные системы обработки информации и управления» / Иркут. нац. исслед. техн. ун-т ; сост. Р. Д. Гутгарц. - Электрон. дан. - Иркутск : ИРНИТУ, 2018. | | | 1. Проектирование информационных систем : учебник и практикум для академического бакалавриата / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук ; под общ. ред. Д. В. Чистова. – М. : Издательство Юрайт, 2016. | | | 1. Рудинский И.Д. Технология проектирования автоматизированных систем обработки информации и управления : учебное пособие для вузов / И. Д. Рудинский. - Москва : Горячая линия - Телеком, 2011. | | | | | |  |

Графическая часть на \_\_\_\_\_\_\_\_\_\_\_\_\_\_ листах.

Дата выдачи задания « 20 » сентября 2025 г.

|  |  |  |  |
| --- | --- | --- | --- |
| Задание получил |  |  | Д.И. Гончаренко |
|  | подпись |  | И.О. Фамилия |

Дата представления проекта руководителю « 20 » декабря 2025 г.

|  |  |  |  |
| --- | --- | --- | --- |
| Руководитель курсового проектирования |  |  | Т.А. Баталова |

**Содержание**

[Введение 4](#_Toc217123143)

[1 Анализ 5](#_Toc217123144)

[1.1 Цель и задачи проекта 5](#_Toc217123145)

[1.2. Обзор предметной области: Студия музыкального образования 5](#_Toc217123146)

[1.3. Обзор существующих программных средств 6](#_Toc217123147)

[1.4. Процесс AS IS vs TO BE 7](#_Toc217123149)

[1.5. Описание вариантов использования 10](#_Toc217123152)

[1.6. Выработка требований и постановка задачи 10](#_Toc217123154)

[2 Проектирование 13](#_Toc217123158)

[2.1. Выбор и обоснование средств проектирования и реализации. 13](#_Toc217123159)

[2.2. Проектирование архитектуры приложения 15](#_Toc217123162)

[2.3. Проектирование хранилища данных 18](#_Toc217123165)

[2.4. Проектирование пользовательского интерфейса 20](#_Toc217123170)

[3 Разработанная система 23](#_Toc217123173)

[3.1 Порядок авторизации пользователя в системе 23](#_Toc217123174)

[3.2 Проектные решения для интерфейса 25](#_Toc217123175)

[Заключение 32](#_Toc217123176)

[Список использованных источников 33](#_Toc217123177)

Введение

Обучение игре на духовых музыкальных инструментах представляет собой комплексный образовательный процесс, требующий организации со стороны студии. Этот процесс включает индивидуальные и групповые занятия, подбор репертуара, управление расписанием занятий, учёт посещаемости и прогресса.

В настоящее время многие студии, особенно малого и среднего размера, управляют этими процессами фрагментарно. Учёт клиентов и оплат ведётся в табличных редакторах (например, Excel) или простых CRM, расписание составляется вручную, а коммуникация с учениками и передача материалов происходят через множество каналов: мессенджеры, электронную почту, файлообменники. Такой подход приводит к целому ряду системных проблем: потерям информации, ошибкам в расписании и оплатах, низкой прозрачности учебного процесса для всех его участников и значительной административной нагрузке на руководителей и преподавателей.

Проект информационной системы «Ритм» нужен для того, чтобы изменить эту ситуацию. Система представляет собой единую веб-платформу, предназначенную для комплексной автоматизации всех ключевых бизнес-процессов студии обучения музыке. «Ритм» объединяет в себе цифровой инструмент для управления клиентами, расписанием, финансами, образовательным контентом и коммуникациями.

Внедрение системы направлено на переход от хаотичного, ручного управления к структурированному, цифровому, что позволит студии повысить свою операционную эффективность и качество обслуживания клиентов.

**1** Анализ

1.1 Цель и задачи проекта

**Цель:** Создание единой информационной системы для комплексной автоматизации управления студией по обучению игре на духовых музыкальных инструментах, обеспечивающей повышение эффективности учебного процесса и качества обслуживания.

**Задачи:**

1. Разработать модуль учёта и управления пользователями (ученики, преподаватели, администраторы).
2. Реализовать систему гибкого планирования и управления расписанием занятий.
3. Внедрить модуль финансового учёта, включающий интеграцию с платёжными системами, контроль оплат и формирование отчётов.
4. Разработать инструменты для отслеживания успеваемости и посещаемости учеников (электронный журнал)
5. Реализовать административную панель для анализа ключевых показателей деятельности студии (отчеты).

1.2. Обзор предметной области: Студия музыкального образования

**Проблема:** Основными проблемами студии являются разрозненность данных (клиенты в таблицах, расписание в календарях, материалы в почте), ведущая к ошибкам и потере информации. Ручное управление расписанием вызывает накладки, а коммуникация через мессенджеры — нестабильность и потерю истории. Финансовый учёт непрозрачен, контроль прогресса учеников — фрагментарен, а административная нагрузка — чрезмерна.

**Участники процесса и их роли в системе:**

Ученик: Основной потребитель услуг. Через личный кабинет получает доступ к записи на занятия, библиотеке материалов, системе коммуникации с преподавателем и возможность оплаты.

Преподаватель: Ключевой исполнитель. Управляет своим расписанием, загружает учебные материалы для своих учеников, взаимодействует с ними в рамках системы.

Администратор: Управляющее звено. Осуществляет полный контроль: регистрирует пользователей, назначает роли, формирует общее расписание, управляет данными об аудиториях и оборудовании, контролирует финансовые потоки, генерирует аналитические отчёты и модерирует контент.

**Актуальность:** Актуальность обусловлена трендом на цифровизацию услуг и ростом конкуренции. Система централизует все процессы, автоматизирует работу и предоставляет аналитику, это повышает качество сервиса, снижает операционные издержки и создаёт преимущество для студии.

1.3. Обзор существующих программных средств

Перед разработкой ИС был проведён анализ доступных на рынке программных решений, которые потенциально могли бы быть использованы для управления музыкальной студии. Рассмотрены три основные категории: системы управления обучением (LMS), CRM-системы для услуг и универсальные инструменты планирования.

1. **Системы управления обучением (LMS):**

**Примеры**: Moodle, Google Classroom, GetCourse.

**Преимущества**: Имеют развитый функционал для организации учебного процесса: размещение материалов, отправка заданий, тестирование, ведение электронного журнала. Хорошо подходят для структурирования контента и отслеживания академического прогресса.

**Недостатки**: Слабо адаптированы под коммерческие аспекты работы студии**.** В них отсутствует или крайне ограничен функционал для управления расписанием с учётом аудиторий и преподавателей, интеграции с платёжными системами, финансовой отчётности и комплексного учёта клиентов как плательщиков. Интерфейс часто избыточен и сложен для задач индивидуального обучения музыке.

1. **CRM и системы записи на услуги:**

**Примеры**: Яндекс.Услуги, Zoon, Mindbox, Simplybook.me.

**Преимущества**: Сфокусированы на привлечении клиентов, онлайн-записи и управлении клиентской базой. Имеют встроенные календари, системы SMS/e-mail уведомлений и часто интегрированы с платёжными шлюзами. Просты в начальном использовании.

**Недостатки**: Их функционал поверхностен для образовательного процесса. Нет инструментов для ведения детальной успеваемости, системной работы с учебными материалами (ноты, аудиозаписи), структурированной обратной связи от преподавателя. Ограничены возможности для внутренней аналитики по загрузке педагогов и эффективности обучения.

1. **Универсальные инструменты планирования и коммуникации:**

**Примеры**: Google Календарь, мессенджеры (WhatsApp, Telegram).

**Преимущества**: Бесплатность или низкая стоимость, высокая гибкость и привычность для пользователей. Позволяют быстро организовать отдельные процессы.

**Недостатки**: Использование набора разрозненных инструментов приводит к фрагментации информации: расписание в одном месте, оплаты в другом, материалы в третьем, коммуникация — в четвёртом. Отсутствует единая база данных, что провоцирует потерю информации, ошибки и крайне затрудняет получение целостной картины работы студии. Требует постоянной ручной синхронизации.

**Вывод:** Проведённый анализ подтверждает наличие рыночной ниши. Существующие решения либо слишком академичны и не покрывают бизнес-задачи студии (LMS), либо слишком поверхностны и не поддерживают глубину учебного процесса (CRM для услуг), либо приводят к хаосу при комбинировании (набор универсальных инструментов). Ценность разработки ИС «Ритм» заключается в создании интегрированной платформы, которая целенаправленно объединяет ключевые функции управления коммерческой студией: от записи и оплаты до глубокого отслеживания учебного прогресса и работы с профессиональным контентом в едином пространстве.

Процесс AS IS vs TO BE

Для анализа текущих бизнес-процессов студии по обучению игре на духовых инструментах и обоснования необходимости разработки ИС «Ритм» была выполнена их формализация. Моделирование в нотации BPMN (Business Process Model and Notation) наглядно демонстрирует различия между существующей (AS IS) и целевой (TO BE) схемами работы, позволяя выявить проблемные точки и определить требования к автоматизации.

1.4.1 Модель процесса AS IS

**Основные этапы текущего процесса (AS IS):**



Рисунок 1.1 – модель AS IS

**Ключевые проблемы процесса AS IS:**

* Информация разбита между бумажными носителями, Excel, мессенджерами и личными записями преподавателей.
* Процесс записи требует постоянных звонков и ручных сверок, что отнимает время и ведет к ошибкам (двойное бронирование).
* Клиент не может самостоятельно проверить расписание, записаться или увидеть историю платежей.
* Сведение данных из разных источников для анализа загрузки, финансов и успеваемости — трудоемкий ручной процесс.
* Материалы и обратная связь теряются в переписке, нет единой истории взаимодействия с учеником.

1.4.2 Модель процесса TO BE

**Основные этапы целевого процесса (TO BE):**



Рисунок 1.2 – модель TO BE

**Преимущества процесса TO BE:**

* Вся информация о клиентах, расписании, финансах и учебном процессе хранится в единой защищенной базе данных.
* Устранение ручных звонков, сверок и переносов данных. Автоматические уведомления и напоминания.
* Круглосуточный онлайн-доступ для клиентов к расписанию, записи, материалам и финансовой истории.
* Руководство получает оперативные инструменты для анализа ключевых показателей бизнеса и качества обучения.
* Создание единого цифрового пространства для взаимодействия ученика и преподавателя с сохранением всей истории.

**1.4.3 Вывод:**  
 Переход от модели **AS IS** к модели **TO BE** с внедрением ИС «Ритм» представляет собой качественную цифровую трансформацию бизнес-процессов студии. Это позволяет устранить ключевые операционные недостатки, снизить административную нагрузку, минимизировать человеческие ошибки и значительно повысить уровень сервиса для клиентов. Целевая модель демонстрирует современный, эффективный и масштабируемый формат управления, создающий устойчивое конкурентное преимущество на рынке.

Описание вариантов использования

Для описания функциональности веб-приложения была построена диаграмма вариантов использования (Use Case Diagram), показывающая взаимодействие актёров с системой.

1.5.1 Описание вариантов использования

**Актёры:**

* **Ученик:** пользователь, который записывается на занятия через веб-интерфейс.
* **Преподаватель:** сотрудник, который загружает учебные материалы, проводит занятия.
* **Администратор:** сотрудник, который занимается управлением расписанием, материалами, формирует отчеты.



Рисунок 1.3 – Диаграмма использования системы

Выработка требований и постановка задачи

На основе анализа предметной области, существующих решений, моделей процессов и вариантов использования можно сформулировать требования к создаваемой системе «Ритм».

1.6.1 Функциональные требования к системе

**Управление пользователями и аутентификация:**

* Регистрация новых пользователей (учеников, преподавателей).
* Ролевой доступ (Ученик, Преподаватель, Администратор).
* Двухфакторная аутентификация (опционально).
* Восстановление пароля.

**Управление расписанием и записью:**

* Создание и редактирование расписания занятий.
* Отображение свободных слотов для записи.
* Онлайн-запись ученика на занятие с выбором времени.
* Автоматическая проверка на конфликты (занятость преподавателя/аудитории).

**Учебный процесс и контроль прогресса:**

* Личный кабинет ученика.
* Электронный журнал для преподавателя.

**Финансовый и договорной учет:**

* Формирование счетов за обучение на основе расписания и тарифов.
* Интеграция с внешними платежными системами для онлайн-оплаты.
* Автоматический учет поступивших платежей.
* Генерация финансовых отчетов.

**Управление ресурсами и контентом:**

* Учет аудиторий и музыкального оборудования.
* Каталог учебных программ и материалов с разграничением прав доступа.
* Модерация загружаемого контента администратором.

1.6.2. Нефункциональные требования

* Данные пользователей и файлов должны быть защищены от потери
* Система должна обеспечивать доступ 24/7
* Отклик интерфейса не более 2 секунд на основные операции
* Авторизация с защитой персональных данных
* Интерфейс должен быть интуитивно понятным для всех пользователей

1.6.3 Постановка задачи

**Задача проекта:**  
Разработать веб-ориентированную информационную систему (ИС) для комплексной автоматизации управления студией по обучению игре на духовых музыкальных инструментах.

**Система должна обеспечить:**

* Управление жизненным циклом клиента (от регистрации до завершения обучения).
* Оперативное и гибкое планирование учебного процесса.
* Эффективное взаимодействие между всеми участниками (администрация, преподаватели, ученики).
* Прозрачный финансовый учет и контроль.
* Централизованное хранение и управление учебными материалами.
* Автоматизированную аналитику для принятия управленческих решений.

2 Проектирование

2.1. Выбор и обоснование средств проектирования и реализации.

В процессе выполнения лабораторной работы и проектирования информационной системы были выбраны современные и широко применяемые средства проектирования и реализации, обеспечивающие наглядность проектных решений, простоту разработки, масштабируемость и соответствие требованиям проекта.

2.1.1. Средства проектирования

* **Mermaid / PlantUML**

Для проектирования структуры и логики информационной системы использовались языки описания диаграмм Mermaid и PlantUML. Данные инструменты позволяют формировать UML-диаграммы (диаграммы вариантов использования, классов, последовательностей и компонентов) в текстовом формате, что упрощает их редактирование и интеграцию в документацию.

**Обоснование выбора:**

* поддержка UML-нотаций;
* наглядное представление архитектуры системы;
* возможность быстрого внесения изменений;
* удобная интеграция с Markdown-документами и средами разработки;
* отсутствие необходимости в сложных графических редакторах.

2.1.2. Средства реализации

**Frontend**

* **Vanilla JavaScript, HTML5, CSS3**

Клиентская часть системы реализуется с использованием стандартных веб-технологий без применения сторонних фреймворков.

**Обоснование выбора:**

* отсутствие зависимости от внешних библиотек;
* высокая производительность;
* контроль над всей логикой пользовательского интерфейса;
* совместимость со всеми браузерами.

**Backend**

* **PHP 7.4 и выше**

Серверная логика информационной системы реализуется на языке PHP.

**Обоснование выбора:**

* широкая распространённость в веб-разработке;
* хорошая интеграция с MySQL/MariaDB;
* поддержка большинства веб-хостингов;
* достаточная производительность для информационных систем учебного уровня.

**Хранилище данных**

* **MySQL / MariaDB**

Для хранения данных используется реляционная система управления базами данных MySQL либо её совместимая версия MariaDB.

**Обоснование выбора:**

* надёжность и устойчивость;
* поддержка SQL-стандарта;
* высокая скорость обработки запросов;
* широкая документация;
* удобство интеграции с PHP.

**Веб-сервер**

* **Apache или Nginx (через XAMPP / OpenServer)**

Для развертывания локальной среды разработки применяется Apache или Nginx в составе XAMPP либо OpenServer.

**Обоснование выбора:**

* простота установки и настройки;
* наличие встроенных инструментов администрирования;
* возможность быстрого развертывания проекта;
* поддержка PHP и MySQL.

**Система контроля версий**

* **Git с использованием GitHub**

Для управления версиями исходного кода используется система Git с удалённым репозиторием на GitHub.

**Обоснование выбора:**

* контроль изменений в коде;
* возможность отката к предыдущим версиям;
* хранение резервных копий проекта;
* стандарт в современной разработке.

**Тестирование API**

* **Postman или Insomnia**

Для тестирования HTTP-запросов и серверных API использовались инструменты Postman либо Insomnia.

**Обоснование выбора:**

* удобный интерфейс для отправки запросов;
* поддержка различных HTTP-методов;
* ускорение отладки серверной логики;
* визуальный контроль корректности ответов сервера.

**Среда разработки**

* **Visual Studio Code**

Разработка велась в современной среде разработки, поддерживающей веб-технологии.

**Обоснование выбора:**

* подсветка синтаксиса и автодополнение кода;
* встроенные средства отладки;
* поддержка Git;
* повышение производительности разработки.

Выбранные средства проектирования и реализации являются оптимальными для разработки информационной системы в рамках работы. Они обеспечивают наглядность проектных решений, простоту реализации, надёжность хранения данных и соответствие современным требованиям к разработке веб-приложений.

2.2. Проектирование архитектуры приложения

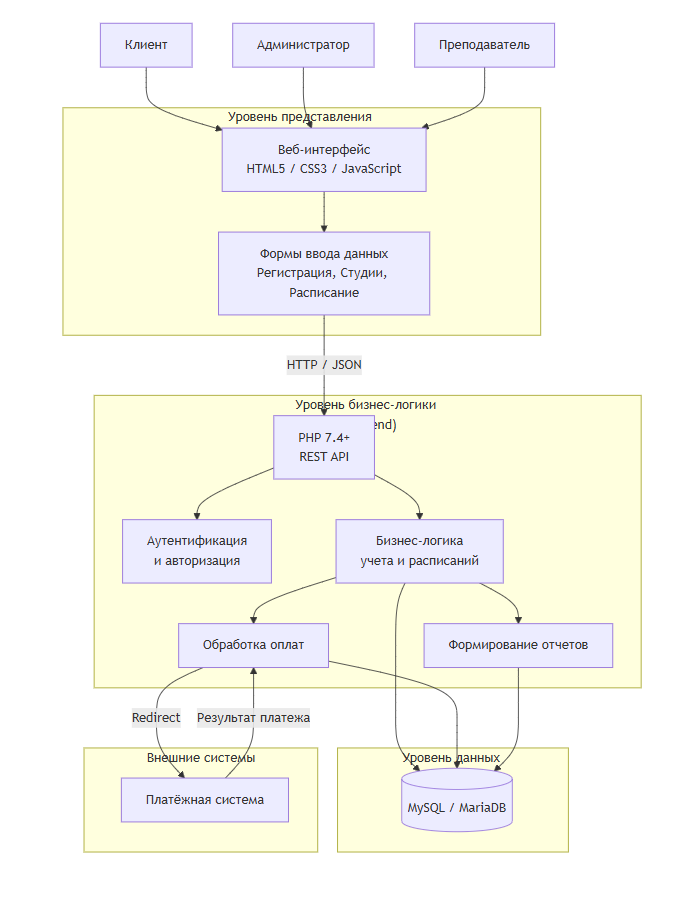
Веб-система «РИТМ» реализуется на основе классической трёхуровневой архитектуры, включающей

• Уровень представления (Frontend)

• Уровень бизнес-логики (Backend)

• Уровень данных (Database)

2.2.1. Схема архитектуры

  
Рисунок 2.1 – Схема архитектуры веб-системы

2.2.2 Описание архитектуры

**Уровень представления (Frontend)**

Уровень представления реализован с использованием HTML5, CSS3 и Vanilla JavaScript и отвечает за взаимодействие пользователя с системой.

Функции уровня:

* отображение интерфейсов (регистрация, студии, расписание, оплата, статистика);
* ввод данных пользователем;
* отправка запросов к серверу;
* отображение результатов обработки данных.

**Уровень бизнес-логики (Backend)**

Backend реализован на языке PHP 7.4+ и выполняет основную обработку данных системы.

Функции уровня:

* аутентификация и авторизация пользователей;
* реализация бизнес-правил:
  + уникальность e-mail;
  + проверка статусов студий и оборудования;
  + ограничения при работе с расписанием;
  + фиксация оплат;
* формирование данных для аналитических отчётов;
* взаимодействие с внешней платёжной системой.

**Уровень данных (Database)**

Для хранения данных используется реляционная СУБД MySQL / MariaDB.

В базе данных хранятся:

* данные клиентов;
* информация о студиях и оборудовании;
* расписание занятий;
* данные об оплатах;
* финансовые показатели для аналитики.

**Внешние сервисы**

Для обработки оплат используется внешняя платёжная платформа, на которую пользователь перенаправляется для безопасного ввода платёжных данных.  
Система «РИТМ» получает от платёжного сервиса подтверждение результата транзакции и сохраняет данные в базе.

2.3. Проектирование хранилища данных

2.3.1 Обоснование выбора архитектуры

Выбор трёхуровневой архитектуры обусловлен следующими факторами:

1. **Соответствие предметной области**  
   Архитектура хорошо подходит для информационных систем учёта и управления, таких как система студии обучения.
2. **Разделение ответственности**  
   Каждый уровень выполняет строго определённые функции, что упрощает разработку, отладку и сопровождение.
3. **Масштабируемость**  
   Возможна замена или развитие любого уровня без переработки всей системы.
4. **Безопасность**  
   Клиентская часть не имеет прямого доступа к базе данных, все операции проходят через серверную бизнес-логику.

2.3.2. Общая характеристика хранилища данных

Для веб-системы «РИТМ» предусмотрено использование реляционной базы данных на основе СУБД MySQL / MariaDB. База данных предназначена для централизованного хранения информации, необходимой для выполнения всех функциональных задач системы: учета клиентов, студийных помещений, оборудования, расписания занятий, оплат и анализа финансовых показателей.

Проектирование базы данных выполнено с соблюдением принципов нормализации до третьей нормальной формы (3НФ), что обеспечивает:

* минимизацию избыточности данных;
* целостность и согласованность информации;
* удобство сопровождения и масштабирования системы.

2.3.3. Основные сущности системы

Таблица 2.1.– Основные сущности и атрибуты

|  |  |
| --- | --- |
| **Сущность** | **Основные атрибуты** |
| Пользователь | id, email, пароль, роль, статус, дата регистрации |
| Студия | id, название, вместимость, адрес, описание, стоимость, статус |
| Оборудование | id, студия\_id, тип\_id, состояние\_id, название, описание, количество |
| Тип оборудования | id, наименование |
| Состояние оборудования | id, наименование |
| Расписание (занятие) | id, студия\_id, ученик\_id, преподаватель\_id, дата, время, тип, статус, примечание |
| Оплата | id, пользователь\_id, занятие\_id, сумма, дата, статус |
| Статус студии | id, наименование |
| Финансовые данные | формируются на основе оплат |

2.3.4. ER-модель данных

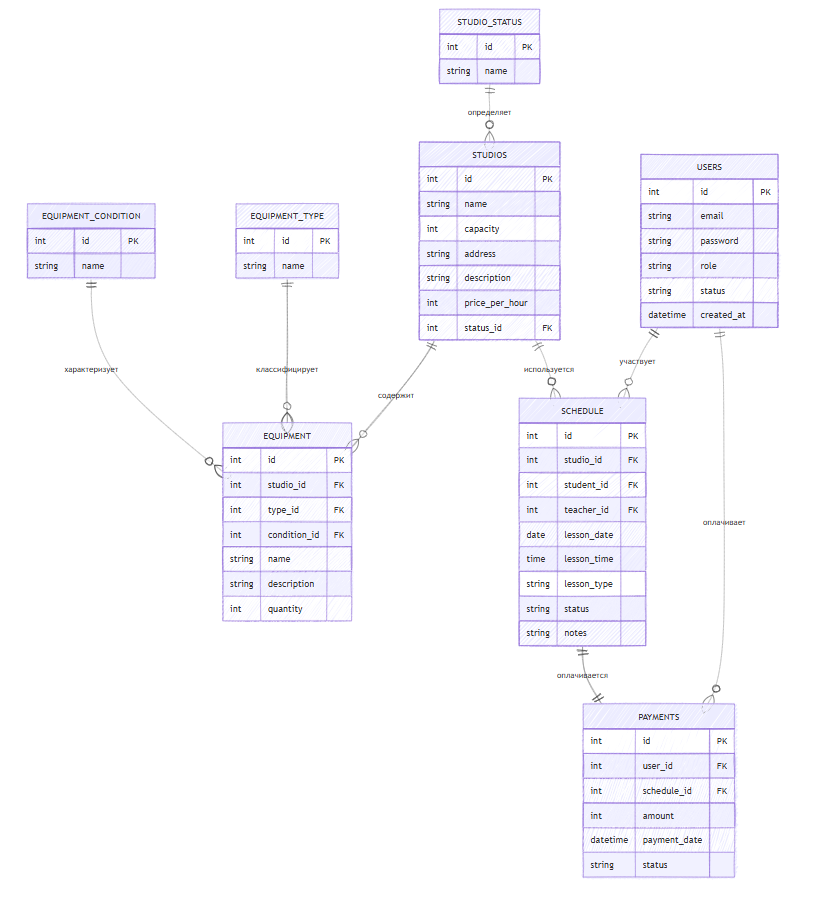


Рисунок 2.2 – ER-модель

**Основные связи между сущностями**

* Пользователь **может иметь** несколько записей на занятия
* Студия **содержит** оборудование
* Студия **участвует** в расписании
* Занятие **связано** с учеником и преподавателем
* Каждое занятие **может иметь** одну оплату
* Оплата **принадлежит** конкретному пользователю

2.4. Проектирование пользовательского интерфейса

2.4.1 Верхнеуровневое определение экранов

Пользовательский интерфейс веб-системы «РИТМ» разделён на четыре основных функциональных раздела. Такое разделение отражает бизнес-логику предметной области и обеспечивает удобство работы для администраторов и преподавателей студии.

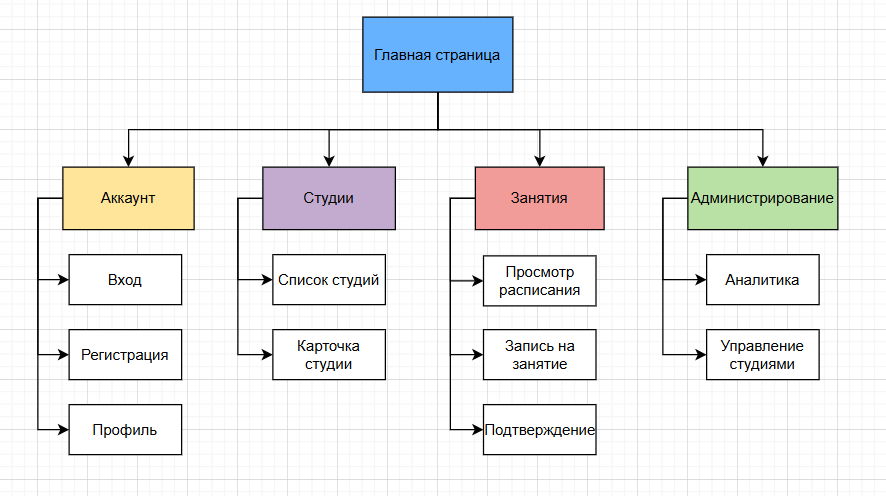


Рисунок 2.3. – Верхнеуровневое определение экранов системы

2.4.2 Назначение экранов и описание

Таблица 2.2 – Описание экранов

|  |  |  |  |
| --- | --- | --- | --- |
|  | Название | Поля ввода для валидации | Описание экрана и поведение |
| 1 | Аккаунт |  |  |
| 1.1 | Вход | Email (формат: user@domain)  Пароль (>8 символов) | Система отображает форму ввода учётных данных пользователя. После нажатия кнопки «Войти» выполняется клиентская валидация введённых данных. В случае обнаружения ошибок валидации отображаются текстовые сообщения об ошибках, а отправка формы блокируется. В случае успешной авторизации пользователь автоматически перенаправляется на экран профиля. |
| 1.2 | Регистрация | Email (формат: user@domain)  Пароль (>8 символов)  Имя (≥ 2 символов)  Фамилия (≥ 2 символов) | Система отображает форму для создания новой учётной записи. После нажатия кнопки «Зарегистрироваться» система проверяет уникальность email-адреса и корректность данных. В случае ошибок выводится поясняющее сообщение. При успешной регистрации пользователь автоматически авторизуется в системе и перенаправляется на экран профиля. |
| 1.3 | Профиль | - | Отображает персональные данные пользователя, его роль в системе. |
| 2 | Студии |  |  |
| 2.1 | Список студий | - | Экран отображает перечень студий, доступных пользователю. Пользователь может выбрать студию для просмотра подробной информации. |
| 2.2 | Карточка студии | - | Экран содержит подробную информацию о студии, включая описание студии, оборудование, вместимость. При редактировании данных студии выполняется проверка корректности введённых значений. После сохранения данных система обновляет информацию на экране и отображает уведомление об успешном выполнении операции. |
| 3 | Занятия |  |  |
| 3.1 | Просмотр расписания | Дата (ДД.ММ.ГГГГ) | Экран отображает календарь занятий в выбранном временном диапазоне. Пользователь может переключать даты и фильтровать данные. При смене периода выполняется повторная загрузка данных. В случае отсутствия занятий отображается соответствующее сообщение. |
| 3.2 | Запись на занятие | Дата (ДД.ММ.ГГГГ) Время (ЧЧ: ММ - ЧЧ: ММ) | Экран предназначен для записи ученика на выбранное занятие или свободный временной слот. При открытии экрана система отображает информацию о доступных датах, времени и студиях. Пользователь выбирает подходящий временной интервал и подтверждает намерение записаться. При выборе параметров система в реальном времени проверяет доступность выбранного слота и отсутствие конфликтов с уже запланированными занятиями. После корректного заполнения полей и нажатия кнопки «Записаться» система сохраняет предварительную запись и выполняет переход на экран подтверждения записи с оплатой. В случае ошибок или недоступности выбранного времени пользователь получает уведомление с пояснением причины. |
| 3.3 | Подтверждение | Номер карты (16 цифр)  Срок действия (ММ/ГГ)  CVV (3 цифры) | Подтверждения выбранного занятия и выполнения оплаты. Пользователю отображается сводная информация о записи: дата, время, стоимость занятия. После подтверждения выполняется проверка корректности введённых платёжных данных и инициируется процесс оплаты. В случае успешной оплаты система фиксирует запись как подтверждённую, отображает уведомление об успешной операции и автоматически перенаправляет пользователя на экран расписания. При ошибке оплаты система отображает соответствующее сообщение и предоставляет возможность повторить попытку. |
| 4 | Администрирование |  |  |
| 4.1 | Аналитика | - | На этой странице администратор выбирает студию, настраивает фильтр. После указания необходимых параметров администратор выбирает действие «Сформировать отчет», после этого формируется отчет и располагается в блоке «Готовый отчет» |
| 4.2 | Управление студиями | Название студии (>0)  Адрес (>0) Вместимость (>0) Стоимость/час (>0)  Описание помещения (>0) | Администратору предоставляется форма для ввода основных реквизитов студии. При заполнении формы осуществляется клиентская и серверная валидация данных. После нажатия кнопки «Создать студию» система проверяет уникальность названия студии. В случае успешного добавления студия сохраняется в базе данных, и администратор автоматически перенаправляется на экран управления студиями. |

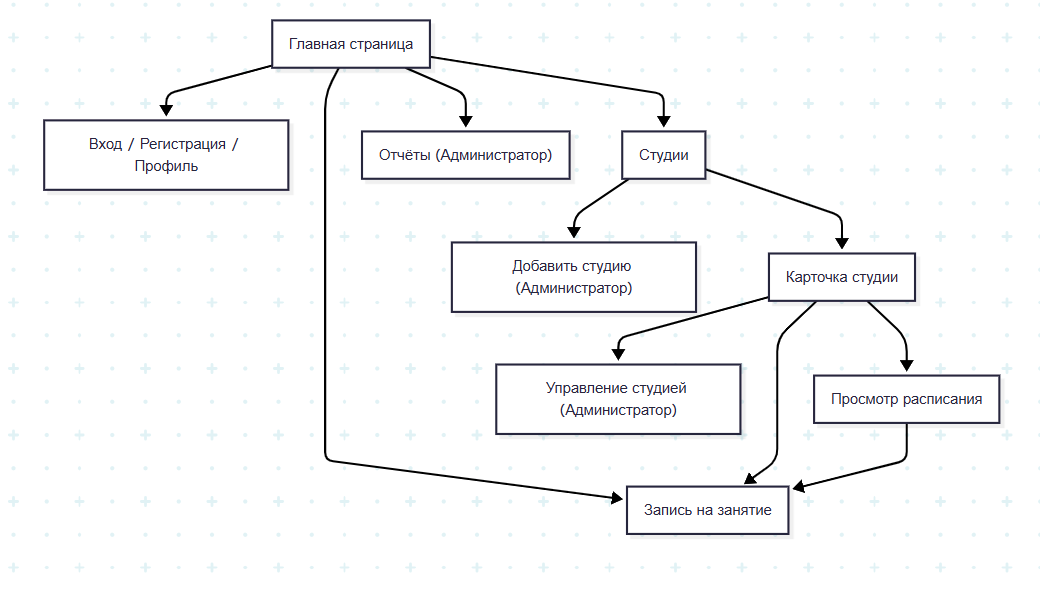
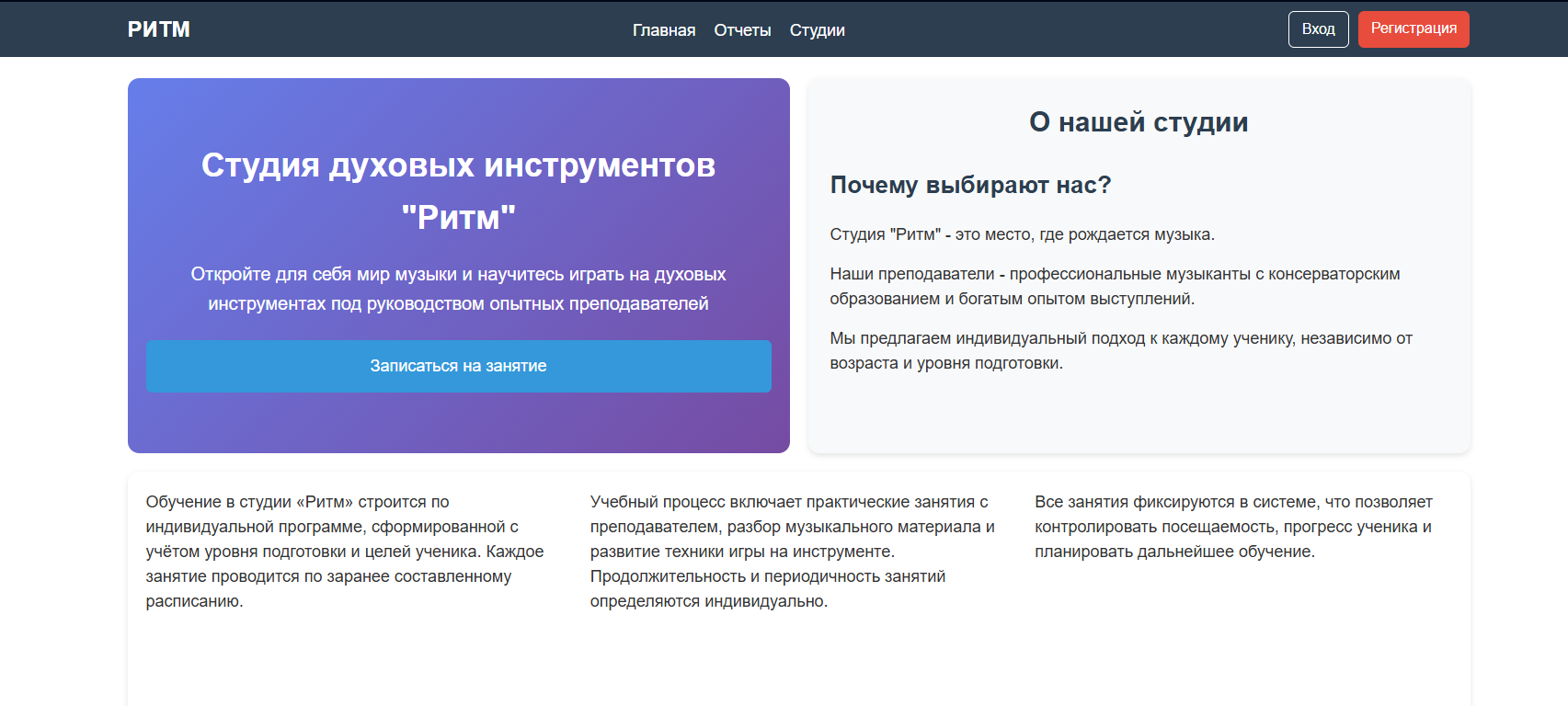


Рисунок 2.5 – Карта экранов

3 Разработанная система

3.1 Порядок авторизации пользователя в системе

Авторизация пользователей в системе выполняется через специальную веб‑форму входа, доступ к которой осуществляется по ссылке «Войти» в верхней части интерфейса главного экрана системы (см. рисунок 3.1).

  
Рисунок 3.1 – Главный экран системы

На странице авторизации отображаются поля для ввода адреса электронной почты и пароля, а также кнопка подтверждения входа (см. рисунок 3.2)

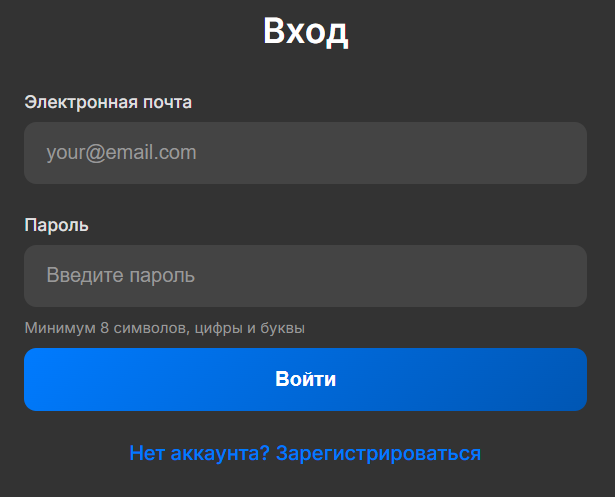


Рисунок 3.2 – Форма авторизации пользователя

Порядок авторизации пользователя в системе включает следующие шаги:

* пользователь открывает веб‑интерфейс системы и переходит по ссылке «Войти» на страницу авторизации;
* в соответствующие поля формы e‑mail и пароль, указанные при регистрации;
* после нажатия кнопки входа введённые данные передаются на сервер для проверки;
* сервер выполняет поиск учётной записи по введённому логину и проверяет корректность пароля;
* при успешной проверке создаётся пользовательская сессия, определяется роль пользователя после чего пользователь перенаправляется в основной интерфейс системы с доступом к соответствующим разделам;
* при неверном логине или пароле на странице выводится сообщение об ошибке авторизации с предложением повторить ввод данных (см. рисунок 3.3).

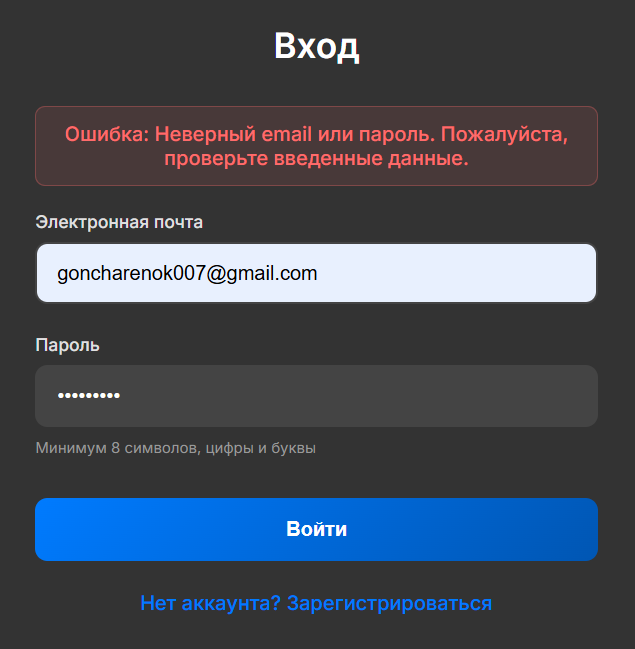


Рисунок 3.3 – Сообщение о неверном логине/пароле

Форма авторизации также содержит ссылку для перехода к регистрации нового пользователя, что позволяет клиентам, не имеющим учётной записи, самостоятельно создать доступ к системе (см. рисунок 3.4).

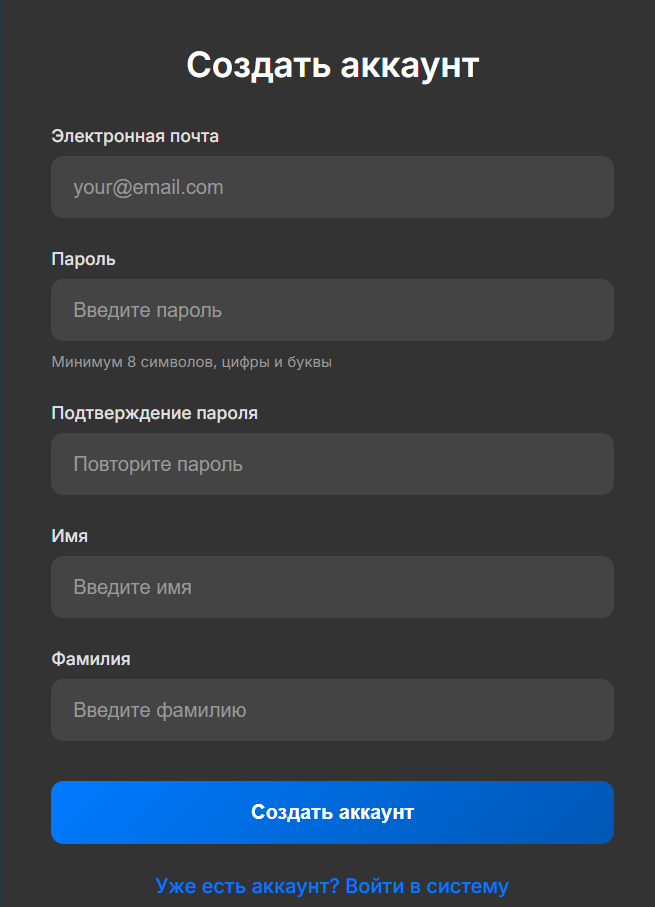


Рисунок 3.4 – Форма регистрации пользователя

3.2 Проектные решения для интерфейса

После успешной авторизации пользователя верхнее меню изменится и будет включать в себя пункты, отражённые на рисунках 3.5 и 3.6.

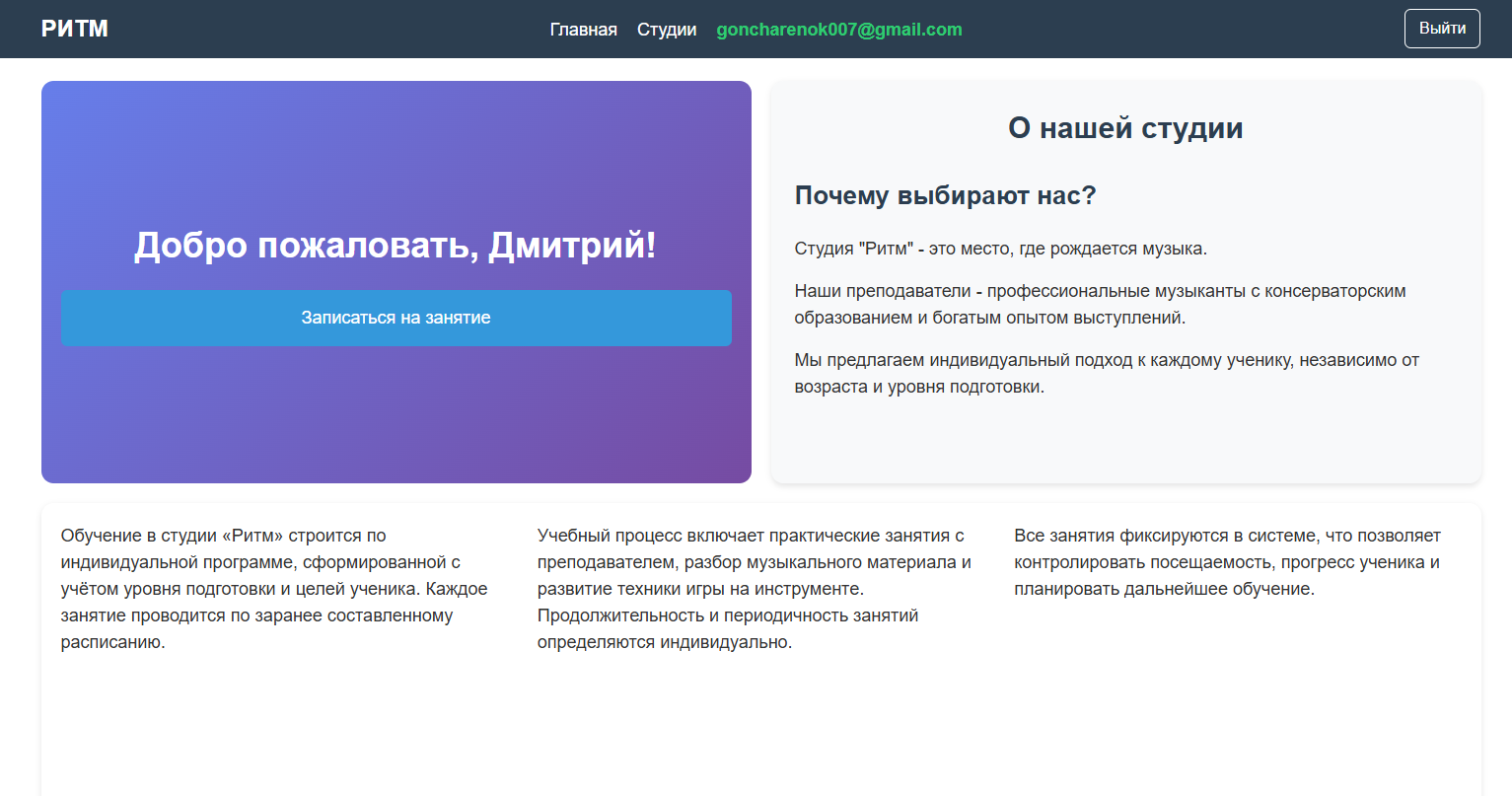


Рисунок 3.5 – Главная страница от лица пользователя

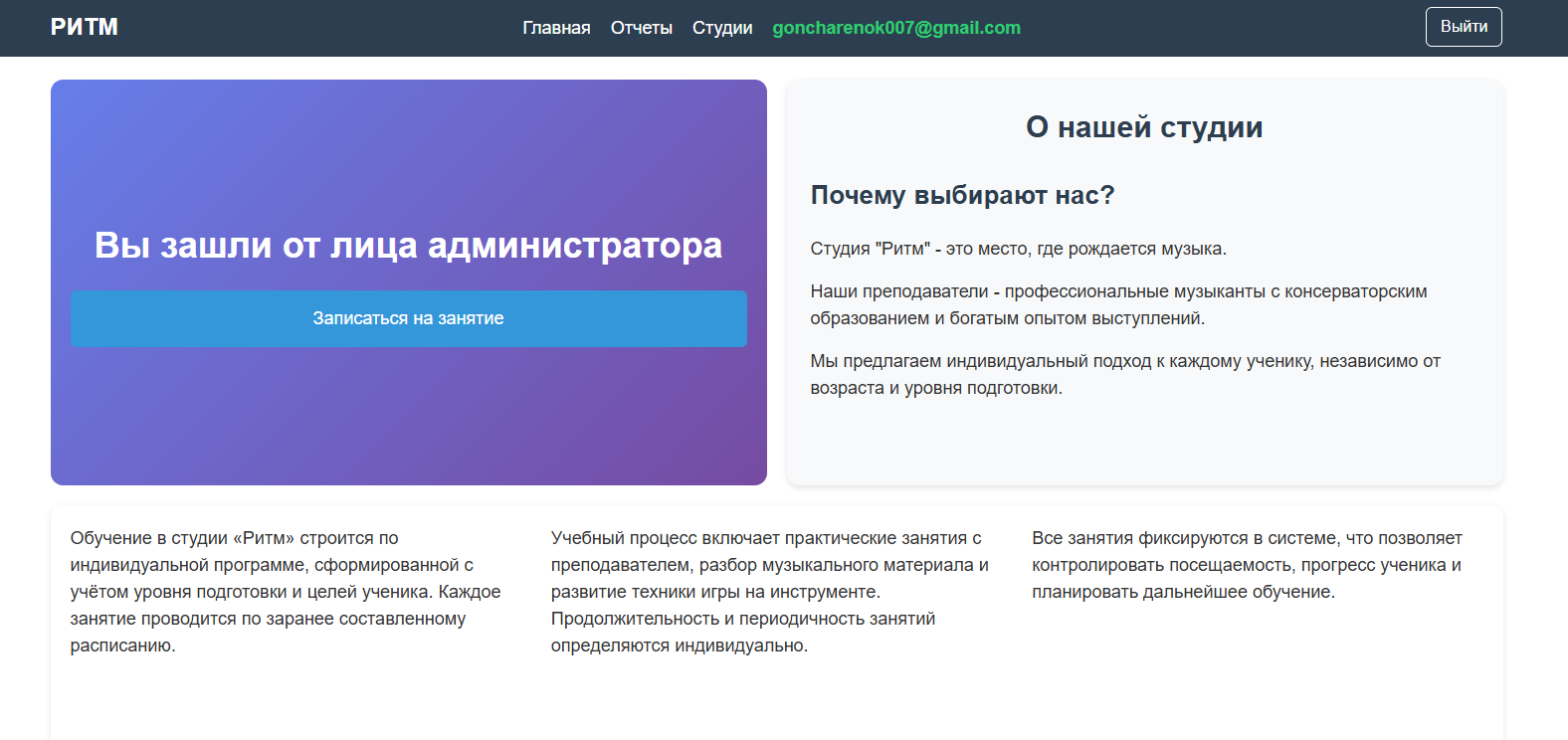


Рисунок 3.6 – Главная страница от лица администратора

На данной странице в верху располагается меню, включающее следующие возможности:

* «Главная» – кнопка для перехода на главную страницу.
* «Ваш email» – кнопка для перехода к личному кабинету.
* «Отчёты» – кнопка для перехода к странице формирования отчётов о деятельности (не доступна обычным пользователям). На ней можно сформировать отчеты.
* «Студии» – кнопка для перехода на страницу с существующими студиями.

Для перехода на страницу со студиями нужно нажать на кнопку «Студии» или «Записаться на занятие» (см. рисунок 3.7).

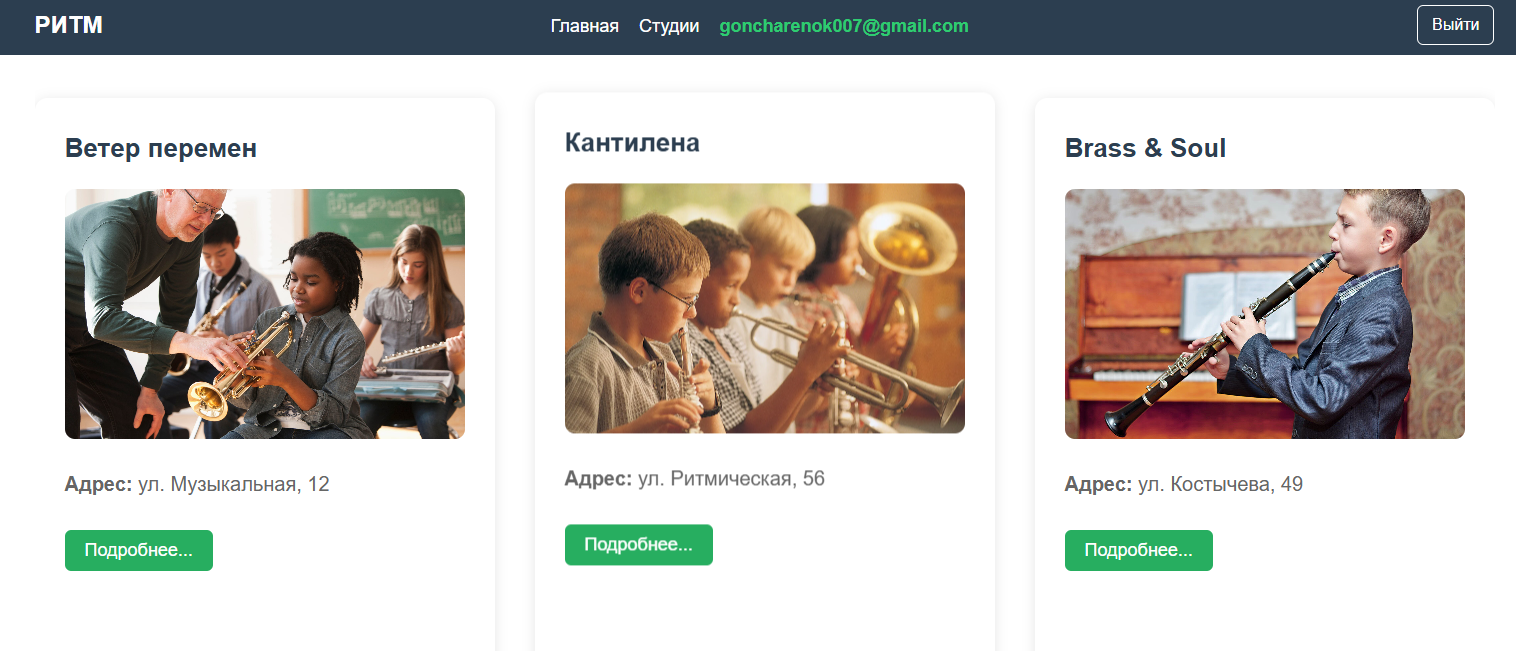


Рисунок 3.7 – Страница доступных студий

На этой странице отображается список всех студий. Каждая студия представлена в виде карточки, содержащей информацию об адресе и названии.

На странице для администратора доступна дополнительная функция с добавлением студии (см. рисунок 3.8):

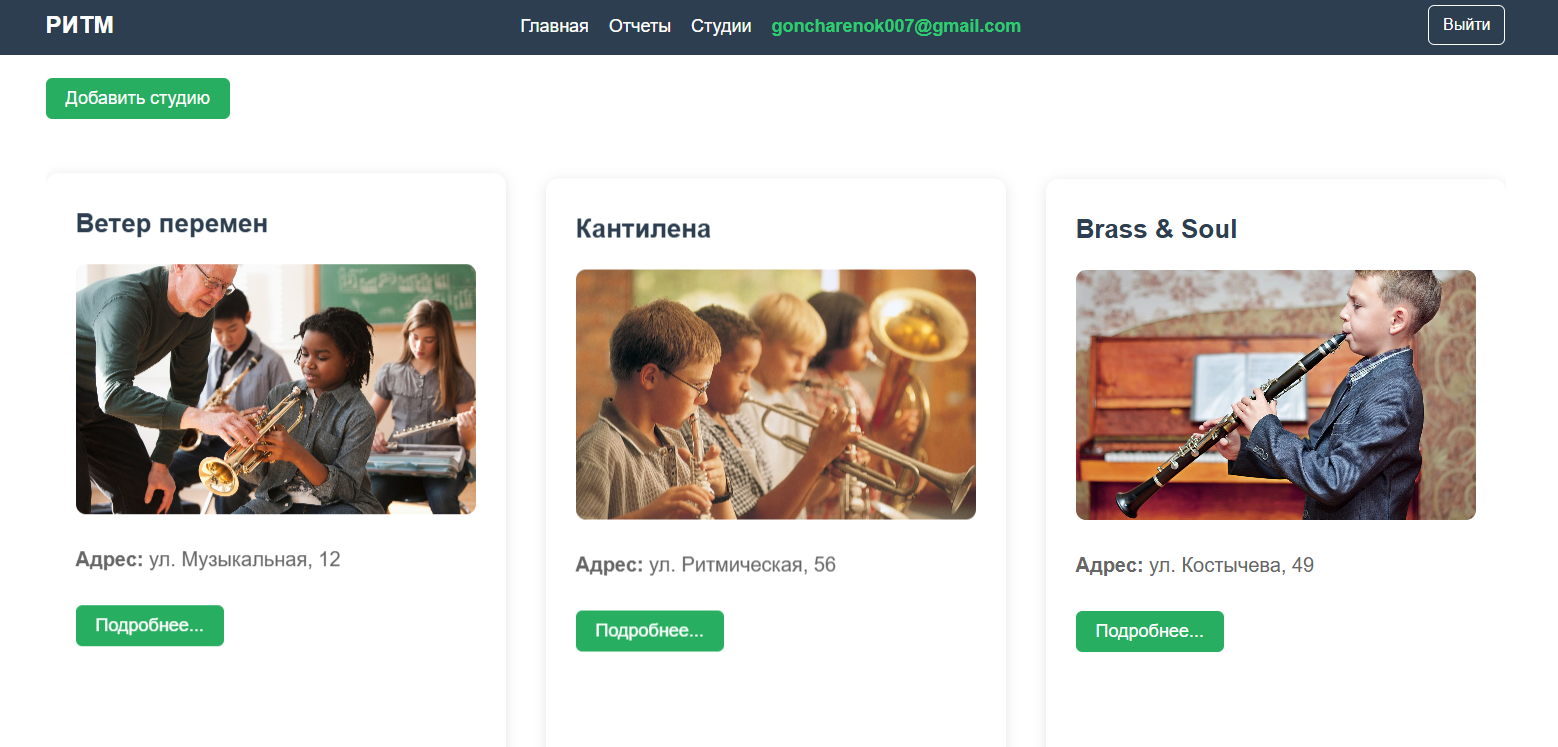


Рисунок 3.8 – Страница доступных студий от лица администратора

При нажатии кнопки «Добавить студию» открывается страница с полной информацией для добавления новой студии (см. рисунок 3.9). Студию нельзя добавить, пока не будут заполнены все поля.

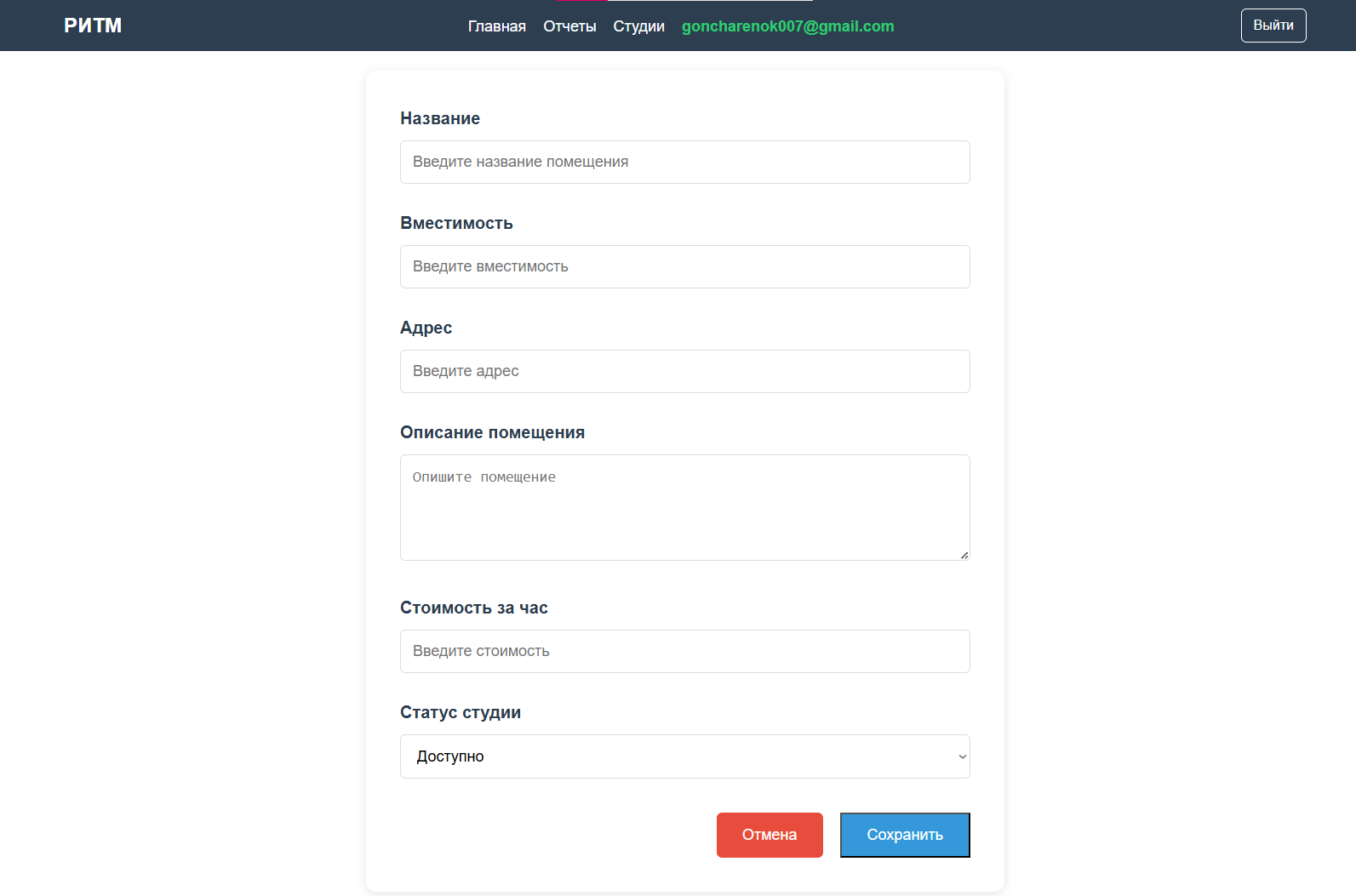


Рисунок 3.9 – Страница с добавлением новой студии

При успешном добавлении новая студия отображается на странице (см. рисунок 3.10).

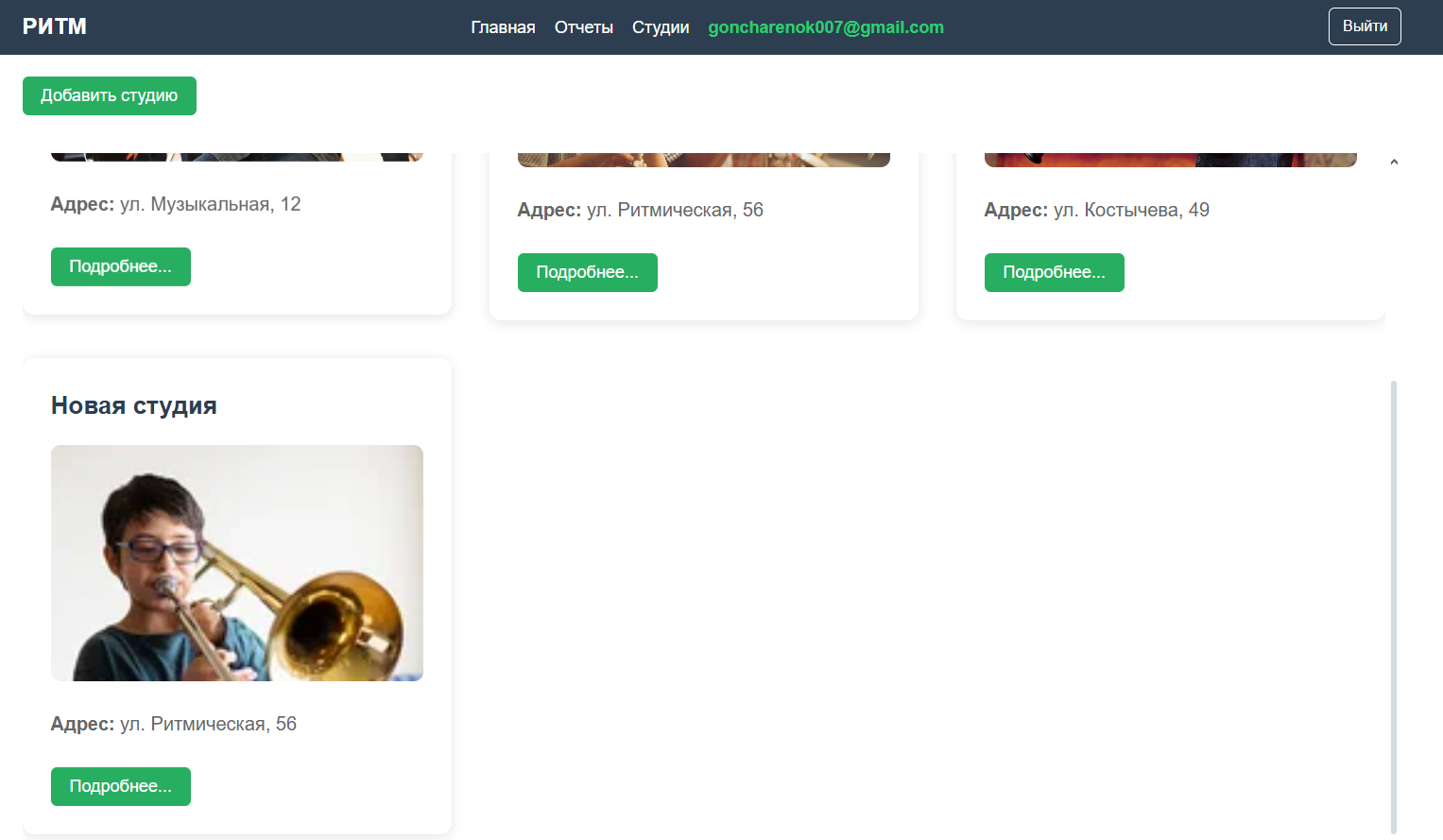


Рисунок 3.10 – Отображение добавленной студии

При нажатии кнопки «Подробнее» открывается страница с выбранной студией. На этой странице отображается список оборудования, описание студии и вместимость помещения. (см. рисунок 3.11);

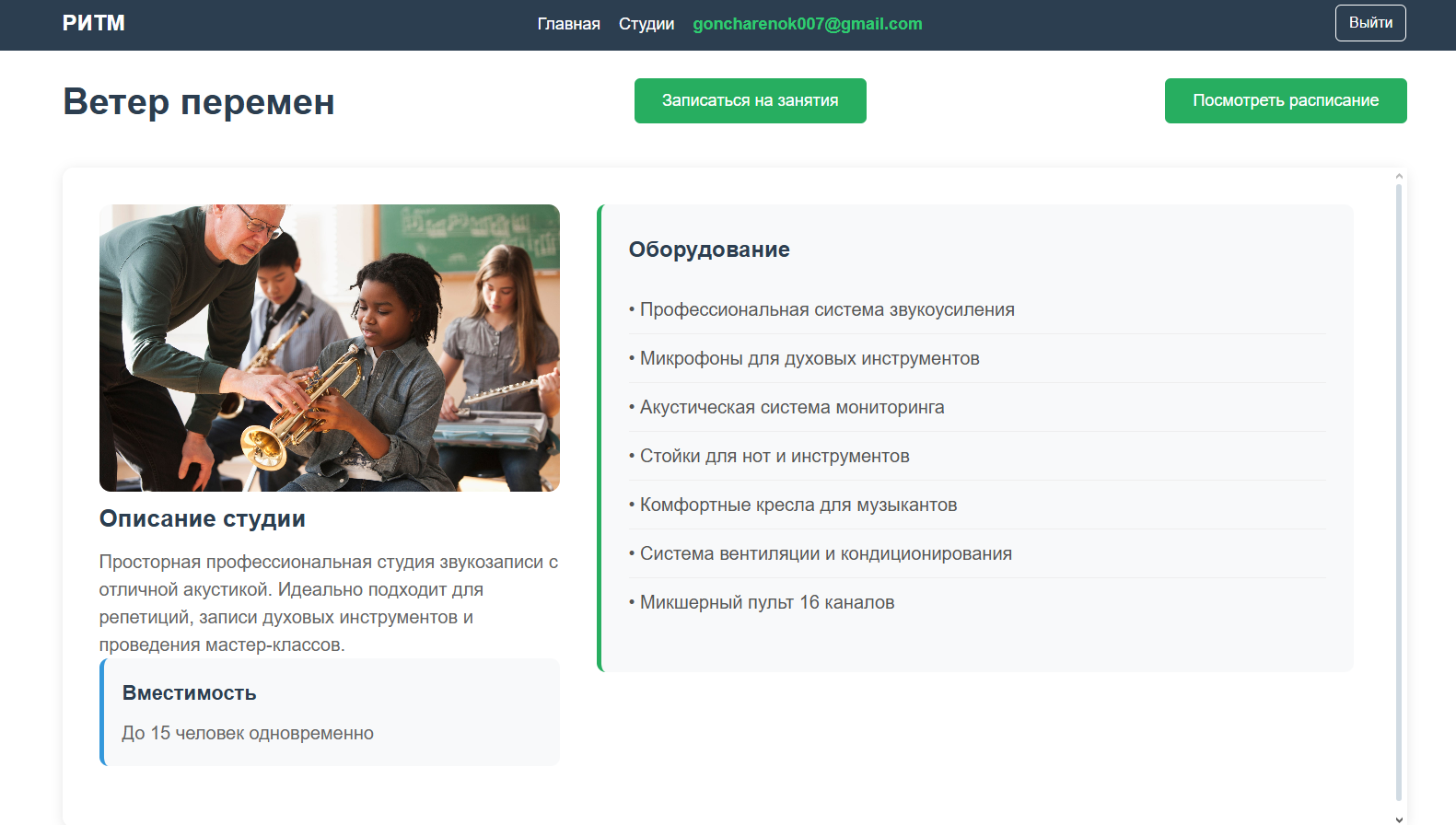


Рисунок 3.11 – Страница студии

При нажатии кнопки «Записаться на занятие» открывается форма с интерактивным календарем (см. рисунок 3.12), где красным цветом помечены даты, когда студия закрыта или все временные слоты заняты. Под календарем расположены кнопки с временными интервалами, что позволяет пользователю быстро выбрать удобное время

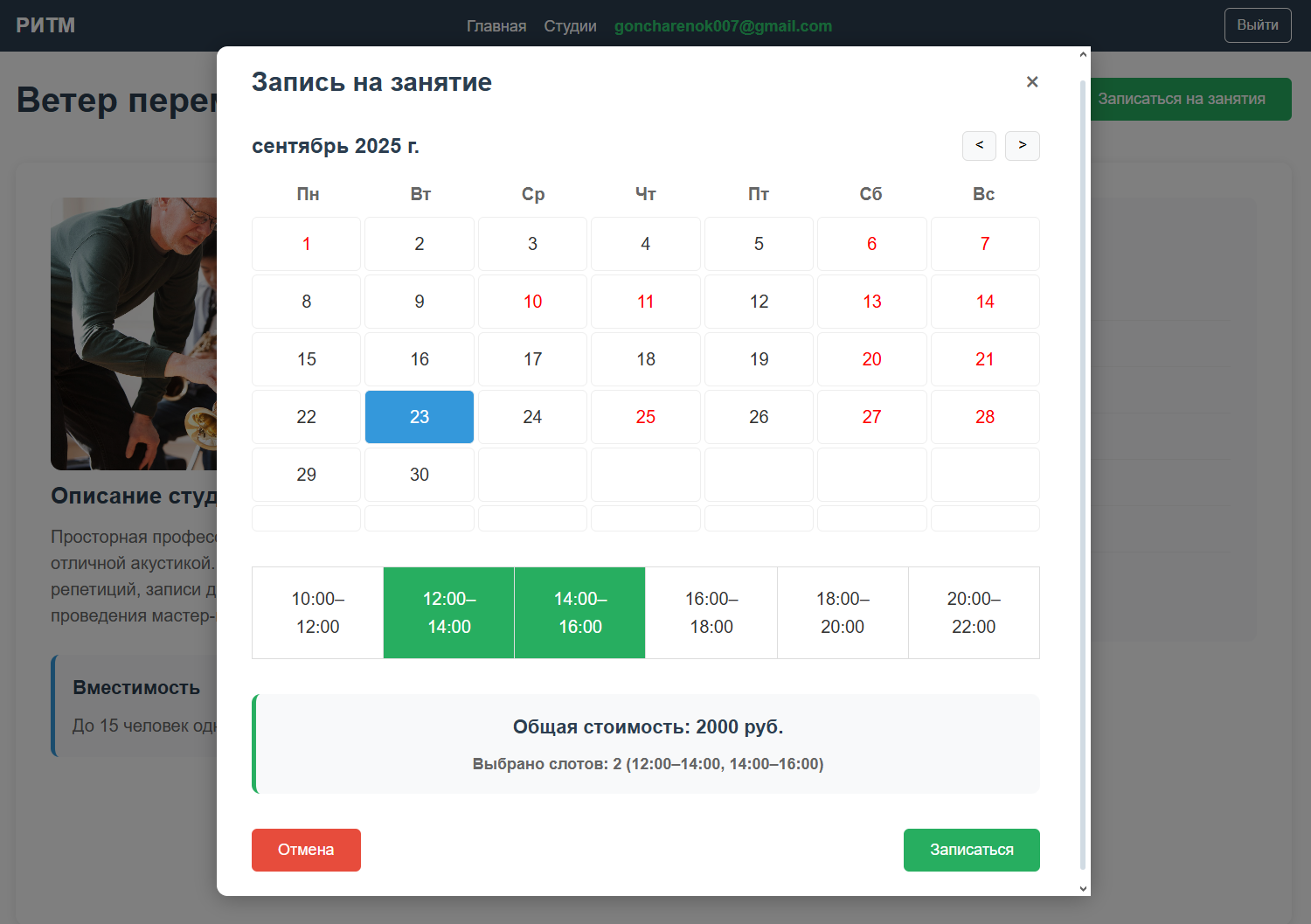
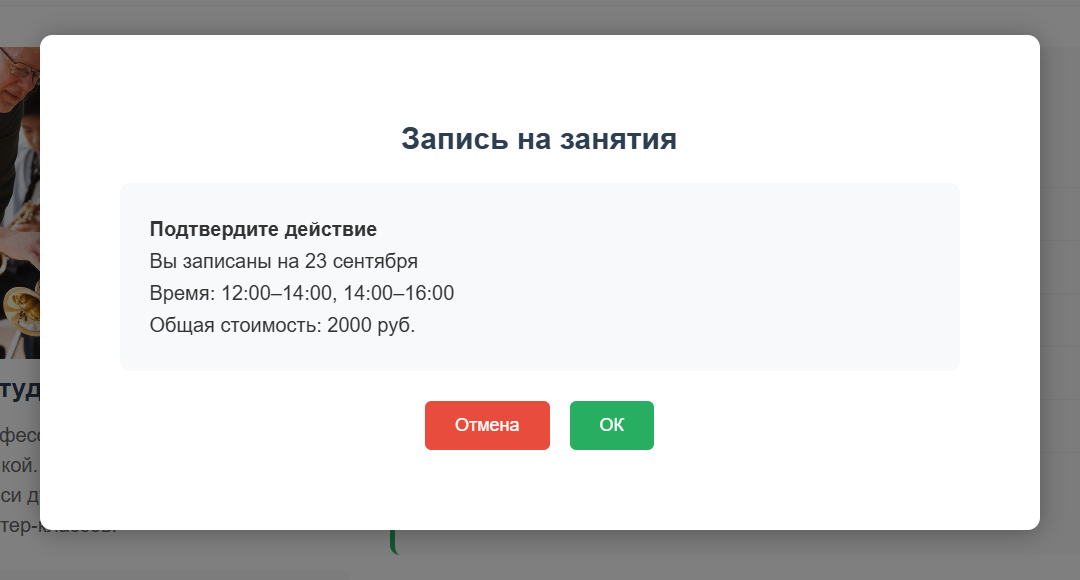


Рисунок 3.12 – Форма записи на занятие

После выбора студии, даты и времени открывается форма для подтверждения записи (см. рисунок 3.13). На форме выводятся основные параметры записи: дата, время и стоимость. Пользователь может либо подтвердить запись, нажав кнопку «Ок», либо отказаться, нажав кнопку «Отмена».​

  
Рисунок 3.13 – Форма записи на занятие

При нажатии на кнопку «Расписание занятий», открывается форма (см. рисунок 3.14), на который видно полностью расписание на неделю для конкретной студии, с возможностью листать недели и, нажав кнопку «Записаться на занятие», произвести запись.

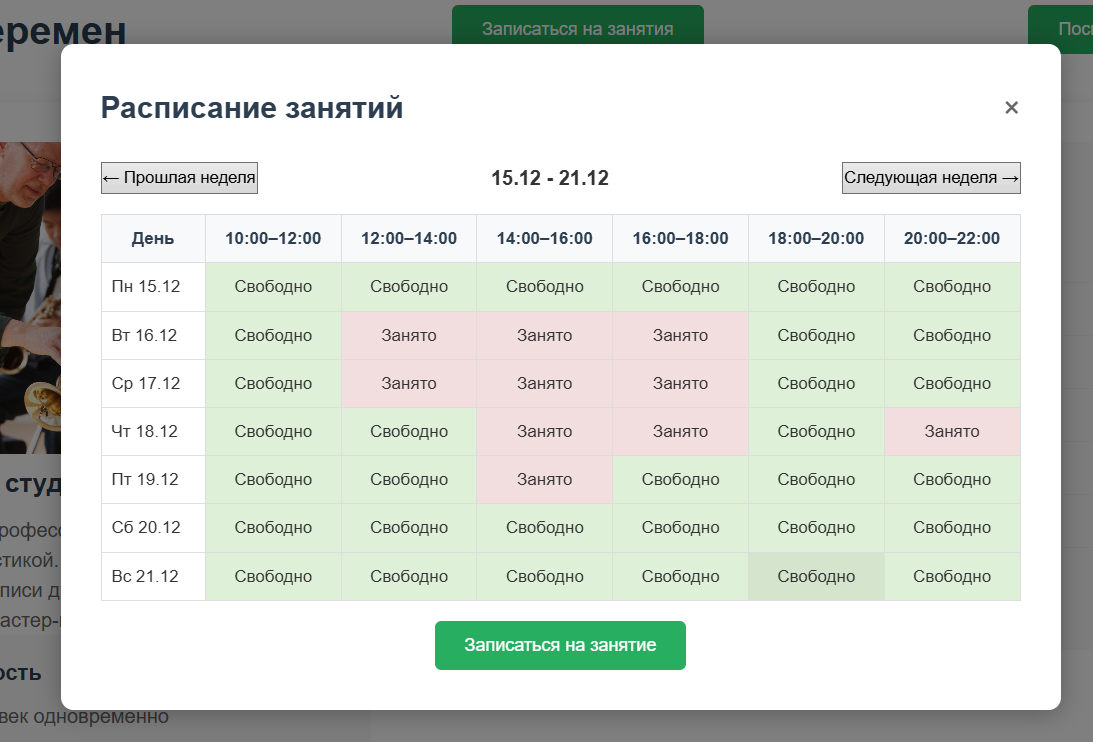


Рисунок 3.14 – Форма расписания занятия

Для получения отчета финансовых показателей необходимо быть зарезервированным в системе с ролью администратор. Администратор при выборе элемента «Отчеты» в навигационном меню сайта переходит на страницу с отчетами (см. рисунок 3.15).

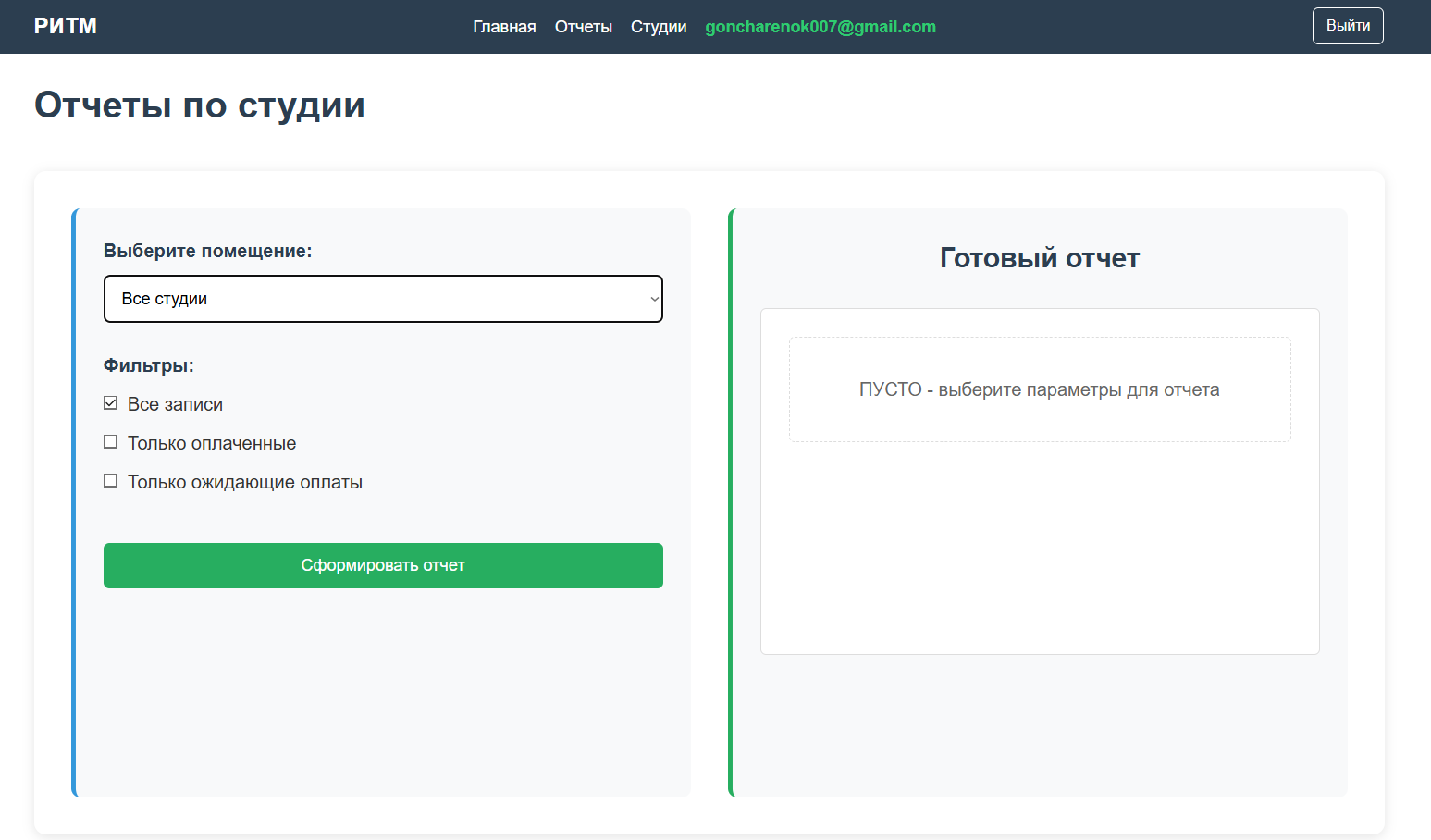


Рисунок 3.15 – Страница с пустым отчетом

На этой странице администратор выбирает студию, настраивает фильтр. После указания необходимых параметров администратор выбирает действие «Сформировать отчет», после этого формируется отчет и располагается в блоке «Готовый отчет» (см. рисунок 3.16).



Рисунок 3.16 – Страница с готовым отчетом

Заключение

В ходе выполнения курсовой работы были разработаны проектные решения для веб-ориентированной информационной системы «РИТМ», предназначенной для автоматизации деятельности музыкальной студии. Основной целью проекта являлось проектирование архитектуры системы, обеспечивающей управление студиями, расписанием занятий, учётом учеников и административными функциями.

В процессе работы была спроектирована многоуровневая архитектура приложения, разработана структура пользовательского интерфейса и карта экранов, а также выполнено проектирование базы данных с приведением её к третьей нормальной форме. Принятые проектные решения обеспечивают целостность данных, удобство использования системы и возможность её дальнейшего развития.

Практическая значимость проекта заключается в упрощении процессов планирования и учёта занятий, повышении эффективности работы студии и создании единого информационного пространства для пользователей и администраторов. Разработанная система имеет потенциал для расширения функциональности и внедрения в деятельность музыкальных студий и образовательных центров.

Список использованных источников

1. ГОСТ Р 7.0.100-2018. Библиографическая запись. Библиографическое описание. Общие требования и правила составления [Электронный ресурс]. – URL: [вставить ссылку на стандарт] (дата обращения: 17.12.2025).
2. Гутгарц Р.Д. Проектирование автоматизированных систем обработки информации и управления : учебное пособие для вузов. — Москва : Юрайт, 2024. — 351 с.
3. Проектирование АСОИУ [Электронный ресурс] : методические указания по выполнению курсового проекта / сост. Р. Д. Гутгарц. — Иркутск : ИРНИТУ, 2018. — URL: [вставить ссылку на методичку] (дата обращения: 17.12.2025).
4. Проектирование информационных систем : учебник и практикум для академического бакалавриата / Д. В. Чистов и др. ; под общ. ред. Д. В. Чистова. — Москва : Юрайт, 2022. — 315 с.
5. Гвоздева Т.В. Проектирование информационных систем. Стандартизация : учебное пособие / Т. В. Гвоздева, Б. А. Баллод. — 2-е изд., стер. — Санкт-Петербург : Лань, 2021. — 256 с.