Техническое задание на разработку Веб-приложения «Танцевальная студия»

1. Введение
   1. Цель

Целью разработки данной системы является автоматизация процессов управления танцевальной студией, включая регистрацию клиентов, управление расписанием занятий, записи на занятие и обновление актуальной информации.

* 1. Актуальность и назначение проекта

В условиях современного рынка танцевальных услуг существует высокая конкуренция, что требует от студий внедрения эффективных инструментов управления. Автоматизированная информационная система позволит:

* Ускорить процессы регистрации клиентов и управления расписанием, что повысит уровень обслуживания и удовлетворенность клиентов.
* Обеспечить удобный доступ к информации для клиентов и преподавателей, что повысит вовлеченность и лояльность.
  1. Основные пользователи системы

Система предназначена для использования сотрудниками компании:

Администратор - создание и управление направлениями танцев, управление пользователями, регистрация преподавателя, управление информацией сайта.

Преподаватель – запись посетителя на занятие, просмотр расписания, просмотр персональных данных с возможностью редактирования.

Посетитель – просмотр расписания, просмотр каталога направлений и преподавателей, оставление отзыва, просмотр персональных данных с возможностью редактирования.

1. Цели и задачи проекта
   1. Основные цели

* Автоматизация и централизованное хранение данных по направлениям и пользователям.
  1. Конкретные задачи
* разработка функциональности для администратора;
* разработка функциональности для пользователя;
* разработка функциональности для преподавателя.

1. Требования к системе
   1. Функциональные требования
2. Разработка функциональности для администратора:

* создания, редактирования и удаления направлениями танцев;
* просмотр данных пользователей;
* регистрирование преподавателей;
* обновление информации на сайте.

1. Разработка функциональности для пользователя:

* просмотр доступных направлений танцев;
* просмотр расписания;
* запись на занятие, при успешной записи должно приходить письмо с подтверждением на почту;
* регистрация;
* возможность оставления отзыва о студии;
* возможность редактирования персональных данных.

1. Разработка функциональности для преподавателя:

* просмотр своего расписания;
* просмотр персональных данных с возможностью редактирования.
  1. Нефункциональные требования
* производительность: отклик интерфейса не более 2 секунд при загрузке направлений танцев.

1. Требования к пользовательскому интерфейсу
   1. Основные экраны

* главная страница: отображение списка направлений танцев, форма для оставления отзыва;
* страница направления**:** подробная информация о направлении, описание, преподаватели данного направления;
* страница с контактными данными: месторасположение в виде карты, контактная информация;
* личный кабинет пользователя: информация о пользователе, история записей, расписание занятий;
* панель преподавателя: просмотр расписания;
* панель администратора: просмотр данных пользователя, регистрация преподавателя, управление направлениями, управление данными страниц.
  1. Навигация
* система должна включать меню навигации для пользователя с разделами: "Главная", "Страницы", "Контакты", "Личный кабинет";
* система должна включать меню навигации для администратора с разделами: "Пользователи", "Направления", "Данные страницы";
* система должна включать меню навигации для преподавателя с разделами: "Мое расписание", "Личный кабинет".
  1. Юзабилити
* Простой и понятный интерфейс для быстрого доступа к направлениям и расписанию для записи на занятие.

1. Требования к технической реализации
   1. Языки и технологии

* backend: Laravel, JavaScript.
* frontend: Laravel.
* база данных: SQL.
  1. Архитектура системы
* хранение данных в реляционной базе данных
  1. Интеграции
* внешние уведомления: интеграция с сервисом email-уведомлений

1. Требования к безопасности
   1. Аутентификация и авторизация

* авторизация: разграничение прав доступа — администраторы, преподаватели, посетители.
  1. Шифрование данных
* хранение паролей: хэширование через bcrypt.
  1. Политики доступа
* Защита от SQL-инъекций и XSS.

1. Ограничения и допущения
   1. Технические ограничения
   2. Финансовые ограничения

* бюджет проекта: до 100 ₽, включая тестирование и документацию.
  1. Сроки выполнения
* полная реализация проекта — 1,5 месяца с начала разработки.

1. Требования к тестированию и приемке
   1. Типы тестирования

* функциональное тестирование: проверка всех реализованных функций;
* тестирование безопасности: проверка на уязвимости.
  1. Критерии приемки
* полное выполнение всех функциональных и нефункциональных требований;
* прохождение не менее 75% тестов по результатам приемочных испытаний.

1. Требования к документации
   1. Пользовательская документация

* руководство пользователя с инструкциями по навигации сайта, записи на занятия, просмотра и редактирования персональных данных.
  1. Техническая документация
* описание архитектуры, структура базы данных, API.

1. План реализации
   1. Этапы разработки
2. анализ и проектирование — 2 недели;
3. разработка прототипа — 1 неделя;
4. полноценная разработка и тестирование — 5 недели;
5. внедрение и обучение — 2 недели.
   1. Сроки выполнения этапов
6. дата начала: 28 октября 2024 года;
7. дата завершения: 20 декабря 2024 года.

10.3 **Ответственные лица**

Менеджер проекта: Миронова Н.Е.

Ведущий разработчик: Миронова Н.Е.

Диаграмма UseCase

1 UML диаграммы

UML (с английского аббревиатура расшифровывается как UnifiedModelingLanguage — унифицированный язык моделирования) — это способ наглядно описать архитектуру, проектирование и реализацию комплексных программных систем.

Диаграмма прецедентов (диаграмма вариантов использования) -диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Диаграмма прецедентов представлена на рисунке 1.

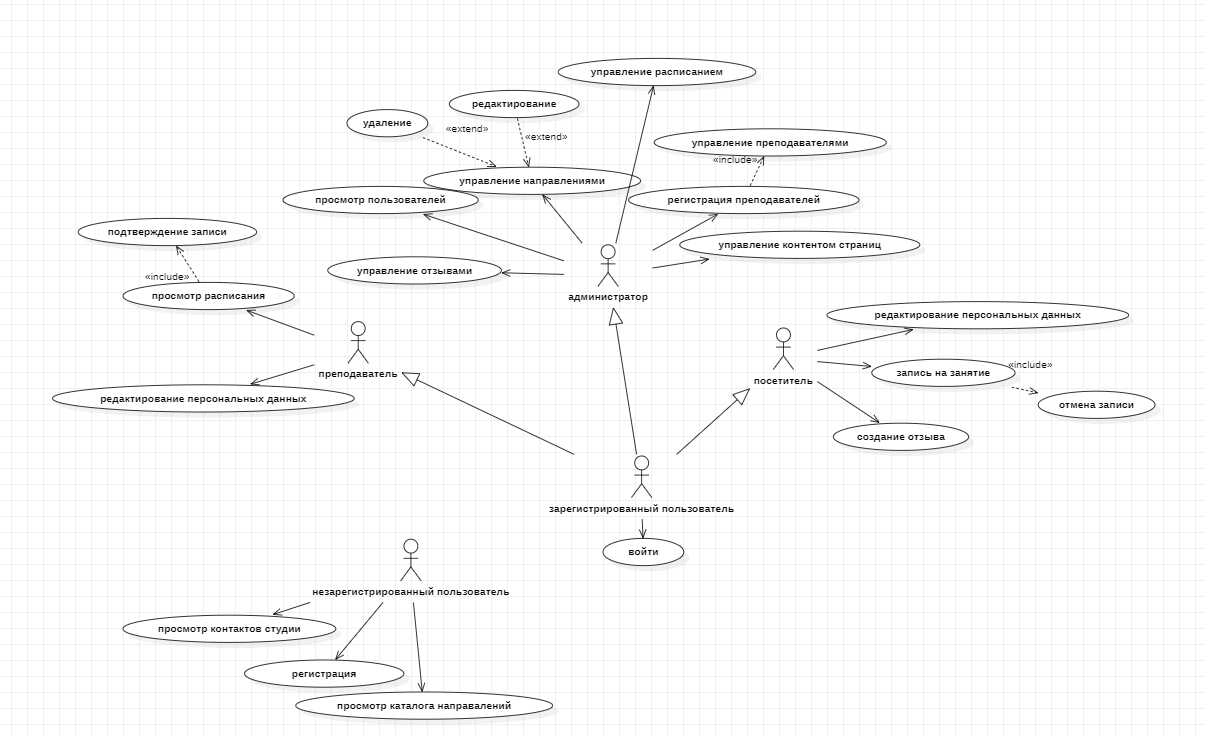


Рисунок 1 — Диаграмма прецедентов

2 Описание структуры базы данных

Схема отношений базы данных представлена на рисунке 2.

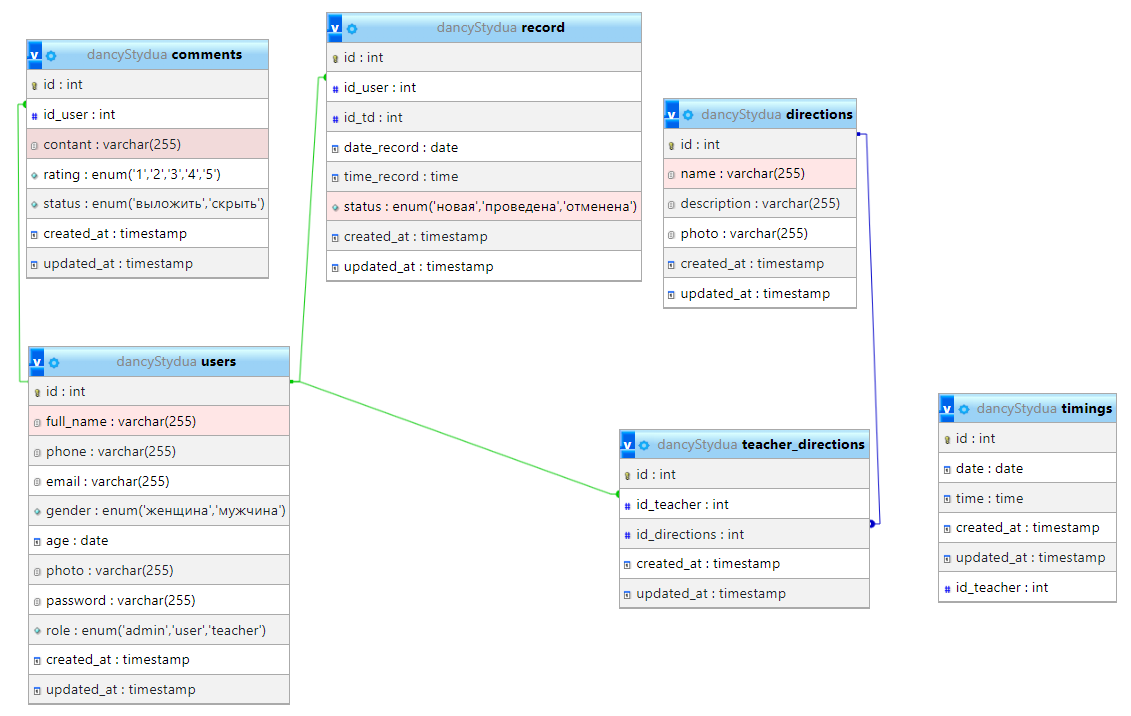


Рисунок 2 – Схема отношений базы данных

Структура таблицы users представлена в таблице 2.1.

Таблица 2.1. - Users(Пользователи)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код пользователя | Id | INTEGER(4) | Первичный ключ, уникальное значение |
| Почта пользователя | email | VARCHAR(255) | Обязательное поле, уникальное значение |
| ФИО | Full\_ame | VARCHAR(255) | Обязательное поле |
| Телефон | phone | VARCHAR(255) | Обязательное поле |
| Пол | gender | ENUM(женщина, мужчина) | Обязательное поле |
| Возраст | age | Date | Обязательное поле |
| Фото | photo | VARCHAR(255) | NULL |
| Пароль | password | VARCHAR(255) | Обязательное поле |
| Роль | role | ENUM(admin, user,teacher) | Обязательное поле |

Структура таблицы directions представлена в таблице 2.2.

Таблица 2.2 - Directions (Направления)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код направления | Id | INTEGER(4) | Первичный ключ,  уникальное значение |
| Название | name | VARCHAR(255) | Обязательное поле |
| Описание | description | VARCHAR(255) | Обязательное поле |
| Фото | photo | VARCHAR(255) | NULL |

Структура таблицы record представлена в таблице 2.3.

Таблица 2.3 – Record(запись)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код записи | Id\_record | INTEGER(4) | первичный ключ, уникальное значение |
| Код пользователя | Id\_user | INTEGER(4) | Внешний ключ (к Users) |
| Код\_направления преподавателя | Id\_td | INTEGER(4) | Внешний ключ (к teacher\_direction) |
| Дата записи | Date\_record | DATE | Обязательное поле |
| Время записи | Time\_record | TIME | Обязательное поле |
| Статус | status | ENUM(новая, проведена, отменена) | Обязательное поле |

Структура таблицы comment представлена в таблице 2.4.

Таблица 2.4-Comment(отзывы)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код отзыва | id | INTEGER(4) | Первичный ключ,уникальное значение |
| Код пользователя | Id\_user | INTEGER(4) | Внешний ключ (к Users) |
| Описание | contant | VARCHAR(255) | Обязательное поле |

Структура таблицы teacher\_direction представлена в таблице 2.5.

Таблица 2.5- Teacher\_direction (Направления преподавателей)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код | id | INTEGER(4) | Первичный ключ,уникальное значение |
| Код\_преподавателя | Id\_teacher | INTEGER(4) | Внешний ключ (к Users) |
| Код направления | id\_direction | INTEGER(4) | Внешний ключ (к directions) |

Структура таблицы timing представлена в таблице 2.6.

Таблица 2.6- Timing (Расписание)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код | id | INTEGER(4) | Первичный ключ,уникальное значение |
| Дата записи | record | DATE | Обязательное поле |
| Время записи | time | TIME | Обязательное поле |
| Преподаватель | Id\_teacher | INTEGER(4) | Внешний ключ (teacher\_direction) |