Техническое задание на разработку Веб-приложения «Аренда автомобилей»

1. Введение
   1. Цель

Целью разработки данной системы является автоматизация процессов управления арендой автомобилей, включая регистрацию клиентов, управление каталогом автомобилей, бронирование, а также обновление актуальной информации о доступных автомобилях и условиях аренды.

* 1. Актуальность и назначение проекта

В условиях современного рынка аренды автомобилей существует высокая конкуренция, что требует от компаний внедрения эффективных инструментов управления. Автоматизированная информационная система позволит:

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

Система предназначена для использования сотрудниками компании:

Администратор — создание и управление каталогом автомобилей, управление пользователями, обработка бронирований, управление информацией сайта.

Клиент — просмотр каталога автомобилей, бронирование автомобиля, оставление отзыва, просмотр персональных данных с возможностью редактирования.

1. Цели и задачи проекта
   1. Основные цели

* Автоматизация и централизованное хранение данных по автомобилям и пользователям.
  1. Конкретные задачи
* разработка функциональности для администратора.
* разработка функциональности для клиента.

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

* создание, редактирование и удаление записей об автомобилях;
* просмотр данных пользователей и истории бронирований;
* обработка запросов на аренду, включая подтверждение или отклонение;
* обновление информации на сайте;
* просмотр доступных автомобилей и их статуса (доступен/занят).

1. Разработка функциональности для клиента:

* просмотр доступных автомобилей с фильтрами по параметрам (марка, модель, цена и т.д.);
* бронирование автомобиля;
* регистрация и вход в личный кабинет.
  1. Нефункциональные требования
* производительность: отклик интерфейса не более 2 секунд при загрузке списка автомобилей.

1. Требования к пользовательскому интерфейсу
   1. Основные экраны

* главная страница: отображение списка доступных автомобилей с фильтрами;
* страница автомобиля: подробная информация о выбранном автомобиле, его характеристиках и условиях аренды;
* личный кабинет клиента: информация о клиенте, история бронирований;
* панель администратора: управление автомобилями, статистика бронирования, пользователями и данными страниц, просмотр заявок на аренду и управление ими.
  1. Навигация
* система должна включать меню навигации для клиента с разделами: "Главная", "Автопарк", "Контакты", "Личный кабинет";
* система должна включать меню навигации для администратора с разделами: "Категории", "Машины", "Аренда","Пользователи","Статистика" .
  1. Юзабилити
* простой и понятный интерфейс для быстрого доступа к каталогу автомобилей и процессу бронирования.

1. Требования к технической реализации
   1. Языки и технологии

* backend:Laravel.
* frontend: Laravel.
* база данных: SQL .
  1. Архитектура системы
* хранение данных в реляционной базе данных.

1. Требования к безопасности
   1. Аутентификация и авторизация

* авторизация: разграничение прав доступа — администраторы, клиенты.
  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.
   1. Ответственные лица

Менеджер проекта: Низамутдинов К.И.

Ведущий разработчик: Низамутдинов К.И.

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.

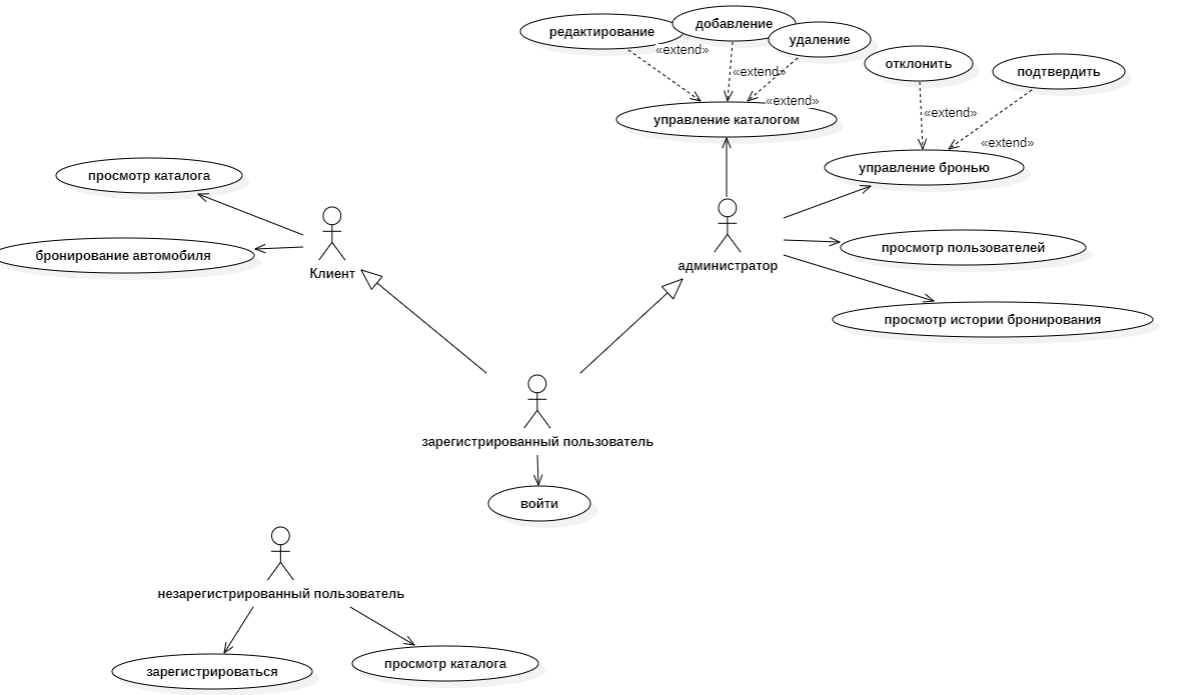


Рисунок 1 — Диаграмма прецедентов

2 Описание структуры базы данных

Схема отношений базы данных представлена на рисунке 2.

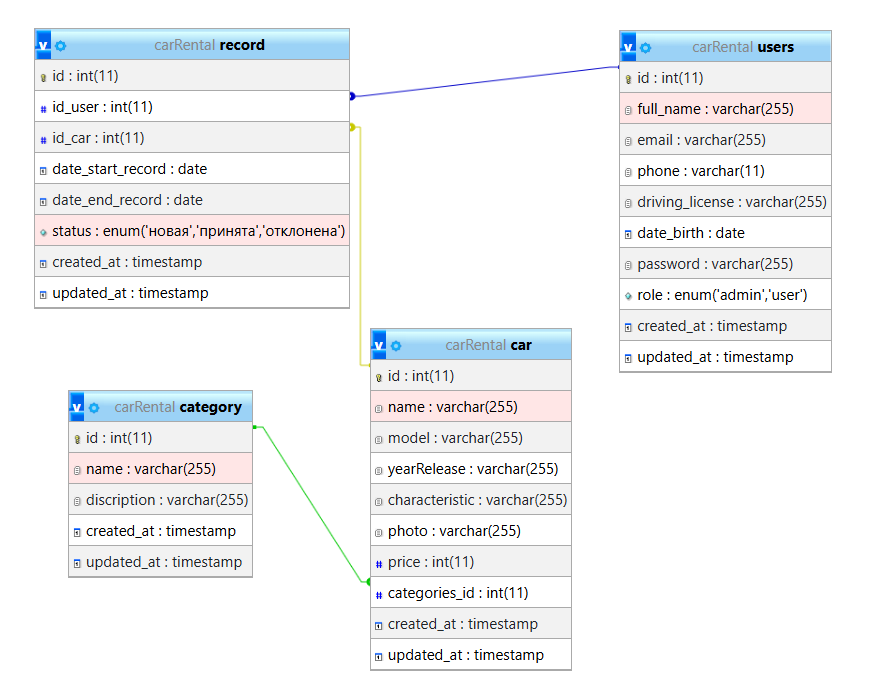


Рисунок 2 – Схема отношений базы данных

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 |
| Водительское удостоверение | Driving\_license | INTEGER(4) | Обязательное поле |
| Пароль | password | VARCHAR(255) | Обязательное поле |
| Роль | role | ENUM(admin, user) | Обязательное поле |

Структура таблицы record представлена в таблице 2.2.

Таблица 2.2 – 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\_машины | Id\_car | INTEGER(4) | Внешний ключ (к car) |
| Дата начала | date\_start\_record | DATE | Обязательное поле |
| Дата окончания | date\_end\_record | DATE | Обязательное поле |
| Статус | status | ENUM(новая, принята, отклонена) | Обязательное поле |

Структура таблицы car представлена в таблице 2.3.

Таблица 2.3 – Car(машина)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код | Id | INTEGER(4) | первичный ключ, уникальное значение |
| Название | name | VARCHAR(255) | Обязательное поле |
| Модель | model | VARCHAR(255) | Обязательное поле |
| Дата выпуска | yearRelease | VARCHAR(255) | Обязательное поле |
| Характеристики | characteristic | VARCHAR(255) | Обязательное поле |
| Фото | photo | VARCHAR(255) | NULL |
| Цена | price | INTEGER(4) | Обязательное поле |

Структура таблицы category представлена в таблице 2.4.

Таблица 2.3 – Category (Категории)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код | Id | INTEGER(4) | первичный ключ, уникальное значение |
| Название | name | VARCHAR(255) | Обязательное поле |
| Описание | discription | VARCHAR(255) | Обязательное поле |