МИНИСТЕРСТВО ОБРАЗОВАНИЯ и науки РЕСПУБЛИКИ БАШКОРТОСТАН

Государственное автономное ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

УФИМСКИЙ КОЛЛЕДЖ СТАТИСТИКИ, ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

ОТЧЕТ ПО ИНДИВИДУАЛЬНОМУ ПРОЕКТУ

На тему «Детейлинг и автомойка автомобилей»

По УП 09.01 Учебная практика

Выполнил студент группы 20ВЕБ1

Миргалиев Дамир Фидарисович

Проверил(а) преподаватель информатики

и программирования

Дмитриева Елизавета Константиновн

2023

СОДЕРЖАНИЕ

|  |  |
| --- | --- |
|  | лист |
| 1 Описание предметной области | 3 |
| 2 Описание входной информации | 4 |
| 3 Описание выходной информации  4 UML диаграммы | 4  5 |
| 5 Логическое моделирование | 5 |
| 6 Описание структуры базы данных | 5 |
| 7 Контрольный пример | 9 |
| 8 Общие требования к программному продукту | 10 |
| 9 Карта сайта | 13 |
| 10 Физическая схема | 13 |
| 11 Описание модулей | 13 |
| 12 Протоколы тестирования  13 Руководство пользователя | 15  20 |
| Приложение А. UML диаграммы | 26 |
| Приложение Б. Даталогическая модель БД  Приложение В. Физическая схема сайта | 27  29 |
| Список литературы | 31 |

1. Описание предметной области

Автомойка «Carte Blanche» занимается предоставление услуг по ремонту, мойке автомобилей. Клиенту необходимо необходимо выбрать нужную услугу и тем самым подать заявку. В случае принятия заявки, клиенту необходимо принести свой автомобилей в сервисный цент. После принятия заявки, техника передается мастеру, который занимается ее ремонтом. По окончании работ формируется документ, где описывается информация о технике, исправленной проблеме и, в случае физической неисправности, замененные детали.

В данной информационной системе присутствуют такие сущности, как:

* клиент;
* администратор.

О пользователе хранится следующая информация:

* код пользователя;
* имя пользователя;
* номер телефона;
* роль;
* почта;
* пароль;

В заявке храниться следующая информация

* код заявки;
* статус заявки;
* код клиента;
* код услуги;

В услуги храниться следующая информация

* код услуги;
* название;
* описание;
* фото.

В системе должна быть предусмотрена работа следующих ролей:

* клиент;
* администратор.

Клиент подает заявку на предоставляемую услугу.

Администратор просматривает заявки, изменяет их статус, занимается предварительным осмотром техники, добавляет и редактирует услуги.

Ограничения системы:

* администратор не может отклонить заявку после ее принятия;
* администратор не может подать заявку;
* клиент не может отправить заявку не авторизованным в системе.

1. Описание входной информации

Входная информация служит для осуществления деятельности системы, в которую входит заявка на предоставление услуги.

Описание входных документов представлено в таблице 2.1.

Таблица 2.1 – описание входных документов

|  |  |  |
| --- | --- | --- |
| Наименование данных | Дата поступления документа | Откуда поступает документ |
| Данные о заявки | При подаче | Клиент |
| Данные о услуге | При подаче | Администратор |

1. Описание выходной информации

Выходная информация - информация, которая возникает в результате обработки человеком или устройством входной информации.

В процессе деятельности автомойки формируется документ об окончании работ.

Описание выходных документов представлено в таблице 3.1.

Таблица 3.1 – описание выходных документов

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование данных | Периодичность выдачи документа | Кол-во экз. | Куда передаются |
| Данные об окончании работ | Один раз | 1 | Клиенту |

1. Use Case диаграмма (диаграмма прецедентов)

Диаграмма прецедентов – это диаграмма, которая позволяет визуализировать различные типы ролей и их взаимодействия с системой.

В данной диаграмме существует 4 объекта:

* актер;
* случай использования;
* система;
* пакет.

Актер в диаграмме прецедентов – это любая сущность, которая взаимодействует с системой. Это может быть человек, организация или внешняя система.

Случай использования представляет собой действие внутри системы.

Система используется для определения сферы применения.

Пакет является дополнительным параметром, который используется в сложных диаграммах. Пакеты используются для группировки большого количества случаев использования.

Каждая группа пользователей на диаграмме вариантов использования изображается человечком.

Диаграмма прецедентов представлена в приложении А, рисунок А.1

1. Логическое (даталогическое) проектирование

Под даталогической понимается модель, отражающая логические взаимосвязи между элементами данных безотносительно их содержания и физической организации.

Даталогическая модель представлена в приложении Б, рисунок Б.1

1. Описание структуры базы данных

База данных — это место для хранения данных. Используется в том числе в [клиент-серверной архитектуре](https://habr.com/ru/post/495698/).

Таблица 7.1 – структура таблицы «roles» (роли)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описания поля | Тип данных | Размер поля | Тип ключа (  PK – первичный, FK – внешний) |
| id | Уникальный код роли | BIGINTEGER | 8 | PK |
| type\_role | Наименование роли | VARCHAR | 150 |  |
| created\_ad | Дата создания | TIMESTAMP | 4 |  |
| updated\_ad | Дата обновления | TIMESTAMP | 4 |  |

Таблица 7.2 – структура таблицы «statuses» (статус заявки)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описания поля | Тип данных | Размер поля | Тип ключа (  PK – первичный, FK – внешний) |
| id | Уникальный код статуса | BIGINTEGER | 8 | PK |
| status\_type | Название статуса | VARCHAR | 255 |  |
| created\_ad | Дата создания | TIMESTAMP | 4 |  |
| updated\_ad | Дата обновления | TIMESTAMP | 4 |  |

Таблица 7.3 – структура таблицы «users» (пользователи)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описания поля | Тип данных | Размер поля | Тип ключа (  PK – первичный, FK – внешний) |
| id | Уникальный код пользователя | BIGINTEGER | 8 | PK |
| name | Имя пользователя | VARCHAR | 250 |  |
| telephone | Номер телефона пользователя | VARCHAR | 250 |  |
| role\_id | Код роли | BIGINTEGER | 8 | FK |
| email | Почта пользователя | VARCHAR | 250 |  |
| password | Пароль пользователя | VARCHAR | 250 |  |
| created\_ad | Дата создания | TIMESTAMP | 4 |  |
| updated\_ad | Дата обновления | TIMESTAMP | 4 |  |

Таблица 7.4 – структура таблицы «services» (услуги)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описания поля | Тип данных | Размер поля | Тип ключа (  PK – первичный, FK – внешний) |
| id | Уникальный код услуги | BIGINTEGER | 8 | PK |
| service\_name | Наименование услуги | VARCHAR | 250 |  |
| service\_desc | Описание услуги | VARCHAR | 250 |  |
| service\_photo | Фото услуги | VARCHAR | 250 |  |
| created\_ad | Дата создания | TIMESTAMP | 4 |  |
| updated\_ad | Дата обновления | TIMESTAMP | 4 |  |

Таблица 7.5 – структура таблицы «applications» (заявки)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описания поля | Тип данных | Размер поля | Тип ключа (  PK – первичный, FK – внешний) |
| id | Уникальный код заявки | BIGINTEGER | 8 | PK |
| status\_app | Код услуги | BIGINTEGER | 8 | FK |
| user\_id | Код клиента | BIGINTEGER | 8 | FK |
| service\_id | Код услуги | BIGINTEGER | 8 | FK |

Продолжение таблицы 7.6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| created\_ad | Дата создания | TIMESTAMP | 4 |  |
| updated\_ad | Дата обновления | TIMESTAMP | 4 |  |

1. Контрольный пример

Таблица 8.1 – таблица «role» (роль)

|  |
| --- |
| Наименование роли |
| Администратор |
| Клиент |

Таблица 8.2 – таблица «statuses» (статус заявки)

|  |
| --- |
| Наименование статуса |
| На рассмотрении |
| Принято |
| Отклонённое |

Таблица 8.3 – таблица «users» (пользователи)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Номер телефона | Роль | Почта | Пароль |
| Анатолий | 89170418423 | Клиент | anta@yandex.ru | Antka123 |
| Дамир | 89174112423 | Администратор | damir@gmail,com | Damir2 |
| Лена | 89271112423 | Клиент | Lena2@gmail.com | Lena2 |

Таблица 8.4 – структура таблицы «services» (услуги)

|  |  |  |
| --- | --- | --- |
| Наименование услуги | Описание услуги | Фото услуги |
| Мойка | Мойка автомобиля | Moika.jpg |
| Ремонт кузова | Ремонт кузова и всего автомобиля | Remint.jpg |
| Мойка салона | Мойка салона | salona.jpg |

Таблица 8.5 – структура таблицы «applications» (заявки)

|  |  |  |
| --- | --- | --- |
| Статус | Клиент | Услуга |
| Принятые | Лена | Мойка |
| Отклоненно | Анатолий | Ремонт кузова и всего автомобиля |
| На рассмотрении | Анатолий | Мойка салона |

1. Общие требования к программному продукту
   1. Введение
2. Краткое описание проекта

Carte Blanche занимается предоставляется услуг по обслуживаю автомобилей клиентов.

Основным назначением проекта является работа с заявками на предоставление услуг.

1. Цели и задачи разработки

Carte Blanche c создается с целью:

- предоставления услуг по ремонту, мойку, диагностику автомобилей;

- повышения качества (полноты, точности, достоверности, своевременности, согласованности) информации.

1. Ссылки на связанные документы или проекты

https://github.com/YsxcS1GMA/carteblanche

* 1. Функциональные требования

1. Подробное описание функциональности, которую должен предоставлять программный продукт.
2. Входы и выходы системы.

Входы:

- заявка пользователя на услугу.

Выходы:

- документ об окончании работ.

1. Взаимодействие с другими системами.

Отсутствует

* 1. Требования к интерфейсу

1. Описание пользовательского интерфейса.

1.1 главная страница:

- логотип компании;

- баннер компании;

- описание деятельности автомойки;

- оказываемые услуги;

- контактная информация.

1.2 страница услуг:

- все услуги на сайте;

- кнопка “Записаться”.

1.3 страница мастера:

- информация о мастерах;

1. Требования к навигации и взаимодействию пользователя с системой.

2.1 навигационное меню:

- фиксированное меню для быстрого доступа ко всем разделам (О нас, Услуги, Контакты, мастера, Авторизация и Регистрация, Выход).

2.2 просмотр оказываемых услуг:

- возможность просмотра оказываемых услуг и подать заявку

2.3 адаптивность:

- отзывчивый дизайн для различных разрешений экрана.

2.4 аутентификация и профили:

- возможность регистрации и авторизации пользователя;

2.5 уведомления:

- оповещение пользователя о вводе неправильных данных.

* 1. Требования к производительности

1. Ожидаемые показатели производительности.

Пропускная способность: 100;

Время ответа сервера: 2мс;

Время отображения: 2мс.

1. Требования к загрузке системы при различных нагрузках.

Запросов в секунду: не более 1000 в секунду;

Пользователей в системе:

Активных – 4 человек.

Зарегистрированных – 100 человек.

Одновременно работающих – 4 человек.

* 1. Требования к безопасности

1. Требования к управлению доступом и аутентификации.

Доступ к учетным записям администратора и клиента разграничивается c помощью механизма Laravel – Middleware.

1. Меры безопасности для защиты данных.

Пароли пользователей автоматически хешируются при регистрации.

1. Требования к обеспечению конфиденциальности.

Требования к обеспечению конфиденциальности основываются на: Закон РФ от 27.12.1991 N 2124-1(ред. от 13.06.2023)»О средствах массовой информации»

1. Карта сайта

HTML карта – это статическая веб-страница, которая включает все основные и подстраницы вашего сайта в соответствии с их иерархией. Проще говоря, это что-то вроде оглавления книги, которое разработано для сайта, чтобы помочь посетителям легко находить страницы на вашем ресурсе.

Карта сайта представлена на рисунке 1.10.1.



Рисунок 10.1 – Карта сайта

1. Физическая схема

Схема базы данных — это структура и организация базы данных, которая определяет ее таблицы, поля, связи, ограничения и типы данных. Он служит образцом для организации данных и доступа к ним, предоставляя план действий для разработчиков, администраторов и пользователей при работе с базой данных. Схема необходима для эффективного и действенного управления данными, что приводит к повышению производительности базы данных и упрощению обслуживания.

Физическая схема базы данных представлена в приложении В.

1. Описание модулей

При запуске программного продукта выполняется проект «Carte Blanche», после которого выполняются следующие процедуры, которые представлены в таблице 12.1. Они показывает на какие модули и функции состоит системы для своей работы.

Таблица 12.1 – Описание модулей

|  |  |  |
| --- | --- | --- |
| Модуль | Метод | Назначение |
| AdminController.php | admin\_application | Открытие страницы со всеми заявками |
| confirm | Принятие заявки |
| deny | Отклонение заявки |
| admin\_services | Открытие страницы со всеми услугами |
| services\_create | Добавление новой услуги |
| services\_update | Редактирования услуги |
| services\_delete | Удаление услуги |
| authController.php | auth | Открытие страницы с авторизацией |
| auth\_user | Авторизация пользователя |
| reg | Открытие страницы с регистрации |
| exit | Выход из учетной записи |
| reg\_user | Регистрация пользователя |
| ApplicationController.php | services | Открытие страницы с авторизацией мастера |
| Application\_create | Авторизация мастера |
|  | - |

1. Протоколы тестирования

Тестирование программного продукта является неотъемлемой частью разработки системы. Его проводят для достижения наилучшей работоспособности и исправления ряда ошибок, которые могут возникнуть при взаимодействии с системой.

Обычно тестирование ПО проводится на всех этапах жизненного цикла: с момента разработки ПО до его запуска в эксплуатацию.

Цель тестирования - проверка соответствия ПО предъявляемым требованиям, обеспечение уверенности в качестве ПО, поиск очевидных ошибок в программном обеспечении.

Для выявления ошибок в работе системы были проведены следующие тесты.

Общая информация о тестировании приведена в таблице 13.1.

Таблица 13.1 – Общая информация о тестировании

|  |  |
| --- | --- |
| Общая информация о тестировании | |
| Название теста | Tests\_Service |
| Номер версии | 1.0 |
| Имя тестера | Дамир Миргалиев |
| Дата тестирования | 12.12.2023 |

Тестирование авторизации при вводе некорректных данных приведено в таблице 13.2.

Таблица 13.2 - Тестирование авторизации при вводе некорректных данных

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | T\_AU\_1 |
| Приоритет тестирования | Высокий |
| Название тестирования | Проверка авторизации в модуле authController.php с вводом некорректных данных |
| Резюме испытания | Необходимо добиться корректного поведения системы, при вводе неправильных данных при авторизации сотрудника |
| Шаги тестирования | 1. Открыть страницу авторизации пользователя; 2. Ввести в поля «Email» и «Пароль» неправильные данные; 3. Нажать на кнопку «Вход». |
| Данные тестирования | Email: sdgsdg;  Пароль: sdgsdggsd. |
| Ожидаемый результат | Сайт должен вывести сообщение о неправильном вводе данных |
| Фактический результат | Вывелось сообщение «Неверная почта или пароль». Результат на рисунке 13.2. |
| Предпосылки | Открытие сайта |
| Постусловия | Работа системы происходит корректно |
| Статус (Pass/Fall) | Pass |



Рисунок 13.2 – Ошибка при вводе некорректных данных авторизации

Тестирование авторизации при вводе корректных данных приведено в таблице 13.3.

Таблица 13.3 - Тестирование авторизации при вводе корректных данных

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | T\_AU\_2 |
| Приоритет тестирования | Высокий |
| Название тестирования | Проверка авторизации в модуле authController.php с вводом корректных данных |
| Резюме испытания | Необходимо добиться корректного поведения системы, при вводе правильных данных при авторизации пользователя |
| Шаги тестирования | 1. Открыть страницу авторизации пользователя; 2. Ввести в поля «Email» и «Пароль» неправильные данные; 3. Нажать на кнопку «Вход». |
| Данные тестирования | Email: damir2@gmail.com  Пароль: Gthcbr22827. |
| Ожидаемый результат | Сайт должен перенаправить пользователя на главную страницу. |
| Фактический результат | Пользователя перенаправило на главную страницу. |
| Предпосылки | Открытие сайта |
| Постусловия | Работа системы происходит корректно |
| Статус (Pass/Fall) | Pass |

Тестирование подачи заявки с незаполненными полями ввода данных 13.4.

Таблица 13.4 - Тестирование регистрации с заполненными полями ввода данных

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | T\_REG\_1 |
| Приоритет тестирования | Высокий |
| Название тестирования | Проверка регистрации в модуле authController.php с вводом корректных данных |
| Резюме испытания | Необходимо добиться корректного поведения системы, при заполненных полях ввода данных |
| Шаги тестирования | 1. Открыть страницу регистрации пользователя; 2. Ввести в поля формы верные данные 3. Нажать на кнопку «Зарегистрироваться». |
| Данные тестирования | Имя: Дамир  Номер телефона: 89170318423  Email: damir2@gmail.com  Пароль: Gthcbr22827  Повторите пароль: Gthcbr22827. |
| Ожидаемый результат | Сайт должен перекинуть на главную страницу |
| Фактический результат | Сайт перекидывает пользователя на главную страницу. |
| Предпосылки | Открытие сайта |
| Постусловия | Работа системы происходит корректно |
| Статус (Pass/Fall) | Pass |

Тестирование подачи заявки с заполненными полями ввода данных 13.5.

Таблица 13.5 - Тестирование регистрации с незаполненными полями ввода данных

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | T\_APP\_2 |
| Приоритет тестирования | Высокий |

Продолжение таблицы 13.5

|  |  |
| --- | --- |
| Название тестирования | Проверка регистрации в модуле authController.php с вводом некорректных данных |
| Резюме испытания | Необходимо добиться корректного поведения системы, при незаполненных полях ввода данных |
| Шаги тестирования | 1. Открыть страницу регистрации пользователя; 2. Ввести в поля формы верные данные 3. Нажать на кнопку «Зарегистрироваться». |
| Данные тестирования | Имя: Дамир21  Номер телефона: 8918423цуц  Email: damir2@gmail.com  Пароль: Gthcbr22827  Повторите пароль: G122827. |
| Ожидаемый результат | Сайт должен выдать ошибку о полях номера телефона и пароля |
| Фактический результат | Вывод ошибку в поле номера телефона «Только цифры» и в поле пароля «Пароли не совпадают». Результат на рисунке 13.5 |
| Предпосылки | Открытие сайта |
| Постусловия | Работа системы происходит корректно |
| Статус (Pass/Fall) | Pass |

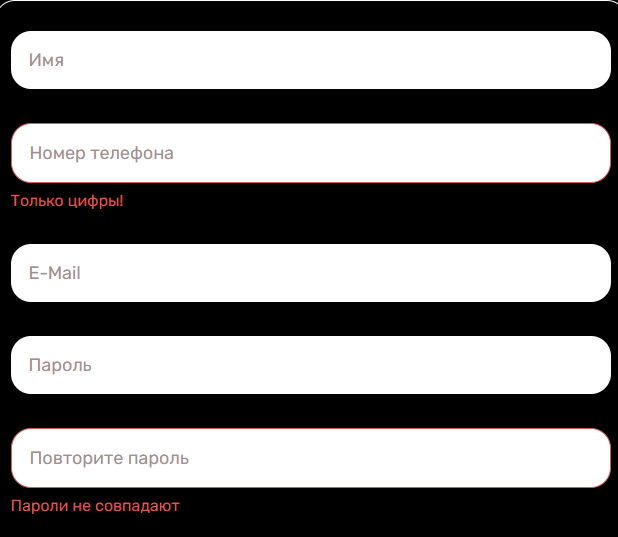


Рисунок 13.5 – Ошибка при вводе некорректных данных регистрации

1. Руководство пользователя

Главной целью руководства пользователя является обучение работы с системой.

Данный программный продукт предназначен для просмотра и изменения информации, которая хранится в базе данных, а точнее регистрации нового пользователя, изменения статуса заявки, добавления услуг и их редактирование, а также подачи заявки от клиента.

Минимальные требования к аппаратному обеспечению для производительной работы программного продукта:

* операционная система Windows 10;
* разрядность системы: x64;
* ОЗУ: от 2Гб.;
* монитор;
* клавиатура;
* компьютерная мышь.

Для запуска и комфортного использования программного продукта пользователю необходимы базовые знания работы с браузером и операционной системой Windows.

Подготовка системы к работе:

* перейти на страницу сайта по его URL.

После перехода по URL, пользователя перекидывает на главную страницу (рисунок 14.1).

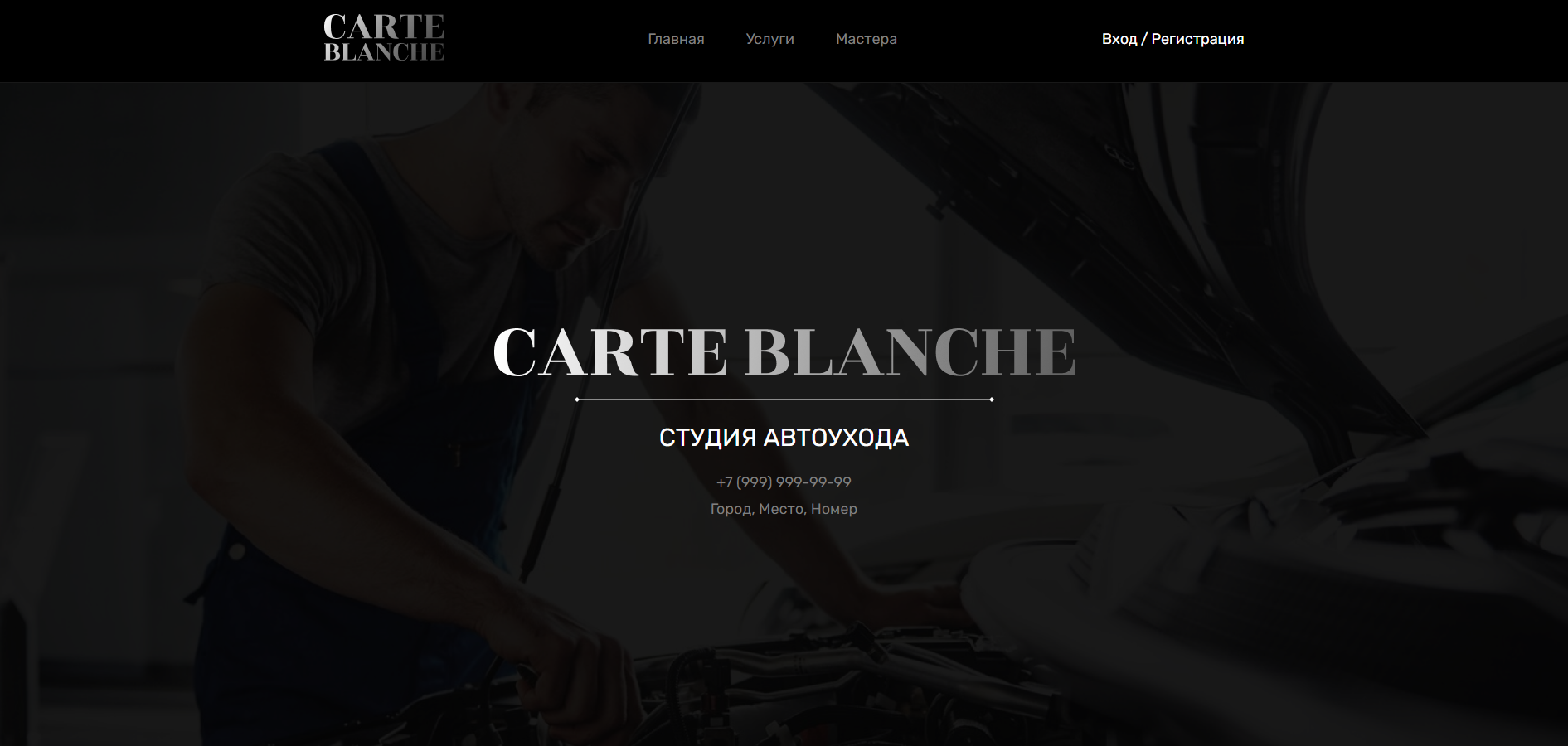


Рисунок 14.1 – Главная страница

Нажав на ссылку “Услуги”, на шапке сайта, пользователя перекинет на блок с выбором услуг для автомобиля (рисунок 14.2).

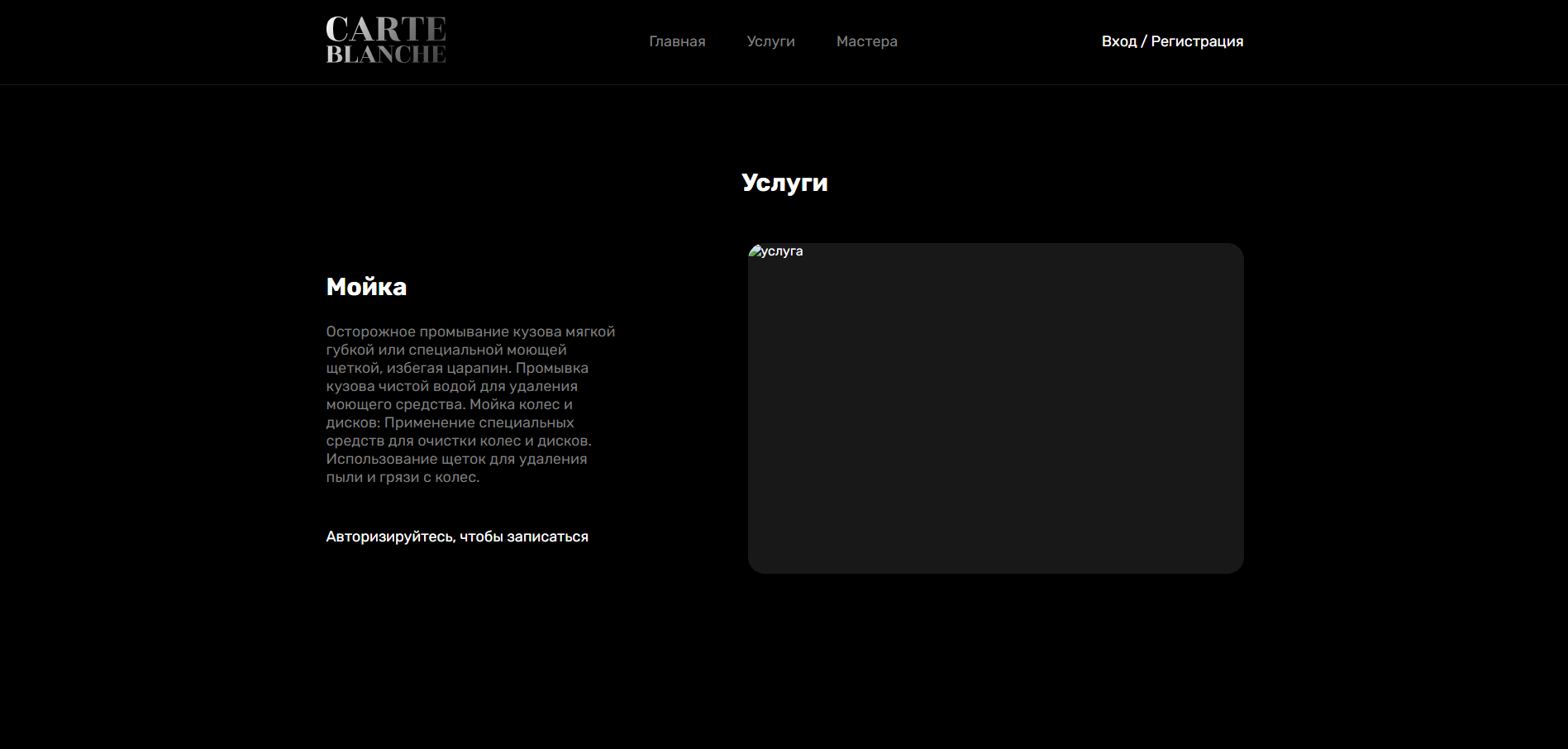


Рисунок 14.2 – Категории услуг

Для того, чтобы пользователь смог оставить заявку, ему нужно авторизоваться в системе. Сделать это можно, нажав на ссылку “Вход” (рисунок 14.3).

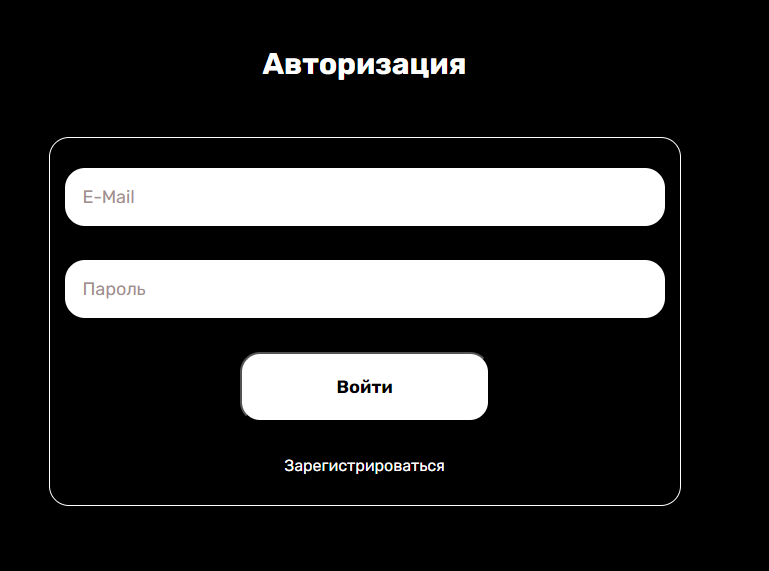


Рисунок 14.3 – Страница авторизации пользователя

Если же у пользователя нет учетной записи, он всегда может зарегистрироваться в системе. Сделать это можно нажав на ссылку “Регистрация” (рисунок 14.4).



Рисунок 14.4 - Страница регистрации

Чтобы подать заявку на услугу, нужно нажать на кнопку “Записаться” (рисунок 14.5).

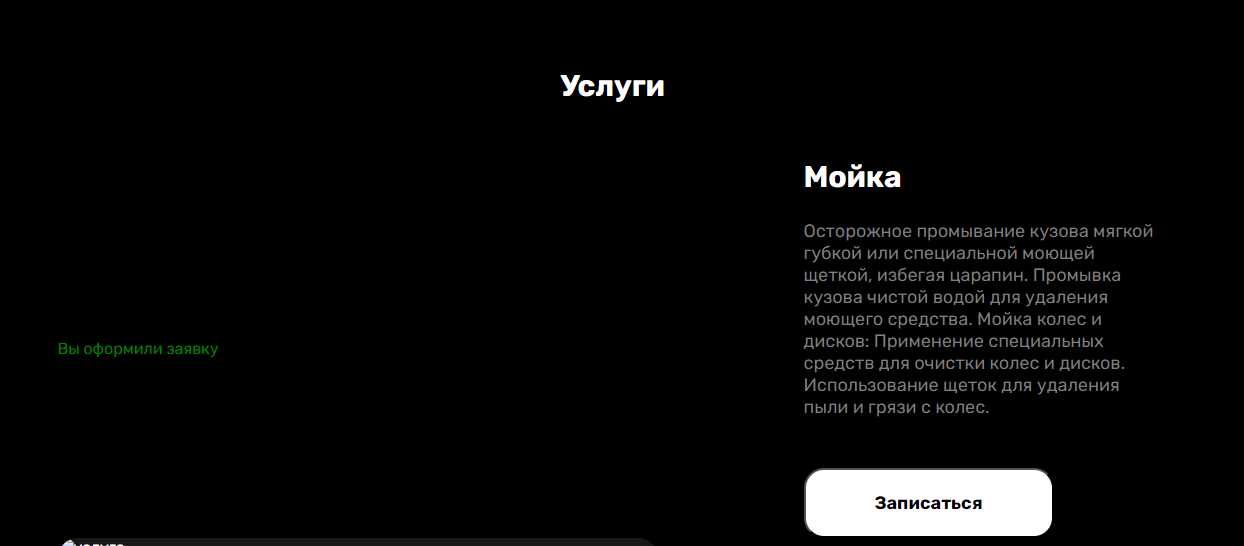


Рисунок 14.5 - Страница с кнопкой «Записаться»

Клиент также может просмотреть информацию о мастерах. Для этого в шапке ему нужно перейти по ссылке «Мастера» (рисунок 14.6).



Рисунок 14.6 - Страница с мастерами

Выйти из учетной записи можно при помощи ссылки «Выход», которая находится в навигационной панели сайта.

Для авторизации администратора сайта нужно также зайти на страницу авторизации и войти в свой аккаунт.

Администратору доступны страницы «Управление услуг» и «Управление заявками».

Для того, чтобы администратор добавить новую услугу, ему нужно перейти на страницу «Управление услуг», ввести в форму данные о услуги и нажать на кнопку «Добавить на сайт» (рисунок 14.7).

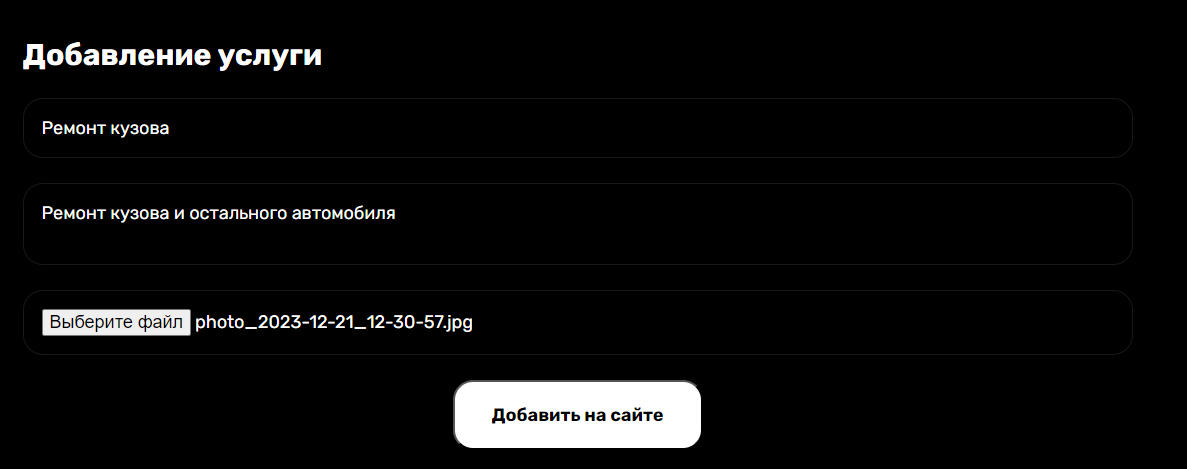


Рисунок 14.7 – Добавление услуг

Для того, чтобы администратор добавить смог редактировать услугу, ему нужно перейти на страницу «Управление услуг», ввести в форму данные о услуги и нажать на кнопку «Редактировать услугу» , также тут можно удалить услугу нажав на кнопку «Удалить услугу» (рисунок 14.8).

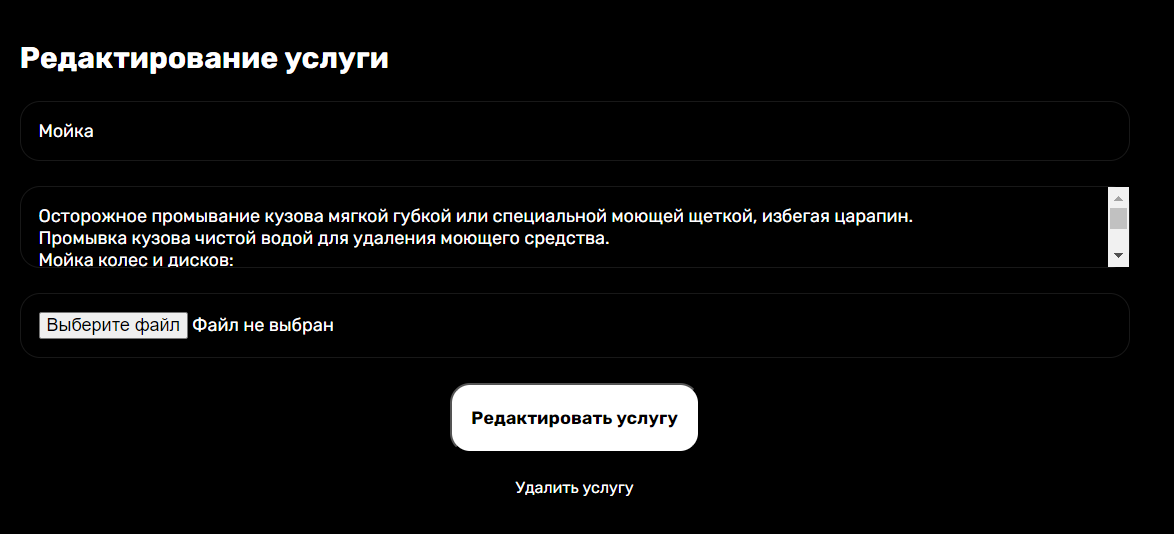


Рисунок 14.8 – Редактировать услуги

Для того, чтобы администратор принять или отклонить заявку, ему нужно перейти на страницу «Управление заявками», на ней выводиться вся информация об новых заявках. Для принятие или отклонения ему нужно выбрать соответствующие кнопки «принять» и «отклонить». В случае принятие в столбце «Тип услуги, статус» измениться на «принять» , «отклонено» (рисунок 14.9).

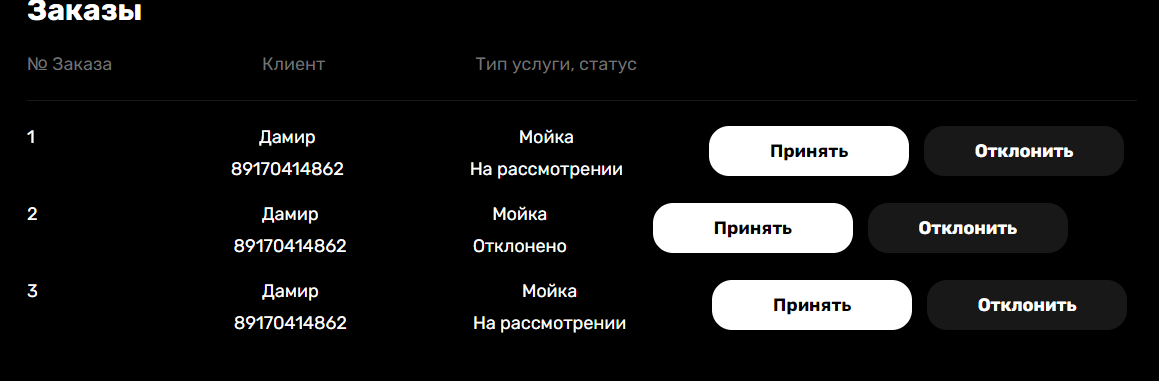


Рисунок 14.9 – Управление заявками

Приложение А

UML диаграммы

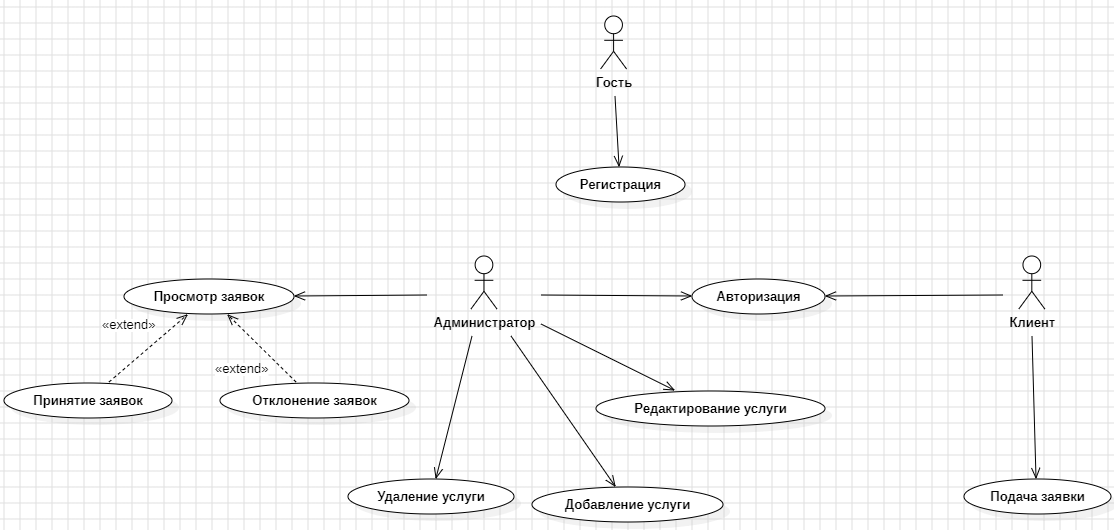


Рисунок А.1 – диаграмма прецедентов

Приложение Б

Даталогическая модель базы данных

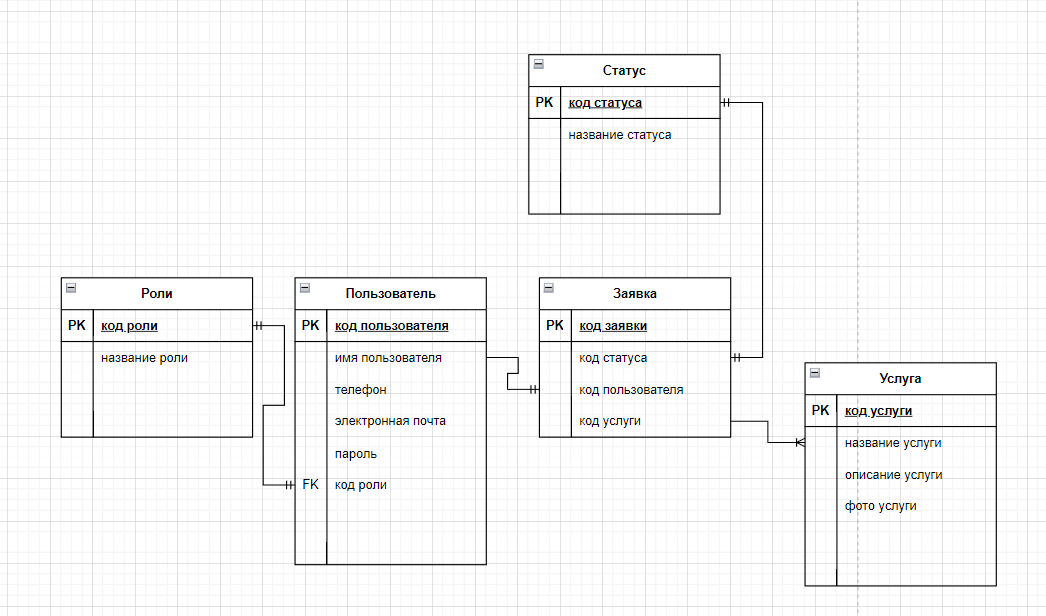


Рисунок Б.1 – даталогическая модель базы данных

Продолжение приложения Б

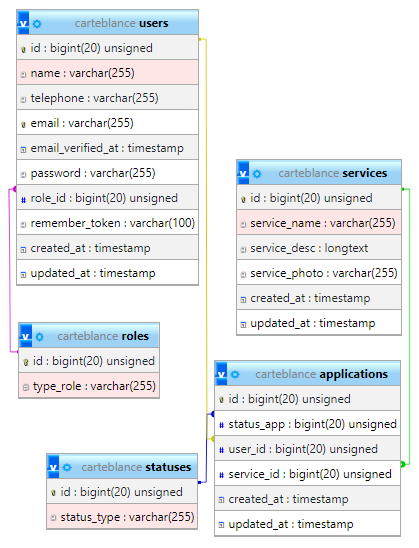


Рисунок Б.2 – даталогическая модель базы данных

Приложение В



Рисунок В.1 – физическая модель БД для клиента

Продолжение приложения В

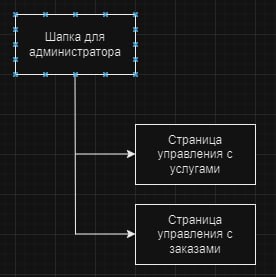


Рисунок В.2 – физическая модель БД для администратора

СПИСОК ЛИТЕРАТУРЫ

1. ГОСТ 19.001–77. ЕСПД. Общие положения.
2. ГОСТ 19.003–80. ЕСПД. Схемы алгоритмов и программ. Обозначения условные графические. – Заменен на ГОСТ 19.701-90.
3. Википедия: офиц. сайт. URL: http://ru.wikipedia.org/wiki/ (дата обращения: 3.09.2023).
4. Язык запросов SQL: офиц. сайт. URL: https://sql-language.ru (дата обращения: 15.09.2023).
5. Хабр: офиц. сайт. URL: https://habr.com/ru/all/ (дата обращения: 24.09.2023).
6. Htmlbook: офиц. сайт. URL: http://htmlbook.ru/ (дата обращения: 5.09.2023).
7. Html5 book: офиц. сайт. URL: http://html5book.ru/ (дата обращения: 5.09.2023).
8. Metanit: офиц. сайт. URL: https://metanit.com/ (дата обращения: 15.10.2023).
9. W3School: офиц. сайт. URL: https://www.w3schools.com/ (дата обращения: 28.10.2023).
10. itProger: офиц. сайт. URL: https://itproger.com/ (дата обращения: 03.11.2023).
11. Хекслет: офиц. сайт. URL: https://ru.hexlet.io/ (дата обращения: 04.11.2023).
12. Денвик: офиц. сайт. URL: https://denvic.ru/ (дата обращения: 21.11.2023).
13. JavaScript: офиц. сайт. URL: https://learn.javascript.ru/first-steps/ (дата обращения: 07.09.2023).
14. Bootstrap: офиц. сайт. URL: https://getbootstrap.com/ (дата обращения: 5.09.2023).
15. Laravel: офиц. сайт. URL: https://laravel.com/ (дата обращения: 5.09.2023).