СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 2](#_Toc743698690)

[1. ПРЕДПРОЕКТНЫЙ АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 2](#_Toc696183317)

[1.1 Описание предметной области 3](#_Toc239367833)

[1.2 Функции организационной структуры 4](#_Toc1798060372)

[1.3 Описание потоков данных 9](#_Toc703317967)

[1.4 Рассмотрение аналогичных проектных решений 16](#_Toc597232730)

[1.5 Обоснование необходимости разработки ИС 18](#_Toc1130367476)

[2. АНАЛИЗ И ПРОЕКТИРОВАНИЕ ИС 19](#_Toc794592904)

[2.1. Разработка архитектуры сайта, выбор технологий реализации ИС 20](#_Toc1593287555)

[2.2. Определение структуры и функциональности ИС 21](#_Toc1675520983)

[1.1. Техническое обеспечение информационной системы 25](#_Toc1776869909)

[3. РАЗРАБОТКА ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ ИС 25](#_Toc1367023935)

[3.1. Концептуальное проектирование БД 25](#_Toc300437659)

[3.2. Логическое проектирование БД 25](#_Toc2092578358)

[3.3. Физическое проектирование БД 25](#_Toc1096495518)

[4. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ИС 25](#_Toc1341367349)

[4.1. Структура программного обеспечения 25](#_Toc893263041)

[4.2. Запросы к базе данных 25](#_Toc762623823)

[4.3. Описание пользовательского интерфейса 25](#_Toc1217869637)

[4.4. Описание интерфейса администратора 25](#_Toc1958472482)

[ЗАКЛЮЧЕНИЕ 25](#_Toc1353105374)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 25](#_Toc285525149)

ТЕРМИНЫ, ОПРЕДЕЛЕНИЯ, ИСПОЛЬЗУЕМЫЕ СОКРАЩЕНИЯ

|  |  |
| --- | --- |
| Обозначение | Описание |
| ИС | Информационная система типографии |
| *IDEF*0 | Графическая нотация, предназначенная для формализации и описания бизнес-процессов |
| *DFD* | Диаграмма потоков данных |
| БД | База данных |
| Система | Информационная система |
| СУБД | Система управления базами данных |
| ИБ | Информационная база |
| *ERD* | Диаграмма «сущность-связь» |

# ВВЕДЕНИЕ

В представленной курсовой работе будет показан процесс проектирования информационной системы (сайта), которая в дальнейшем будет использоваться для автоматизации бизнес-процессов при подборе запчастей для легковых автомобилей.

Разрабатываемая информационная система реализуется на базе автобюро и предназначена для взаимодействия с клиентами организации: привлечение клиентов, упрощение процесса записи клиентов на ремонт, автоматизация ведения истории ремонта клиентов.

Актуальность выбранной темы объясняется взрывным ростом уровня автомобилизации, так, например, если в 2000 году в РФ на 1000 человек насчитывалось не более 130 автомобилей, то к 2018 году это число увеличилось до 309. Техническим обслуживанием транспорта как раз и занимаются автобюро (автосервисы). Если подойти к анализу рынка таких организаций, то можно прийти к заключению, что у подавляющего большинства либо вовсе отсутствуют ИС, либо эти ИС предоставляют лишь ознакомительную информацию и не несут никакого функционала.

В качестве основных целей создания информационной системы автобюро можно выделить следующие:

* увеличение прибыли организации, за счет упрощения механизма проведения записи на оказание услуг;
* освобождение сотрудников от однотипной работы, либо существенное сокращение времени ее выполнения;
* увеличение размера клиентской базы, за счет использования различных способов общения и взаимодействия с ними;
* уменьшение количества ведения бумажных документов.

# **ПРЕДПРОЕКТНЫЙ АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

## Описание предметной области

В рамках курсовой работы принято решение объединить как одно предприятие автосервис и автомагазин для наибольшего охвата функционала и проектирования полноценной ИС для работы автомобильного бюро.

Автосервис [1]– это специализированная мастерская, которая оказывает специализированные услуги, связанные с техническим обслуживанием и ремонтом автомобилей. Основной целью функционирования автосервиса является обеспечение надлежащего функционирования автомобилей, устранение всевозможных технических неисправностей и, как следствие, сохранение жизней людей, обеспечение безопасности на дорогах.

Автомагазин – это торговое заведение или магазин, специализирующийся на продаже автомобильных запчастей, аксессуаров, инструментов, масел и других товаров, связанных с автомобилями. Также автомагазины могут предоставлять услуги по обслуживанию, ремонту или установке деталей на автомобили. Это место, где владельцы автомобилей могут приобрести необходимые компоненты для технического обслуживания или улучшения своих транспортных средств.

Деятельность организации осуществляется следующим образом. Каждый клиент имеет возможность зарегистрироваться в системе для этого вводятся следующие данные: ФИО, дата рождения, номер телефона, email, серия и номер паспорта. Также клиент имеет возможность зарегистрировать в системе автомобиль, для этого необходимо также заполнить необходимые поля, а именно: марка, модель, регистрационный знак, VIN, цвет и год выпуска. Все эти данные будут храниться в БД, а пользователь на их основе сможет записываться на ремонт своего автомобиля.

Информационная система «Автобюро» разработана для клиентов автобюро. Деятельность компании осуществляется следующим образом. Каждый клиент имеет возможность зарегистрироваться в системе для этого вводятся следующие данные: ФИО, дата рождения, номер телефона, email, серия и номер паспорта. Также клиент имеет возможность зарегистрировать в системе автомобиль, для этого необходимо также заполнить необходимые поля, а именно: марка, модель, регистрационный знак, VIN, цвет и год выпуска. Все эти данные будут храниться в БД, а пользователь на их основе сможет записываться на ремонт своего автомобиля.

Процесс записи на ремонт начинается с выбора вида необходимых услуг из перечня проводимых работ. В основной перечень проводимых работ входит:

* диагностика неисправностей различных агрегатов автомобиля;
* замена технических жидкостей в автомобиле;
* ремонт и обслуживание двигателя;
* ремонт и обслуживание системы кондиционирования;
* ремонт подвески и рулевого управления;
* развал-схождение;
* ремонт КПП;
* шиномонтаж и балансировка колес;
* ремонт электрооборудования.

После заполнения вида проводимых услуг производится предварительная оценка стоимости, если клиента устраивает, то ему необходимо выбрать удобное для него время из предоставленных вариантов. Также у клиента имеется возможность добавления к заказу фирменных автозапчастей.

## Функции организационной структуры

Организационная структура – это система, которая определяет точную иерархию подразделений, отделов и должностных лиц внутри организации, их отношения и распределение ответственности между ними. Она определяет способ организации работы, коммуникации, принятия решений и координации деятельности внутри организации. Если говорить об анализе организационной структуры предприятия, то она является важным аспектом в понимании и оценки эффективности функционирования организации. Также среди целей проведения анализа организационной структуры можно выделить: выявление избыточности (недостатка) ответственности и полномочий; определение слабых мест в процессе коммуникации между подразделениями и отдельными сотрудниками внутри подразделений; определение эффективности использования ресурсов предприятия, и многое другое. Организационно автобюро можно разделить на следующие отделы: отдел администрации, мастерская и склад. Схема организационной структуры автобюро представлена на рисунке 1.1.

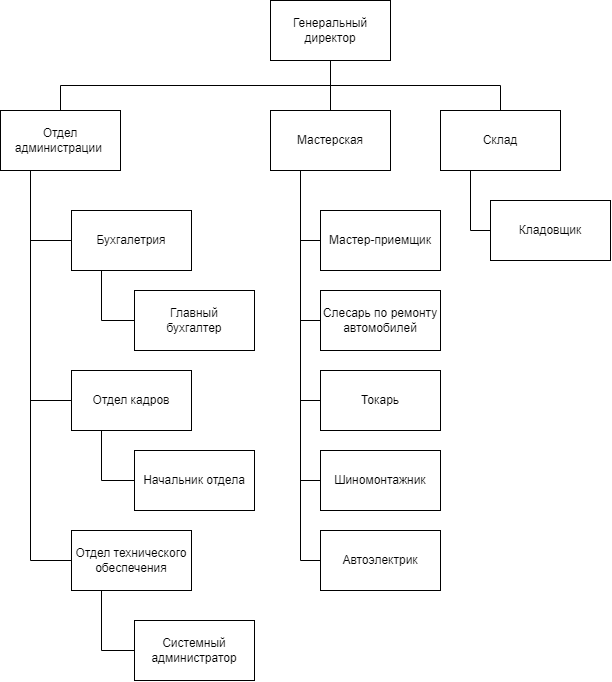


Рисунок 1.1 – схема организационной структуры автосервиса

За отдел администрации отвечает директор. В подчинении у него находятся: бухгалтерия, отдел кадров и отдел технического обеспечения. Главным в бухгалтерии является главный бухгалтер, он ведет всю необходимую бухгалтерскую статистику по финансовым состояниям предприятия. Таким, как расчет и выплата заработной платы сотрудникам, расчет статей расходов и прибыли. В отделе кадров также существует начальник отдела, он ведет набор на предприятие новых сотрудников. В отделе технического обеспечения работает системный администратор, он занимается налаживанием работы ИТ-инфраструктуры предприятия, налаживанием работы ИС, наполнением БД.

За мастерскую отвечает мастер-приемщик. В его обязанности входит работа с клиентами, контроль за ходом ремонта, а также ведение инвентаризации рабочего инструмента. В его подчинении находятся матера по ремонту, у каждого из которых есть своя специализация.

За склад отвечает кладовщик. В его обязанности входит все, что касается склада, а именно прием, выдача, учет и правильное хранение запчастей и рабочего инвентаря на складе. Также в обязанности кладовщика входит взаимодействие с поставщиками и заказ необходимых запчастей и рабочего инвентаря.

Далее рассмотрим функциональную структуру автобюро. Функциональная структура – это тип организационной структуры, при котором организация организована на основе функциональных областей или отделов. В функциональной структуре подразделения группируются по функциональным областям, таким как производство, маркетинг, финансы, ресурсы человека и т.д. Каждое подразделение выполняет специфические функции, связанные с его областью компетенции.

Функциональная структура изображена на рисунке 1.2.

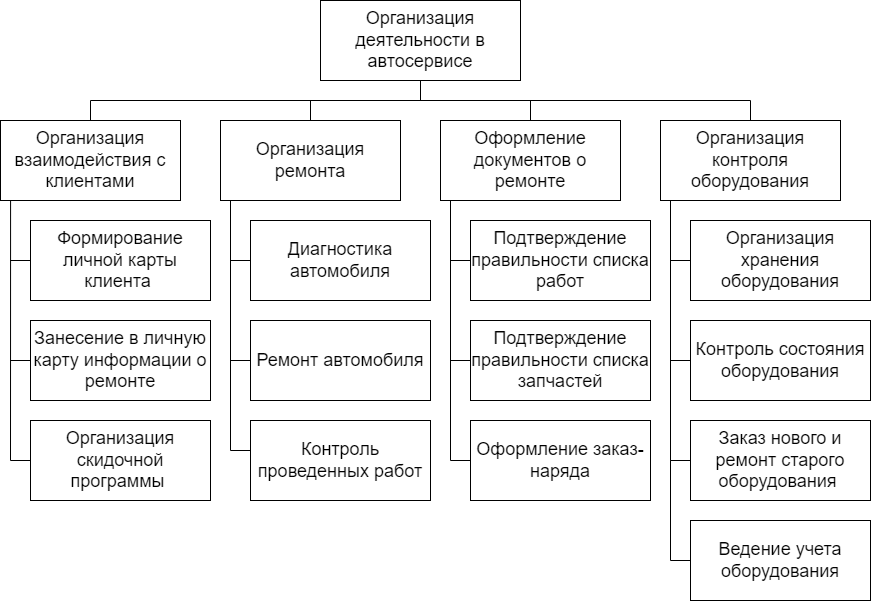


Рисунок 1.2 – схема функциональной структуры автобюро

В конце необходимо рассмотреть организационную структуру системы автобюро с точки зрения матричной модели. Матричная модель организационной структуры представляет собой комбинацию функциональной и проектной структур. Матричная модель позволяет обеспечить открытый обмен информацией между отделами и сотрудниками внутри этих отделов. Матричная модель представлена в таблице 1.3. Для удобства построения была созданы вспомогательные таблицы (таблица 1.1, таблица 1.2) с элементами и функциями организационной структуры, а также их обозначениями.

Таблица 1.1 – Обозначение элементов организационной структуры

|  |  |
| --- | --- |
| Обозначение | Описание |
| W0 | Директор автобюро |
| Бухгалтерия | |
| W1.1 | Главный бухгалтер |
| Отдел кадров | |
| W2.1 | Начальник отдела |

Продолжение таблицы 1.1

|  |  |
| --- | --- |
| Отдел технического обеспечения | |
| W3.1 | Системный администратор |
| Мастерская | |
| W4.1 | Мастер-приемщик |
| W4.2 | Слесарь по ремонту автомобилей |
| W4.3 | Токарь |
| W4.4 | Шиномонтажник |
| W4.5 | Автоэлектрик |
| Склад | |
| W5.1 | Кладовщик |

Таблица 1.2 – Обозначение элементов функциональной структуры

|  |  |
| --- | --- |
| Обозначение | Описание |
| Организация взаимодействия с клиентами – F1 | |
| F1.1 | Формирование личной карты клиента |
| F1.2 | Занесение в личную карту информации о ремонте |
| F1.3 | Организация скидочной программы |
| Организация ремонта – F2 | |
| F2.1 | Диагностика автомобиля |
| F2.2 | Ремонт автомобиля |
| F2.3 | Контроль проведенных работ |
| Оформление документов о ремонте – F3 | |
| F3.1 | Подтверждение правильности списка работ |
| F3.2 | Подтверждение правильности списка запчастей |
| F3.3 | Оформление заказ-наряда |

|  |  |
| --- | --- |
| Организация контроля оборудования – F4 | |
| F4.1 | Организация хранения оборудования |
| F4.2 | Контроль состояния оборудования |
| F4.3 | Заказ нового и ремонт старого оборудования |
| F4.4 | Ведение учета оборудования |

Таблица 1.3 – Матричная модель организационной структуры

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Работы | Исполнители | | | | | | | | | |
| W0 | W1.1 | W2.1 | W3.1 | W4.1 | W4.2 | W4.3 | W4.4 | W4.5 | W5.1 |
| F1 |  |  |  |  | + |  |  |  |  |  |
| F1.1 |  |  |  | + | + |  |  |  |  |  |
| F1.2 |  |  |  | + | + |  |  |  |  |  |
| F1.3 |  |  |  |  | + |  |  |  |  |  |

Продолжение таблицы 1.3

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F2 |  |  |  |  | + | + | + | + | + |  |
| F2.1 |  |  |  |  | + |  |  |  |  |  |
| F2.2 |  |  |  |  | + | + | + | + | + |  |
| F2.3 |  |  |  |  | + |  |  |  |  |  |
| F3 |  |  |  |  | + |  |  |  |  |  |
| F3.1 |  |  |  |  | + | + | + | + | + |  |
| F3.2 |  |  |  |  | + |  |  |  |  | + |
| F3.3 |  | + |  |  | + |  |  |  |  |  |
| F4 |  |  |  |  | + |  |  |  |  | + |
| F4.1 |  |  |  |  |  |  |  |  |  | + |
| F4.2 | + |  |  |  | + |  |  |  |  | + |
| F4.3 |  |  |  |  |  |  |  |  |  | + |
| F4.4 | + |  |  |  |  |  |  |  |  | + |

## Описание потоков данных

Для описания потоков данных автобюро были построены функциональные диаграммы, основанные на технологии моделирования IDEF0[2]. Методология IDEF0 разработана для описания бизнес-процессов и операций в организациях. В качестве главной цели данной нотации можно выделить описание функциональных возможностей системы и взаимодействий между функциями. Она также может быть использована для определения требований к системе и оптимизации бизнес-процессов.

Чтобы правильно построить контекстную диаграмму необходимо определить главный бизнес-процесс системы, а также его роли. В качестве основного процессов системы был выбран процесс организации деятельности автобюро. Входными данными были выбраны: сам клиент и его неисправный автомобиль. Выходными данными для главного бизнес-процесса получаем следующее: клиент, исправный автомобиль, рекомендации по будущей эксплуатации, заказ-наряд, запись о ремонте в карте клиента. В качестве управления бизнес-процесса выступают: устав компании, инструкции по ремонту, а также постановление Правительства РФ от 11.04.2001 N 290 (ред. от 31.01.2017) "Об утверждении Правил оказания услуг (выполнения работ) по техническому обслуживанию и ремонту автомототранспортных средств". В роли механизмов выступают: мастер-приемщик, мастера по ремонту, кладовщик, техническое оборудование. Контекстная диаграмма главной функции автосервиса изображена на рисунке 1.3.

Далее необходимо произвести разбиение главного процесса на более мелкие подпроцессы. Для этого декомпозируем процесс А0 на процессы А1, А2 и А3. Эти разбиения продемонстрированы диаграммой декомпозиции главного бизнес-процесса. Диаграмма представлена на рисунке 1.4.

Процесс А1 – приемка автомобиля на ремонт. На входе процесса находится обращение клиента, а также неисправный автомобиль. Выполнение процесса происходит под управлением устава компании, инструкций по ремонту и постановления правительства РФ от 11.04.2001. В результате работы получают выходные ресурсы: акт дефектовки, информация об автомобиле и сам неисправный автомобиль после проведенной дефектовки.

Процесс А2 – выполнение работ с автомобилем. На вход процесс получает акт дефектовки, информацию об автомобиле и сам неисправный автомобиль после проведенной дефектовки. Выполнение процесса происходит под управлением устава компании, инструкций по ремонту и постановления правительства РФ от 11.04.2001, а также с применением мастеров по ремонту, кладовщика и технического оборудования. В результаты работы получаются выходные ресурсы – список работ, список запчастей, исправный автомобиль, запись о ремонте в тех. карте клиента.

Процесс А3 – выдача автомобиля клиенту. На вход процесс получает список работ, список запчастей, исправный автомобиль. Выполнение процесса происходит под управлением устава компании и постановления правительства РФ от 11.04.2001, а также с применением мастера-приемщика и технического оборудования. В результате работы на выходе процесс получает: исправный автомобиль, рекомендации по будущей эксплуатации, заказ-наряд.

Процесс А1 можно декомпозировать на три процесса. Процесс А1.1 – передача автомобиля клиентом в автобюро, процесс А1.2 – дефектовка автомобиля, А1.3 – заполнение тех. карты автомобиля. Диаграмма представлена на рисунке 1.5.

Процесс А2 можно декомпозировать на три процесса. Процесс А2.1 –заказ запчастей, необходимых для ремонтных работ, А2.2 – проведение ремонтных работ, А2.3 – заполнение заказ-наряда. Декомпозиция процесса «Выполнение работ с автомобилем» представлена на рисунке 1.6.

Процесс А3 можно декомпозировать на три процесса. Процесс А3.1 – осмотр автомобиля клиентом, А3.2 – оплата проведенных работ, А3.3 – передача автомобиля клиенту. Декомпозиция процесса «Выдача автомобиля клиенту» представлена на рисунке 1.7.

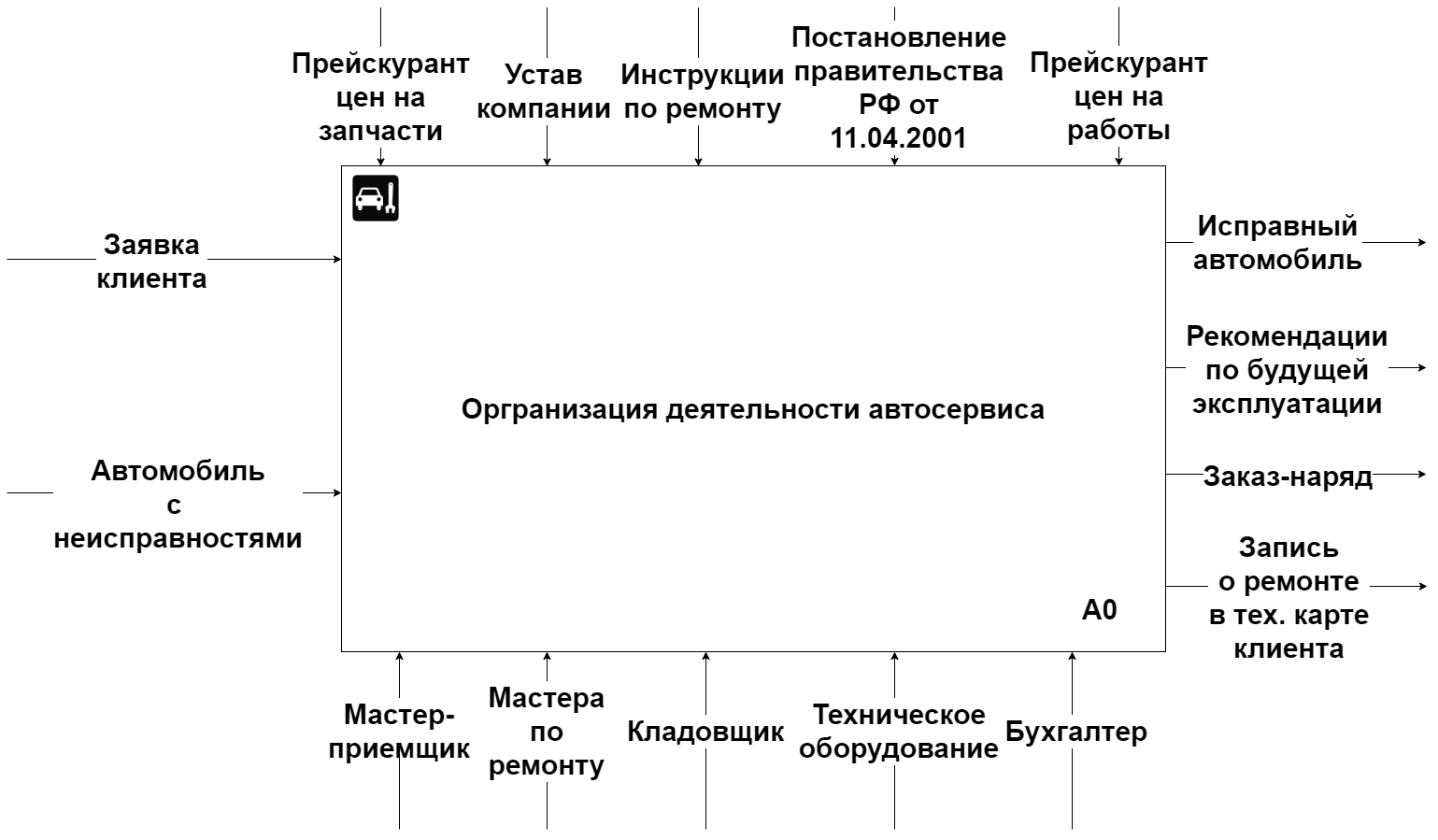


Рисунок 1.3 – Контекстная диаграмма главной бизнес-функции «Организация деятельности автосервиса»

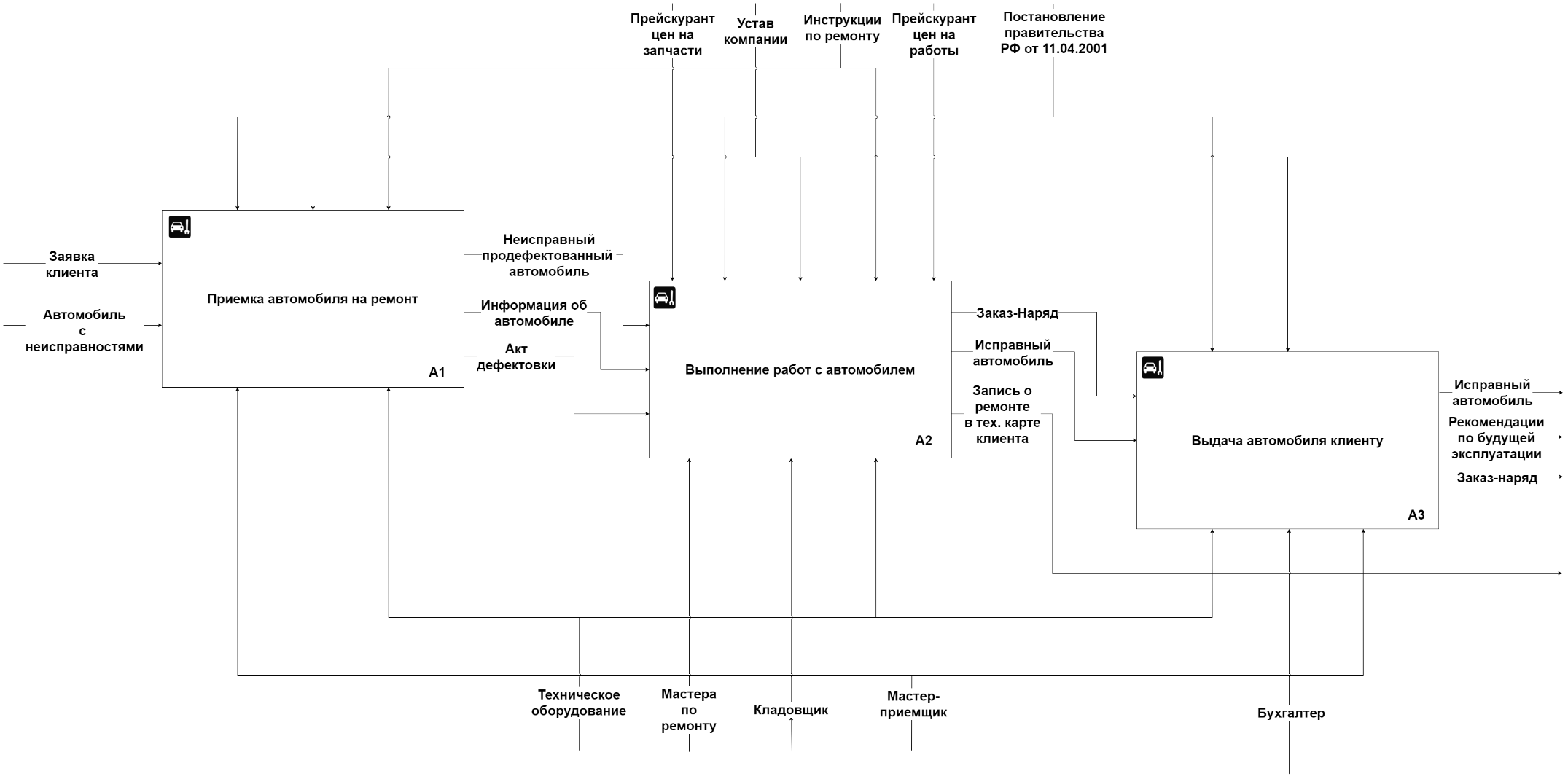
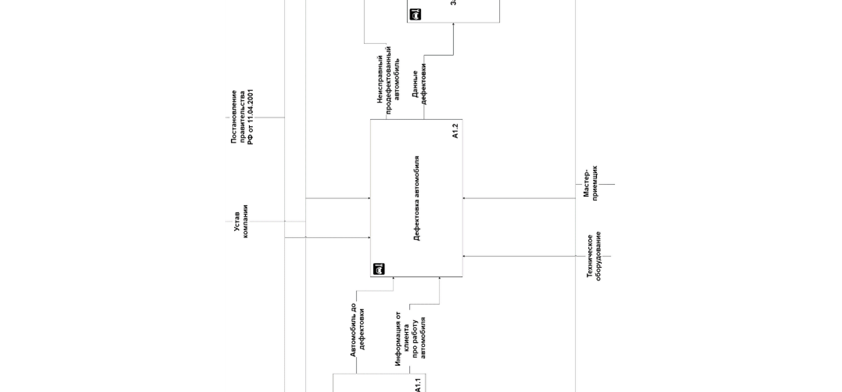


Рисунок 1.4 – Диаграмма декомпозиции бизнес-функции «Организация деятельности автобюро»

Рисунок 1.5 – Диаграмма декомпозиции бизнес-функции «Приемка автомобиля на ремонт»

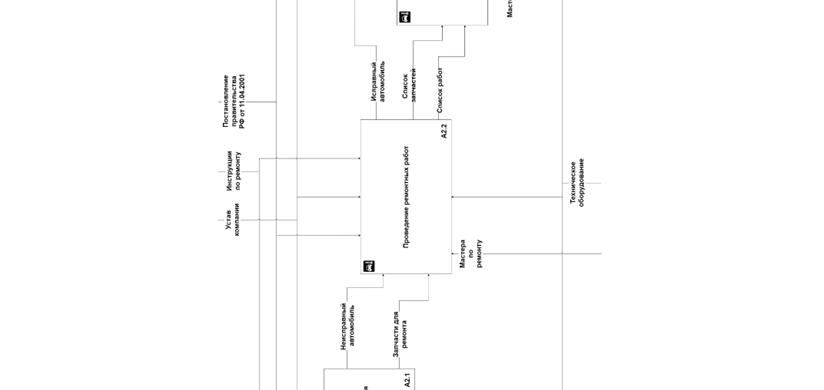


Рисунок 1.6 – Диаграмма декомпозиции бизнес-функции «Выполнение работ с автомобилем»

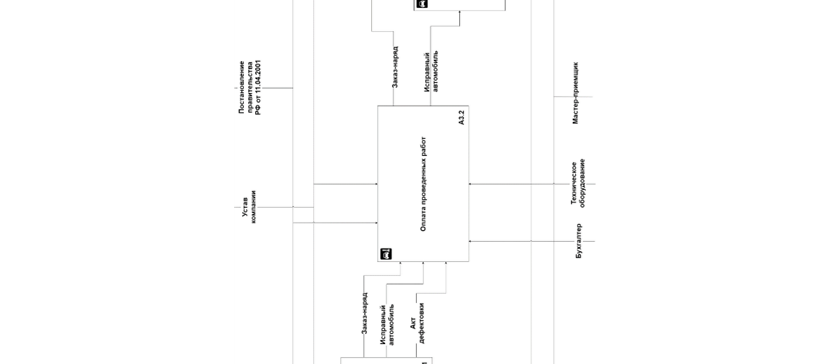


Рисунок 1.7 – Диаграмма декомпозиции бизнес-функции «Выдача автомобиля клиенту»

В конце данного раздела проведем рассмотрение структуры проведения ремонта на схеме в стандарте DFD [3]. Нотация DFD позволяет представить, как данные перемещаются от источников к получателям в рамках системы, и описать функции, которые обрабатывают эти данные. Основными элементами данной нотации являются:

* Процессы – это функция или операция, которая выполняется над данными. Они обозначаются прямоугольниками с текстовыми метками. Процессы могут принимать входные данные, обрабатывать их и генерировать выходные данные.
* Потоки данных, они представляют перемещение данных между процессами, хранилищами данных и внешними сущностями. Они обозначаются стрелками, также могут иметь текстовые метки, которые указывают на типы данных, которые передаются по этому потоку. Потоки бывают: входные, выходные и внутренние.
* Внешние сущности – это внешние источники или получатели данных, которые непосредственно взаимодействуют с системой. Графически изображаются в виде прямоугольников с текстовыми метками.
* Хранилища данных – это некоторые места, где данные могут быть сохранены или извлечены. Графически изображаются прямоугольниками с текстовыми метками.

Основными элементами проектируемой системы являются следующие компоненты:

* внешние сущности – «Поставщик»;
* работы – «Прием в ремонт», «Составление списка запчастей», «Выполнение ремонта», «Получение комплектующих», «Оформление заказ-наряда».
* хранилища данных «Списки работ», «Заказы на комплектующие», «Сведения о сотрудниках», «Заказ-наряды».

Диаграмма потоков данных представлена на рисунке 1.8.

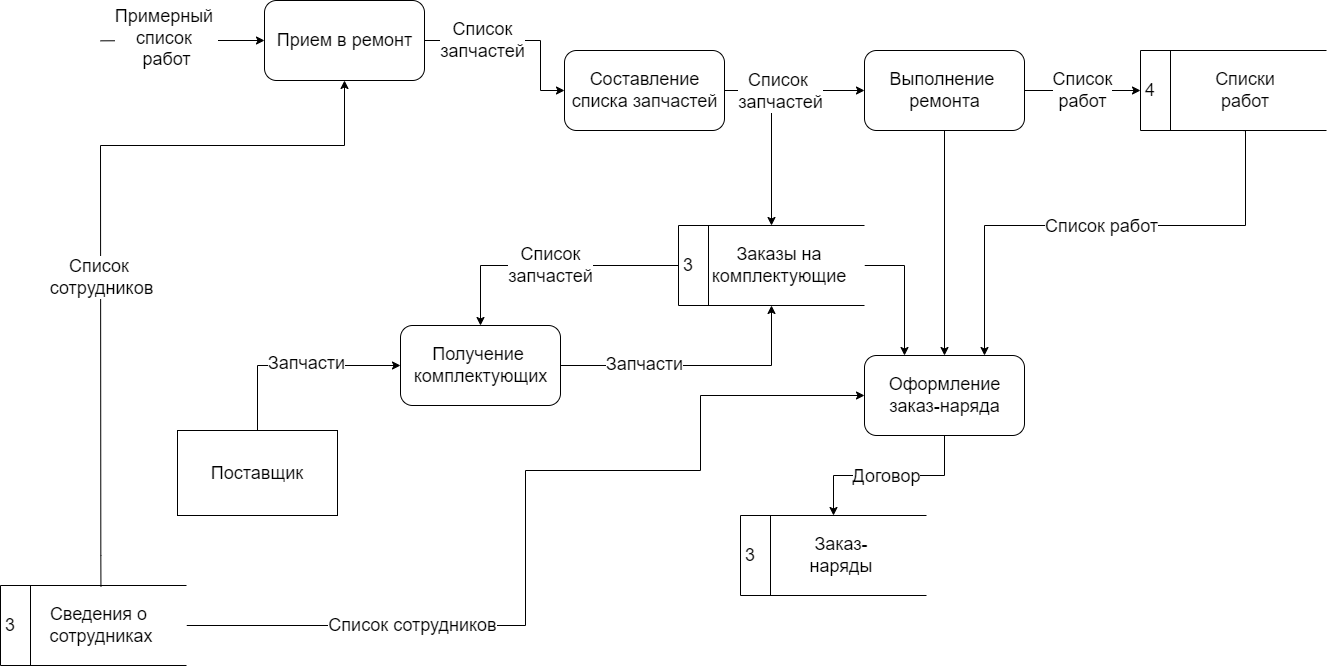


Рисунок 1.8 – Диаграмма потоков данных

## Рассмотрение аналогичных проектных решений

Перед этапом проектирования и разработки собственной ИС необходимо проанализировать уже существующие ИС, а также выявить их достоинства и недостатки. Также, благодаря, такому сравнительному анализу можно понять, какие особенности и стратегии могут быть применены при проектировании собственной ИС.

1. Первым этапом сравнительного анализа является определение конкурирующих или похожих ИС. Оптимальным количеством для проведения анализа является от трех до пяти аналогичных ИС. Для поиска можно использовать специальные поисковые системы и ключевые слова, связанные с предметной областью, разрабатываемой ИС. В результате данного шага были выбраны:

* система «zimwerk» [4]
* система «На-ходу» [5]
* система «Гольф-Авто» [6]

1. Вторым этапом является определение характеристик трафика. Это позволяет понять, количеств посещений, среднее время посещений, процент посетителей, которые не заходят далее основной страницы, а также откуда приходят посетители. Среди характеристик трафика можно выделить следующие: общее количество посещений за месяц; средняя продолжительность посещения; процент отказов (количество людей, которые не зашли дальше основной страницы). Среди всех источников трафика можно выделить основные: переходы из поисковых систем, прямые заходы, переходы из социальных сетей. Результатом данного шага является таблица 1.4.

Таблица 1.4 – Характеристики трафика рассматриваемых ИС

|  |  |  |  |
| --- | --- | --- | --- |
| Характеристика | Рассматриваемыe ИС | | |
| «zimwerk» | «На-ходу» | «Гольф-Авто» |
| Общее количество посещений за месяц | 17 300 человек | 2 600 человек | 10 000 человек |
| Средняя продолжительность посещения | 00:01:30 | 00:00:40 | 00:00:51 |
| Процент отказов | 56.88% | 86.38% | 75.10% |
| Переходы из поисковых систем | 67 % | 54 % | 52 % |
| Переходы из соц. сетей | 24 % | 42 % | 40 % |
| Остальное | 9 % | 4 % | 8 % |

1. Далее необходимо рассмотреть удобство использования и дизайн интерфейса. На этом шаге рассматривается: удобство навигации, дизайн и визуальная привлекательность; адаптивность и отзывчивость интерфейса; наличие необходимых разделов; читабельность и доступность контента. Результатом данного шага является таблица 1.5.

Таблица 1.5 – Характеристики дизайна рассматриваемых ИС

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | Рассматриваемыe ИС | | |
| «zimwerk» | «На-ходу» | «Гольф-Авто» |
| Дизайн | 7 | 5 | 8 |
| Удобство навигации | 6 | 5 | 7 |
| Адаптивность интерфейса | 5 | 5 | 5 |
| Читабельность | 5 | 6 | 4 |

1. На четвертом шаге произведем рассмотрение функциональной составляющей рассматриваемых ИС. На этом шаге проверяется наличие онлайн-записи; каталогов товаров и запчастей; прейскуранта работ; отзывов клиентов; онлайн-чатов. Результатом данного шага является таблица 1.6.

Таблица 1.6 – Характеристики функциональности рассматриваемых ИС

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | Рассматриваемыe ИС | | |
| «zimwerk» | «На-ходу» | «Гольф-Авто» |
| Наличие прейскуранта цен | - | + | + |
| Наличие формы заказа звонка | + | + | + |
| Наличие личного кабинета пользователя | - | + | - |
| Наличие интерактивного чата с поддержкой | - | + | + |
| Наличие возможности просмотра каталога автозапчастей | - | - | - |
| Наличие истории заявок на ремонт | - | + | - |
| Наличие страницы с отзывами клиентов | - | - | + |

В результате проведенного анализа была получена ценная информация об успешных стратегиях и неудачных решениях, рассматриваемых ИС. Были выделены сильные и слабые стороны будущих конкурентных решений. Во время проведения анализа были рассмотрены такие аспекты, как: характеристики трафика, удобство использования и дизайн, функциональные возможности рассматриваемых ИС.

## Обоснование необходимости разработки ИС

В ходе выполнения предпроектного анализа была обоснована актуальность выбранной темы, а также необходимость разработки ИС. Были описаны потоки данных и бизнес-процессы, были построены функциональные диаграммы, основанные на технологии моделирования IDEF0, DFD. В рамках рассмотрения аналогичных проектных решений были выявлены их достоинства и недостатки, которые могут быть использованы в рамках проектирования разрабатываемой ИС. Опираясь на предпроектный анализ можно, утверждать, что разработка ИС «Автосервис» является целесообразной.

Для реализации ИС необходимо выполнить следующее:

* разработать концепцию, определить основные компоненты, модули и их взаимодействие;
* исследовать технологии и инструменты, которые подходят для реализации ИС;
* разработать структуру разрабатываемой ИС, состав функциональных и обеспечивающих подсистем;
* определить требования к аппаратному и программному обеспечению, исходя из функциональных требований ИС.

# **АНАЛИЗ И ПРОЕКТИРОВАНИЕ ИС**

## Разработка архитектуры сайта, выбор технологий реализации ИС

В ходе проведения предпроектного анализа предметной области были выделены цели разработки ИС, а также были проанализированы конкурентные решения. Исходя всего вышесказанного, разрабатываемая ИС будет представлять собой онлайн систему, а именно Web-сайт. После выбора вида приложения необходимо разобраться с выбором технологий разработки. В качестве платформы для разработки был выбран фреймворк Django [7], а в качестве языка программирования – Python [8].

Django - это высокоуровневый веб-фреймворк, который был разработан на языке программирования Python. Django предоставляет инструменты и функциональность для быстрой и эффективной разработки веб-приложений. Огромной список достоинств данного фреймворка объясняет его выбор: удобная система для работы с формами, и url-маршрутами; использование системы шаблонов, которая помогает разделять логику и представление; встроенные механизмы аутентификации и авторизации; встроенный интерфейс администратора для удобного взаимодействия с базой данных и многое другое.

В качестве СУБД была выбрана PostgreSQL [9]. PostgreSQL – это открытая объектно-реляционная система управления базами данных, она обладает хорошей производительностью и большим количеством возможностей.

В результате данного подраздела должна быть построена схема функционирования информационной системы. Перед процессом построения данной схемы необходимо определиться с некоторыми аспектами, один из таких – особенности архитектуры, разрабатываемой ИС. Разрабатываемая ИС будет базироваться на клиент-серверной архитектуре (двухуровневая), где клиентские устройства (браузеры) взаимодействуют с сервером. Данная архитектура работает по следующему принципу, клиентский уровень будет включать пользовательские устройства, на которых находятся программы клиенты, отвечающие за интерфейс пользователя. Все прикладные программы будут сосредоточены на сервере приложений. Клиентский и серверный уровень будут взаимодействовать по следующему принципу: клиентский уровень посылает запросы на сервер, а серверный уровень обрабатывает эти запросы и отправляет обратно необходимые данные. Такая архитектура является более логичной для архитектуры вида «клиент-сервер».

Django использует шаблон проектирования, который включает в себя:

* Модель. В Django модель определяется в виде класса Python и представляет собой таблицу в базе дынных.
* Шаблон. Шаблоны отвечают за отображение данных вашего приложения и создание пользовательского интерфейса.
* Представление. Представление является связующим звеном между моделью и шаблоном. В Django представления обрабатывают запросы от клиентского уровня, извлекает необходимые данные из модели и передает их в шаблон для отображения.

Схема функционирования ИС показана на рисунке 2.1.

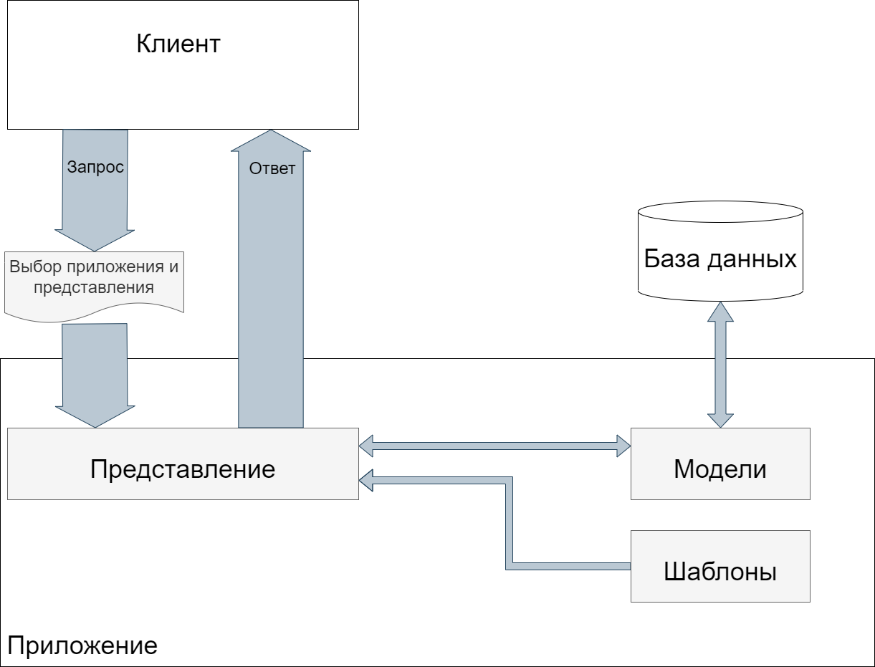


Рисунок 2.1 – схема функционирования ИС автобюро

## Определение структуры и функциональности ИС

Обычно, при проектировании ИС выделяют функциональные и обеспечивающие подсистемы [10]. Функциональные подсистемы, это те, которые выполняют особые функции, связанные с определенными бизнес-процессами и необходимостями организации, например, подсистема управления складом.

В разрабатываемой системе должны быть реализованы следующие функциональные подсистемы:

* подсистема авторизации и регистрации;
* подсистема формирования заказа;
* подсистема новостного форума;
* подсистема администрирования базы данных;

Подсистема авторизации предназначена для доступа пользователей к системе, обеспечение требований защиты данных, хранящихся в системе.

Подсистема оформления заказа представляет собой форму заказа, в которой клиент выбирает услуги, которые необходимы для его автомобиля. После завершения работы с этой формой, она отправляется менеджеру на подтверждение. Менеджер проверяет все данные, назначает время и отправляет информацию клиенту, а также мастеру-приемщику.

Подсистема новостного форума. Для дополнительного привлечения клиентов на сайте будет реализован новостной портал с комментариями. В нем будут публиковаться новости по заранее определенным темам, а посетители сайта смогут оставлять свое мнение в виде комментариев.

Подсистема администрирования базы данных необходима для загрузки и удаления контента ИС, а также редактирования необходимой информации.

В отличие от функциональных, обеспечивающие подсистемы, это те, которые обеспечивают базовую инфраструктуру, необходимую для функционирования ИС.

В разрабатываемой системе должны быть реализованы следующие обеспечивающие подсистемы:

* информационное обеспечение;
* техническое обеспечение;
* программное обеспечение;

Подсистема информационного обеспечения. Данная подсистема отвечает за сбор, хранение, обработку и предоставление информации, которая нужна для правильного функционирования ИС.

Подсистема технического обеспечения. Данная подсистема должна включать в себя физические компоненты и технологии, которые обеспечивают инфраструктуру и обеспечивают работы ИС

Подсистема программного обеспечения. Эта подсистема включает в себя совокупность программных компонентов, которая обеспечивает функциональность ИС.

Структурная схема разрабатываемой информационной системы показана на рисунке 2.2.

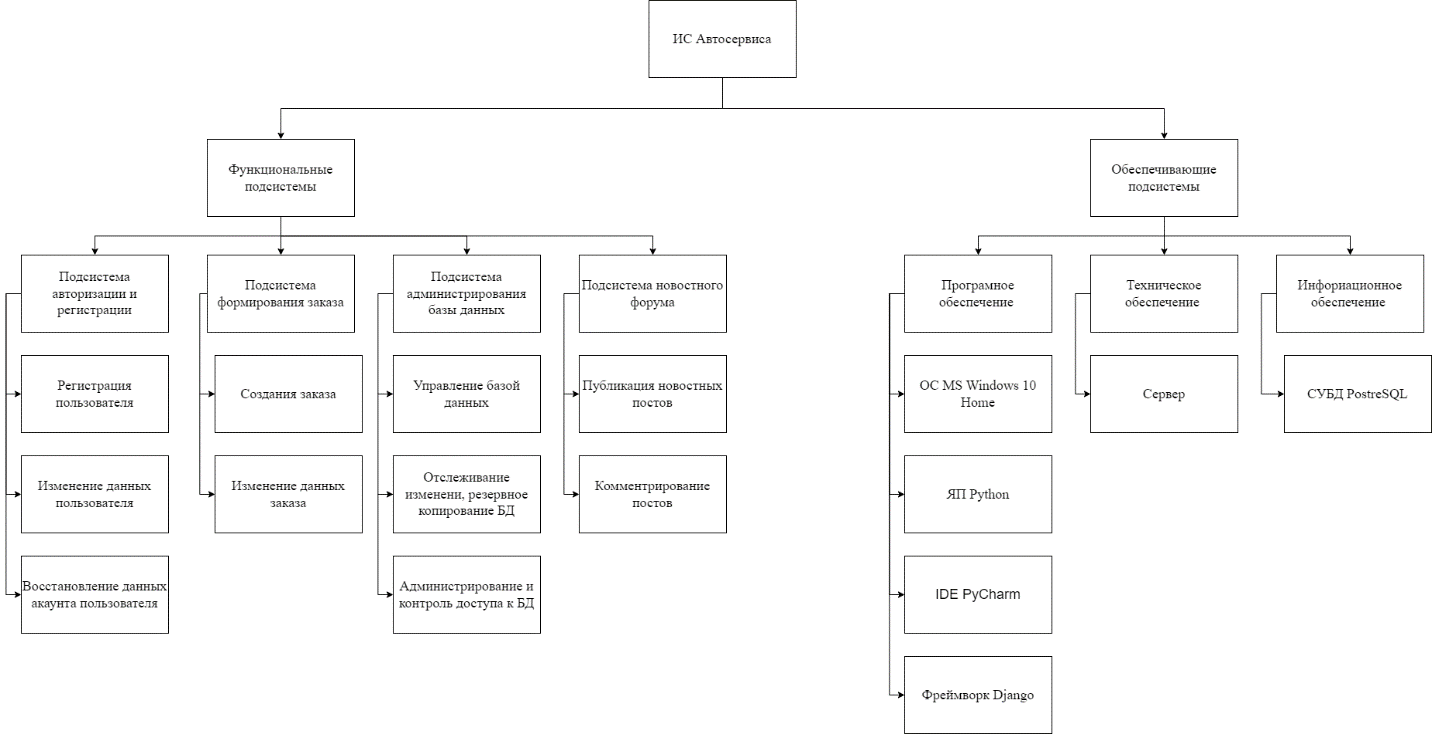


Рисунок 2.2 – Структурная схема ИС автосервиса

Для описания функциональности ИС был выбран язык UML. UML (Unified Modeling Language) - это стандартизированный язык моделирования, который используется для визуализации, проектирования, спецификации и документирования системного проектирования программного обеспечения. UML предоставляет набор графических символов и нотаций, которые позволяют разработчикам создавать абстрактные модели системы, отражающие ее структуру, функции и поведение. В качестве вида диаграммы для описания функциональности ИС была выбрана диаграмма прецедентов.

Диаграмма прецедентов (Use Case Diagram) - это одна из типичных диаграмм, используемых в рамках UML для моделирования функциональных требований системы с точки зрения ее взаимодействия с акторами (actors), которые могут быть пользователями, внешними системами или другими компонентами. Для разрабатываемой ИС субъектами являются:

* суперпользователь
* авторизованный пользователь
* неавторизованный пользователь

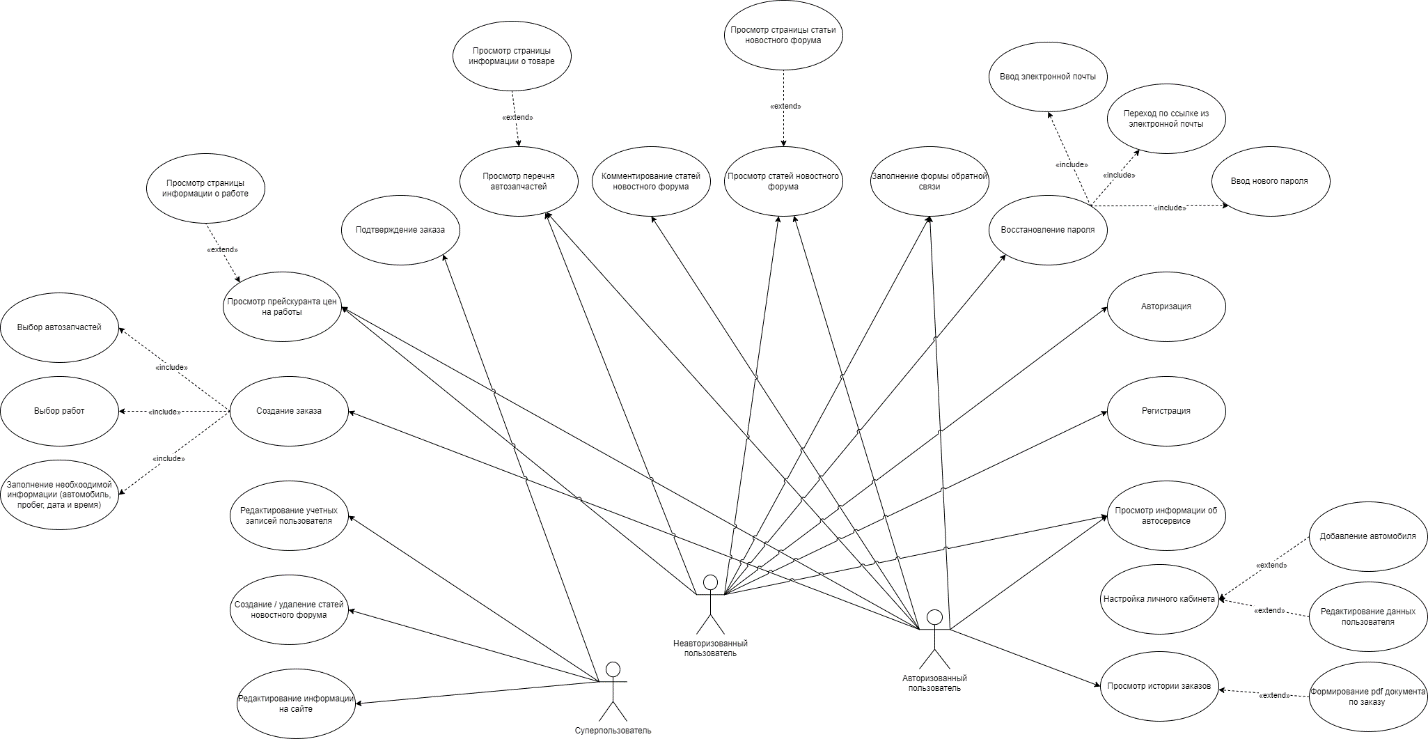
Диаграмма прецедентов для разрабатываемой ИС представлена на рисунке 2.3.

Рисунок 2.3 – Диаграмма прецедентов

## Техническое обеспечение информационной системы

Объем данных проекта предполагается довольно большим. Для размещения проектов больших объемов обычно используются специальные центры обработки данных (ЦОД). Так, готовую информационную систему можно разместить на готовом ЦОД, а можно на своем собственном.

Дата-центр, или центр (хранения и) обработки данных (ЦОД/ЦХОД) – это специализированное здание для размещения (хостинга) серверного и сетевого оборудования и подключения абонентов к каналам сети Интернет. Чтобы обеспечить полноценную бесперебойную работу всех функций информационной системы, потребуется комплекс технических средств, удовлетворяющий потребностям. Для этого было принято решение использовать группу из трёх серверов, характеристики каждого из которых: - процессор - Intel Xeon E-2224G: 4 х 3.5 ГГц;

- оперативная память – DDR4, 64 Гб, 2.666 ГГц;

- жесткий диск – SATA 3, 1 ТБ;

- картридер;

- сетевой адаптер – 100 Мбит.

Таких характеристик технических средств информационной системы будет достаточно для полноценного функционирования системы. Под полноценным функционирование понимается:  гарантированное выполнение всего комплекса программного обеспечения в случае сбоя или выхода из строя части оборудования;  защиту данных от несанкционированного доступа.

Общий вид СКС центра обработки данных информационной системы тематических блогов представлен на рисунке 2.3.

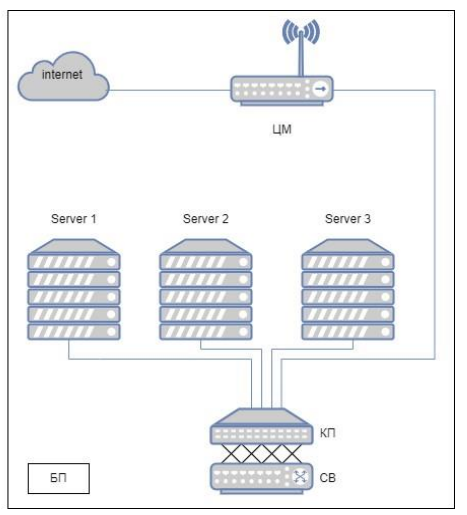


Рисунок 2.3 – Общий вид СКС центра обработки данных

# **РАЗРАБОТКА ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ ИС**

## Концептуальное проектирование БД

В общем виде процесс проектирования базы данных подразделяется на следующие этапы:

* концептуальное проектирование;
* логическое проектирование;
* физическое проектирование;

Концептуальное проектирование базы данных является первым шагом. На данном этапе определяются основные сущности, их атрибуты и типы связей между ними. На данном этапе еще нет никакой привязки к конкретным структурам данных и технологиям хранения. Графическим инструментом на этапе концептуального проектирования является диаграмма «сущность-связь». Диаграмма «сущность-связь» позволяет визуализировать сущности, их атрибуты и связи между ними. Элементы, которые включает диаграмма:

* Сущность. Сущности представляют объекты или концепции, которые хранятся в базе данных. Каждая сущность отражается на диаграмме прямоугольником, в котором указывается название сущности.
* Атрибут. Атрибуты представляют характеристики или свойства сущностей, они отображаются внутри прямоугольника сущности.
* Связь. Связи определяют виды отношений между сущностями, они показывают, как именно одна сущность связана с другой. Связи изображаются линиями, которые соединяют связанные сущности.

На диаграмме сущность-связь изображаются различные виды связей:

* один-к-одному;
* один-ко-многим;
* многие-ко-многим.

Концептуальная модель разрабатываемого приложения представлена на рисунке 3.1.

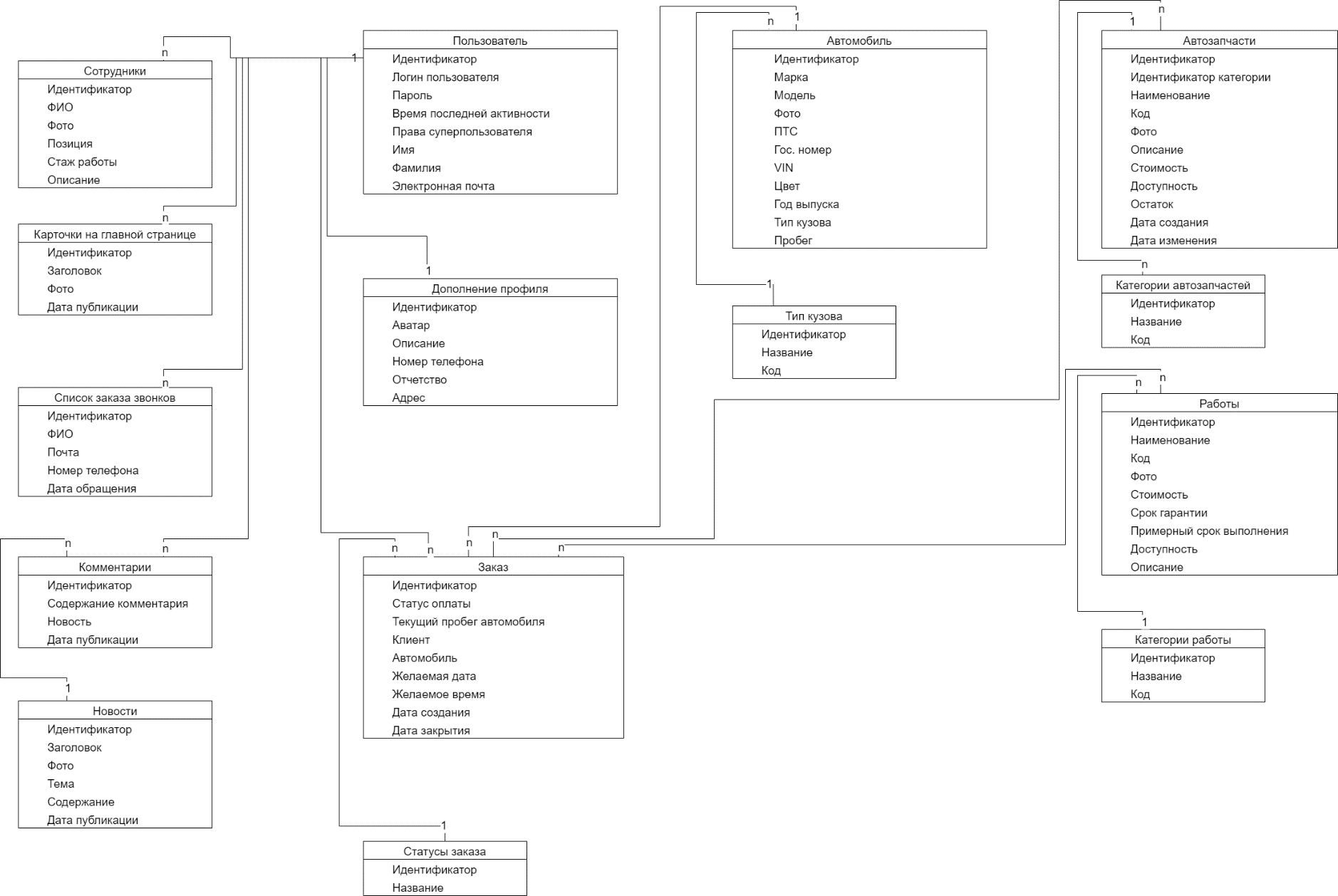


Рисунок 3.1 – Концептуальная модель ИС

## Логическое проектирование БД

Логическое проектирование базы данных является следующим шагом после концептуального проектирования и представляет собой процесс перевода концептуальной модели в логическую модель. В логической модели базы данных определяются таблицы, их атрибуты, связи и правила целостности данных.

Первым этапом построения логической модели является определение сущностей и их атрибутов, опираясь на концептуальную модель.

Вторым этапом идет определение первичных ключей. На каждой таблице определяется первичный ключ, который идентифицирует каждую запись в таблице БД.

На третьем этапе идет определение внешних ключей. Внешние ключи связывают таблицы между собой. Внешний ключ представляет собой атрибут, который ссылается на первичный ключ в другой таблице. Также внешний ключ определяет связи между собой.

Последним этапом необходимо провести нормализацию, это необходимо, чтобы устранить избыточность данных. В этот процесс также может входить разделение крупных таблиц на более мелкие и более связанные между собой таблицы. Это позволит обеспечить высокую структурную целостность и эффективность базы данных.

Логическая модель разрабатываемого приложения представлена на рисунке 3.2.

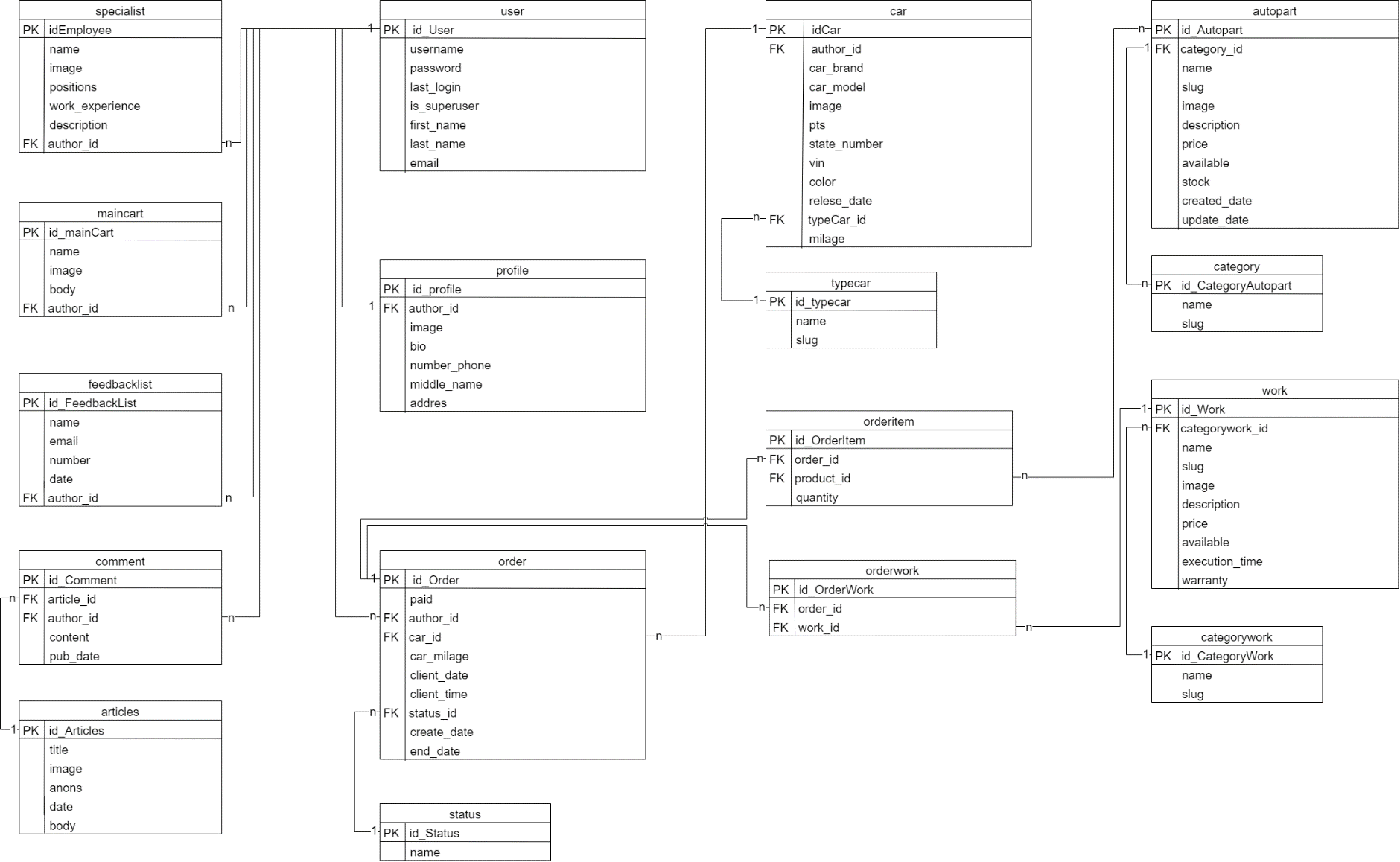


Рисунок 3.2 – Логическая модель ИС

## Физическое проектирование БД

На этапе физического проектирования происходит преобразование логической модели базы данных в реализуемую физическую структуру на выбранной платформе СУБД. Именно поэтому физическое проектирование проводится после выбора и с учетом особенностей выбранной СУБД.

В качестве платформы для физического проектирования была выбрана СУБД PostgreSQL. PostgreSQL – это открытая объектно-реляционная система управления базами данных, она обладает хорошей производительностью и большим количеством возможностей.

На таблицах 3.2 – 3.17 отображена информация об основных элементах, используемых в информационной системе, а также о типе данных и их назначениях.

Таблица 3.1 – Перечень таблиц базы данных

|  |  |  |
| --- | --- | --- |
| №  п/п | Название таблицы | Описание таблицы |
| 1 | *specialist* | Таблица, хранящая информацию о сотрудниках |
| 2 | *maincart* | Таблица, хранящая информацию о карточках главной страницы |
| 3 | *feedbacklist* | Таблица, хранящая информацию о списке обратной связи |
| 4 | *comment* | Таблица, хранящая информацию о комментариях в постах |
| 5 | *articles* | Таблица, содержащая информацию о постах |
| 6 | *user* | Таблица, содержащая информацию о пользователях |
| 7 | *profile* | Таблица, содержащая информацию о комментариях |
| 8 | *order* | Таблица, содержащая информацию о заказах |
| 9 | *status* | Таблица, содержащая информацию о статусе  заказа |
| 10 | *car* | Таблица, содержащая информацию об автомобилях |
| 11 | *typecar* | Таблица, содержащая информацию о типах кузова автомобилей |
| 12 | *orderitem* | Таблица, содержащая информацию о запчастях в заказе |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 13 | *orderwork* | Таблица, содержащая информацию о работах в заказе |
| 14 | *autopart* | Таблица, содержащая информацию об автозапчастях в заказе |
| 15 | *category* | Таблица, содержащая информацию о категориях автозапчастей |
| 16 | *work* | Таблица, содержащая информацию о всех работах |
| 17 | *categorywork* | Таблица, содержащая информацию о категориях работ |

В таблицах 3.2- 3.17 представлено описание полей отдельных таблиц ИБ с указанием ограничений.

Таблица 3.2 – Описание таблицы specialist

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Назначение | Примечания |
| *Id\_Employee* | *IntegerField* | Первичный ключ | *PRIMARY KEY AUTOINCREMENT* |
| *name* | *CharField(250)* | ФИО | - |
| *image* | *FileField* | Фото | - |
| *positions* | *CharField(250)* | Специализация | - |
| *work\_experience* | *IntegerField* | Стаж работы | - |
| *description* | *TextField* | О себе | - |
| *author\_id* | *IntegerField* | ID автора | *FOREIGN KEY* |

Таблица 3.3 – Описание таблицы maincart

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Назначение | Примечания |
| *Id\_MainCart* | *IntegerField* | Первичный ключ | *PRIMARY KEY AUTOINCREMENT* |
| *name* | *CharField(250)* | Заголовок | - |
| *image* | *FileField* | Фото | - |
| *body* | *CharField(250)* | Тело записи | - |
| *author\_id* | *IntegerField* | ID автора | *FOREIGN KEY* |

Таблица 3.4 – Описание таблицы feedbacklist

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Назначение | Примечания |
| *Id\_MainCart* | *IntegerField* | Первичный ключ | *PRIMARY KEY AUTOINCREMENT* |
| *name* | *CharField(250)* | ФИО | - |
| *email* | *CharField(250)* | Почта | - |
| *number* | *CharField(250)* | Телефон | - |
| *date* | *DateField* | Дата заявки | - |
| *author\_id* | *IntegerField* | ID автора | *FOREIGN KEY* |

Таблица 3.5 – Описание таблицы comment

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Назначение | Примечания |
| *Id\_Comment* | *IntegerField* | Первичный ключ | *PRIMARY KEY AUTOINCREMENT* |
| *article\_id* | *IntegerField* | ID поста | *FOREIGN KEY* |
| *author\_id* | *IntegerField* | ID автора | *FOREIGN KEY* |
| *content* | *CharField(250)* | Комментарий | *-* |
| *pub\_date* | *DateField* | Дата комментария | *-* |

Таблица 3.6 – Описание таблицы articles

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Назначение | Примечания |
| *Id\_Article* | *IntegerField* | Первичный ключ | *PRIMARY KEY AUTOINCREMENT* |
| *title* | *CharField(250)* | Заголовок | *-* |
| *image* | *FileField* | Превью | *-* |
| *anons* | *CharField(250)* | Тема | *-* |
| *date* | *DateField* | Дата | *-* |
| *body* | *TextField* | Содержание | *-* |

Таблица 3.7 – Описание таблицы user

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Назначение | Примечания |
| *Id\_User* | *IntegerField* | Первичный ключ | *PRIMARY KEY AUTOINCREMENT* |

|  |  |  |  |
| --- | --- | --- | --- |
| *username* | *CharField(250)* | Логин | *-* |
| *password* | *CharField(250* | Пароль | *-* |
| *last\_login* | *DateField* | Дата посл. активн. | *-* |
| *is\_superuser* | *BooleanField* | Права суперполь. | *-* |
| *first\_name* | *CharField(250)* | Имя | *-* |
| *last\_name* | *CharField(250)* | Фамилия | *-* |
| *email* | *CharField(250)* | Почта | *-* |

Таблица 3.8 – Описание таблицы profile

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Назначение | Примечания |
| *Id\_Profile* | *IntegerField* | Первичный ключ | *PRIMARY KEY AUTOINCREMENT* |
| *author\_id* | *IntegerField* | ID пользователя | *FOREIGN KEY* |
| *image* | *FileField* | Превью | *-* |
| *bio* | *TextField* | О себе | *-* |
| *number\_phone* | *CharField(250)* | Номер телефона | *-* |
| *middle\_name* | *CharField(250)* | Отчество | *-* |
| *addres* | *CharField(250)* | Адрес | *-* |

Таблица 3.9 – Описание таблицы order

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Назначение | Примечания |
| *Id\_Order* | *IntegerField* | Первичный ключ | *PRIMARY KEY AUTOINCREMENT* |
| *paid* | *BooleanField* | Оплата | *-* |
| *author\_id* | *IntegerField* | ID пользователя | *FOREIGN KEY* |
| *car\_id* | *IntegerField* | ID автомобиля | *FOREIGN KEY* |

Продолжение таблицы 3.9

|  |  |  |  |
| --- | --- | --- | --- |
| *car\_milage* | *CharField(250)* | Пробег авто | *-* |
| *client\_date* | *DateField* | Дата клиента | *-* |
| *client\_time* | *DateField* | Время клиента | *-* |
| *status\_id* | *IntegerField* | ID статуса заказа | *FOREIGN KEY* |
| *create\_date* | *DateField* | Дата создания | *-* |

Таблица 3.10 – Описание таблицы status

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Назначение | Примечания |
| *Id\_Status* | *IntegerField* | Первичный ключ | *PRIMARY KEY AUTOINCREMENT* |
| *Name* | *CharField(250)* | Наименование | *-* |

Таблица 3.11 – Описание таблицы car

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Назначение | Примечания |
| *Id\_Car* | *IntegerField* | Первичный ключ | *PRIMARY KEY AUTOINCREMENT* |
| *author\_id* | *IntegerField* | ID пользователя | *FOREIGN KEY* |
| *car\_brand* | *CharField(250)* | Марка | *-* |
| *car\_model* | *CharField(250)* | Модель | *-* |
| *image* | *FileField* | Фото | *-* |
| *pts* | *CharField(250)* | Номер ПТС | *-* |
| *state\_number* | *CharField(250)* | Гос. Номер | *-* |
| *vin* | *CharField(250)* | VIN | *-* |
| *color* | *CharField(250)* | Цвет | *-* |
| *relese\_date* | *DateField* | Дата выпуска | *-* |
| *typeCar\_id* | *IntegerField* | ID типа кузова | *FOREIGN KEY* |
| *milage* | *IntegerField* | Пробег авто | *-* |

Таблица 3.12 – Описание таблицы typecar

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Назначение | Примечания |
| *Id\_TypeCar* | *IntegerField* | Первичный ключ | *PRIMARY KEY AUTOINCREMENT* |

Продолжение таблицы 3.12

|  |  |  |  |
| --- | --- | --- | --- |
| *name* | *CharField(250)* | Наименование | *-* |
| *slug* | *CharField(250)* | Код | *-* |

Таблица 3.13 – Описание таблицы orderitem

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Назначение | Примечания |
| *Id\_OrderItem* | *IntegerField* | Первичный ключ | *PRIMARY KEY AUTOINCREMENT* |

|  |  |  |  |
| --- | --- | --- | --- |
| *order\_id* | *IntegerField* | ID заказа | *FOREIGN KEY* |
| *product\_id* | *IntegerField* | ID автозапчасти | *FOREIGN KEY* |
| *quantity* | *PositiveIntegerField* | Количество | *-* |

Таблица 3.14 – Описание таблицы orderwork

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Назначение | Примечания |
| *Id\_OrderWork* | *IntegerField* | Первичный ключ | *PRIMARY KEY AUTOINCREMENT* |
| *order\_id* | *IntegerField* | ID заказа | *FOREIGN KEY* |
| *work\_id* | *IntegerField* | ID работы | *FOREIGN KEY* |

Таблица 3.15 – Описание таблицы autopart

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Назначение | Примечания |
| *Id\_Autopart* | *IntegerField* | Первичный ключ | *PRIMARY KEY AUTOINCREMENT* |
| *category\_id* | *IntegerField* | ID категории | *FOREIGN KEY* |
| *name* | *CharField(250)* | Наименование | *-* |
| *slug* | *CharField(250)* | Код | *-* |
| *image* | *FileField* | Изображение | *-* |
| *description* | *CharField(250)* | Описание | *-* |
| *price* | *DecimalField* | Стоимость | *-* |
| *available* | *BooleanField* | Доступность | *-* |
| *stock* | *PositiveIntegerField* | Остаток | *-* |
| *created\_date* | *DateField* | Дата создания | *-* |
| *update\_date* | *DateField* | Дата издания | *-* |

Таблица 3.16 – Описание таблицы category

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Назначение | Примечания |
| *Id\_CategoryAutopart* | *IntegerField* | Первичный ключ | *PRIMARY KEY AUTOINCREMENT* |
| *name* | *CharField(250)* | Наименование | *-* |
| *slug* | *CharField(250)* | Код | *-* |

Таблица 3.17 – Описание таблицы work

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Назначение | Примечания |
| *Id\_Work* | *IntegerField* | Первичный ключ | *PRIMARY KEY AUTOINCREMENT* |
| *categorywork\_id* | *IntegerField* | ID категории | *FOREIGN KEY* |
| *name* | *CharField(250)* | Наименование | *-* |
| *slug* | *CharField(250)* | Код | *-* |
| *image* | *FileField* | Изображение | *-* |
| *description* | *CharField(250)* | Описание | *-* |
| *price* | *DecimalField* | Стоимость | *-* |
| *available* | *BooleanField* | Доступность | *-* |
| *execution\_time* | *IntegerField* | Время выполнения | *-* |
| *warranty* | *IntegerField* | Срок гарантии | *-* |

Таблица 3.18 – Описание таблицы categorywork

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип данных | Назначение | Примечания |
| *Id\_CategoryWork* | *IntegerField* | Первичный ключ | *PRIMARY KEY AUTOINCREMENT* |
| *name* | *CharField(250)* | Наименование | *-* |
| *slug* | *CharField(250)* | Код | *-* |

Далее необходимо выполнить расчет максимального объема информационной базы. При расчете необходимо учитывать типы полей моделей, количество моделей, количество записей. Для этого воспользуемся следующими формулами:

,

где – коэффициент блокирования ;

– максимальный объем i-го объекта ИБ, i=1,N;

N – количество объектов ИБ

,

где – кол-во атрибутов i-го объекта ИБ;

– размер в байтах j-го атрибута i-го объекта;

– макс. кол-во экз. i-го объекта ИБ с учетом логических связей.

Для расчета объема информационной базы также необходимо построить вспомогательную таблицу (таблица 3.19), которая будет содержать все таблицы ИБ, максимальное количество записей, количество различных видов атрибутов.

Таблица 3.19 – Описание атрибутов всех таблиц ИБ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *№* | *Таблица* | *Максимальн. количество записей* | *Количество атрибутов CharField* | *Количество атрибутов IntegerField* | *Количество атрибутов DateField* | *Объем прочих атрибутов* |
| *1* | *specialist* | *500* | *2* | *3* | *0* | *355* |
| *2* | *maincart* | *50* | *2* | *2* | *0* | *100* |
| *3* | *feedbacklist* | *100000* | *3* | *2* | *1* | *0* |
| *4* | *comment* | *100000* | *1* | *3* | *1* | *0* |
| *5* | *articles* | *10000* | *2* | *1* | *1* | *355* |
| *6* | *user* | *10000* | *4* | *1* | *1* | *1* |
| *7* | *profile* | *10000* | *3* | *2* | *0* | *355* |
| *8* | *order* | *100000* | *1* | *4* | *3* | *1* |
| *9* | *status* | *100* | *1* | *1* | *0* | *0* |

Продолжение таблицы 3.19

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *10* | *car* | *20000* | *7* | *3* | *1* | *100* |
| *11* | *typecar* | *100* | *2* | *1* | *0* | *0* |
| *12* | *orderitem* | *100000* | *0* | *4* | *0* | *0* |
| *13* | *orderwork* | *100000* | *0* | *3* | *0* | *0* |
| *14* | *autopart* | *100000* | *3* | *2* | *2* | *109* |
| *15* | *category* | *100* | *2* | *1* | *0* | *0* |
| *16* | *work* | *1000* | *3* | *4* | *0* | *109* |
| *17* | *categorywork* | *100* | *2* | *1* | *0* | *0* |

Проведем расчет максимального объема для каждой таблицы ( ), индекс i равен номеру таблицы по порядку в таблице 3.18.

байт

байт

байт

байт

байт

байт

байт

байт

байт

байт

байт

байт

байт

байт

ба йт

ба йт

байт

Далее проведём расчет максимального объема ИБ:

ба йт

В конце переведем полученное значение в Мб

байт = 1217,52 Мб

Проведенный в этом разделе расчет является скорее теоретическим, полученное значение возможно достичь только при максимальном заполнении всех таблиц базы данных.

# **ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ИС**

## Структура программного обеспечения

Для программной реализации ИС использовалась IDE PyCharm. Согласно официальному сайту [11], PyCharm — это IDE для профессиональной разработки на Python. PyCharm поддерживает современные фреймворки для веб-разработки: Django, Flask, Google App Engine, Pyramid и web2py. Помимо Python, PyCharm поддерживает JavaScript, CoffeeScript, TypeScript, Cython, SQL, HTML/CSS, языки шаблонов, AngularJS, Node.js и другие технологии.

PyCharm — это кроссплатформенная интегрированная среда разработки для языка программирования Python, разработанная компанией JetBrains на основе IntelliJ IDEA. Предоставляет пользователю комплекс средств для написания кода и визуальный отладчик.

Серверное приложение написано на Python, с помощью фреймворка Django. Согласно информации, из официального сайта, Django - это веб-фреймворк на Python высокого уровня, который поощряет быструю разработку и чистый, прагматичный дизайн. Созданный опытными разработчиками, он берет на себя большую часть хлопот по веб-разработке, поэтому вы можете сосредоточиться на написании своего приложения без необходимости изобретать велосипед. Это бесплатно и с открытым исходным кодом. Django - свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC. Проект поддерживается организацией Django Software Foundation. Сайт на Django строится из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и подключаемыми. Это одно из существенных архитектурных отличий этого фреймворка от некоторых других. Для работы с базой данных Django использует собственный ORM, в котором модель данных описывается классами Python, и по ней генерируется схема базы данных.

На рисунке 4.1 представлена структура проекта ИС.

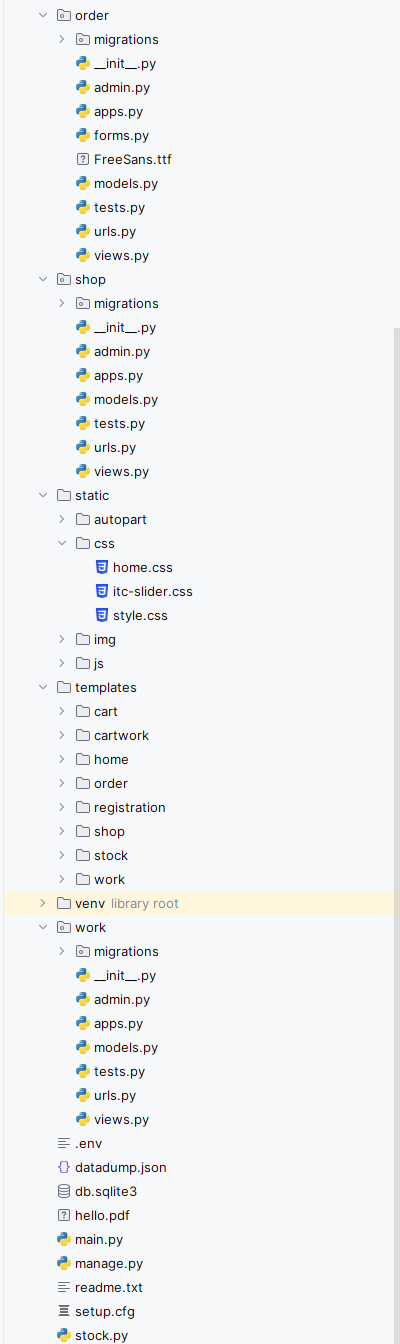
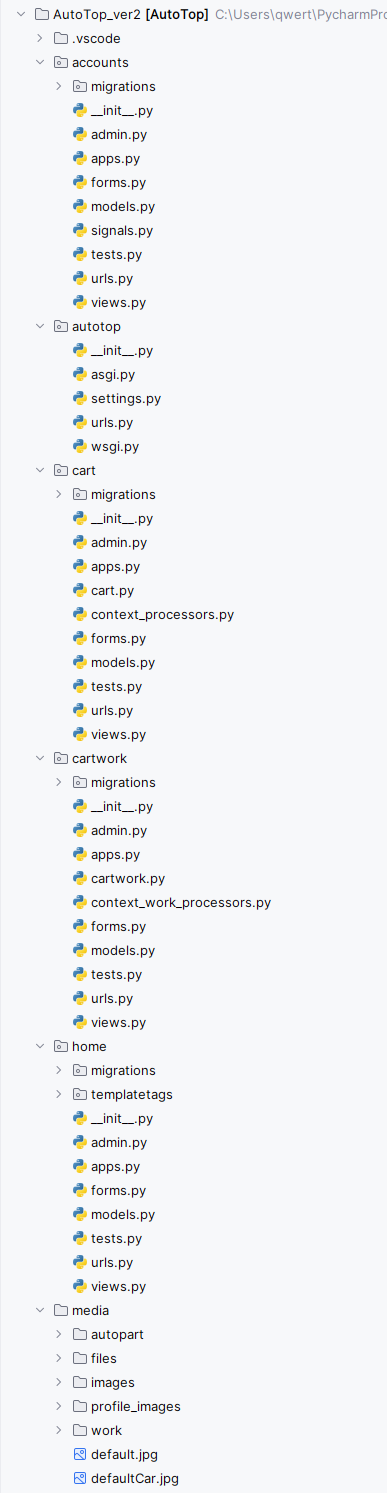


Рисунок 4.1 – Структура проекта ИС

## Запросы к базе данных

На рисунках 4.2-4.10 продемонстрированы за к реализованной физической модели данных информационной системы.

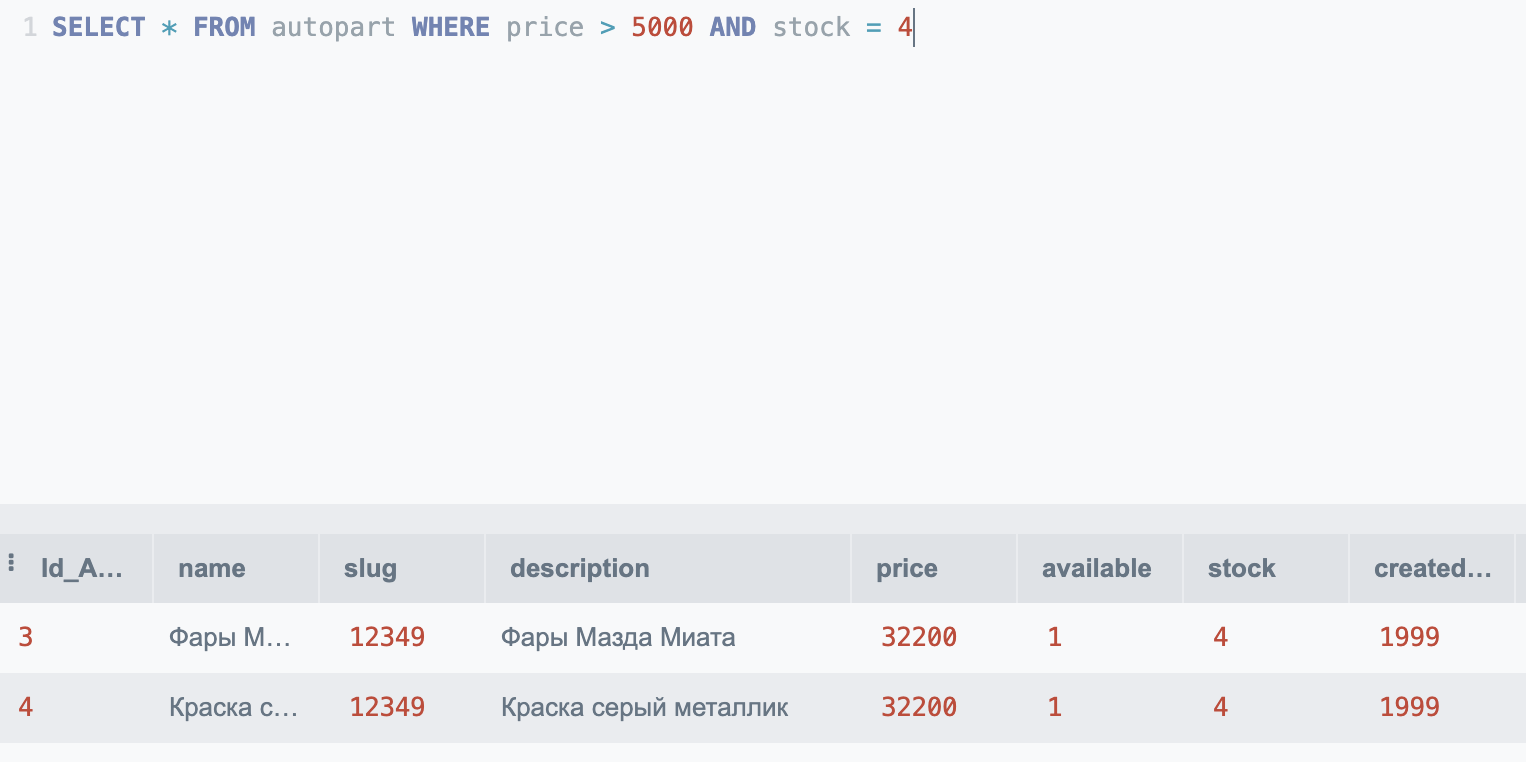


Рисунок 4.2 – Запрос к таблице autopart с условием “price > 5000 and stock = 4”



Рисунок 4.3 – Запрос к таблице autopart для вывода суммы количества деталей на складе с кодом “12349”



Рисунок 4.4 – Запрос к таблице autopart для вывода средней цены на запчасти в наличии

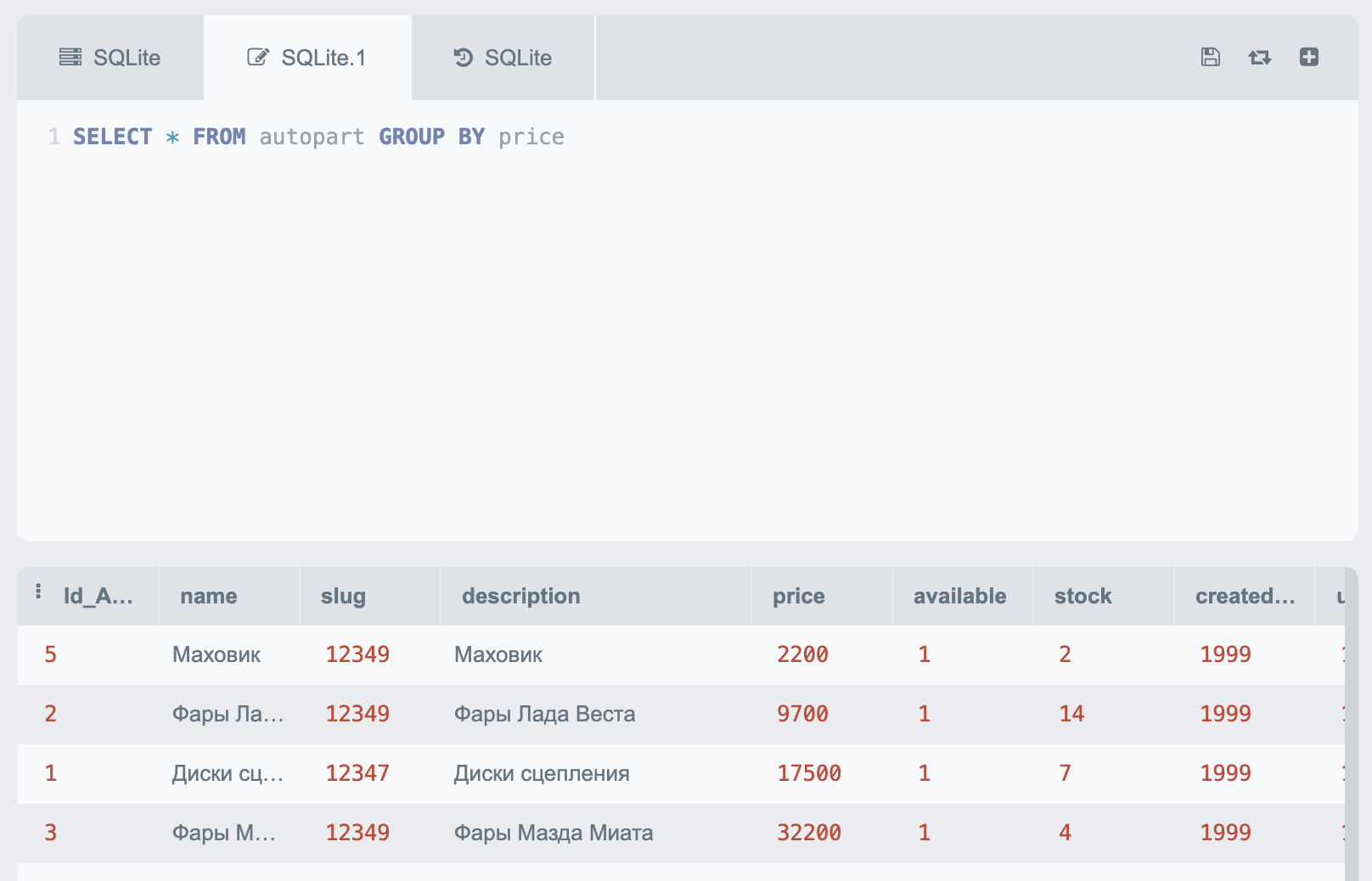


Рисунок 4.5 – Запрос к таблице autopart для вывода всех товаров с группировкой по цене

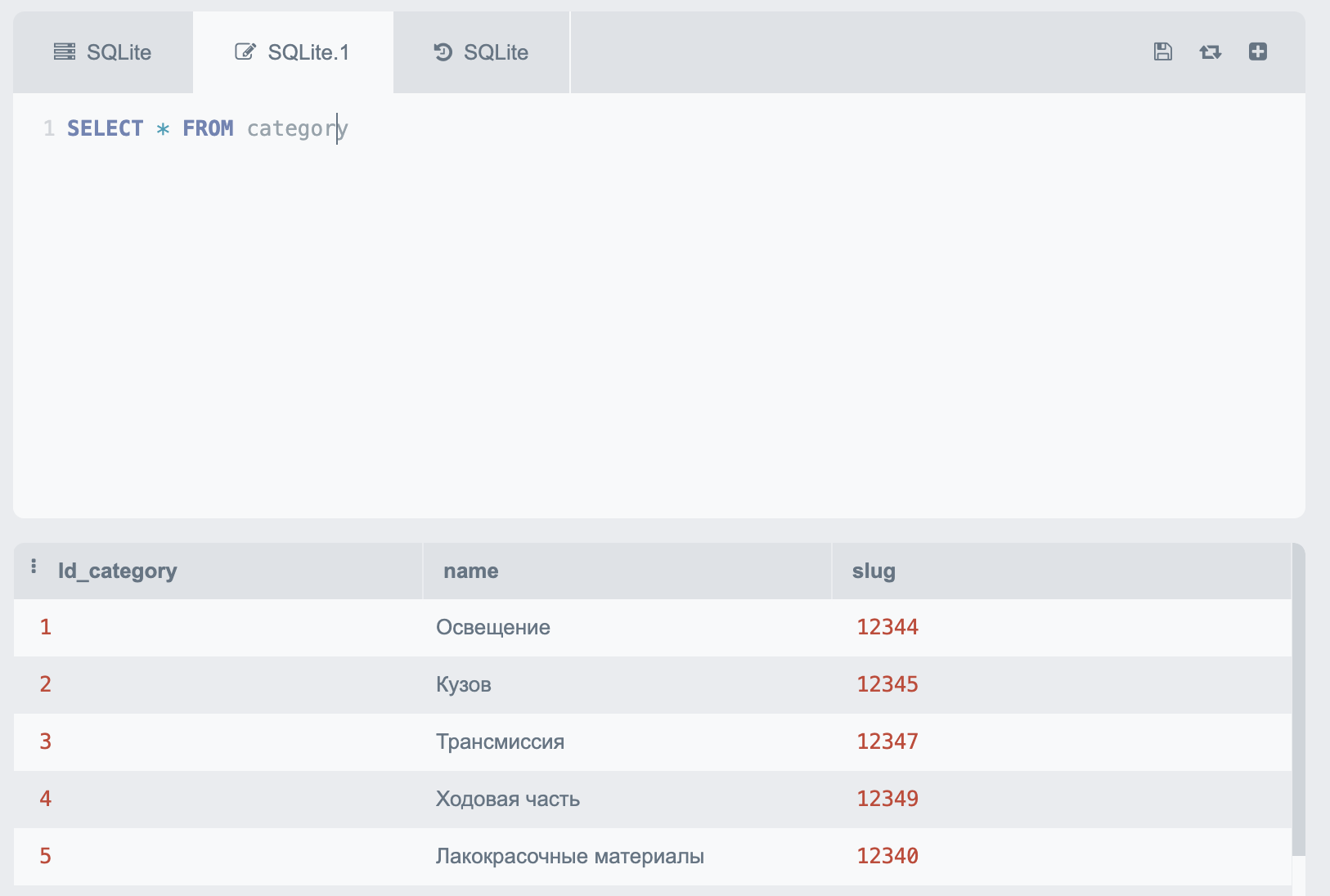


Рисунок 4.6 – Запрос к таблице autopart для вывода всех товаров категорий товаров

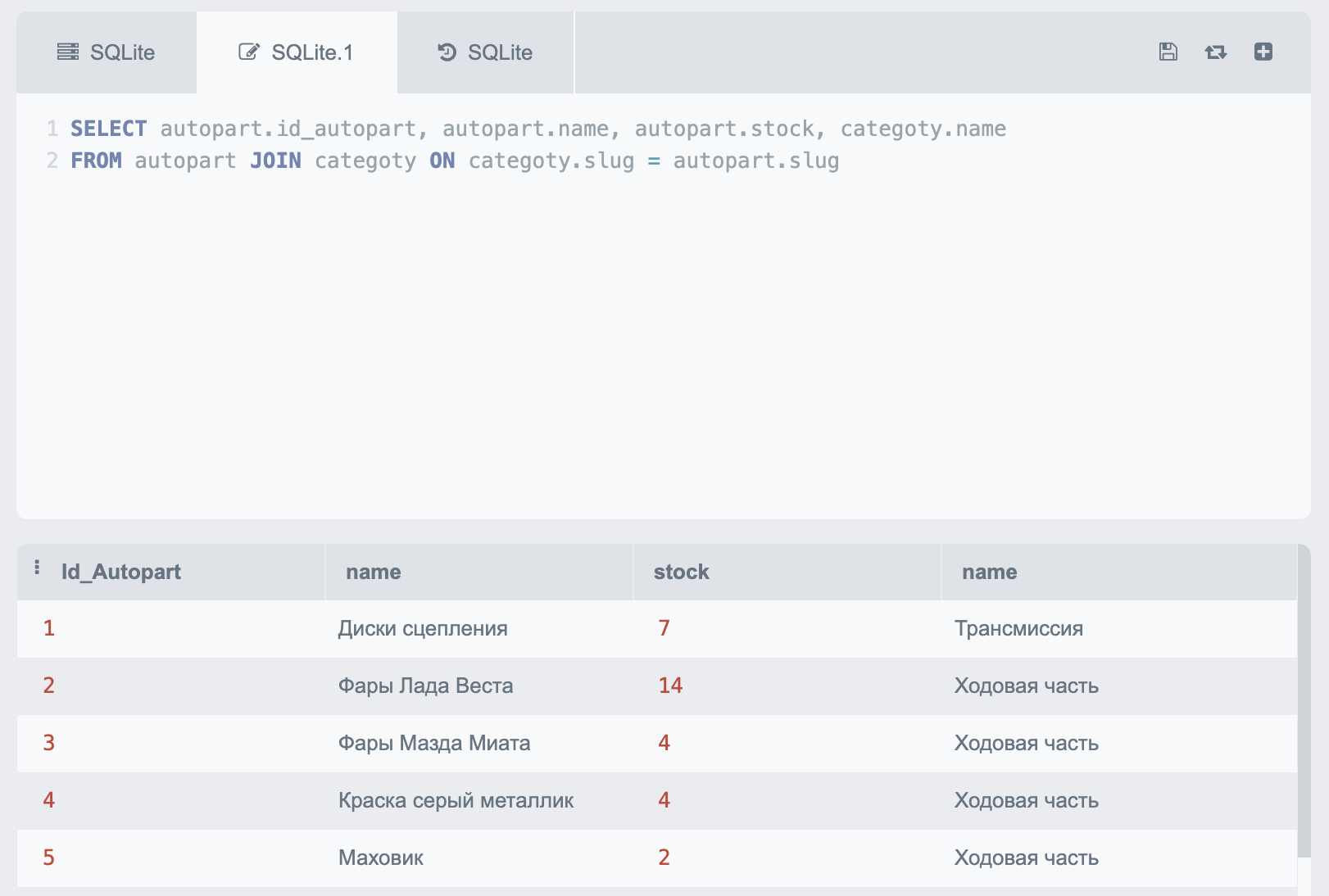


Рисунок 4.7 – Запрос к таблицам autopart и category с объединением по условию индентичности кодов



Рисунок 4.8 – Запрос к таблице car с условием вывода списка машин, их госномера и пробега в порядке возрастания пробега

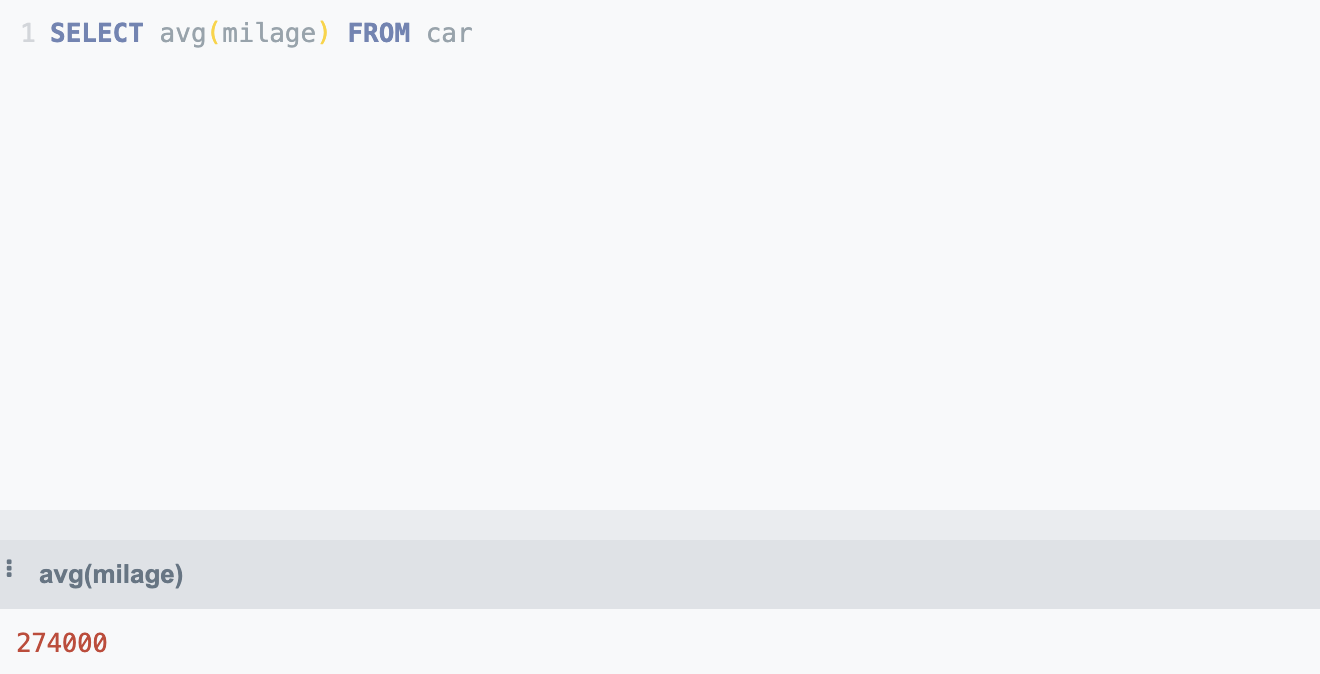


Рисунок 4.9 – Запрос к таблице car с условием вывода среднего пробега всех машин в базе данных

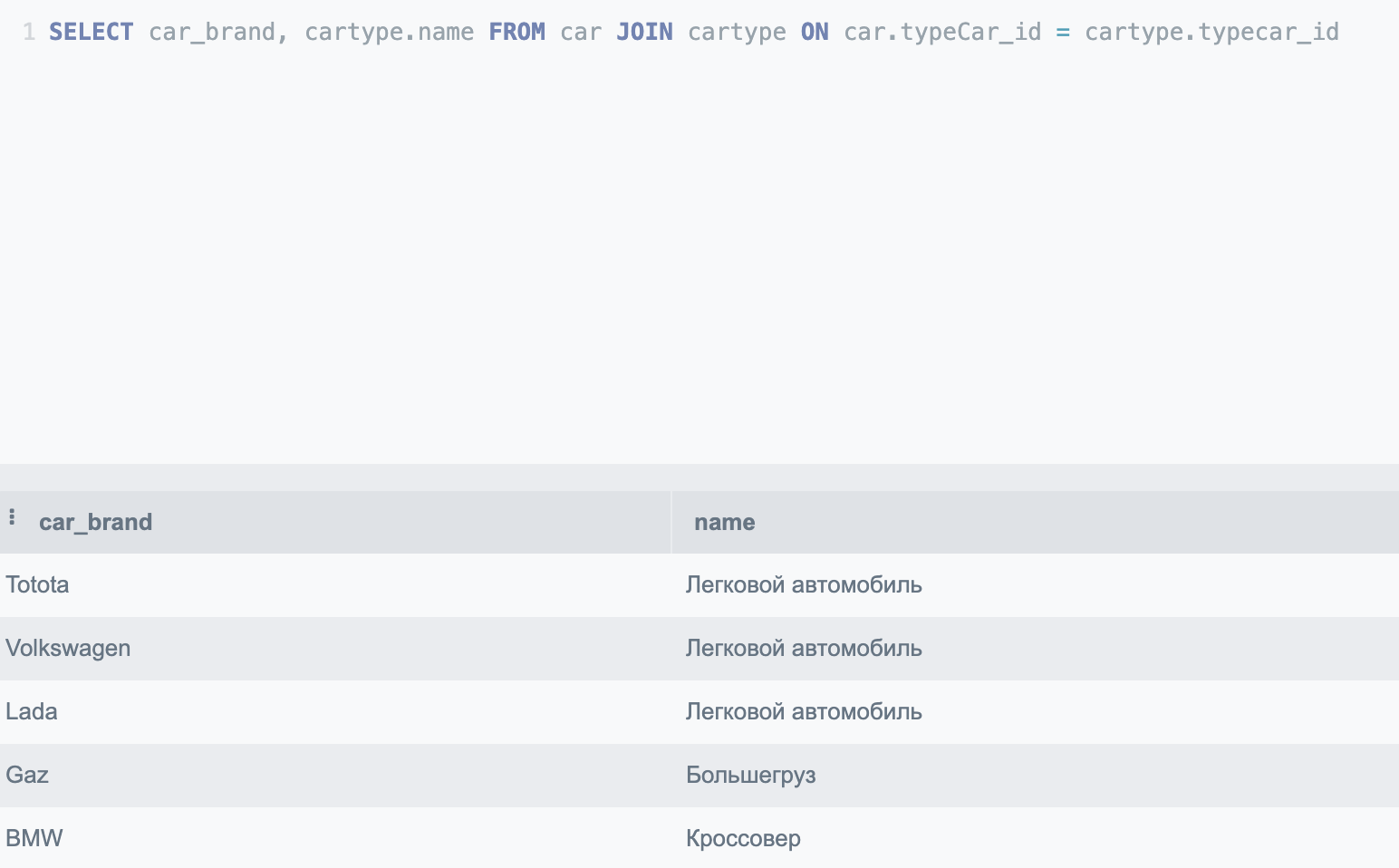


Рисунок 4.10 – Запрос к таблице car с объединением с таблицей cartype для вывода соответсвий марки автомобиля и наименования типа автомобиля

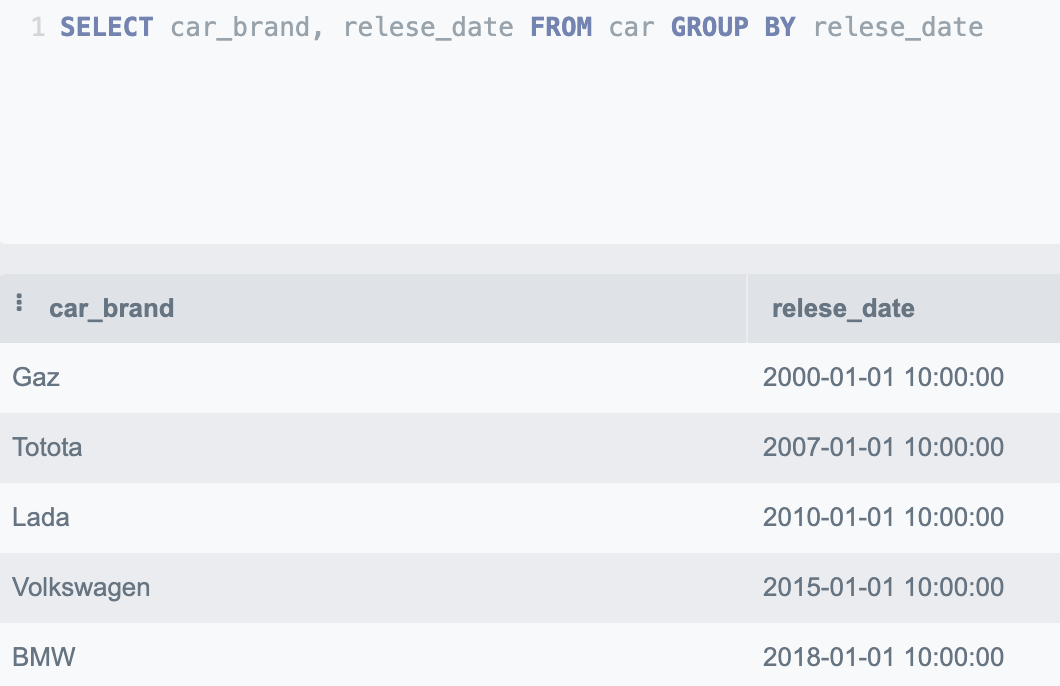


Рисунок 4.11 – Запрос к таблице car для вывода года выпуска автомобиля для определения самых новых и старых машин в базе данных

## Описание пользовательского интерфейса

## Описание интерфейса администратора

# ЗАКЛЮЧЕНИЕ

В результате написания ВКР, был разработан функциональный и удобный сайт автобюро, который позволяет клиентам легко и удобно взаимодействовать с автобюро, получать информацию о услугах, записываться на сервис и отслеживать историю обслуживания своих автомобилей. В ходе выполнения работы были пройдены следующие шаги:

* проведено описание предметной области;
* описаны функции организационной структуры;
* проведено описание потоков данных и бизнес процессов, с применением диаграмм нотации idef0;
* выполнено рассмотрение аналогичных проектных решений;
* определена структура функциональных и обеспечивающих подсистем ИС;
* спроектированы концептуальная и логическая модель ИС
* спроектирована физическая модель ИБ, произведен расчет максимального объема информационной базы;
* разработана программная реализация ИС с применением фреймворка Django.

Внедрение проекта информационной системы позволит увеличить приток клиентов, а также автоматизировать процесс записи и ведения документации. В дальнейшем возможно расширение функциональности сайта, добавление новых модулей и интеграция с другими системами, чтобы еще больше улучшить опыт пользователей и повысить эффективность работы автосервиса.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. КонсультантПлюс. Постановление Правительства РФ от 11.04.2001 N 290 (ред. от 31.01.2017) "Об утверждении Правил оказания услуг по ТО и ремонту автомототранспортных средств" - [Электронный ресурс] - URL: <https://www.consultant.ru/document/cons_doc_LAW_31220/4672ea2ced01b7a5f0b1b7747241ba9494758cb9/> (дата обращения: 10.02.2023)
2. Владымцев Н.В., Извольская И.В., Принцип моделирования бизнес-процессов в стандарте IDEF0, 2008,.140с.;
3. Калянов Г. Н., Концептуальная модель DFD-технологии, 2017. 35с,;
4. Автосервис «zimwerk» - [Электронный ресурс] - URL: <https://zimwerk.ru/msk> (дата обращения: 22.03.2023)
5. Автосервис «На ходу» - [Электронный ресурс] - URL: <https://novocherkassk.na-hodu.ru/> (дата обращения: 23.03.2023)
6. Автосервис «Гольф авто» - [Электронный ресурс] - URL: <http://www.golfauto.ru/> (дата обращения: 24.03.2023)
7. The web framework for perfectionists with deadlines | Django - [Электронный ресурс] - URL: <https://www.djangoproject.com/> (дата обращения: 11.04.2023)
8. Welcome to Python.org - [Электронный ресурс] - URL: <https://www.python.org/> (дата обращения: 22.04.2023)
9. PostgreSQL: The World's Most Advanced Open Source Relational Database - [Электронный ресурс] - URL: <https://www.postgresql.org/> (дата обращения: 30.04.2023)
10. О.Г. Инюшкина, Проектирование информационных систем (на примере методов структурного системного анализа), 2014, 240с.;
11. PyCharm: The Python IDE for Professional Developers - [Электронный ресурс] - URL: <https://www.jetbrains.com/pycharm/> (дата обращения: 22.05.2023)