**Научно-исследовательская часть**

**2.1 Постановка задачи проектирования**

Разрабатываемая база данных будет предназначена для реализации корпоративного приложения, со следующей функциональностью.

* Учет прихода и продажи автомобилей;
* Учет прихода и продажи товаров для автомобилей (аксессуары);
* Проведение необходимых документов;
* Хранение информационной базы по клиентам, автомобилям и аксессуарам;
* Возможность рассчитать кредит;

База данных будет содержать информацию о автомобилях, их производителе, модели и комплектации. Также, она позволит узнавать информацию о аксессуарах которые имеются в наличие у автосалона. Пользователь сможет добавлять товары в базу данных, создавать договоры и заявки, вести базу данных клиентов.

При входе в систему каждый сотрудник должен иметь возможность регистрации и авторизации, для отслеживания проведения договоров. На главном экране должны быть представлены все главные разделы, в которых можно создавать заявки на поставку, проводить документы, записывать или узнавать информацию и рассчитывать рассрочку.

Работа приложения должна начинаться с окна авторизации, где сотрудник должен указать свой логин и пароль. Так же должна быть кнопка регистрации.

После успешной авторизации открывается основное окно (Рисунок 1) с рабочей областью и разделами. В разделах должны быть реализованы кнопки управления.

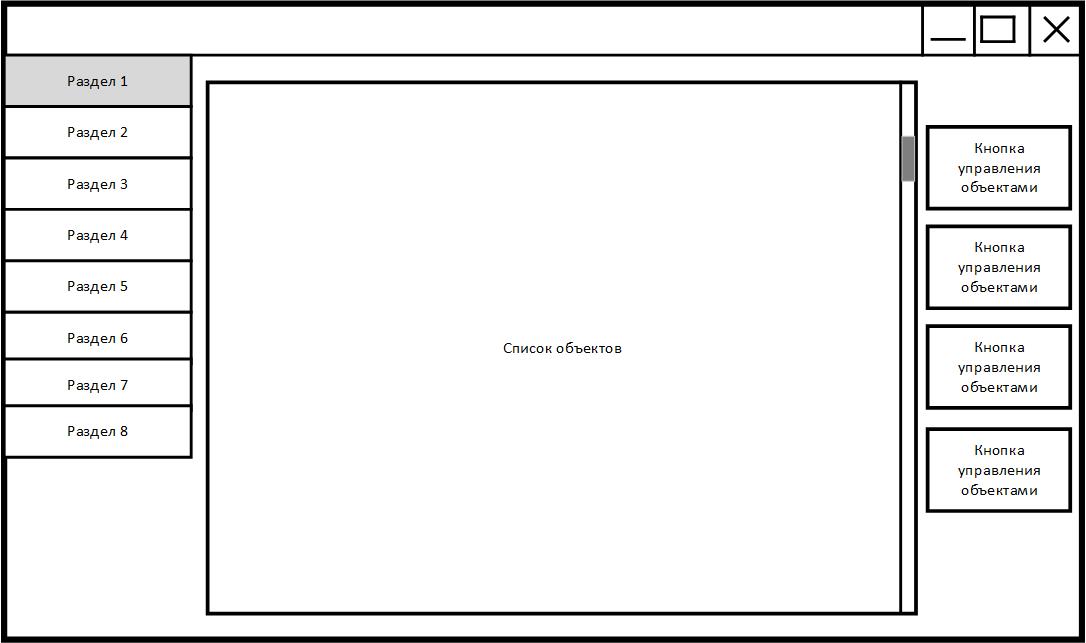
****

Рисунок 2.1 – Макет главного рабочего экрана приложения

**2.2 Описание предметной области.**

Автосалон — это некий аналог магазина, в котором представлены существующие образцы автомобилей. Автосалон осуществляет автомобильную дилерскую логистику, планирование производственного заказа на выпуск автомобилей, посредством официального дилерского договора.

Тема продаж и покупок автомобилей очень популярна в наше время, так как сам предмет – автомобиль пользуется большой популярностью у населения. Для облегчения поиска нового автомобиля или места сбыта старого авто во многих городах организуются специальные стоянки или закрытые магазины, называемые автосалоном, в которых клиенты могут подобрать себе автомобиль по вкусу а также избавиться от старого авто, выставив его на продажу.

Основные функции автосалона:

* осуществляет продажу новых автомобилей;
* совместно с дистрибьютором формирует производственную программу завода-производителя посредством производственного заказа на выпуск автомобилей. Формированием заказа на выпуск автомобилей с учётом предпочтений и вкусов клиентов в дилерском центре занимается диспонент;
* осуществляет предпродажную подготовку автомобиля и его послепродажное и гарантийное обслуживание — прохождение во время гарантийного периода планового технического обслуживания (замена эксплуатационных материалов, техническая диагностика, гарантийный ремонт и так далее);
* осуществляет продажу специальных банковских продуктов по кредитованию и страхованию проданных автомобилей;
* осуществляет продажу оригинальных (фирменных) запасных частей. Для этого на территории демонстрационного зала оборудуется специальное рабочее место специалиста по запасным частям для подбора и заказа нужной детали по каталогу или выдачи со склада дилерского центра.

Основными элементами предметной области являются поставляемые автомобили, которые обозначены следующими атрибутами – комплектация, цвет, vin номер. Также выделены основные составляющие автомобиля – производитель, модель. Данная предметная область содержит информацию о покупателях, сотрудниках, заявках, договорах и оплате клиента

Предполагаемая база данных должна обеспечивать работу автосалона по учету продаж и поставок автомобилей различных видов, база данных будет относиться к классу база данных управления предприятием, т. к. она будет обслуживать конкретный один автосалон, а не систему автосалонов в целом. Приложение упрощает процедуру поиска необходимой информации о товарах и ценах на них. С ее помощью сотрудник может легко узнать информацию о поставленном или проданном товаре.

Приложение упрощает процедуру поиска необходимой информации о товарах и ценах на них. С ее помощью сотрудник может легко узнать информацию о поставленном или проданном товаре, клиент может просматривать существующий в наличии модельный ряд авто, руководитель отслеживать деятельность сотрудников и динамику продаж.

**Описание понятий предметной области:**

* Производитель — автозавод, компания, фирма, занимающаяся разработкой, изготовлением или сборкой автомобилей.
* Модель автомобиля — это название или номер, которые производители используют для идентификации и продажи ряда похожих автомобилей.
* **Комплектация автомобиля — это определенный набор опций, которые завод изготовитель, устанавливает в свои автомобили по умолчанию.**
* Автомобиль - моторное дорожное и внедорожное транспортное средство, используемое для перевозки людей и грузов.
* Аксессуары – это вещи, которые обеспечивают комфорт в автомобиле
* Договор – это документ купли-продажи, который заключается между автосалоном и клиентом.
* Клиент – человек, которому предоставляются услуги автосалона.
* Сотрудник – человек, который работает в автосалоне и имеет доступ к системе.
* Заявки – документ, в котором четко прописаны характеристики товаров, заказываемых автосалоном.
* Журнал событий – таблица с сообщениями которые происходят в системе.

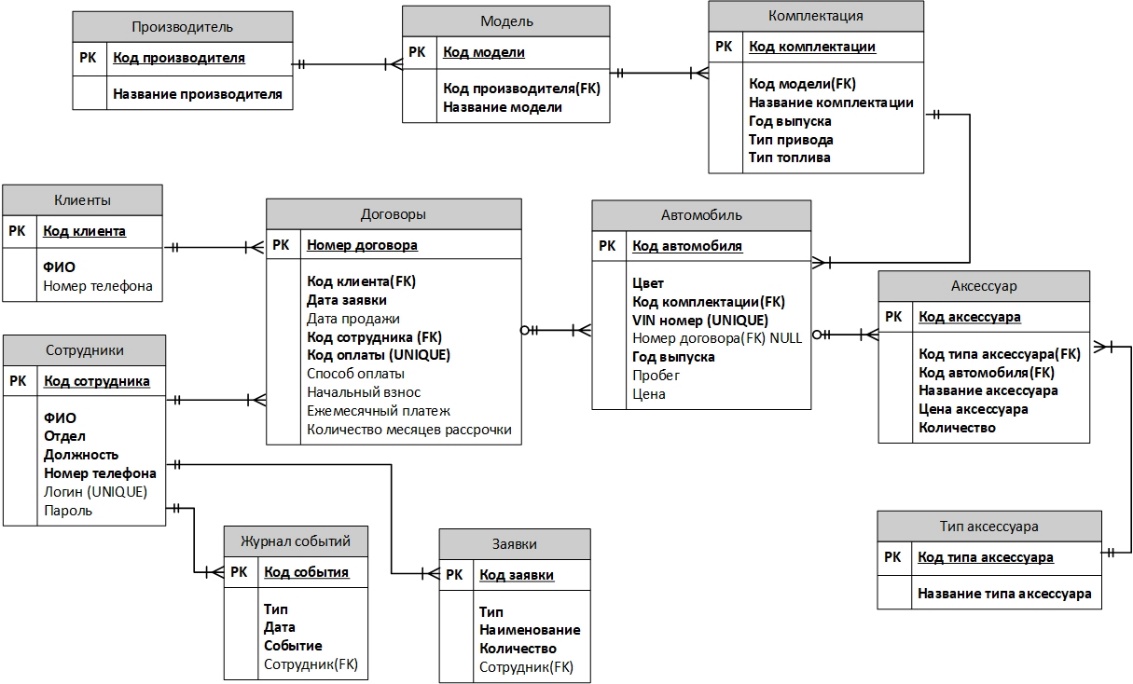


Рисунок 2.2 Логическая схема базы данных

**2.3 Анализ аналогов.**

**«АвтоСалон»**

Программа «АвтоСалон» предназначена для автоматизации деятельности автосалонов, торгующих автомобилями, в части, касающейся ведения учета, продаж автомобилей и формирования договорных документов. Также предусмотрено формирование и печать документов, специфичных для данного вида деятельности (справка-счет, ПТС, транзитный номер).

Возможности системы:

* Печать документов на бланках строгой отчетности (справка-счет, ПТС, транзитный номер);
* Неограниченное количество рабочих мест в рамках одной локальной сети;
* Сколько угодно много предприятий (торгующих организаций);
* Полная мультивалютность;
* Ведение различных справочников (клиенты, автомобили, комплектации и т.д.);
* Ведение Склада и Журнала продаж;
* Формирование различных отчетов и статистики;

**«Альфа-Авто»**

Программный продукт «Альфа-Авто» предназначен для комплексной автоматизации учета на предприятиях автобизнеса (магазинов оптовой и розничной продажи запчастей, автосервисов и станций технического обслуживания автомобилей, автосалонов и дилерских центров).

С помощью «Альфа-Авто» можно:

* оформлять заказы и продажи автомобилей;
* выполнять предпродажную подготовку автомобилей;
* оказывать услуги по ремонту и обслуживанию автомобилей;
* вести оптовую и розничную торговлю запасными частями;
* организовать работу склада;
* учитывать оплаты и отслеживать состояние взаиморасчетов с покупателями и поставщиками;

Конфигурация позволяет:

1. Учет запчастей:

* оптовая продажа запчастей;
* розничная продажа запчастей;
* работа по заказам запчастей;
* закупки запчастей;
* организация внутрифирменного товародвижения;

1. Автосервис:

* планирование ресурсов;
* оформление ремонтов;
* автоматизированное рабочее место сотрудника автосервиса;

1. Автосалон:

* клиентские и складские заказы;
* покупка и продажа автомобилей;
* предпродажная подготовка автомобилей;
* сделки «Trade-in»;
* ответственное хранение автомобилей;

1. Финансовый блок:

* оплаты покупателей и поставщикам;
* ведение взаиморасчетов с подотчетными лицами;
* бюджетирование;

1. Обмен данными:

* обмен данными с бухгалтерскими системами;
* обмен данными с каталогами производителей;

Пользователи программы имеют возможность быстро формировать необходимые документы. Руководство может оперативно получать и использовать данные о различных аспектах деятельности компании. Система предоставляет информацию, необходимую для принятия управленческих решений.

**2.4 Перечень задач, подлежащих решению в процессе разработки.**

1. Определить состав и структуру данных;
2. разработка пользовательского интерфейса;
3. Определить архитектуру приложения;
4. разработка и реализация базы данных;
5. разработка функционала автоматизированной информационной системы;
6. реализация программного обеспечения системы;
7. Тестирование и проверка разработанной автоматизированной информационной системы и базы данных.
8. Исправление выявленных ошибок.

**2.5 Обоснование выбора инструментов и платформы для разработки.**

Для выполнения курсовой работы были выбраны:

1. Операционная система Windows 10.

Данная операционная система является одной из самых распространенных на данный момент, она обладает так же рядом существенных преимуществ. Одним из критериев выбора Windows 10 является то, что она установлена на многих предприятиях

Это единый (для всех программ-приложений Windows), стандартизированный и интуитивно-понятный интерфейс взаимодействия с пользователем. Т.е., даже начинающему пользователю персонального компьютера, не требуется особых усилий для освоения навыков работы с данным программным средством.

Причины выбора Операционной системы Windows 10:

1. Простота эксплуатации. Даже неподготовленному человеку легко разобраться в принципах работы операционной системы, ведь все интуитивно понятно, также система не требует знаний в программировании.

2. Большое количество программ сторонних компаний выпускают именно под данную операционную систему. Есть как платные, так и бесплатные версии.

3. Большая часть выпускаемого оборудования для ПК выпускается с поддержкой ОС Windows. Веб-камеры, сканеры, принтеры, игровые манипуляторы и так далее. Все драйвера, в первую очередь, выходят под данную ОС, и лишь по прошествии большого промежутка времени выходят под другие операционные системы.

4. Большая распространённость. Данная операционная система, по состоянию на апрель 2020 года, заняла второе место в общей мировой статистике использования ОС.

5. Различные формы и виды оформления. Кроме стандартных вариантов ОС позволяет сторонним программам вносить изменения во внешний вид рабочего стола, папок, заставок и так далее.

2. Язык программирования С#

В настоящее время C # и C ++ чаще всего используются для создания коммерческих и бизнес-приложений. Эти языки подходят для большинства программистов. Но на самом деле развитие производительности не было правильным. Например, процесс создания приложения в C ++ часто занимает больше времени, чем применение эквивалента в Visual Basic. Существует язык, который в настоящее время разрабатывается из-за потери привычной эластичности и необходимости программистов на C # / C ++. Эти решения не очень удобны для программистов и предлагают значительно меньше возможностей. Они не ориентированы на текущее взаимодействие с системой и часто несовместимы с существующими методами разработки для Интернета. Большинство разработчиков хотят использовать современные языки, которые позволяют писать, читать и разрабатывать программы с простотой Visual Basic, обеспечивая при этом мощь и гибкость C ++ для доступа ко всем функциям системы. Взаимодействовать с существующими стандартными веб-приложениями

Основные преимущества языка:

1. По умолчанию C # запрещает прямое управление памятью, возвращая различные типы систем и сборку мусора. Работа напрямую с памятью все еще существует в специальном режиме кода «опасности», но это должно быть четко объявлено. По этой причине в C # активно используется только один поставщик доступа «.».
2. В C # нет методов множественного наследования. Но вместо этого предлагается использовать создания нескольких интерфейсов. Хотя многие мнения о наследовании разные. Но отсутствие этого механизма в C # должно, по крайней мере, облегчить разработку компилятора.
3. Система типов C # представляет систему типов. NET как большинство современных современных языков. Важной особенностью системы этого типа является четкое разделение всех категорий на типы значений и ссылочные типы.
4. В C # можно организовать данные в структуру, состоящую из переменных любого типа или распределения, состоящего из множества переменных одного типа. Важной особенностью распределения в C # является необходимость явного преобразования в базовый тип, если вы хотите интерпретировать значения из перечисления в числа

Кроме того, следующие языковые механизмы могут быть отнесены к особенностям языка:

1. Исключения
2. Встроенный механизм контроля версий (Цель этого механизма - гарантировать, что пользователям придется изменять или компилировать старые библиотеки только тогда, когда это абсолютно необходимо.)
3. Возможность отключить контроль переполнения (Ключевые слова, которые не проверены)

Исходя их этого, C # является ориентированным на программирование для платформы. NET и используется как для прототипирования программ и для разработки крупномасштабных приложений

C # поддерживает .NET Framework 4, что позволяет использовать опыт разработки приложений для устройств.

Исходя из всего вышеперечисленного, можно прийти к выводу, что язык программирования C#, несомненно, подходит для разработки автоматизированной информационной системы.

3. Объектно-реляционная система управления базами данных PostgreSQL

Из характеристик СУБД, которые могут определить выбор, одной из важнейших является модель данных. Теоретически любую информацию можно представить в виде реляционной модели. Сила реляционных баз данных в том, что эта модель очень хорошо подходит для предприятий, которые располагают немалыми средствами для активного внедрения передовых систем. Эта модель имеет наиболее проработанное математическое основание и хорошо проработанные стандарты. Реляционная модель данных отличается большой гибкостью с точки зрения изменения структуры данных. Здесь можно менять физическую структуру данных, не переписывая приложения.

Выбор СУБД напрямую связан с анализом и выбором аппаратных ресурсов. Стандартные тесты проводятся на современных быстродействующих системах, в то время как целью клиента может быть просто работоспособная система в рамках отведенного бюджета. Не менее важны и взаимодействие с унаследованными системами, и возможности переноса накопленных данных на новую систему.

Важным критерием выбора с точки зрения перспектив становится наличие эффективных средств разработки. Такие средства, обладающие удобным интерфейсом, позволяют специалистам предприятия самостоятельно и быстро настраивать информационные системы в соответствии с требованиями бизнеса.

PostgreSQL - это мощная объектно-реляционная система управления базами данных с открытыми исходными текстами. Она разрабатывается на протяжении более 15 лет и улучшает архитектуру, чем завоевала репутацию надежной, интегрированной и масштабируемой СУБД. Она запускается на всех основных платформах, включая Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), и Windows. Она полностью соответствует ACID, имеет полную поддержку ключей, объединений, представлений, триггеров, и хранимых процедур (на разных языках). Она включает большинство типов данных SQL92 и SQL99, включая integer, numeric, boolean, char, varchar, date, interval, и timestamp. Она также поддерживает хранение больших двоичных объектов (BLOB's), включая картинки, звук, или видео. Она имеет API для C# C/C++, Java, Perl, Python, Ruby, Tcl, ODBC и др.

Являясь СУБД класса предприятия, PostgreSQL предоставляет такие особенности как Multi-Version Concurrency Control (MVCC), восстановление по точке во времени, табличное пространство, асинхронная репликация, вложенные транзакции (точки сохранения), горячее резервирование, планировщик/оптимизатор запросов, и упреждающее журналирование на случай поломки. Он поддерживает международные кодировки, в том числе и многобайтовые, при использовании различных кодировок можно использовать сортировку и полнотекстовый поиск, различать регистр. Большое количество подконтрольных данных и большое число одновременно работающих пользователей, тем не менее, не сильно влияет на масштабируемость системы. Есть действующие PostgreSQL системы, которые управляют более чем 4 терабайтами данных.

Средства обеспечения целостности данных включают составные первичные ключи, внешние ключи с поддержкой запрета и каскадирования изменений/удалений, проверку ограничений (constraints), ограничения уникальности и ограничения на непустые значения.

PostgreSQL может выполнять хранимые процедуры, написанные на различных языках программирования, включая Java, Perl, Python, Ruby, Tcl, C/C++, и собственном PL/pgSQL, аналогичном Oracle's PL/SQL. В стандартную библиотеку функций включены сотни встроенных функций — от базовых математических и строковых операций до криптографических функций и функций, обеспечивающих совместимость с Oracle. Триггеры и хранимые процедуры могут быть написаны на Cи и загружены в базу данных в качестве библиотеки, позволяя тем самым расширять ее возможности. Также PostgreSQL включает средства разработки, позволяющие создавать пользовательские типы данных вместе с функциями и операторами, описывающими их поведение. Как результат, созданы и могут быть добавлены к системе различные типы данных — от геометрических и пространственных примитивов до типов данных, определенных в ISBN/ISSN (International Standard Book Number/International Standard Serial Number).

3. Среда разработки Microsoft Visual Studio 2019

Microsoft Visual Studio 2019— это набор инструментов разработки, основанных на использовании компонентов, и других технологий для создания мощных, производительных приложений. Кроме того, среда Visual Studio оптимизирована для совместного проектирования, разработки и развертывания корпоративных решений. Visual Studio предоставляет средства для проектирования, разработки b отладки.

Среда разработки Visual Studio представляет собой полный набор средств разработки для создания веб-приложений ASP.NET, XML (веб-службы), настольных приложений и мобильных приложений. Visual Basic, Visual C# и Visual C++ используют единую интегрированную среду разработки (IDE), которая позволяет совместно использовать средства и упрощает создание решений на базе нескольких языков. Можно создать обычные приложения Microsoft Windows и приложения с архитектурой "клиент-сервер" с помощью использования конструкторов в Visual Studio.

Почему именно эта версия продукта, потому что в нем есть такие удобные нововведения как:

1. Visual Studio IntelliCode повышает эффективность разработки программного обеспечения с помощью искусственного интеллекта (ИИ). Для создания рекомендаций IntelliCode анализирует 2000 проектов с открытым кодом на GitHub
2. Рефакторинг В C# есть много новых удобных возможностей рефакторинга, которые помогают упорядочить код. Они отображаются как предложения со значком лампочки и включают такие действия, как перемещение элементов в интерфейс или базовый класс, настройку пространств имен в соответствии со структурой папок.