Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

Специальность 09.02.07 Информационные системы и программирование

**ОТЧЕТ**

**ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

**(преддипломной)**

Выполнил:

обучающийся учебной группы № 4ИСИП-519

--Алиев Нурлан Фарман оглы

*(И.О. Фамилия)*

Проверил:

руководитель практики от колледжа:

*(И.О. Фамилия)*

**Москва**

**20­­­­23**

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](file:///C:\Users\ADMIN\Downloads\diplom.docx#_Toc134730631)

[ГЛАВА 1 ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ 6](file:///C:\Users\ADMIN\Downloads\diplom.docx#_Toc134730632)

[1.1 Описание предметной области 6](file:///C:\Users\ADMIN\Downloads\diplom.docx#_Toc134730633)

[1.2 Характеристика инструменталльных средств 8](file:///C:\Users\ADMIN\Downloads\diplom.docx#_Toc134730634)

[1.3 Анализ применяемых магазинами автозапчастей информационных систем для поставок и продаж 13](file:///C:\Users\ADMIN\Downloads\diplom.docx#_Toc134730635)

[ГЛАВА 2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ 19](file:///C:\Users\ADMIN\Downloads\diplom.docx#_Toc134730637)

[2.1 Анализ требований и разработка спецификаций 19](file:///C:\Users\ADMIN\Downloads\diplom.docx#_Toc134730638)

[ЗАКЛЮЧЕНИЕ 24](file:///C:\Users\ADMIN\Downloads\diplom.docx#_Toc134730641)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 26](file:///C:\Users\ADMIN\Downloads\diplom.docx#_Toc134730642)

# ВВЕДЕНИЕ

Создание автоматизированных информационных систем позволяет эффективно обрабатывать большие потоки информации, которые имеют определенную структуру, зависящую от особенностей места применения. Это позволяет осуществлять индивидуальный подход к автоматизации именно тех функций, которые выполняются автоматизируемым подразделением. Внедрение на предприятии автоматизированных систем управления позволяет значительно сократить время выполнения работ, повысить их точность и облегчить труд специалистов.

В данном проекте будет разработана информационной система по автомобильных запчастей автоваза. Эффективность деятельности сотрудников будет увеличена за счет автоматизированного учета и обработки данных о поступлении и отгрузке запчастей, автоматическом формировании счет-фактуры и товарно-транспортной накладной, построения отчетности за любой промежуток времени.

Актуальность темы проекта обусловлена необходимостью создания нового программного продукта, которое позволит создать универсальный инструмент по автоматизации учета поставок и отпуска запчастей или любых других товаров для предприятий малого и среднего масштаба. Система позволит создать централизованное хранилище данных для учета информации о запчастях, реализовать возможность оперативного получения и поиска информации, формировать отчетность для принятия обоснованных управленческих решений руководством организации.

Отличительной особенностью системы, которая будет разработана в настоящем проекте, является возможность учета как поступления, так и отгрузки запчастей клиентам с одновременным формированием сопроводительных документов. Система разрабатывается для работы в локальной сети, но при необходимости может быть использована и в качестве однопользовательского приложения, так как зачастую склад не подключен с локальной сети предприятия и размещение базы данных на сервере невозможна. Кроме того, разработка и внедрение системы автоматизации учета поставок и отпуска запчастей позволит максимально устранить случайные ошибки, возникающие при ручном вносе данных, минимизировать недостатки существующей системы работы компании по учету запчастей, и привнести в нее факторы, положительно влияющие на качество и сроки выполнения реализуемых в ней функций: уменьшение времени выполнения каждой функции; автоматическое создание документации и отчетов; простой и быстрый поиск.

Целью проекта является разработка информационной системы по учету запчастей.

Задачи, которые необходимо решить для достижения поставленной цели:

* изучить предметную область.
* исследовать бизнес-процессы по учету запчастей на предприятии.
* разработать требования к разрабатываемой системе.
* выбрать среду разработки и систему управления базами данных для
* разрабатываемой системы.
* разработать логическую и физическую модели данных.
* создать базу данных и визуальный интерфейс информационной системы.
* предусмотреть защиту информации и разделение прав доступа в информационной системе.

Для достижения поставленных задач будут применяться следующие

инструментальные средства, языки проектирования и разработки

программного обеспечения:

* Система управления реляционными базами данных Microsoft SQL Server;
* Компактная встраиваемая СУБД SQL;
* Сервис для проектирования диаграмм Draw.io;
* Текстовый редактор Microsoft Word;
* Программа подготовки и просмотра презентаций Microsoft PowerPoint.

Объектом исследования являются информационная система автомобильных запчастей автоваза.

**ГЛАВА 1 ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ**

## Описание предметной области

Информационные системы. Классификация информационных систем Первые информационные системы появились в 50-х гг. В эти годы они были предназначены для обработки счетов и расчета зарплаты, а реализовывались на электромеханических бухгалтерских счетных машинах. Это приводило к некоторому сокращению затрат и времени на подготовку бумажных документов. -с гг. знаменуются изменением отношения к информационным системам. Информация полученная из них, стала применяться для периодической отчетности по многим параметрам. Для этого организациям требовалось компьютерное оборудование широкого назначения, способное обслуживать множество функций, а не только считать зарплату, как было ранее. В 70-х - начале 80-х гг. информационные системы начинают широко использоваться в качестве средства управленческого контроля, поддерживающего и ускоряющего процесс принятия решений). Тoгдa пoявилacь кoнцeпция cиcтeм пoддepжки пpинятия peшeний (decision support systems - DDS). Эти cиcтeмы должны были oбecпeчить мeнeджepoв спeциaлизиpoвaннoй и интepaктивнoй пoддepжкoй пpoцeccoв принятия yникaльныx peшeний пpoблeм в peaльнoм, быcтpoизмeняющeмcя миpe. К концу 80-х гг. концепция использования информационных систем вновь изменяется. Развитие мощности (быстродействия) микро-ЭВМ, пакетов прикладных программ и телекоммуникационных ceteй далo тoлчoк к пoявлeнию фeнoмeнa кoнeчнoгo пoльзoвaтeля (end user computing). Они становятся стратегическим источником информации и используются на всех уровнях организации любого профиля. Информационные системы этого периода, предоставляя вовремя нужную информацию, помогают организации достичь успеха в своей деятельности, создавать новые товары и услуги, находить новые рынки сбыта, обеспечивать себе достойных партнеров, организовывать выпуск продукции по низкой цене и многое другое. Инфopмaциoнныe cиcтeмы, пpeднaзнaчeнныe для oбecпeчeния мeнeджepoв инфopмaциeй для пoддepжки принятия эффeктивныx peшeний, нaзывaютcя yпpaвлeнчecкими инфopмaциoнными cиcтeмaми (management information systems - MIS). Информационная система является средой, составляющими элементами которой являются компьютеры, компьютерные сети, программные продукты, базы данных, люди, различного рода технические и программные средства связи и т.д. Информационная система - это организационно-упорядоченная взаимосвязанная совокупность средств, и методов ИТ, а также используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели. Такое понимание информационной системы предполагает использование в качестве основного технического средства переработки информации ЭВМ и средств связи, реализующих информационные процессы и выдачу информации, необходимой в процессе принятия решений задач из любой области. Классификация информационных систем:

Классификация информационных систем по признаку структурированности задач;

Для структурированных задач;

Для частично структурированных или неструктурированных задач:

Создающие управленческие отчеты;

Разрабатывающие альтернативные решения (модельные и экспертные). Классификация информационных систем по функциональному признаку и уровням управления;

производственные системы;

   системы маркетинга;

финансовые и учетные системы;

системы кадров (человеческих ресурсов);

прочие типы, выполняющие вспомогательные функции в зависимости от специфики деятельности фирмы.

Классификация по степени автоматизации:

ручные;

автоматические;

автоматизированные;

Классификация по характеру использования информации:

Информационно-поисковые системы;

Информационно-решающие системы;

  Управляющие ИС;

Советующие ИС;

Классификация по сфере применения:

Информационные системы организационного управления;

ИС управления технологическими процессами (ТП);

  ИС автоматизированного проектирования (САПР);

* 1. Характеристика инструментальных средств

Процессу моделирования баз данных в сфере информационных технологий долго не придавали особого значения. Считалось, что если необходимо протестировать или уточнить код приложения, вполне достаточно просто перебросить данные в любую базу данных. Это мнение глубоко ошибочно. Правильно построенные модели данных не только способствуют повышению производительности приложений за счет более быстрого извлечения и записи данных, но и обеспечивают маштабируемость и гибкость дальнейшей разработки. В наши дни даже секунды решают многое, поэтому пользователи, которым приходится терять время в ожидании ответа от медленных приложений, стремятся найти альтернативные решения. Если вы хотите, чтобы ваши приложения были лучшими в своем роде, придется выделить время и ресурсы на создание моделей данных. В этом техническом описании мы расскажем, что такое моделированием данных, почему оно имеет такое значение, и какие концепции и методы лежат в основе моделирования данных. Базы данных играют очень важную роль в процессе разработки приложений. Большинству приложений необходимо хранить определенные данные для дальнейшего использования. Когда дело доходит до сохранения этих данных, в большинстве случаев выбор самым логичным образом падает на реляционные базы данных. Существуют и другие варианты - плоские файлы, XML, персистентные наборы записей или даже пользовательские форматы файлов, но, ни один из этих вариантов не обладает надежностью и защищенностью базы данных. Базы данных позволяют анализировать собранную информацию при помощи таких инструментов, как отчеты и хранилища данных. Кроме того, большинство реляционных систем управления базами данных (РСУБД) позволяют восстановить базу данных из резервной копии в случае отказа и предлагают способы реализации решений высокой готовности для минимизации времени простоя вследствие отказа. Даже сами по себе эти преимущества могут подтолкнуть к решению об использовании баз данных в качестве решения для хранения и управления информацией при разработке приложений. На самом деле, моделирование баз данных не отличается сложностью - это процесс отображения реальной информации на логическое представление этих данных. Другими словами, как информация о клиентах хранится в модели данных? Здесь многое зависит от специалиста, создающего модель, но есть ряд ключевых концепций, которые помогут такому специалисту принимать правильные решения в процессе разработки. Итак, при моделировании нужно воспринимать данные с точки зрения логики, не заботясь о том, как именно будут выглядеть в базе данных таблицы и столбцы. Такой подход называется логическим моделированием, и его единственная задача - создать модель, представляющую реальные объекты. После создания логической модели можно перейти к проектированию физической модели данных. Разделение логической и физической модели гарантирует создание надежной базы данных. В процессе разработки приложения в самой ранней фазе проекта проектировщики и разработчики обычно тратят много времени на сбор требований к создаваемому приложению. Эту информацию обычно собирают при помощи интервьюирования пользователей и руководителей, а также путем наблюдения за существующей системой (даже если это совершенно неавтоматизированная система). Конечный результат представляет собой довольно детализированный набор требований, включающий прецеденты, диаграммы системы и макеты интерфейса приложения. После того, как требования всех основных заинтересованных в проекте сторон будут удовлетворены, начинается разработка приложения. Одно из первых действий, которое приходится выполнять большинству разработчиков приложений - это создание репозитория для данных, с которыми будет работать новое приложение, другими словами - базы данных приложения. Во многих случаях разрабатывается база данных, которая будет удовлетворять физическим требованиям интерфейса приложения. То есть, при наличии проекта, описывающего способ получения и отображения данных приложением, разработчики приложения могут создать в РСУБД пустую базу данных. Если имеется таблица или набор таблиц, которые полностью определяют все данные, получаемые от интерфейса, то разработчик приложения может начать разработку механизма хранения данных и создание кода, при помощи которого приложение будет взаимодействовать с только что созданной базой данных. Хотя этот подход далек от совершенства, он может работать, особенно для краткосрочных проектов. Если все будет сделано с достаточной тщательностью, то готовая база данных даже может функционировать, при условии небольшого объема хранящихся в ней данных и отсутствия необходимости в изменениях. Однако разработанные этим способом базы данных неизбежно будут испытывать серьезные проблемы с масштабируемостью, и по прошествии некоторого времени их будет чрезвычайно трудно изменить. Известно, что приложения, в конце концов, приходится изменять и даже полностью переписывать, чтобы добавить в них новые (или убрать устаревшие) функции; но при этом обычно нужно сохранить устаревшие данные. Значит, мы приходим к необходимости частичного перепроектирования "устаревшей" базы данных для добавления новых функций без риска потери каких-либо данных. Это ведет к снижению производительности, поскольку зачастую мы добавляем новые структуры базы данных (таблицы и представления), а также сложную логику SQL для представления устаревших и новых данных в рамках все того же интерфейса. Создание эффективной логической модели данных до выполнения любых реальных действий по разработке базы данных поможет предотвратить описанные проблемы и выяснить любые неучтенные требования на уровне данных до того, как будет написана хотя бы строка программного кода. Модели данных создаются на ранних стадиях фазы проектирования, обычно ближе к завершению фазы сбора требований данного проекта. После завершения всех интервью и наблюдений создается модель данных, предназначенная для документирования данных, которыми будет управлять приложение. Эта модель будет логически представлять все порции информации, необходимые приложению, и описывать, как различные порции данных связаны с другими порциями данных. Модель также предусматривает представление данных для пользователей, не обладающих достаточными техническими знаниями, и может быть очень полезна для получения одобрения проекта приложения в целом конечными пользователями. Строго говоря, реляционная база данных представляет собой упорядоченный набор таблиц, в которых хранятся данные. На практике база данных представляет собой набор таблиц, представлений и хранимых процедур (в зависимости от конкретной СУБД), которые хранят данные и выполняют с ними различные действия. Эти структуры определены при помощи встроенного языка программирования данной РСУБД, который обычно представляет собой один из диалектов языка SQL. Любая РСУБД хранит данные в файлах операционной системы и оснащена функциями по управлению файлами, управлению безопасностью и отладке производительности запросов, которые используются для манипуляций с данными. С точки зрения приложения, база данных - это место, откуда поступают данные. В отличие от базы данных, модель данных не является представлением физического хранилища данных. Если база данных определяет способ хранения данных, способ использования реальных отношений между ними для манипулирования данными и обеспечивает программный доступ к данным, то модель данных просто перечисляет, какие данные существуют и как различные биты информации связаны между собой. Хорошо спроектированная модель данных, в конце концов, превращается в логическую схему разрабатываемой базы данных. По этой причине модели данных обязательно должны быть платформенно-независимыми; и любая модель данных может использоваться для создания физической базы данных в Oracle 10g, SQL Server 2005 или MySQL. И все же не следует думать, что при моделировании не нужно учитывать, с какой РСУБД будет работать приложение. В ряде ситуаций предварительная осведомленность о РСУБД, которая будет использоваться для управления базой данных, может повлиять на процесс моделирования данных. Эффективное моделирование данных обеспечивает высокую производительность работы РСУБД, Во-первых, выполнение стандартных правил моделирования данных поможет вам устранить алогичности данных, например, их дублирование, что в конечном итоге поможет избежать необходимости встраивания в приложение дополнительной логики для обработки этих алогичностей. Кроме того, при хранении данных в структурированном формате ядро запросов может найти и извлечь данные быстрее, чем в том случае, если они хранятся в плоском файле или являются плохо структурированными. Это обусловливает более высокую производительность вашего приложения или отчетов.

* 1. Анализ применяемых магазинами автозапчастей информационных систем для учёта поставок и продаж.

Информационные системы для отдела продаж – это большое подспорье в деле аналитики. Автоматизация процесса сбора данных избавляет специалистов по продажам от множества рутинных обязанностей, дает возможность сформировать информационную базу, сделать выводы о том, какие позиции пользуются наибольшим покупательским спросом. Система поможет оперативно найти «слабые» моменты в работе организации и исправить их [14]. Широко известными на рынке информационных систем являются следующие поставщики ИС: − Intrum; − Битрикс24; − Terrasoft Creatio; − Amo; 11 − Mango; − Мой склад; − ClientBase; − Salesap; − YCLIENTS; − Envybox. Далее рассмотрим более детально каждую из них. Intrum представляет следующие возможности: − Автоматизация ваших бизнес процессов. Все этапы работы с клиентом идут по заранее выстроенной четкой схеме. Эталонные образцы помогут быстро запустить бизнес процессы, а с помощью гибкой настройки можно адаптировать их под свой бизнес. − Учет продаж и воронка продаж. Ведение продаж по стадиям. − IP-телефония. Подключение к системе любого вашего провайдера телефонии. Настройка сценария обработки вызовов и скриптов продаж прямо в ИС. − Интеграция ИС и сайта. Онлайн консультант для общения с клиентом. ИС интегрирована со многими социальными сетями, что даёт возможность автоматизации онлайн-рекламы через бизнес-процессы. − Конструктор отчетов. Создание отчетов по метрикам эффективности бизнеса. Настройка их вывода на рабочий стол сотрудников индивидуально для каждого, в зависимости от должности и выполняемых ими задач в компании. − Автоматическая генерация документов и документооборот. Создание документов в два клика из карточки клиента или сделки на основе бланков. Можно гибко настраивать бланки шаблонов документов, в т. ч. оформлять их в фирменном стиле вашей компании. Битрикс24 входит в перечень лучших систем для малого бизнеса. Она оснащена необходимым инструментарием: 12 − «облачным» хранилищем «Битрикс Диск» с возможностью хранения документов, ограничения доступа к папкам, поиска файлов по наименованию и синхронизации с рабочим ПК; − специальным почтовым сервисом, интегрируемым с корпоративной почтой; − шаблонами для построения сайтов; − таблицами для учета сделок, контактов, анализа графика продаж; − телефонией. В сравнении с системами другого образца данная программа настроена на оптимизацию рабочего процесса внутри компании, учет уровня продаж. Преимущества Terrasoft Creatio: − удобный интерфейс; − предоставление клиенту всех данных о товаре, помощь в оформлении заказа и послепродажном обслуживании; − построение графиков продаж при помощи дэшбордов. Amo — удобная коммуникационная система способна автоматически формировать запросы для отправки уведомлений клиентам по электронной почте, в чате, а также через форму поддержки на сайте и любой мессенджер. Специальные опции помогают не только донести предложение до клиента, но и увидеть его реакцию. Софт подходит для контакт-центров, торговых организаций. Mango — система для продаж, интегрированная с телефонией. Звонки, таблицы и чаты открываются в 1 окне. Все данные хранятся в «облаке», команды перераспределяются между членами коллектива. Руководители имеют возможность контролировать деятельность подразделений компании. Дополнительные функции — построение графика продаж, составление отчетов и их анализ. Круглосуточно работает техническая поддержка. Мой склад — это система, совмещающая в себе функции ведения учета продаж, закупок, контроля финансов предприятия, товаров на складе, базы 13 данных поставщиков и покупателей. Сферы применения — оптовые и розничные торговые организации, интернет-магазины. Софт интегрирован с 1C. Список опций: − создание карточек товаров; − формирование уведомлений о текущих акциях, проводимых бонусных программах; − ведение статистики продаж; − инструменты для распечатки документов — накладных, чеков. В ClientBase содержится только все самое необходимое для организации малого бизнеса. Установка и обслуживание проводятся без участия программистов, что позволяет снизить издержки. Список основных функций: − бизнес-конструктор для построения таблиц, решающих те или иные задачи; − улучшенная IP-телефония; − сервис рассылок; − шаблон для формирования базы данных. Salesap — автоматизированная система, созданная для организации деловых процессов, ведения финансового учета, складирования, кассы. Прирост прибыли происходит за счет таких инструментов для ведения бизнеса: − аналитика; − средства для ускоренной обработки заявок; − модуль для распределения звонков между сотрудниками. Благодаря специальной опции можно проконтролировать состояние воронки продаж. Софт подходит для мелких торговых компаний, контактцентров, агентств недвижимости, салонов красоты и стоматологий. YCLIENTS — совместное использование склада, клиентской базы, аналитических таблиц и графиков, касс, системы уведомлений, учета финансов и отработки лояльности. Программа оснащена телефонией, которую легко 14 установить, не обладая специальными знаниями. Она находит свое применение в салонах красоты, медицинских учреждениях, розничных и юридических компаниях, фитнес-клубах, обучающих центрах. Envybox — программа помогает организовать обратные звонки для приема большего количества вызовов в течение рабочего дня, высылать приглашения клиентам в режиме онлайн, персонализировать обращения специалистов в чате, генерировать клиентскую базу и автоматически публиковать на сайте уведомления о проходящих акциях. Envybox внедряется мгновенно, работать с ней могут менеджеры без специальных знаний. Рассмотрим далее CRM для отделов продаж. CRM-система для отдела продаж — это программа для отдела продаж, которая помогает управлять отношениями с клиентами, следить за сделками, за работой менеджеров и не только. Она решает множество проблем, с которыми сталкиваются отделы продаж [41]. Приведём небольшой список основных проблем: − Ошибки из-за человеческого фактора; − Потеря клиентов и отсутствие повторных продаж; − Отсутствие продаж при полной загрузке менеджеров; − Невозможно отследить, почему нет продаж и уходят клиенты; − Слабая аналитика не позволяет следить за всеми бизнес-процессами. Рассмотрим популярные CRM-системы для отдела продаж: − Простой бизнес; − Мегаплан; − Pipedrive; − RetailCRM. Рассмотрим более детально известные CRM-ситемы. Простой бизнес Возможности сервиса: Учет клиентов — ведение клиентской базы, воронка продаж, история сделок, напоминания. 15 Управление проектами — планировщик задач, калькулятор, диаграмма Ганта, проектные файлы. Управление персоналом — учет рабочего времени, внутренняя телефония, права доступа, база знаний, работа с удаленными сотрудниками. Документооборот — шаблоны документов, электронная подпись, сканирование, отправка по факсу и email. Склад и бухгалтерия — учет доходов и расходов, электронная отчетность, управление складом. Встроенные сервисы — IP-телефония, видеоконференции, чат, рассылки. «Мегаплан» — простая и легкая в применении CRM программа с удобным интерфейсом и гибкими настройками. За каждым клиентом закрепляется персональный менеджер, для оперативного решения возникающих вопросов работает бесплатная техподдержка [41]. Есть интеграция с 1С. Основные возможности программы: − Активатор продаж; − Карточки клиента; − Диаграмма Ганта; − Воронка продаж и список задач; − Интеграция с IP-телефонией; − Объёмный архив для хранения данных; − Историю операций от заключения договора до завершения сделки; − Интуитивную схему управления проектами. В «Мегаплане» есть конструктор CRM, позволяющий настроить систему под нужды бизнеса — можно отключить лишние функции и добавить нужные. Рабочий стол «Мегаплана» позволяет контролировать ход дел и следить за эффективностью сотрудников. Pipedrive — мощная зарубежная CRM для малого бизнеса. Работает с 2010 года, охват — 140 стран, поддерживает 12 языков. Основной акцент здесь делается на контроле продаж и простоте работы с системой. Все сделки отслеживаются в визуальной воронке продаж, которая легко настраивается под 16 нужды клиента. Воронка представляет собой мини-карточки, которые можно перетаскивать в зависимости от того, на каком этапе находится сделка. Преимущества: − Тестовый период 14 дней; − Интуитивно понятный интерфейс; − Отлично подходит для небольших компаний с фокусом на отдел продаж. − Недостатки: − Не такой большой функционал, как у некоторых других конкурентов; − В справочной службе нет инструкций на русском; − Сам сервис и компания иностранная и оплата только в долларах; − Некоторые функции подходят больше на американский рынок. RetailCRM подходит для интернет-магазинов. Функции: − обработка заказов; − отправка предложений клиентам для их удержания и возврата; − организация продаж через социальные сети и мессенджеры; − автоматическое продвижение товаров; − обработка и оформление заказов в едином окне; хранение клиентских данных в «облаке».

Подводя итог по рассмотрению компаний, предлагающих автоматизированные информационные системы для оптимизации процесса продажи, занесем данные в сравнительную таблицу 1.1 (см.на следующей странице).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Компания | Интеграция с 1С | Количество пользователей | Доступность | Наличие документов для печати | Тестовый период и стоимость |
| Intrum | Да | Нет ограничений | Менеджерам | Да | 14 дней |
| Битрикс24 | Да | Нет ограничений | Клиентам, руководителям, менеджерам | Нет | 30 дней |
| Amo | Да | От 1 | Менеджерам, клиентам | Да | 14 дней |
| Mango | Нет | Любое | Всем | Нет | 14 дней |
| Мой склад | Да | Бесплатно - для 1 человека | Руководителям, менеджерам | Нет | 14 дней |
| ClientBase | Да | От 1 | Всем сотрудникам | Нет | 14 дней |
| Sale Sap | Нет | Любое | Всем | Да | 14 дней |
| YCLIENTS | Нет | Любое | Менеджерам | Да | 7 дней |
| Envybox | Да | Любое | Всем | Да | 7 дней |
| Sales Creatio | Да | От 1 | Клиентам, сотрудникам | Нет | 14 дней |

# 

# ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ

## 2.1 Анализ требований и определение спецификаций ПО

Спецификации данного проекта определяются следующими диаграммами из списка:

* диаграмма состояний;
* диаграмма «Сущность-связь» (ER-диаграмма);
* диаграмма классов;
* диаграмма компонентов;

На рисунке 1 представлена диаграмма «сущность-связь» дипломного проекта по теме «Разработка информационной системы автомобильных запчастей автоваза.», построенная с использованием Microsoft SQL Server Management Studio.

На рисунке 1 представлена диаграмма состояний для разработки программного обеспечения предметной области «Разработка информационной системы автомобильных запчастей автоваза».

Изображение выглядит как диаграмма

Автоматически созданное описание  
Рисунок 1. Диаграмма состояний

На рисунке 2 представлена диаграмма «сущность-связь» дипломного

проекта по теме «Разработка информационной системы автомобильных запчастей автоваза», построенная с использованием веб-сервиса dbdiagram.io.

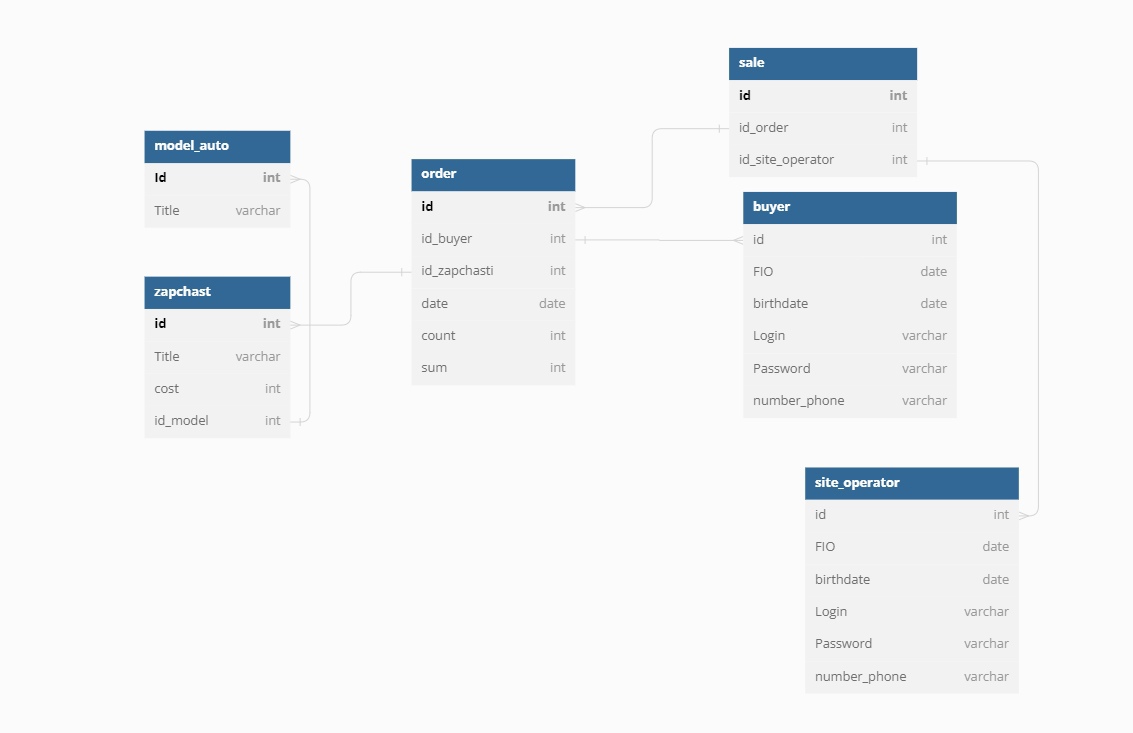


Рисунок 2. Диаграмма «сущность-связь»

На рисунке 3 представлена диаграмма классов дипломного проекта по

теме «Разработка информационной системы автомобильных запчастей автоваза».

## Изображение выглядит как диаграмма Автоматически созданное описание

## Рисунок 3. Диаграмма классов

Ниже представлен код для построения диаграммы классов.

@startuml

class model\_auto {

+ int Id

+varchar Title

+ void просмотр model\_auto()

}

class buyer {

+ int Id

+ varchar FIO

+ date birthdate

+ varchar Login

+ varchar Password

+ varchar nimber\_phone

+void регистрация в системе buyer()

+void просмотр buyer()

}

class Sale {

+ int Id

+ int id\_order

+int id\_site\_operator

+ void просмотр Sale()

}

class site\_operator {

+ int Id

+ varchar FIO

+ date birthdate

+ varchar Login

+ varchar Password

+ varchar nimber\_phone

+void регистрация в системе site\_operator()

+void просмотр site\_operator()

}

class zapchast {

+ int Id

+ varchar Title

+ int cost

+ int id\_model

+void просмотр zapchast()

}

class order{

+ int id\_buyer

+ id\_zapchast

+ date date

+int count

+int sum

+ void просмотр order()

}

model\_auto --> zapchast

zapchast --> order

order --> Sale

buyer -->order

site\_operator -->Sale

@enduml

На рисунке 4 представлена диаграмма компонентов для Разработка информационной системы автомобильных запчастей автоваза.

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 4. Диаграмма компонентов

## ЗАКЛЮЧЕНИЕ

Результатом данной дипломной работы является разработанная информационная система по учету запанных частей для автомобилей, выполняющая круг задач, связанных с анализом результатов, с которыми ежедневно приходится сталкиваться, работникам магазина, ведь рассмотрев функционирование данной предметной области мы выяснили, что учет запчастей - это процесс, который продолжается непрерывно. Поэтому данный процесс требует постоянного контроля и четкой организации. Разработанная система управления базой данных обладает следующими качествами:

Простота освоения специалистами, не владеющими языками программирования, что сокращает время на проектирования и уменьшает затраты на разработку системы;

Удобный интерфейс, который позволяет с легкостью освоить управление базой данных;

Обеспечивает автоматизацию обработки данных, что облегчает работу пользователям.

Позволяет быстро получить ответ на любой вопрос по запросу пользователя.

Проведя в ходе исследовательской работы анализ литературы, предметной области и разработку информационной системы, мы можем сделать вывод о том, что результат данного исследования занимает важное место в автоматизации деятельности магазина по учету запасных частей для автомобилей. Таким образом, поставленная нами цель достигнута, задачи выполнены.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Законодательные и нормативные акты:

1. ГОСТ Р 7.0.12-2011 Библиографическая запись. Сокращение слов и словосочетаний на русском языке. Общие требования и правила. – М.: Стандартинформ, 2012. – 61 с. – Текст: непосредственный.
2. ГОСТ 7.1-2003 Библиографическая запись. Библиографическое описание. Общие требования и правила составления. – М.: Стандартинформ, 2010. – 92 с. - Текст: непосредственный.
3. ГОСТ 7.32-2017 Отчет о научно-исследовательской работе. Структура и правила оформления. – М.: Стандартинформ, 2017. – 47 с. - Текст: непосредственный.
4. ГОСТ 7.82-2001 Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления. – М.: ИПК Издательство стандартов, 2001. – 39 с. - Текст: непосредственный.
5. ГОСТ Р 7.0.100-2018 Библиографическая запись. Библиографическое описание. Общие требования и правила составления. – М.: Стандартинформ, 2018. – 122 с. - Текст: непосредственный.
6. ГОСТ Р 7.0.5-2008 Библиографическая ссылка. Общие требования и правила составления. – М.: Стандартинформ, 2008. – 32 с. - Текст: непосредственный.
7. Единая система программной документации. – М.: Стандартинформ, 2005. – 128 с. - Текст: непосредственный.

Учебная и научная литература:

1. Иванова, Г.С. Технология программирования: учебник для студентов вузов обуч. по напр. «Информатика и вычислительная техника» / Г.С. Иванова. – 3-е изд., стер. – Москва: Кнорус, 2018. – 333 с. – Текст: непосредственный. Павловская, Т.А. C#. Программирование на языке высокого уровня: учебник для студентов вузов. – СПб: Питер, 2020. – 432 с. - Текст: непосредственный.
2. Павловская, Т.А. C#. Программирование на языке высокого уровня: учебник для студентов вузов. / Т.А. Павловская. – СПб: Питер, 2020. – 432 с. – Текст: непосредственный.
3. Перлова, О.Н. Проектирование и разработка информационных систем: учебник. / О.Н. Перлова О.П. Ляпина, А.В. Гусева. – 2-е изд, стер. – Москва. Издательский центр «Академия», 2018. – 256 с. – Текст: непосредственный.

Интернет-документы:

1. Altlinux: Discord – [Электронный ресурс]. – URL: https://www.altlinux.org/Discord (дата обращения: 16.10.2021). – Текст: электронный.
2. Draw.io. Интернет-сервис для построения схем и диаграмм Draw.io. – [Электронный ресурс]. – URL: https://www.draw.io/ (дата обращения: 30.09.2020). – Текст: электронный.
3. Lucidchart – [Электронный ресурс]. – URL: https://www.lucidchart.com/pages/ru/visio-%D1%87%D1%82%D0%BE-%D1%8D%D1%82%D0%BE (дата обращения: 17.10.2021). – Текст: электронный.
4. PlantUml. Интернет-сервис для построения UML-диаграмм. – [Электронный ресурс]. – URL: https://plantuml.com/ (дата обращения: 05.10.2020). – Текст: электронный.
5. RuProgi. Аналоги DB Browser for SQLite – [Электронный ресурс]. – URL: https://ruprogi.ru/software/sqlite-database-browser (дата обращения: 17.10.2021). – Текст: электронный.
6. Unetway. SQLite – Введение. – [Электронный ресурс]. – URL: https://unetway.com/tutorial/sqlite (дата обращения: 17.10.2021). – Текст: электронный.
7. Введение в язык C# и .NET Framework. – [Электронный ресурс]. – URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/> (дата обращения: 17.10.2021). – Текст: электронный.
8. Википедия. Microsoft PowerPoint – [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/Microsoft\_PowerPoint (дата обращения: 18.10.2021). – Текст: электронный.
9. Википедия. Microsoft Word – [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/Microsoft\_Word (дата обращения: 17.10.2021). – Текст: электронный.
10. Всё о TeamSpeak. TeamSpeak. – [Электронный ресурс]. – URL: https://ts3shtorm.ucoz.ru/ (дата обращения: 16.10.2021). – Текст: электронный.
11. Документация Microsoft. Интегрированная среда разработки Visual Studio. – [Электронный ресурс]. – URL: https://docs.microsoft.com/ru-ru/visualstudio/get-started/visual-studio-ide?view=vs-2022 (дата обращения: 17.10.2021). – Текст: электронный.
12. Документация Microsoft. Общие сведения об ASP.NET Core. – [Электронный ресурс]. – URL: https://docs.microsoft.com/ru-ru/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-6.0 (дата обращения: 17.10.2021). – Текст: электронный.
13. Документация Microsoft. Общие сведения об ASP.NET Core SignalR – [Электронный ресурс]. – URL: https://docs.microsoft.com/ru-ru/aspnet/core/signalr/introduction?view=aspnetcore -6.0 (дата обращения: 17.10.2021). – Текст: электронный.
14. Инсталлятор Actual Installer. – [Электронный ресурс]. – URL: http://www.actualinstaller.ru/ (дата обращения: 21.10.2021). – Текст: электронный.
15. О программе. Ventrilo – [Электронный ресурс]. – URL: <https://oprogramme.ru/prog.php?id=555> (дата обращения: 22.10.2021). – Текст: электронный.
16. Полное руководство по языку программирования C# 6.0 и платформе .NET 4.6. – [Электронный ресурс]. – URL: <http://metanit.com/sharp/tutorial/> (дата обращения: 27.09.2020). – Текст: электронный.
17. Хабр. Пять простых шагов для понимания JSON Web Tokens (JWT) – [Электронный ресурс]. – URL: https://habr.com/ru/post/340146/ (дата обращения: 17.10.2021). – Текст: электронный.
18. Цифровые новости – [Электронный ресурс]. – URL: https://www.comnews.ru/content/212372/2020-12-24/2020-w52/deystvitelno-li-vash-brandmauer-blokiruet-ispolzovanie-nekotorykh-prilozheniy(дата обращения: 16.11.2021). – Текст: электронный.