

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
«Южно-Уральский государственный университет»
(национальный исследовательский университет)
Высшая школа экономики и управления
Кафедра «Информационные технологии в экономике»

РАБОТА ПРОВЕРЕНА
Рецензент, заведующий сектором
веб-проектов ОГСТ УИ ЮУрГУ
_____ / Сапожников А.И. /
«_22_» ___06___2018__ г.

ДОПУСТИТЬ К ЗАЩИТЕ
Зав. кафедрой, д.т.н., с.н.с.
_____ / Б.М. Суховилов /
«_ ___ » _____ 20 ___ г.

Разработка информационной системы учета работы автосалона

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 09.03.03.2018. 115. ВКР

Руководитель, доцент
_____ / В.А. Конов /
«_ ___ » _____ 20 ___ г.

Автор
студент группы ЭУ-469
/ В.В. Лесняк /
«_ ___ » _____ 20 ___ г.

Нормоконтролер, доцент
/ Е.А. Конова /
«_ ___ » _____ 20 ___ г.

Челябинск 2018

АННОТАЦИЯ

Лесняк В.В. «Разработка информационной системы учета работы автосалона». – Челябинск: ЮУрГУ, ЭиУ-469, 68 с., 28 ил., 3 табл., библиогр. список – 8 наим..

Выпускная квалификационная работа выполнена с целью разработки программного обеспечения, предоставляющего возможности для ведения учёта поступления и продажи автомобилей, авто комплектующих, запасных частей, а так же Web-интерфейса позволяющего предоставлять справочную информацию. В работе показана актуальность выбранной темы, сформулирована цель и задачи.

Разработана система для учёта работы автосалона и просмотра справочной информации через интернет, которая успешно отлажена и протестирована.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
1 ПОСТАНОВКА ЗАДАЧИ.....	7
Выводы по разделу один	7
2 АНАЛИЗ СУЩЕСТВУЮЩИХ СИСТЕМ.....	8
2.1 Товар-Деньги-Товар	8
2.2 ИП: Торговый склад	10
2.3 Microinvest Склад Pro	11
2.4 Subtotal	13
2.5 СуперСклад.....	14
2.6 Большая птица.....	15
2.7 Мой склад	17
2.8 УчетОблако.....	18
2.9 CloudShop.....	19
2.10 1С: Управление торговлей.....	21
Выводы по разделу два	22
3 АНАЛИЗ ПОТОКОВ ДАННЫХ В РАЗРАБАТЫВАЕМОЙ СИСТЕМЕ	22
3.1 Схема работы системы	22
3.2 Структура создаваемой системы.....	23
3.3 Потoki данных внутри приложения	25
3.4 Описание требующихся данных	28
Выводы по разделу три.....	32
4 РАЗРАБОТКА БАЗЫ ДАННЫХ.....	33
4.1 Таблицы	33
4.2 Триггеры	36
4.3 Представления.....	37
4.4 Функции и процедуры.....	38
Выводы по разделу четыре.....	39
5 РАЗРАБОТКА ПРИЛОЖЕНИЯ.....	41

5.1	Среда разработки	41
5.2	Дизайн программы и краткое описание компонентов	41
5.3	Диаграмма классов	41
5.4	Методы классов.....	42
	Выводы по разделу пять	50
6	РАЗРАБОТКА WEB-ИНТЕРФЕЙСА.....	51
	Выводы по разделу шесть	61
7	РАЗРАБОТКА ТЕСТОВ ДЛЯ ПРИЛОЖЕНИЯ	61
7.1	Разработка закодированного теста пользовательского интерфейса	61
7.2	Разработка модульного теста.....	62
	Выводы по разделу семь.....	63
	ЗАКЛЮЧЕНИЕ	64
	БИБЛИОГРАФИЧЕСКИЙ СПИСОК	65

ВВЕДЕНИЕ

Одной из важнейших проблем в учёте работы автосалона является несовершенство программных продуктов. По этой причине автосалонам приходится покупать несколько систем, которые нацелены на «узкий круг» учёта. В связи с этим часто случаются ошибки в данных, так как приходится одновременно вести несколько систем.

Актуальность дипломного проекта в том, что разрабатываемая информационная система должна объединить все необходимые модули учёта в одном программном продукте, то есть исключить трату денежных средств на покупку лицензий нескольких ИС.

Цель дипломного проекта – разработка уникальной информационной системы для учёта работы автосалона.

Для достижения поставленной цели решены следующие задачи:

- проведён анализ предметной области;
- проведён анализ имеющихся систем - конкурентов;
- разработаны диаграммы и модели;
- разработана структура и сама база данных;
- разработан Web-интерфейс;
- проведены тесты приложения.

Данная система нацелена не только на улучшение работы сотрудников автосалона, но и на улучшение связи клиента с менеджером, что играет большую роль в дальнейшем сотрудничестве.

Для создания информационной системы использовались языки программирования C#, HTML5, CSS3 и JavaScript, в качестве системы управления базами данных был использован программный продукт MariaDB. Основной используемый инструментарий для разработки и администрирования приложения – Microsoft Visual Studio 2017 и Adobe Dreamweaver CC 2018.

Базой данных называется система специальным образом организованных данных программных, технических и языковых средств, предназначенных для централизованного накопления и коллективного использования.

Structure Query Language (SQL) – язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных.

HTML (HyperText Markup Language «язык гипертекстовой разметки») — стандартизированный язык разметки документов в сети Internet. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.

1 ПОСТАНОВКА ЗАДАЧИ

Требуется разработать систему для учёта работы салона по продаже автомобилей, со следующей функциональностью.

- Учет прихода и продажи автомобилей;
- Учет прихода и продажи товаров для автомобилей (запасные части, аксессуары, комплектующие);
- Учёт проведения технического обслуживания автомобилей;
- Учет ремонтных работ;
- Просмотр справочного материала через Web-сервис.

Выводы по разделу один

Проведен анализ предметной области, выполнена постановка задачи.

2 АНАЛИЗ СУЩЕСТВУЮЩИХ СИСТЕМ

Проведен анализ информационных систем, сходных по функционалу. Проведена оценка по десятибалльной шкале, результат приведен в таблице 1.

Таблица 1 – Сравнительный анализ систем

Программа	Простота освоения	Возможности функционала	Стабильность	Техническая поддержка
Мой склад	10	8	9	10
CloudShop	9	7	9	8
УчетОблако	9	7	9	8
1С: Управление торговлей	3	10	9	10
СуперСклад	8	7	7	7
Большая птица	7	7	8	7
ИП: Торговый склад	8	9	9	8
Subtotal	7	6	7	8
Microinvest Склад Pro	8	6	7	8
Товар-Деньги-Товар	7	7	7	7
Автосалон 2.0	10	9	10	10

Сухие цифры в таблице являются не слишком весомыми аргументами в вопросе о том, какая же программа лучшая в своем сегменте. Поэтому проанализировав существующие программы, разработаем наилучший вариант для пользователей.

2.1 Товар-Деньги-Товар

Данная программа является крепким середняком в мире софта для учета склада. В ней трудно выделить ярко выраженные плюсы и критические минусы. Ее универсальный функционал «сделан» под ведение складского учета на небольших предприятиях розничной и оптовой торговли. Интерфейс главного окна приведен на рисунке 1.

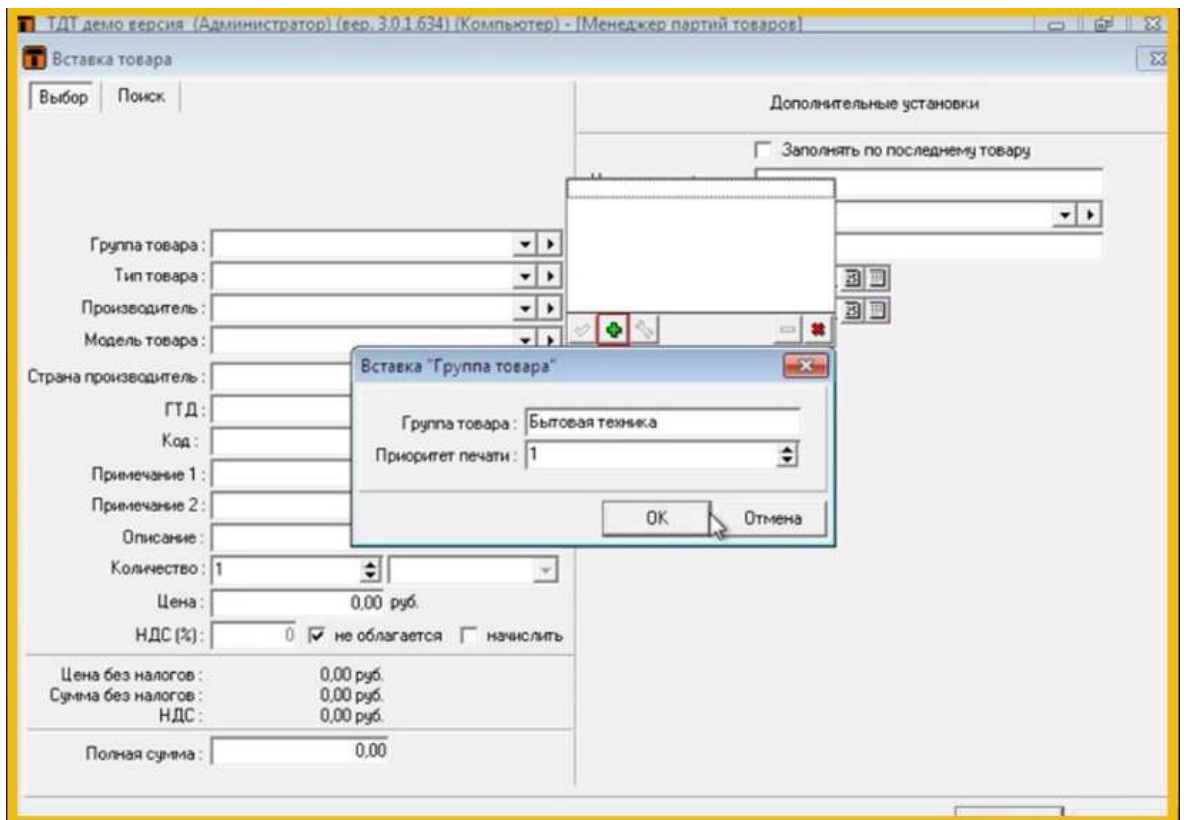


Рисунок 1 – Интерфейс системы «Товар-Деньги-Товар»

Плюсы

- 1 Наличие демо-версии, которой можно пользоваться три месяца.
- 2 Поддержка неограниченного количества складов или торговых точек, а также возможность создания любой собственной структуры предприятия.
- 3 Поддержка работы «ТДТ» на удаленных компьютерах, вне локальных сетей.
- 4 Возможность применения данного софта в сборочном производстве, когда готовый товар изготавливается из заранее приобретенных компонентов.

Минусы

- 1 Иногда в программном коде «ТДТ» встречаются «дыры», которые открывают возможность для махинаций с учетом.
- 2 Бесплатно программа обновляется только полгода, а по истечении этого срока приходится вносить абонентскую плату за каждое обновление.

Цена лицензии

Цена на электронную версию программы для одного рабочего компьютера – 3894 рубля. Однако при необходимости установки данного софта на второй, третий и последующие компьютеры стоимость будет снижаться. Минимальный порог – 2950 рублей (пятый и более рабочий компьютер). Соответственно, общая стоимость «Товар-Деньги-Товар» для пяти рабочих мест составит: $3894 + 3658 + 3422 + 3186 + 2950 = 17\ 110$ рублей.

2.2 ИП: Торговый склад

Эта программа складского и торгового учета – своеобразный старожил рынка, первая ее версия вышла еще в начале «нулевых». Вместе с софтом «ИП: Бухгалтерия» она входит в программный комплекс «Инфо-Предприятие». Не трудно догадаться, что в симбиозе данные программы работают вдвойне эффективнее. Интерфейс главного окна приведен на рисунке 2.

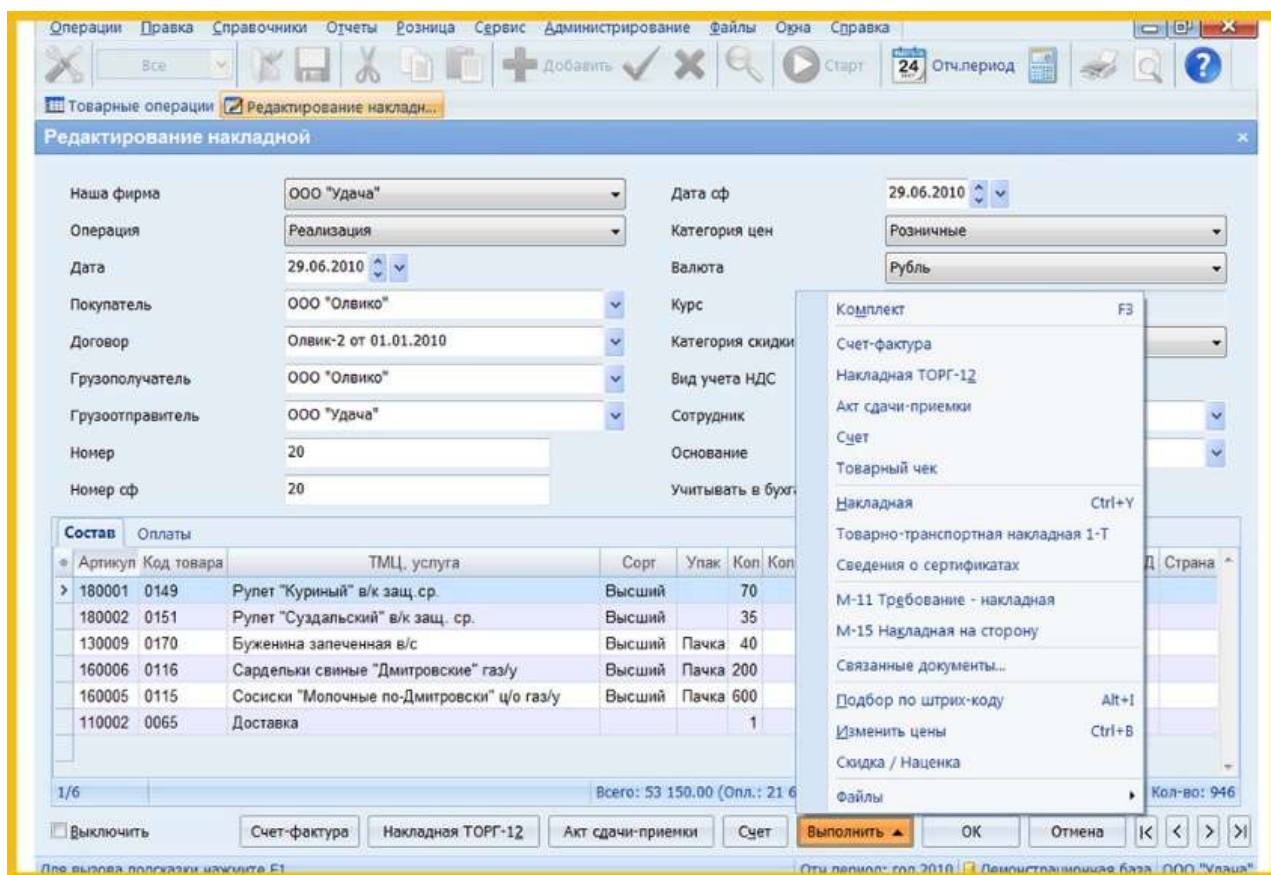


Рисунок 2 – Интерфейс системы «ИП: Торговый склад»

Плюсы

- 1 Стабильная и бесперебойная работа на компьютерных конфигурациях, как старой, так и современной сборки.
- 2 Лучшее соотношение цены и возможностей функционала, в сравнении с аналогами.
- 3 Своевременная техническая поддержка и выход обновлений.
- 4 Наличие бесплатной версии, которая прекрасно подходит для полноценного ведения складского учета на предприятиях малого бизнеса.
- 5 Возможность ручной настройки некоторых аспектов программы без вызова программиста.

Минусы

- 1 Несколько архаичный интерфейс и общая структура программы.
- 2 Недостаточное количество автоматических настроек типовых операций.

Цена лицензии

Версию «Стандарт» можно приобрести за 6900 рублей, «Проф» стоит 11900 рублей. Дополнительные сетевые места для этих версий стоят 1950 и 2900 рублей соответственно. Возможна покупка каждой из версий в рассрочку, разбитую на четыре равных квартальных платежа.

2.3 Microinvest Склад Pro

Гость из Болгарии, программа Microinvest Склад Pro, предназначена для автоматизации складского учета материалов на предприятиях, преимущественно сетевой розничной торговли. Ее уникальность заключается в возможности выстраивания управляемых логических цепей из элементарных операций с последующим тотальным контролем каждого бизнес-процесса.

Интерфейс главного окна приведен на рисунке 3.

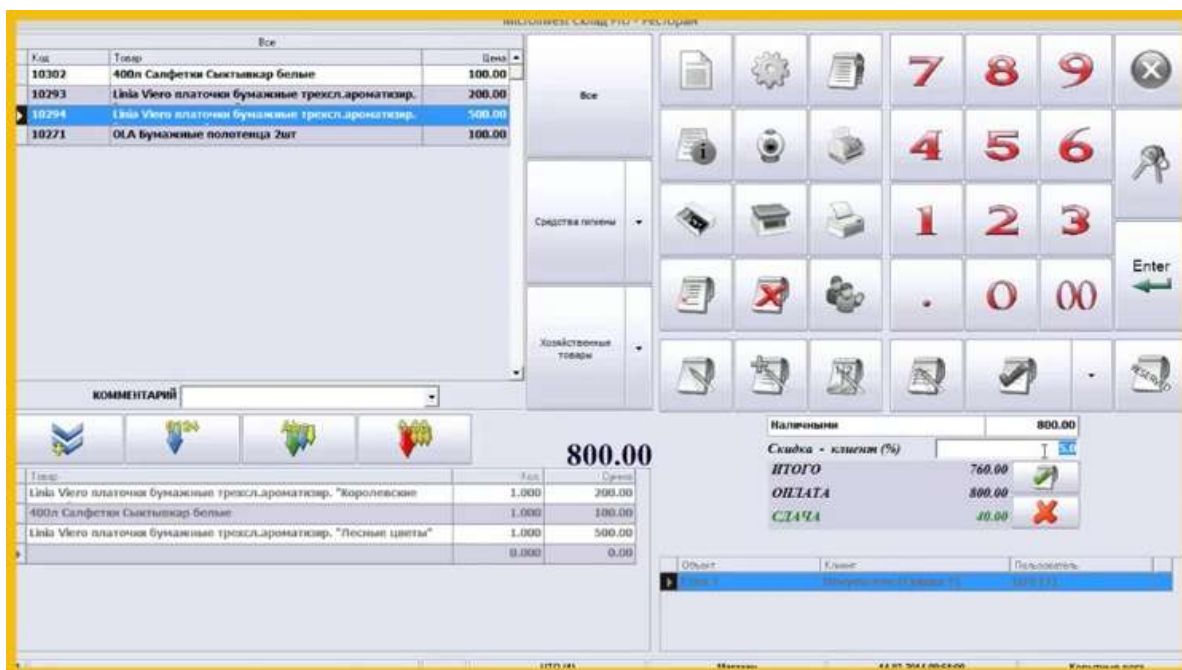


Рисунок 3 – Интерфейс системы «Microinvest Склад Pro»

Плюсы

- 1 Функционал программы идеально подходит для ресторанов и магазинов самообслуживания.
- 2 Не требует тонкой настройки и внесения изменений в систему после инсталляции.
- 3 Присутствует опция синхронизации программы с электронным торговым оборудованием для выдачи чеков.
- 4 Довольно гибкая система настройки программы под конкретные нужды.

Минусы

- 1 Далеко не интуитивный интерфейс.
- 2 Проблемы с внутренним поиском документов и формированием отчетов.
- 3 Относительно высокая стоимость программы.

Цена лицензии

Цена полной версии программы составляет 199 евро или чуть больше 12 000 рублей по курсу на февраль 2017 года.

2.4 Subtotal

Назначением данного онлайн-сервиса, разработанного в культурной столице России, является максимальная автоматизация процессов розничной торговли. Мощный набор инструментов для ведения складского учета позволяет Subtotal составить достойную конкуренцию более специализированным программам и сервисам. Интерфейс главного окна приведен на рисунке 4.

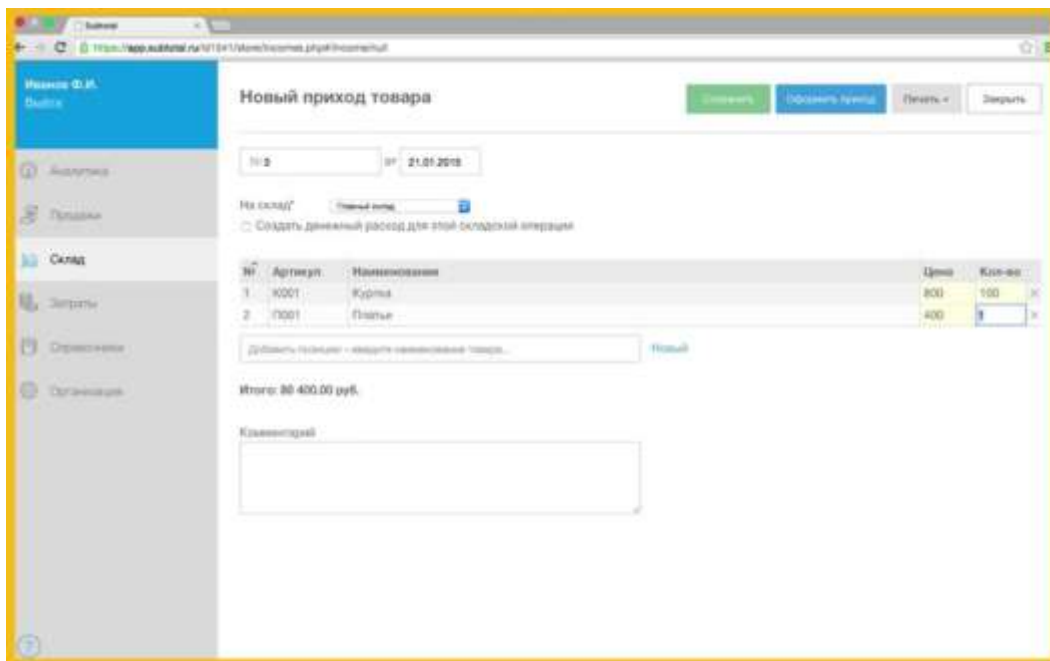


Рисунок 4 – Интерфейс системы «Subtotal»

Плюсы

- 1 Интеграция с бухгалтерским онлайн-сервисом «Мое дело» и пакетом программ 1С: Предприятие.
- 2 Возможность работы в сервисе с планшетов и ноутбуков.
- 3 Поддержка ЕГАИС, что позволяет торговать алкоголем.
- 4 Наличие приложения для создания шаблонов составных товаров.
- 5 Дружелюбный интерфейс и потрясающая легкость освоения программы.

Минусы

- 1 «Молодость» сервиса и, как следствие, отсутствие некоторых специализированных функций.

Цена лицензии

Первый месяц работы в Subtotal не стоит ни копейки. А вот за каждый последующий месяц вносится абонентская плата в размере 1400 рублей при подключении одной торговой точки. Ежемесячная абонентская плата за каждую дополнительную торговую точку составляет 900 рублей.

2.5 СуперСклад

Первая версия программы «СуперСклад» вышла в далеком 1993 году. С тех пор данный софт постоянно совершенствовался, идя в ногу со временем и с законодательством, а в 2016 году была запущена облачная версия «СуперСклада». Секрет такой долгой жизни программы прост – максимальная легкость освоения вкупе с наличием всех нужных функций для ведения складского учета на небольшом предприятии. Интерфейс главного окна приведен на рисунке 5.

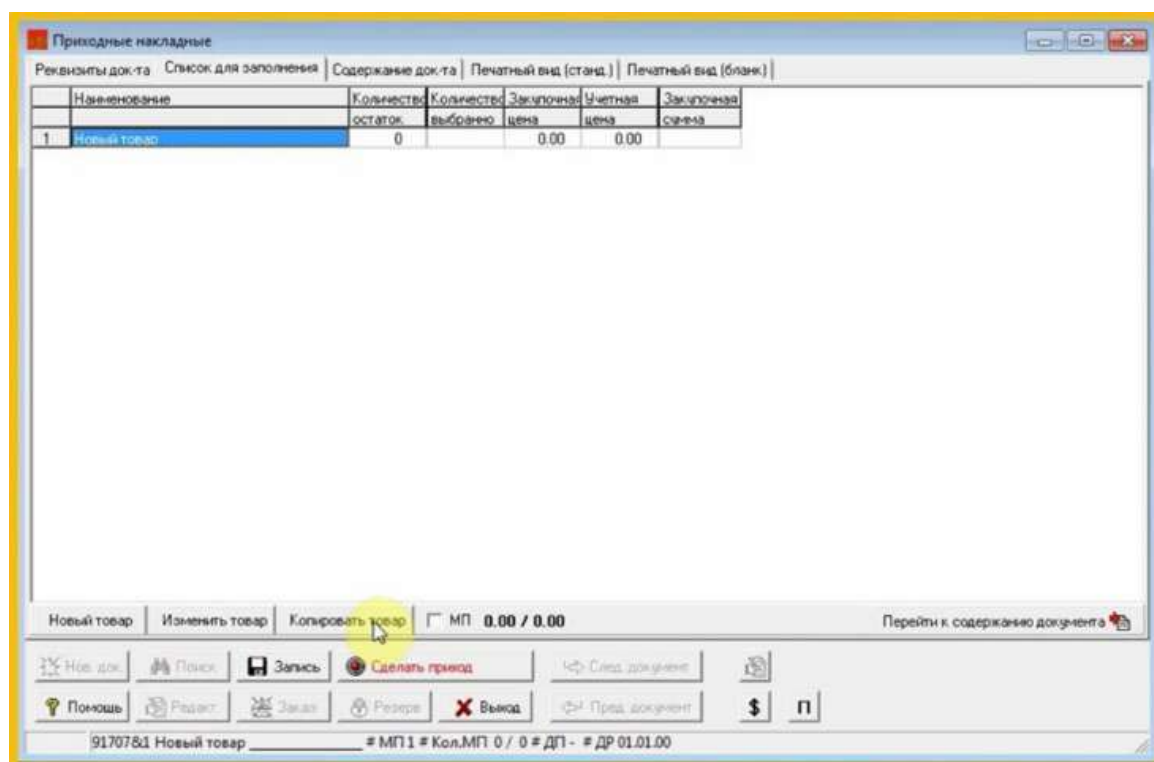


Рисунок 5 – Интерфейс системы «СуперСклад»

Плюсы

1 Научиться пользоваться этой программой за несколько дней сможет даже человек без бухгалтерского образования.

2 Интегрированная опция генератора первичных документов, позволяющая пользователю создавать собственные формы документации.

3 Простая и гибкая настройка разграничения доступа пользователей к отдельным документам и отчетам.

4 Поддержка автоматизированного учета товаров на 100 различных складов.

Минусы

1 Возможностей функционала недостаточно для ведения складского учета на крупных торговых предприятиях.

Цена лицензии

Устанавливаемая версия программы СуперСклад обойдется в 985 рублей, «облачная» – в 345 рублей.

2.6 Большая птица

Онлайн-сервис складского учета «Большая птица» имеет репутацию одной из лучших программ в своем сегменте. Несмотря на свой относительно юный возраст, этот продукт от отечественных программистов из фирмы «Этерон» получил признание многих владельцев малого бизнеса благодаря своей надежности, высокой скорости работы и наличию неплохого функционала. Интерфейс главного окна приведен на рисунке 6.

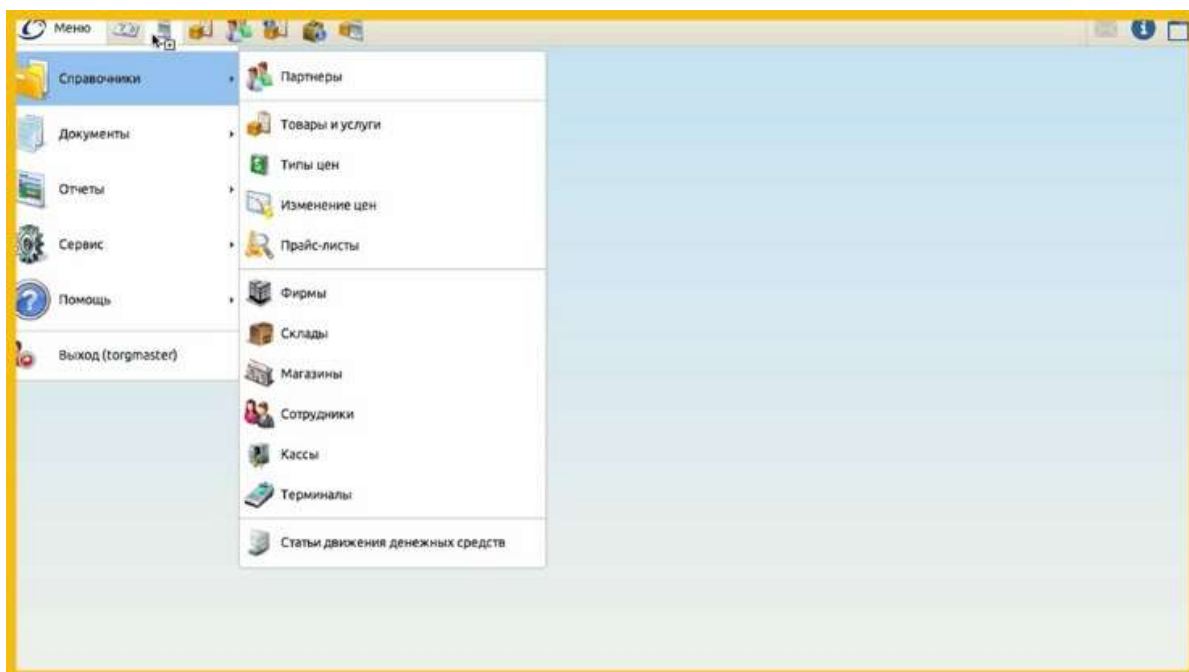


Рисунок 6 – Интерфейс системы «Большая птица»

Плюсы

1 Богатый интерфейс программы, как у устанавливаемых аналогов, позволит быстро сориентироваться в ней пользователям, привыкшим работать с «классикой».

2 Наличие внутри программы практически отдельного приложения для продавца – простого и функционального.

3 Возможность автоматического сохранения последних изменений.

Минусы

1 Местами «сырой» программный код, способствующий возникновению сбоев в работе программы.

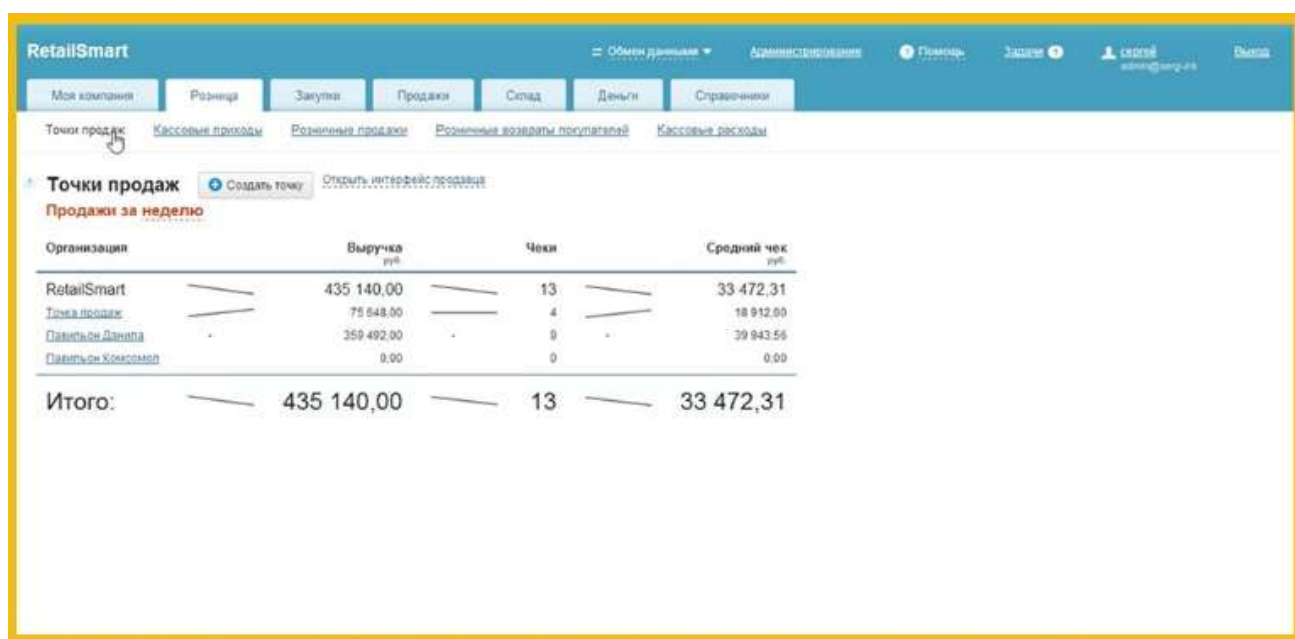
2 Периодически возникающие проблемы в модуле работы с интернет-магазинами.

Цена лицензии

В сервисе есть два тарифных плана – «Колибри» и «Альбатрос». Первый – совершенно бесплатный, но с ограниченными возможностями. За пользование вторым придется заплатить от 790 до 990 рублей в месяц – в зависимости от общего срока подписки (чем он больше, тем меньше стоимость одного месяца).

2.7 Мой склад

Дата рождения данного облачного сервиса складского учета – 2008 год. Уже через пять лет «Мой склад» получил премию «Облако-2013» и был признан лучшим облачным сервисом России. Несмотря ни на что, этот сервис продолжает держать марку и на сегодняшний день заслужил доверие более 700 тысяч организаций малого и среднего бизнеса. Интерфейс главного окна приведен на рисунке 7.



The screenshot shows the 'RetailSmart' web interface. At the top, there is a navigation bar with tabs for 'Моя компания', 'Разница', 'Закупки', 'Продажи', 'Склад', 'Деньги', and 'Справочники'. Below this, there are sub-tabs for 'Точки продаж', 'Кассовые приходы', 'Розничные продажи', 'Розничные возвраты покупателей', and 'Кассовые расходы'. The main content area is titled 'Точки продаж' and includes a sub-section 'Продажи за неделю'. A table displays sales data for various organizations, including 'RetailSmart', 'Точка продаж', 'Павильон Давида', and 'Павильон Колесяев'. The table columns are 'Организация', 'Выручка руб.', 'Чеки', and 'Средний чек руб.'. The total sales for the week are 435,140.00 rubles, 13 checks, and an average check of 33,472.31 rubles.

Организация	Выручка руб.	Чеки	Средний чек руб.
RetailSmart	435 140,00	13	33 472,31
Точка продаж	75 848,00	4	18 912,00
Павильон Давида	359 492,00	9	39 943,56
Павильон Колесяев	0,00	0	0,00
Итого:	435 140,00	13	33 472,31

Рисунок 7 – Интерфейс системы «Мой склад»

Плюсы

- 1 Универсальность и удивительная простота освоения программы.
- 2 Постоянное совершенствование функционала и дружелюбная техподдержка.
- 3 Гибкий API, позволяющий производить интеграцию с другими полезными сервисами и программами.
- 4 Беспрецедентный уровень защиты данных пользователя.
- 5 Поддержка ЕГАИС.
- 6 Доступ к системе с планшетов и смартфонов под управлением iOS и Android.

Минусы

- 1 Отсутствие шаблонов для продаж.
- 2 Отсутствие возможности создания архивов данных.

Цена лицензии

Абонентская плата за тариф «Базовый», рассчитанный на два рабочих места, составляет 1000 рублей в месяц. «Профессиональный» (5 рабочих мест) стоит 2900 рублей, а за самый продвинутый тариф – «Корпоративный» (до 10 сотрудников) придется ежемесячно платить 6900 рублей. Тариф «Бесплатный» для одного рабочего места соответствует своему названию, но имеет ряд существенных ограничений.

2.8 УчетОблако

В данном онлайн-сервисе во главе стоят функции, связанные с движением материальных ценностей и учетом остатков на складе. Простой интерфейс и ориентация на малый бизнес делают «УчетОблако» прекрасным выбором для малых оптово-розничных предприятий. Интерфейс главного окна приведен на рисунке 8.

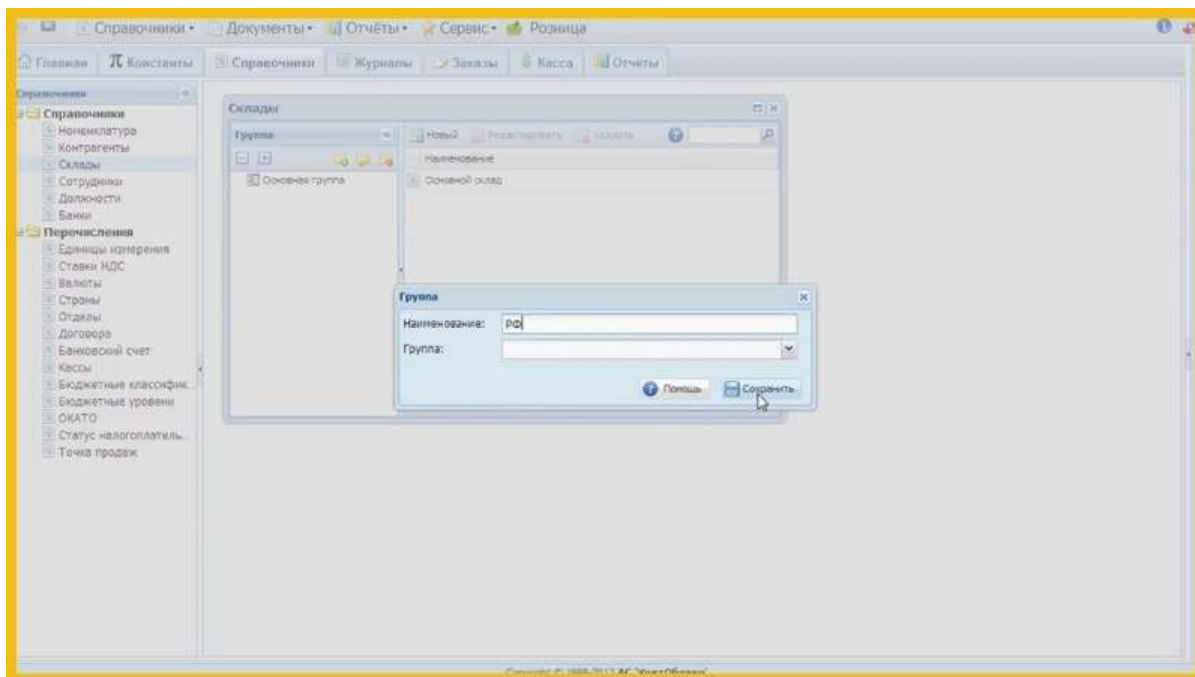


Рисунок 8 – Интерфейс системы «УчетОблако»

Плюсы

- 1 Наличие удобного конструктора печати, позволяющего настраивать печать на ценниках, чеках и документах.
- 2 Высокая стабильность работы даже на слабых компьютерах с низкой скоростью интернет-соединения.
- 3 Возможность объединения в один массив нескольких удаленных торговых точек или складов.

Минусы

- 1 Медленный темп развития сервиса и внедрения новых возможностей.

Цена лицензии

Тариф «Начинающий» абсолютно бесплатен и доступен только для одного пользователя. Что касается тарифа «Предприниматель» (3 пользователя), то ежемесячная абонентская плата за него составляет смешную сумму – от 80 до 100 рублей, в зависимости от длительности подписки. Плата за каждое дополнительное рабочее место – 80 рублей в месяц.

2.9 CloudShop

Молодой, практичный и удобный онлайн-сервис для учета складских запасов, материальных ценностей и клиентов. Пользователи 28 стран мира уже оценили его возможности. Его бесплатность и сбалансированность всех компонентов привлекают все больше владельцев малого и среднего бизнеса. Интерфейс главного окна приведен на рисунке 9.



Рисунок 9 – Интерфейс системы «CloudShop»

Плюсы

- 1 Возможность создания интернет-витрины, где выставляются товары, которые есть в базе данных пользователя.
- 2 Надежная система защиты данных пользователя.
- 3 Дружелюбность интерфейса и гибкость тонких настроек программы.
- 4 Наличие удобного мобильного приложения.

Минусы

- 1 Отсутствие возможности отсрочки платежа кассиру и привязки к ККМ по 54-ФЗ.
- 2 Нет поддержки Bluetooth и USB-чековых принтеров.

Цена лицензии

За работу в CloudShop не нужно платить, но только при условии наличия не более 5 пользователей. Абонентская плата за подключение каждого последующего пользователя составляет 299 рублей.

2.10 1С: Управление торговлей

Мощный и функциональный софт для управления складом от лидера рынка – компании 1С. Главный и самый серьезный конкурент всех программ и сервисов, описанных в данном сравнении. Интерфейс главного окна приведен на рисунке 10.



Рисунок 10 – Интерфейс системы «1С: Управление торговлей»

Плюсы

1 Запредельные функциональные возможности, практически круглосуточная техническая поддержка.

2 Подходит для управления складским учетом на самых крупных предприятиях.

Минусы

1 Данная программа является не слишком дружелюбной к новичкам, а иногда и к опытным пользователям. Зачастую для «доработки» некоторых функций требуется помощь профессионального программиста.

Цена лицензии

Рекомендованная розничная цена базового пакета «1С: Управление торговлей 8» составляет 6700 рублей.

Выводы по разделу два

Выполнен сравнительный анализ существующих систем, выявлены нужные функции.

3 АНАЛИЗ ПОТОКОВ ДАННЫХ В РАЗРАБАТЫВАЕМОЙ СИСТЕМЕ

Для создания информационной системы, смоделируем схему потока данных при работе в приложении. Для этих целей воспользуемся программным продуктом VPwin .

3.1 Схема работы системы

Для начала нужно определиться, кто будет работать с разрабатываемой системой, а также с какими сторонними модулями будет обмениваться данными:

Директор (начальники разного уровня). Лицо, ведущее контроль над работой компании. Имеет доступ ко всем данным в режиме просмотра.

Менеджер по продажам автомобилей. Лицо, оформляющее процесс продажи. Имеет доступ к данным об автомобилях и клиентах в режиме редактирования, к остальным данным в режиме просмотра.

Менеджер технического центра. Лицо, оформляющее процессы приема, выдачи автомобилей. Имеет доступ к данным об автомобилях и клиентах в режиме редактирования, к остальным данным в режиме просмотра.

Менеджер по продаже запчастей и аксессуаров. Лицо, оформляющее процессы продажи автомобильных запчастей и аксессуаров. Имеет доступ к данным об автомобильных запчастях и аксессуарах в режиме редактирования.

Сотрудник технического центра. Лицо, выполняющее работы по автомобилю. Имеет доступ к данным о запчастях и аксессуарах в режиме просмотра.

Покупатель. Лицо, осуществляющее покупку услуг или автомобиля. Имеет доступ к данным об автомобилях в режиме просмотра.

Клиент на ТО. Лицо, осуществляющее передачу автомобиля на ремонт или ТО.

Отдел по поставщикам. Имеет доступ к данным об автомобильных запчастях и аксессуарах в режиме редактирования.

Отдел кадров. Имеет доступ к данным о сотрудниках в режиме редактирования.

Справочник запчастей и аксессуаров.

Схема потоков данных приведена на рисунке 11.

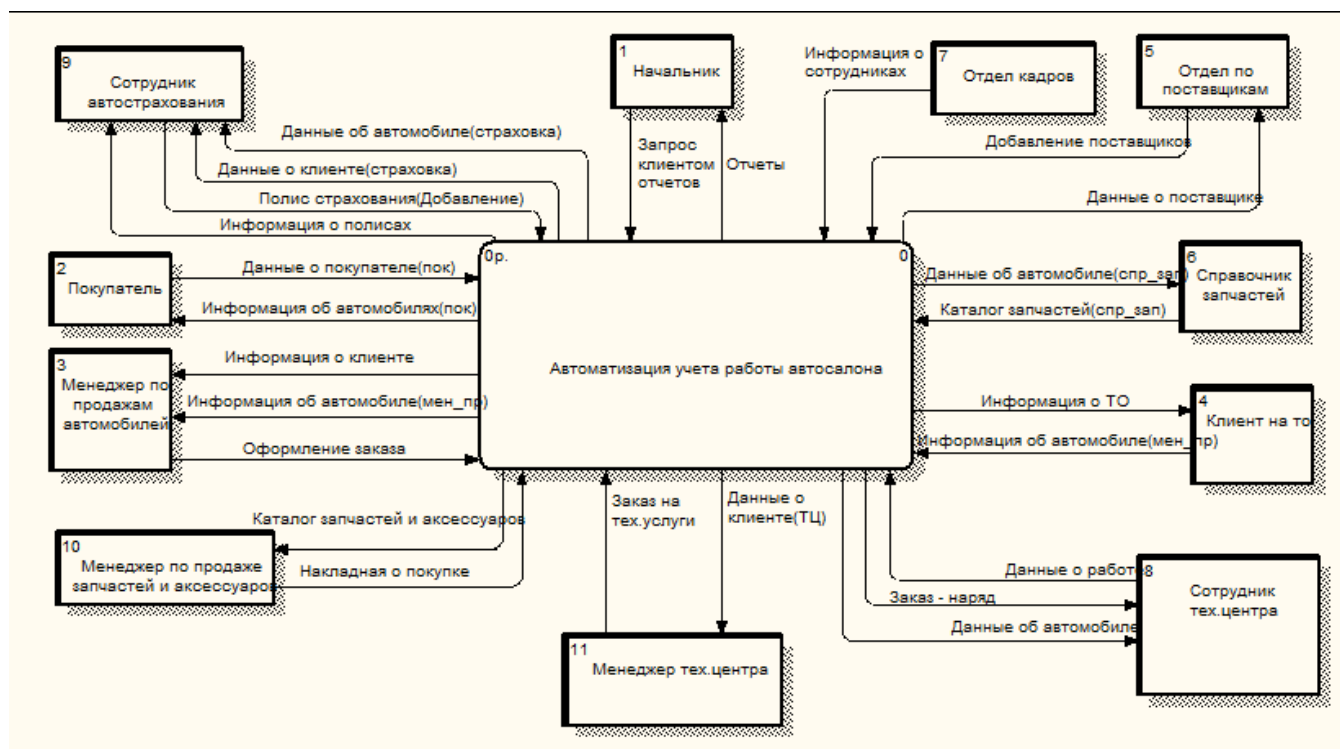


Рисунок 11 – Схема потока данных

3.2 Структура создаваемой системы

В систему входит четыре основных модуля:

Оформление продаж.

Включает в себя:

- оформление продаж автомобилей;
- оформление продаж комплектующих;
- добавление покупателей.

Клиент получает информацию об автомобилях в виде таблиц. Менеджер добавляет клиента в систему (отдельная форма). Менеджер оформляет покупку автомобиля, получая из системы, данные об автомобиле и клиенте. Менеджер оформляет покупку комплектующих, получая из системы таблицу комплектующих.

Оформление услуг

Включает в себя:

- оформление ТО;
- оформление ремонта автомобиля;
- оформление страхового полюса;
- добавление клиентов.

Менеджер добавляет клиента в систему (отдельная форма). Менеджер оформляет услугу ремонта. Менеджер оформляет услугу ТО. Менеджер оформляет услугу страхования.

Формирование отчетов

Включает в себя:

- отчет о заказах;
- отчет о продажах;
- отчет о расходах запчастей и материалов;
- отчет о поставках.

При появлении потребности в отчете, начальник формирует отчет с нужными данными.

Формирование списков

Включает в себя:

- справочник сотрудников;
- справочник запчастей;
- справочник поставщиков.

Все данные заносятся в хранилище данных.

Поток информации внутри приложения приведен на рисунке 12.

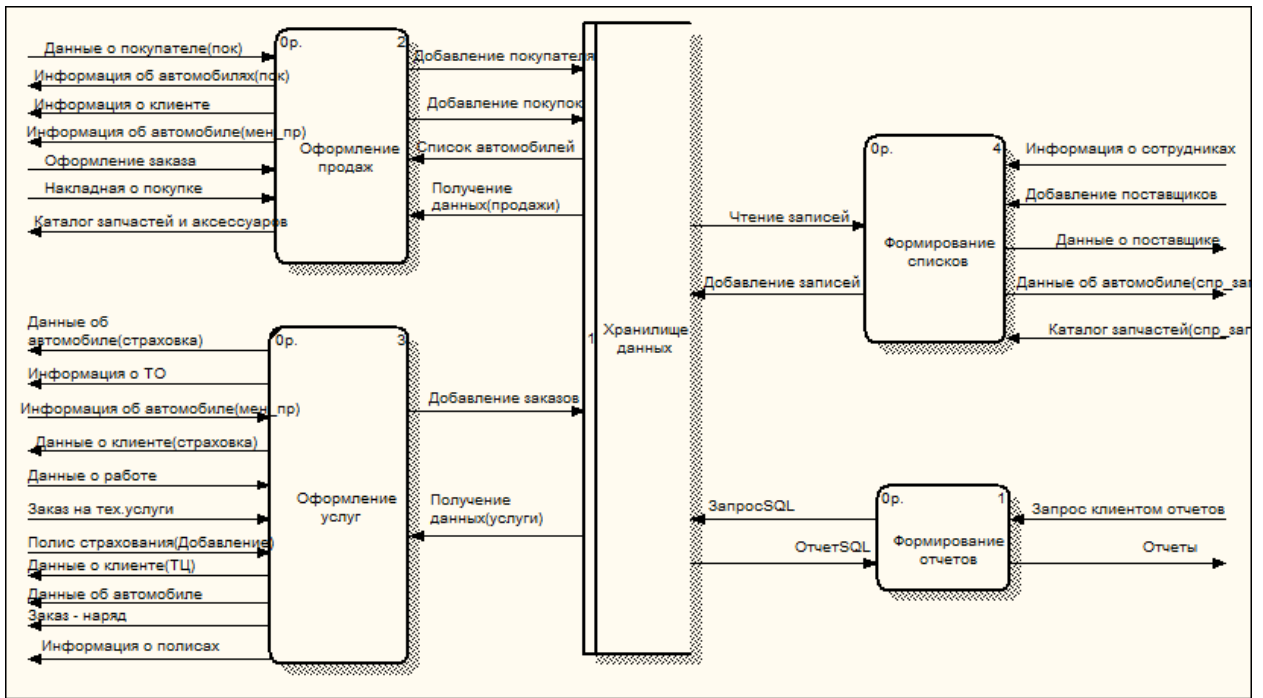


Рисунок 12 – Поток информации внутри приложения

3.3 Поток данных внутри приложения

Оформление продаж

Поток данных при продажах приведен на рисунке 13.

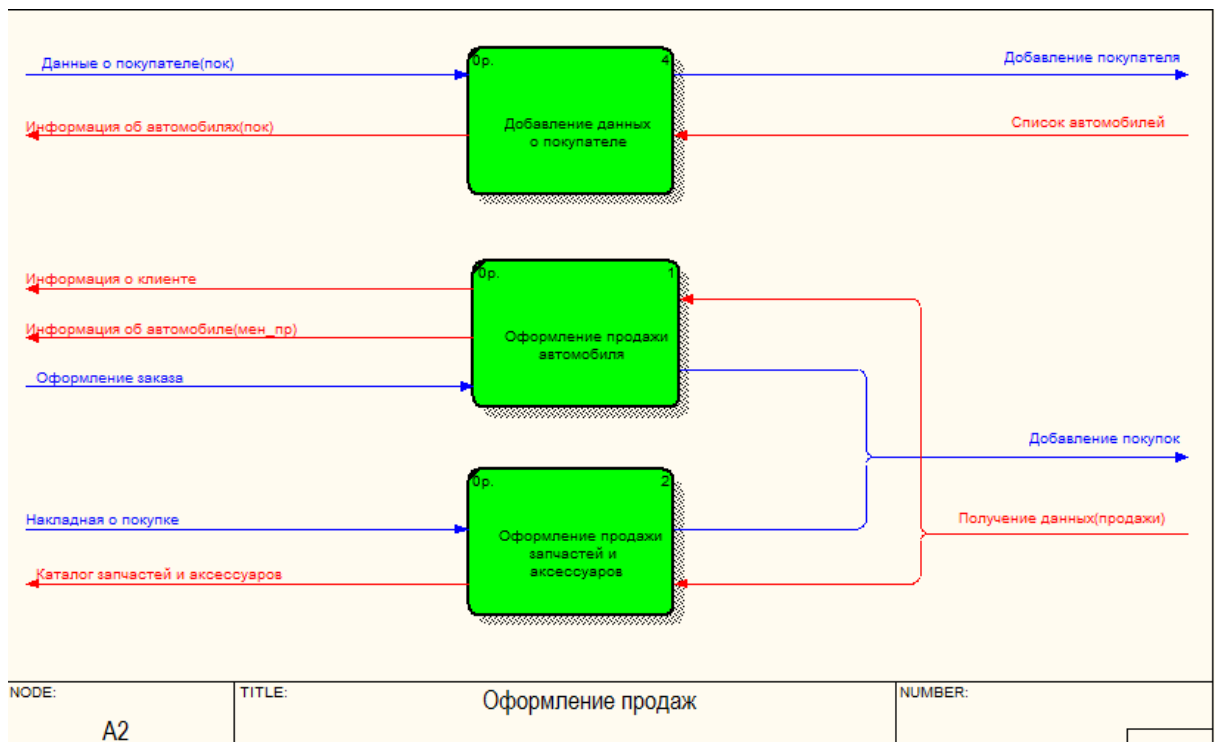


Рисунок 13 – Поток данных при продажах

Данные о покупателе. В системе должны храниться: ФИО, дата рождения, серия и номер паспорта, адрес проживания.

Данные об автомобиле. В системе должны храниться: модель автомобиля, VIN, цвет автомобиля, дата выпуска, цена.

Оформление услуг

Поток данных при оформлении приведен на рисунке 14.

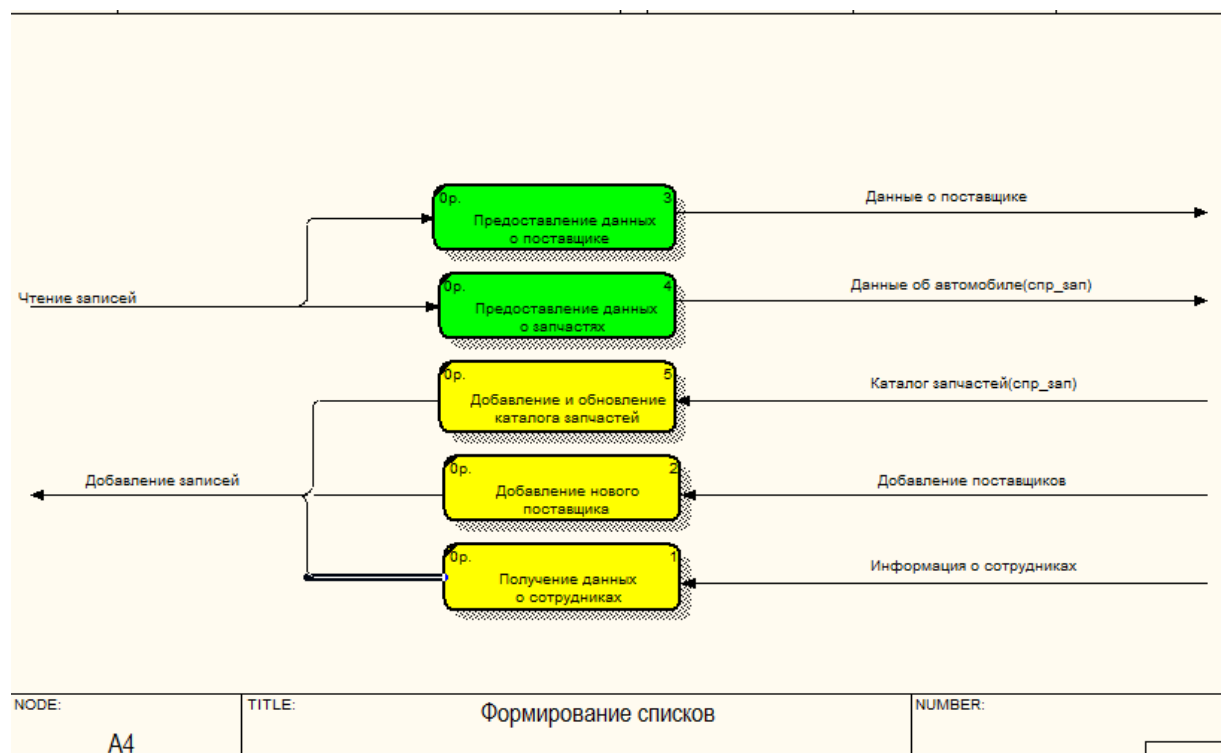


Рисунок 14 – Поток данных при оформлении

Формирование отчетов

Поток данных для отчетов приведен на рисунке 15.

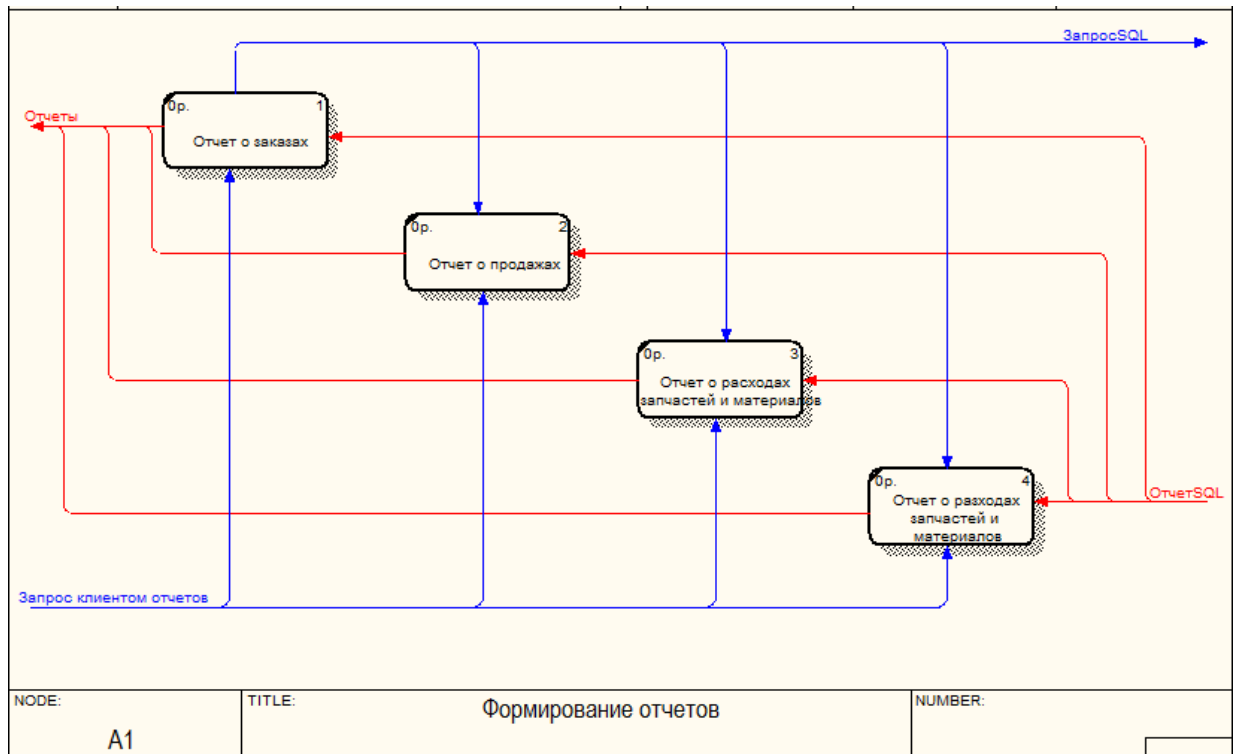


Рисунок 15 – Поток данных для отчетов

Формирование списков

Поток справочной информации приведен на рисунке 16.

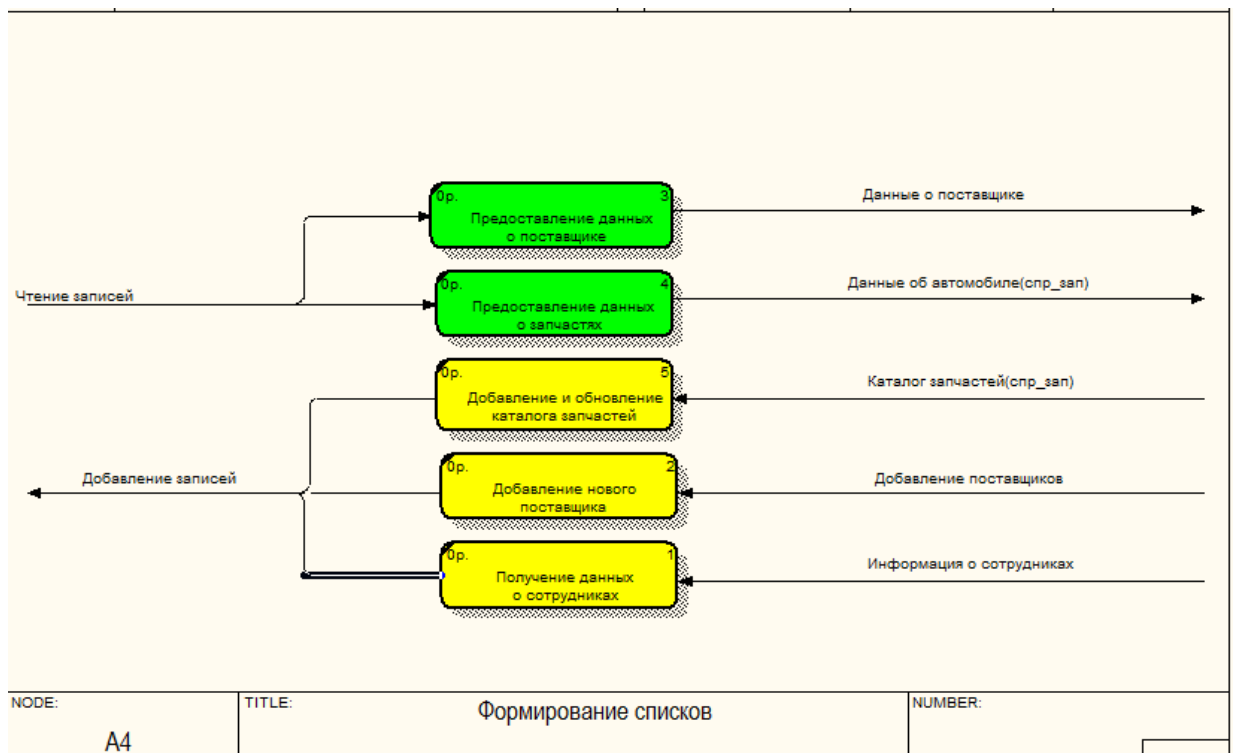


Рисунок 16 – Поток справочной информации

3.4 Описание требующихся данных

Подробное описание информационных потоков в системе описано в таблице 2.

Таблица 2 – Описание потоков данных

Имя	Описание заказчика	Описание разработчика
Данные о клиенте	Данные о клиенте в систему: фамилию, имя, отчество, дату рождения, паспортные данные,	ФИО string(50) ДатаРождения - date Серия№Паспорта -string(15) Адрес - string(100) Телефон - string(15) №ВодУдост. - string(50)
Данные о клиенте(страховка)	Данные о клиенте: ФИО, дата рождения, пол, паспорт, телефон, адрес	ФИО - string(50) Дата рождения - date Пол - string(5) Паспорт - string(15) Телефон - string(15) Адрес - string(100)
Данные о клиенте(ТЦ)	Данные о клиенте в систему: фамилию, имя, отчество, дату рождения, паспортные данные,	ФИО string(50) ДатаРождения - date Серия№Паспорта string(15) - Адрес - string(100) Телефон - string(15) №ВодУдост. - string(50)
Данные о покупателе	Данные о клиенте в систему: фамилию, имя, отчество, дату рождения, паспортные данные,	ФИО string(50) ДатаРождения - date Серия№Паспорта string(15) - Адрес - string(100) Телефон - string(15) №ВодУдост. - string(50)
Данные о покупателе(пок)	Данные о клиенте в систему: фамилию, имя, отчество, дату рождения, паспортные данные,	ФИО string(50) ДатаРождения - date Серия№Паспорта string(15) - Адрес - string(100) Телефон - string(15) №ВодУдостоверения string(50) -
Данные о поставщике	Данные о поставщиках	Наименование - string(50)

Имя	Описание заказчика	Описание разработчика
	Наименование Адрес ИНН Банк БанковскийСчет Телефон	Адрес - string(100) ИНН - string(20) Банк - string(50) БанковскийСчет - string(20) Телефон - string(15)
Данные о работе	Данные о работах Название, количество, цена,	Название - string(100) количество - int64 цена - int64
Данные об автомобиле	Данные о проданных автомобилях. Марка, цвет, комплектация, цена.	Данные передаются в виде таблицы. Организована возможность поиска, выбора, Марка - string(30) Цвет - string(30) Комплектация - string(30) Цена - int64
Данные об автомобиле(спр_зап)	Данные о проданных автомобилях. Марка, комплектация	Марка - string(20) комплектация - string(20)
Данные об автомобиле(страховка)	Данные о проданных автомобилях. Марка, цвет, комплектация, цена.	Данные передаются в виде таблицы. Организована возможность поиска, выбора, Марка - string(30) Цвет - string(30) Комплектация - string(30) Цена - int64
Добавление поставщиков	Данные поставщиков в систему Название Адрес Телефон ИНН Банк Счет в банке	Название (string 100символов) Адрес (string 200символов) Телефон (string 15символов) ИНН (string 20символов) Банк (string 50символов) Счет в банке (string 20символов)
Заказ на тех.услуги	Данные об автомобиле Марка, комплектация, пробег, номер	Марка - string(20) комплектация - string(20) пробег - string(20) номер - string(20)
Запрос клиентом отчетов	Отчет о продажах Отчет о заказах Отчет о расходах	Клиент выбирает из списка необходимый отчет, На форме: Надпись,

Имя	Описание заказчика	Описание разработчика
	запчастей и материалов Отчет об оказанных услугах	Выпадающий список, Кнопки "Построить", "Заккрыть"
Информация о клиенте	Данные о клиенте в систему: фамилию, имя, отчество, дату рождения, паспортные данные,	ФИО string(50) ДатаРождения - date Серия№Паспорта - string(15) Адрес - string(100) Телефон - string(15) №ВодУдостоверения - string(50)
Информация о полисах	Данные о выданных полюсах: Дата, срок, серия-номер, ФИО клиента, ФИО сотрудника, вид, статус	Дата - date срок - string(5) серия-номер - string(20) ФИО клиента - string(50) ФИО сотрудника - string(50) вид - string(20) статус - string(20)
Информация о сотрудниках	Данные о сотрудниках ФИО ДатаРождения Пол Паспорт Должность Отдел Телефон	ФИО - string(50) ДатаРождения - date Пол - string(5) Паспорт - string(10) Должность - string(20) Отдел - string(20) Телефон - string(10)
Информация о ТО	Данные о выполненных работах на ТО Название работы, цена, количество	Название работы - string(100) цена - int64 количество - int64
Информация об автомобиле (мен_пр)	Данные об автомобиле Марка, цвет, комплектация, цена, год	Марка - string(15) цвет - string(15) комплектация - string(15) цена - string(15) год - date
Информация об автомобилях (пок)	Данные об автомобиле Марка, цвет, комплектация, цена, год	Марка - string(15) цвет - string(15) комплектация - string(15) цена - string(15)

Имя	Описание заказчика	Описание разработчика
		год - date
Каталог запчастей и аксессуаров	Данные о товаре Название, артикул, количество, цена, производитель	Название - string(50) артикул - string(50) количество - int64 цена - int64 производитель - string(50)
Каталог запчастей (спр_зап)	Информацию о товаре Таблица от производителя	Таблица
Накладная о покупке	Данные о накладной: Номер, дата, название товара, цена, количество, сотрудник	Номер - string(50) дата - date название товара - string(50) цена - int64 количество - int64 сотрудник - int64
ОтчетSQL		из сервера БД получится ответ в виде SQL
Отчеты	Отчет о продажах Вид: Заголовок, Таблица, Сумма Отчет о заказах Вид: Заголовок, Таблица, Сумма Отчет о расходах запчастей и материалов Вид: Заголовок, Таблица, Количество, Сумма Отчет об оказанных услугах Вид: Заголовок, Таблица, Сумма	Необходимые функции: "Сохранить в", "Печать". На форме: Надпись с название отчета, Таблица с данными, Кнопки "Сохранить", «Печать"
Оформление заказа	Данные об автомобиле и клиенте Фамилию, имя, отчество, дату рождения, паспортные данные, Марка, цвет, комплектация, цена,	Марка - string(15) цвет - string(15) комплектация - string(15) цена - string(15) год - date ФИО string(50) ДатаРождения - date Серия№Паспорта -

Имя	Описание заказчика	Описание разработчика
	год	string(15) Адрес - string(100) Телефон - string(15) №ВодУдостоверения - string(50)
Полис страхования	Данные о полюсе: Серия - номер, тип, вид.	Серия - номер - string(20) тип - string(30) вид - string(30)

Выводы по разделу три

Разработана структура информационной системы, выполнена контекстная диаграмма, поставлены задачи приложения, выполнена структура приложения.

4 РАЗРАБОТКА БАЗЫ ДАННЫХ

4.1 Таблицы

Для разработки базы данных мною была выбрана система управления базами данных MySQL/MariaDB. Создадим все необходимые таблицы.

Подробное описание полей в создаваемых таблицах представлены в таблице 3.

Таблица 3 – Описание структуры создаваемых таблиц

Таблица	Поля
Товар	КодТовара НазваниеТовара АртикулТовара КодПоставщика Тип(запчасть, комплектующая)
Поставщик	КодПоставщика НаименованиеПоставщика АдресПоставщика ИННПоставщика БанкПоставщика БанковскийСчетПоставщика ТелефонПоставщика
Сотрудник	КодСотрудника ФамилияСотрудника ИмяСотрудника ОтчествоСотрудника ДатаРожденияСотрудника ПолСотрудника СерияНомерПаспортаСотрудника ИННСотрудника

Таблица	Поля
	СНИЛС Сотрудника Дата Приема Сотрудника Дата Увольнения Сотрудника Код Должности Код Отдела Коэффициент Телефон Сотрудника E-mail Сотрудника Фотография Сотрудника
Отдел	Код Отдела Название Отдела Телефон Отдела
Должность	Код Должности Название Должности Оклад Должности
Аутентификация	Код Аутентификации Код Сотрудника Логин Пароль NFS_Метка
Автомобиль	Код Автомобиля Код Модели VIN Автомобил Код Цвета Дата Выпуска Автомобил Цена Автомобил Код Поставщика
Модель	Код Модели

Таблица	Поля
	НазваниеМодели КомплектацияМодели КППМодели ДвигательМодели ТипКузоваМодели
Клиент	КодКлиента ФИОКлиента ДатаРожденияКлиента СерияНомерПаспортаКлиента КодГорода КодАдреса Дом_КвартираКлиента ТелефонКлиента ВодительскоеУдостоверениеКлиента
Город	КодГорода НазваниеГорода ОбластьГорода РеспубликаГорода
Адрес	КодАдреса ТипАдрес (улица,проспект,переулок,...) НазваниеАдрес КодГорода
Услуги	КодУслуги НазваниеУслуги СрокГарантииУслуги
Техническое Обслуживание	КодТО НомерТО КодУслуги

Таблица	Поля
	КодМодели
Расход	КодРасход КодТовара КодНакладной Количество Цена
Приход	КодПриход КодТовара КодНакладной Количество Цена
Накладная	КодНакладной Дата Тип КодСотрудника
Работы	КодРаботы НазваниеРабот ОписаниеРабот Ед.Из. Работ КодСотрудника КоличествоРабот ЦенаРабот

На примере рассмотрим SQL- код создания таблицы Адрес:

4.2 Триггеры

Для ведения журнала выполняемых операций (добавление, изменение, удаление) воспользуемся «Триггерами». Триггер — это хранимая процедура,

которая не вызывается непосредственно, а исполняется при наступлении определенного события.

На примере рассмотрим код триггера для таблицы Адрес, приведенный в листинге 1.

Листинг 1 – Триггер для таблицы Адрес

```
CREATE TRIGGER `avtosalon_db`.`address_AFTER_INSERT` AFTER INSERT ON `address` FOR EACH ROW
BEGIN
INSERT INTO operation (idEmployees, OperationDate, OperationType) VALUES(@id_empl, NOW(),
'добавление адреса');
END
CREATE DEFINER=`root`@`localhost` TRIGGER `avtosalon_db`.`address_AFTER_UPDATE` AFTER UPDATE
ON `address` FOR EACH ROW
BEGIN
INSERT INTO operation (idEmployees,OperationDate,OperationType) VALUES(@id_empl,
NOW(), 'изменение адреса');
END
CREATE DEFINER=`root`@`localhost` TRIGGER `avtosalon_db`.`address_AFTER_DELETE` AFTER DELETE
ON `address` FOR EACH ROW
BEGIN
INSERT INTO operation (idEmployees,OperationDate,OperationType) VALUES(@id_empl,
NOW(), 'удаление адреса');
END
```

4.3 Представления

Для корректного отображения информации созданы необходимые Представления:

- registrsale;
- view_address;
- view_arrivalgoods;
- view_authentication;
- view_car_ordered;
- view_car_sold;
- view_car_stock;
- view_city;
- view_client;
- view_color;
- view_consumables;
- view_department;

- view_employees;
- view_invoice;
- view_model;
- view_operation;
- view_position;
- view_provider;
- view_remont;
- view_salegoods;
- view_services;
- view_tehobs;
- view_tehobs_zapis;
- view_tovar;
- view_work.

На примере рассмотрим SQL – код создания «Представления» для таблицы Адрес, приведенный в листинге 2.

Листинг 2 – «Представления» для таблицы Адрес

```
CREATE
  ALGORITHM = UNDEFINED
  DEFINER = `root`@`localhost`
  SQL SECURITY DEFINER
VIEW `view_address` AS
  SELECT
    `address`.`idAddress` AS `idAddress`,
    `address`.`AddressType` AS `AddressType`,
    `address`.`AddressName` AS `AddressName`,
    `city`.`CityName` AS `CityName`
  FROM
    (`address`
  LEFT JOIN `city` ON ((`address`.`idCity` = `city`.`idCity`)))
```

4.4 Функции и процедуры

Для «облегчения» работы разрабатываемого приложения, мною было решено перенести большую часть функций и процедур на сервер MySQL, а именно (на примере данных таблицы Адрес):

Процедура добавления данных

Код процедуры добавления данных приведен в листинге 3.

Листинг 3 – процедура добавления данных

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `ADD_Adress`(IN Type_new nvarchar(45), IN Name_new nvarchar(45), IN id int)
BEGIN
SET @id_empl = id;
INSERT INTO address (AddressType,AddressName) VALUES(Type_new,Name_new);
END
```

Процедура изменения данных.

Код процедуры изменения данных приведен в листинге 4.

Листинг 4 – процедура изменения данных

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `UPD_Adress`(IN idAddress_new int,IN Type_new nvarchar(45), IN Name_new nvarchar(45), IN id int)
BEGIN
SET @id_empl = id;
update address set AddressType = Type_new,AddressName=Name_new where idAddress = idAddress_new;
END
```

Процедура удаления данных.

Код процедуры удаления данных приведен в листинге 5.

Листинг 5 – процедура изменения данных

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `DEL_Adress`(IN idAddress_new int, IN id int)
BEGIN
SET @id_empl = id;
DELETE FROM address where idAddress = idAddress_new;
END
```

В результате получаем ER-диаграмму, представленную на рисунке 17 – ER диаграмма создаваемой базы данных:

Выводы по разделу четыре

Разработана ER-диаграмма, разработана база данных.

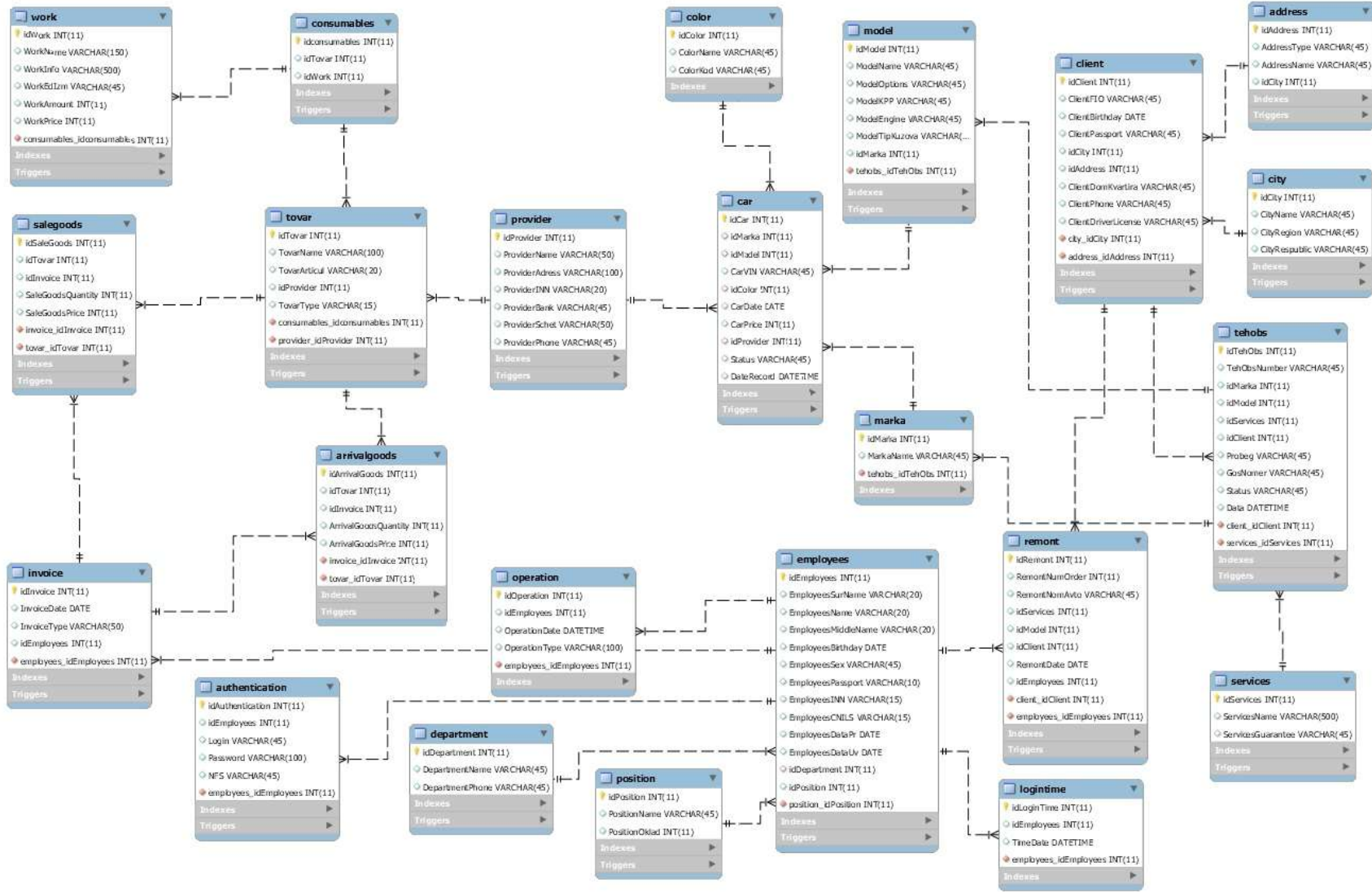


Рисунок 17 – ER диаграмма создаваемой базы данных

5 РАЗРАБОТКА ПРИЛОЖЕНИЯ

5.1 Среда разработки

Для разработки приложения выбрана среда Microsoft Visual Studio Enterprise 2017. Языком для разработки выбран объектно-ориентированный язык программирования C#.

5.2 Дизайн программы и краткое описание компонентов

Для удобства визуального восприятия разработаем шаблон компоновки элементов при котором 80% формы занимает таблица, а 20% необходимые функции.

Модуль «Автомобили»

- 1 Вкладка «В наличии» отображает все автомобили, находящиеся на продаже.
- 2 Вкладка Заказ отображает все автомобили, которые заказал автосалон.
- 3 Вкладка Проданные отображает все автомобили проданные автосалоном.
- 4 Вкладка «Добавить» отображает все элементы, требующиеся для добавления нового автомобиля.
- 5 Вкладка «Редактировать» отображает все элементы выбранной записи допустимые для редактирования.

5.3 Диаграмма классов

На рисунке 18 – Диаграмма классов, мы видим получившуюся диаграмму после разработки приложения.

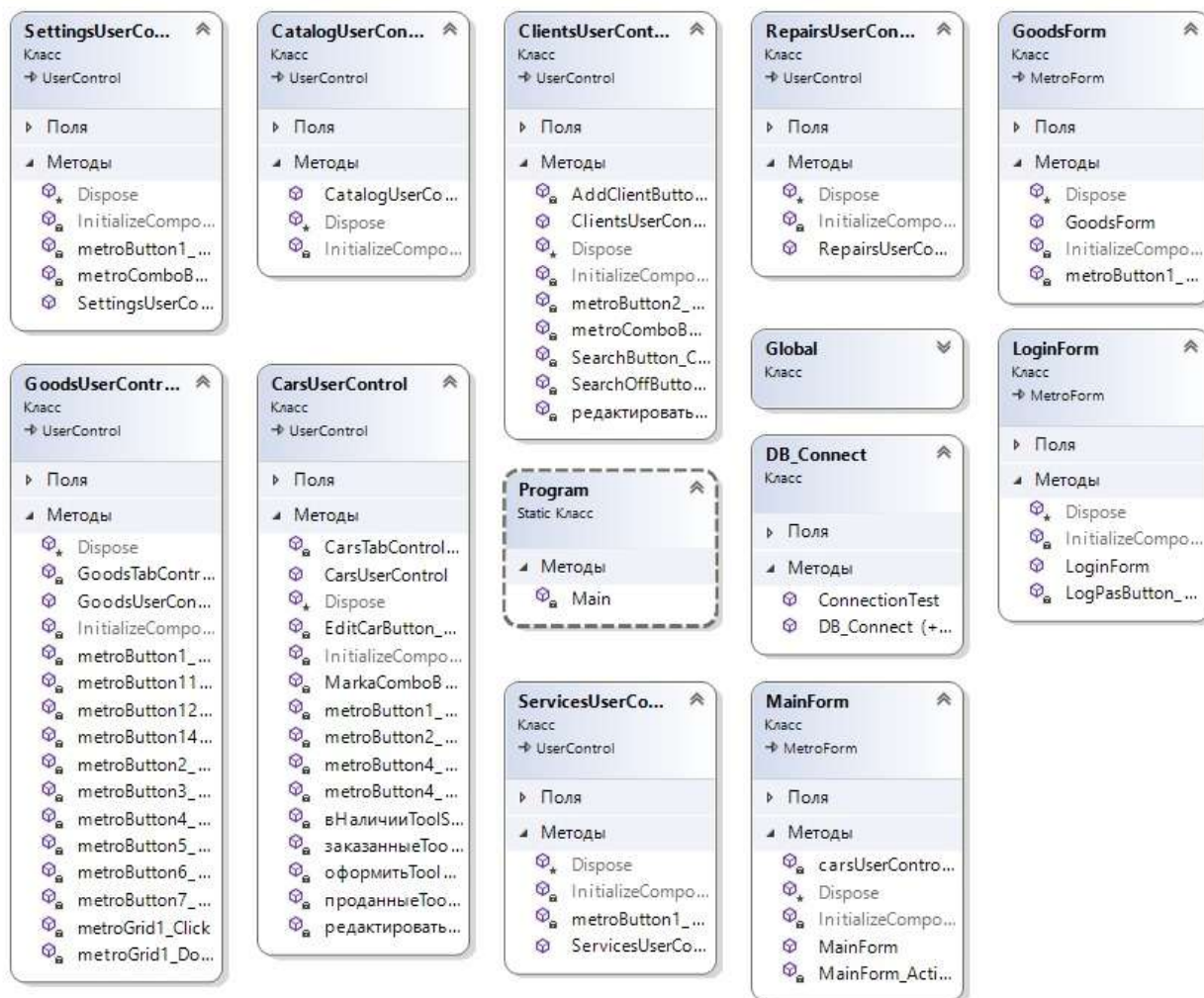


Рисунок 18 – Диаграмма классов

5.4 Методы классов

На примере рассмотрим самые интересные методы этих классов.

Метод `DB_Connect` нужен для создания подключения к базе данных. Конструктор имеет две перегрузки:

- без аргументов – служит для автоматического входа в систему, необходимые данные берутся из *.log файла;
- с аргументом – служит для настройки подключения к базе данных, данные берутся из полей на форме.

Код метода `DB_Connect` приведен в листинге 6.

Листинг 6 – метод DB_Connect

```
public DB_Connect()
{
    constr = "Server=" + DBServer + ";"
           + "Database=" + DBName + ";"
           + "Uid=" + DBUser + ";"
           + "Pwd=" + DBPass + ";"
           + "CharSet = cp1251; ";
    avtosalon_dbConnection = new MySqlConnection(constr);
    try
    {
        avtosalon_dbConnection.Open();
    }
    catch (MySqlException e)
    {
        MessageBox.Show("Нет подключения к серверу" + e.Message);
    }
}

public DB_Connect(string con_str)
{
    avtosalon_dbConnection = new MySqlConnection(con_str);
    try
    {
        avtosalon_dbConnection.Open();
        MessageBox.Show("Статус подключения: Успешно");
    }
    catch (MySqlException e)
    {
        MessageBox.Show("Нет подключения к серверу" + e.Message);
    }
}
```

При загрузке программы автоматически происходит проверка подключения к базе данных при помощи метода `ConnectionTest()`.

Код метода `ConnectionTest()` приведен в листинге 7.

Листинг 7 – метода ConnectionTest()

```
public void ConnectionTest()
```

```

{
    string st = Convert.ToString(avtosalon_dbConnection.State);
    if (Convert.ToString(avtosalon_dbConnection.State) == "Open")
    {
        Global.message = "Статус подключения: Успешно";
        Global.caption = "Подключение к серверу";
        var result = MessageBox.Show(Global.message, Global.caption,
            MessageBoxButtons.OK,
            MessageBoxIcon.Asterisk);
    }
    else
    {
        Global.message = "Статус подключения: Ошибка";
        Global.caption = "Подключение к серверу";
        var result = MessageBox.Show(Global.message, Global.caption,
            MessageBoxButtons.OK,
            MessageBoxIcon.Error);
    }
}
}

```

Для аутентификации сотрудников в программе написан метод проверки соответствия введённых данных, данным из БД.

Код метод аутентификации сотрудников приведен в листинге 8.

Листинг 8 – аутентификация сотрудников

```

private void LogPasButton (object sender, EventArgs e)
{
    try
    {
        queriesTableAdapter1.LogPasCheck(LoginTextBox.Text, PasswordTextBox.Text, out
NFS);

        Global.message = "Ввидите ключ";
        Global.caption = "Инициализация ключа";
        var result = MessageBox.Show(Global.message, Global.caption,
            MessageBoxButtons.OK,
            MessageBoxIcon.Asterisk);

        if (result == DialogResult.OK)//if (NFS != "")
        {
            Global.ID_Empl=LoginTextBox.Text;
            this.Close();
        }
    }
}

```

```

    }
    catch
    {
        Global.message = "Повторите пожалуйста ввод или обратитесь к системному
администратору";
        Global.caption = "Ошибка инициализации";
        var result = MessageBox.Show(Global.message, Global.caption,
                                    MessageBoxButtons.OK,
                                    MessageBoxIcon.Exclamation);
    }
}

```

Внешний вид формы для аутентификации представлен на рисунке 19 – Форма аутентификации

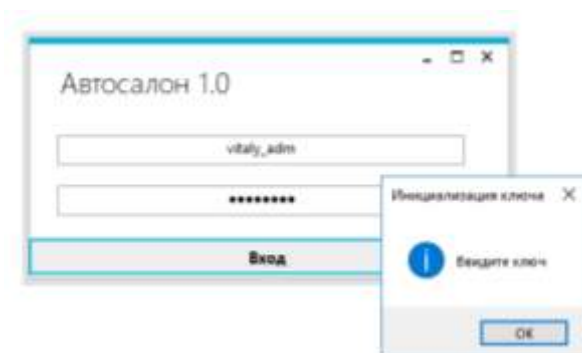


Рисунок 19 – Форма аутентификации

Для удобства написания и сопровождения программного продукта, воспользуемся компонентами UserControl для создания интерфейса программы. На примере рассмотрим CarsUserControl (интерфейс для работы с автомобилями).

Шаблон компоновки представлен на рисунке 20 – Форма «Автомобили»

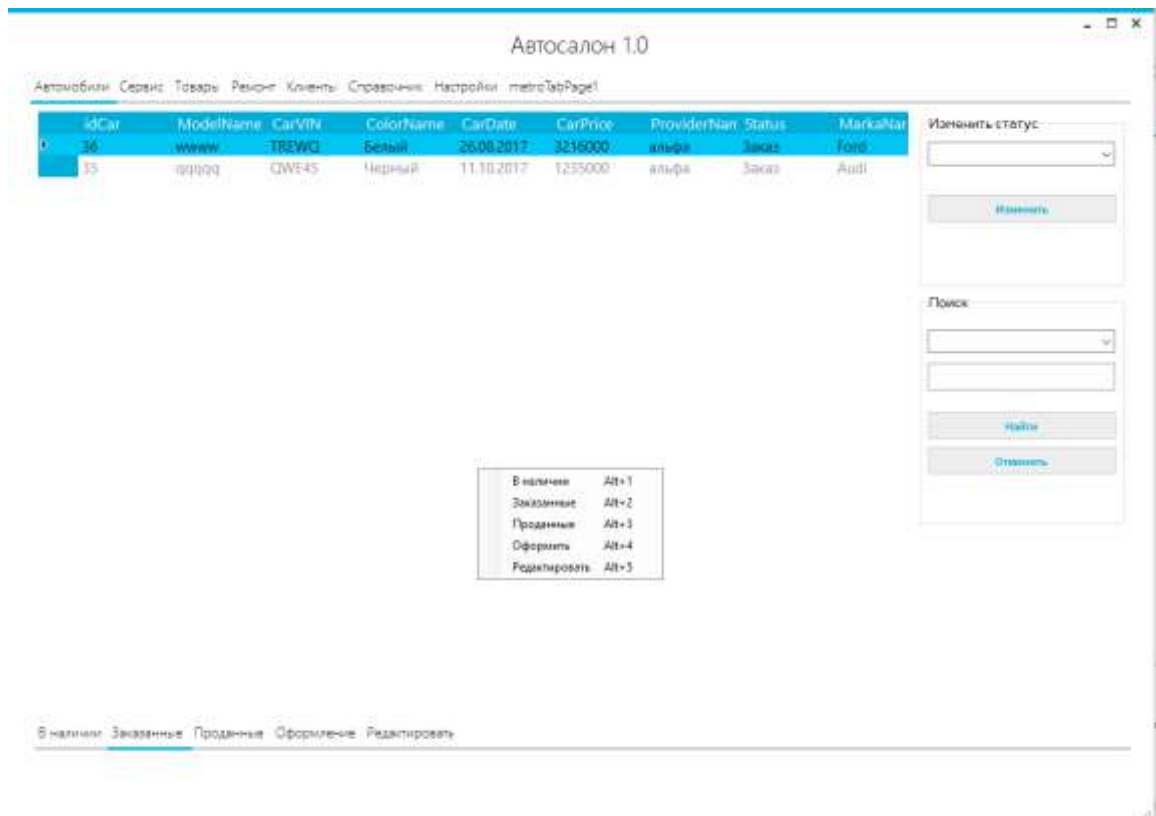


Рисунок 20 – Форма «Автомобили»

На форме располагается несколько вкладок: в наличии, заказанные, проданные, добавление, редактирование. Для удобства работы на вкладках: в наличии, заказанные, проданные добавлена возможность изменения статуса автомобиля – так как это очень экономит время при работе в программе. Также на этих вкладках организован поиск и фильтрация по таблице, переход между вкладками по контекстному меню и с помощью «горячих клавиш».

Код метода добавления записи приведен в листинге 9.

Листинг 9 – метод добавления записи

```
private void CarsADD(object sender, EventArgs e)
{
    queriesTableAdapter1.ADD_Car(
        Convert.ToInt32(ModelGrid.CurrentRow.Cells[0].Value),
        VINTextBox.Text,
        Convert.ToInt32(ColorGrid.CurrentRow.Cells[0].Value),
        Convert.ToDateTime(CarDateTime.Text),
        Convert.ToInt32(PriceTextBox.Text),
        Convert.ToInt32(ProviderGrid.CurrentRow.Cells[0].Value),
        Convert.ToInt32(1),
        Convert.ToInt32(MarkaComboBox.SelectedValue),
```

```

        StatusComboBox.Text,
        Convert.ToDateTime(RecordDateTime.Text));
    }

```

В методе вызывается хранимая процедура, аргументом в которую передаются введённые/выбранные данные с формы.

Код метода редактирования записи приведен в листинге 10.

Листинг 10 – метод редактирования записи

```

private void редактироватьToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (CarsTabControl.SelectedTab.Text == "В наличии")
        tr = (DataRowView)viewcarstockBindingSource.Current;
    if (CarsTabControl.SelectedTab.Text == "Заказанные")
        tr = (DataRowView)viewcarorderedBindingSource.Current;
    if (CarsTabControl.SelectedTab.Text == "Проданные")
        tr = (DataRowView)viewcarsoldBindingSource.Current;
    id = (int)tr["idCar"];
    carBindingSource.Filter = "idCar =" + Convert.ToString(tr["idCar"]);
    modelBindingSource.Position = modelBindingSource.Find("idModel",
Convert.ToInt32(idModelTextBox.Text));
    markaBindingSource.Position = markaBindingSource.Find("idMarka",
Convert.ToInt32(idMarkaTextBox.Text));
    colorBindingSource.Position = colorBindingSource.Find("idColor",
Convert.ToInt32(idColorTextBox.Text));
    providerBindingSource.Position = providerBindingSource.Find("idProvider",
Convert.ToInt32(idProviderTextBox.Text));
    EditStatus.Text = CarsTabControl.SelectedTab.Text;
    CarsTabControl.SelectedTab = CarsTabControl.TabPages["EditCarTabPage"];
}

```

В зависимости от родительской вкладки, происходит выбор источника данных и формируется редактируемая строка. Далее вызывается хранимая процедура, аргументом в которую передаются скорректированные данные с формы.

Код метода редактирования записи приведен в листинге 11.

Листинг 11 – метод редактирования записи

```

private void EditCarButton_Click(object sender, EventArgs e)
{
    queriesTableAdapter1.UPD_Car(
        Convert.ToInt32(metroComboBox1.SelectedValue),

```

```

        id,
        Convert.ToInt32(EditModelGrid.CurrentRow.Cells[0].Value),
        carVINTextBox.Text,
        Convert.ToInt32(EditColorGrid.CurrentRow.Cells[0].Value),
        Convert.ToDateTime(carDateDateTimePicker.Text),
        Convert.ToInt32(carPriceTextBox.Text),
        Convert.ToInt32(EditProviderGrid.CurrentRow.Cells[0].Value),
        Convert.ToInt32(1),
        EditStatus.Text,
        Convert.ToDateTime(dateRecordDateTimePicker.Text)
    );
}

```

Код метода поиска записи приведен в листинге 12.

Листинг 12 – метод поиска записи

```

private void SearchButton_Click(object sender, EventArgs e)
{
    string sear = "";
    switch (SearchBox.SelectedIndex)
    {
        case 1:
            sear = "ClientFIO = '" + SearchTextBox.Text + "'";
            break;
        case 2:
            sear = "ClientBirthday = '" + SearchTextBox.Text + "'";
            break;
        case 3:
            sear = "ClientPassport = '" + SearchTextBox.Text + "'";
            break;
        case 4:
            sear = "CityName = '" + SearchTextBox.Text + "'";
            break;
        case 5:
            sear = "AddressName = '" + SearchTextBox.Text + "'";
            break;
        case 6:
            sear = "ClientPhone = '" + SearchTextBox.Text + "'";
            break;
        default:
            break;
    }
    viewclientBindingSource.Filter = sear;
}

```


Оформление продажи товаров. Менеджер выбирает из списка необходимый товар, после чего происходит открытие новой формы с остатком на складе и ценой выбранного товара, после чего менеджер вводит в поле необходимое количество и нажимает «Добавить».

Форма «Перечень товаров» с вводом количества для продажи представлена на рисунке 21.

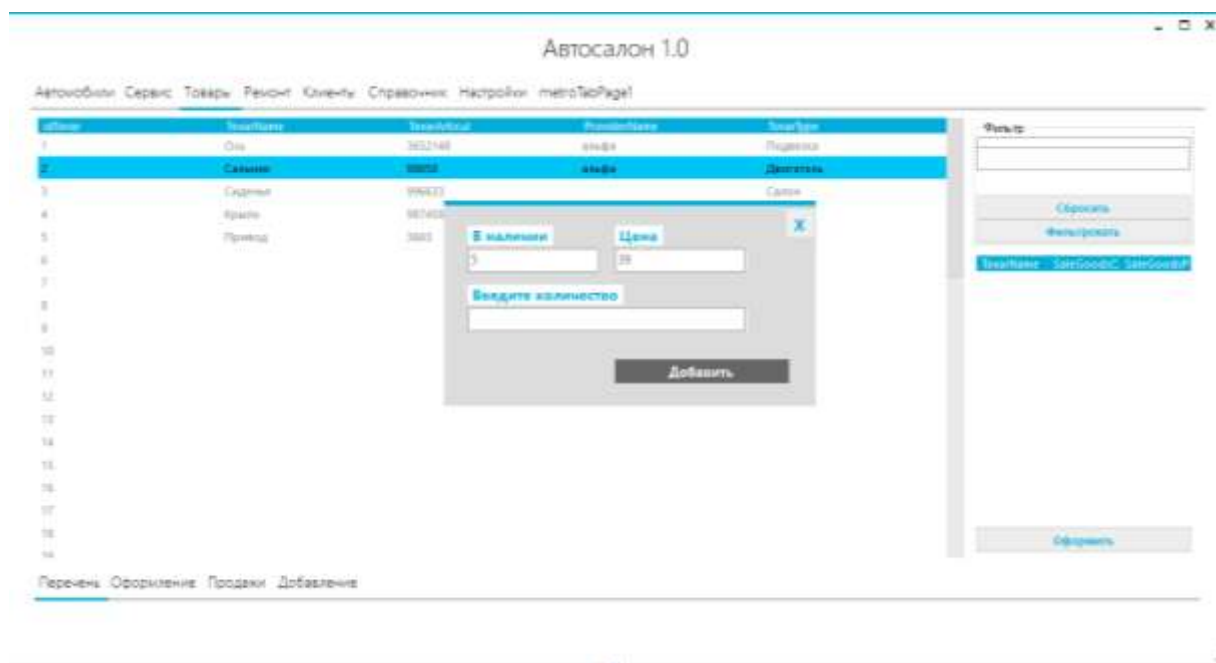


Рисунок 21 – Форма «Перечень товаров» с вводом количества для продажи

Код выбора товара приведен в листинге 14.

Листинг 14 – выбор товара

```
private void metroGrid_DoubleClick(object sender, EventArgs e)
{
    DataRowView tr = (DataRowView)viewtovarBindingSource.Current;
    int id = (int)tr["idTovar"];
    GoodsForm Goods = new GoodsForm(id);
    Goods.ShowDialog();
}
public GoodsForm(int id)
{
    id_goods = id;
    InitializeComponent();
    queriesTableAdapter1.GoodsQuantityPrice(id_goods, out Quantity, out Price);
    metroTextBox2.Text = Price;
}
```

```

        metroTextBox1.Text = Quantity;
    }
private void metroButton1_Click(object sender, EventArgs e)
{
    if (Convert.ToInt32(Quantity) >= Convert.ToInt32(metroTextBox3.Text))
    {
        queriesTableAdapter1.ADD_Salegoods(id_goods,          Convert.ToInt32(id_employee),
Convert.ToInt32(metroTextBox3.Text),                          Convert.ToInt32(metroTextBox2.Text),
Convert.ToInt32(id_employee));
        Form f = new Form();
        Close();
    }
    else
    {
        Global.message = "Выбранное количество больше остатка";
        Global.caption = "Добавление";
        var result = MessageBox.Show(Global.message, Global.caption,
                                     MessageBoxButtons.OK,
                                     MessageBoxIcon.Asterisk); } }

```

Пока менеджер не оформил закрытие накладной, все товары записываются в накладную под номером равным коду менеджера. Это даёт возможность параллельно оформлять всем менеджерам продажи – что очень экономит время клиентов.

Код закрытия накладной приведен в листинге 15.

Листинг 15 – закрытие накладной

```

private void CloseInvoice_Click(object sender, EventArgs e)
{
    queriesTableAdapter1.ADD_Invoice(Convert.ToDateTime(DateTime.Today), "Продажа",
Convert.ToInt32(id_employee));
    queriesTableAdapter1.last_inv(out i_new);
    queriesTableAdapter1.UPD_Inv_Sale(Convert.ToInt32(id_employee),
Convert.ToInt32(i_new));
    metroTextBox3.Text = i_new;
}

```

Выводы по разделу пять

Выбраны программные средства, разработана структура приложения.

6 РАЗРАБОТКА WEB-ИНТЕРФЕЙСА

Для отображения справочной информации созданы информационные страницы(*.html):

- Главная страница для выбора необходимой информации;
- Страница с информацией об автомобилях;
- Страница с информацией о товарах;
- Страница с информацией об услугах;
- Веб-кабинет.

На рисунке 22 представлена главная форма Web-интерфейса.

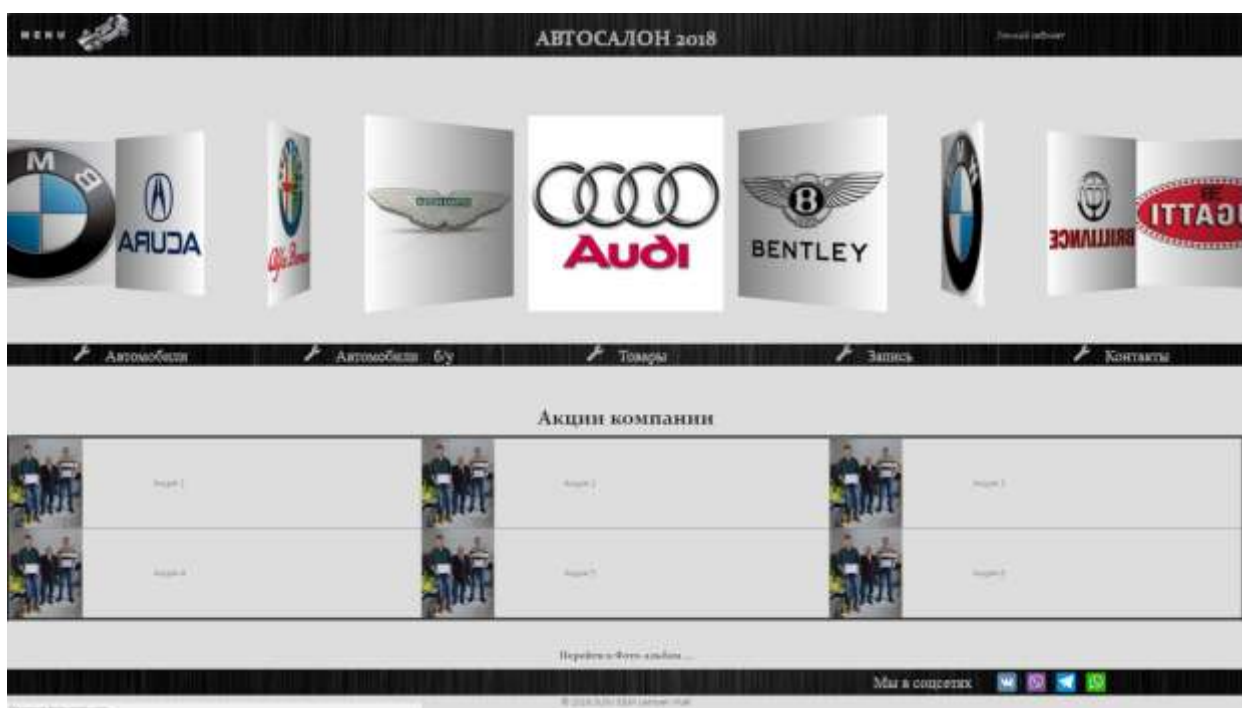


Рисунок 22 – Главная страница для выбора необходимой информации

Для работы с БД из веб-приложения воспользуемся ORM- библиотекой под названием RedBeanPHP. Это ORM слой для PHP 5, создает и поддерживает таблицы на лету, с открытым исходным кодом и BSD - лицензией.

PHP код для подключения библиотеки и установки данных для подключения к БД MYSQL/MariaDB приведен в листинге 16.

Листинг 16 – подключение библиотеки и установка данных для соединения

```

<?php
require'libs/rb-mysql.php';
R::setup( 'mysql:host=localhost;dbname=avtosalon_db','root', '1234' );
R::freeze( true );
?>

```

Для вывода информации о товарах (рисунок 23) создадим PHP страницу, код приведен в листинге 17.

Листинг 17 – вывод информации о товарах

```

<?php
require "libs/db.php";
if ( isset( $_POST[ 'do_next' ] ) and $_POST[ 'do_next' ]<$count = R::count( 'viewgoods' ) )
    {
        $min=$_POST[ 'do_next' ]+100;
        $goods = R::findCollection( 'viewgoods', 'JOIN (select idtovar from viewgoods order by
idtovar limit ?,?) AS b ON b.idtovar = viewgoods.idtovar', [ $min, 100 ] );
    }

    if(isset( $_POST[ 'do_last' ] ) and $_POST[ 'do_last' ]>0 ) {
        $min=$_POST[ 'do_last' ]-100;
        $goods = R::findCollection( 'viewgoods', 'JOIN (select idtovar from tovar order by
idtovar limit ?,?) AS b ON b.idtovar = viewgoods.idtovar', [ $min, 100 ] );
    }
if ( !isset( $_POST[ 'do_next' ] ) and !isset( $_POST[ 'do_last' ] )) {
    $goods = R::findCollection( 'viewgoods', 'JOIN (select idtovar from tovar order by
idtovar limit ?,?) AS b ON b.idtovar = viewgoods.idtovar', [ 0, 100 ] );
}
if ( isset( $_POST[ 'do_search' ] ) and $_POST[ 'Group' ]=='name' ) {
    $goods = R::findCollection( 'viewgoods', 'where name Like ?', [ "%".$_POST[ 'TextSearch'
] . "%" ] );
}
if ( isset( $_POST[ 'do_search' ] ) and $_POST[ 'Group' ]=='articul' ) {
    $goods = R::findCollection( 'viewgoods', 'where articul Like ?', [ "%" . $_POST[
'TextSearch' ] . "%" ] );
}
?>

<form action="goods.php" id="data" method="Post">
    <meta charset="utf-8">
    <font color="#2E2726" face="Constantia, Lucida Bright, DejaVu Serif, Georgia, serif"
size="" +1><strong> &nbsp; Найти&nbsp; &nbsp; &nbsp; </strong> </font>
    <label>
        <input type="radio" name="Group" value="name">
        Наименование</label>

```

```

<label>
  <input type="radio" name="Group" value="articul" >
  Артикул</label>
<br>
<input type="text" name="TextSearch" style="width:300px">

<button type="submit" name="do_search" style="align-content: center">Найти </button>

<br><br>
<table width="100%" border="1">
  <tbody>
    <tr height="30px">
      <th scope="col">#</th>
      <th scope="col">Наименование</th>
      <th scope="col">Артикул</th>
      <th scope="col">Группа</th>
      <th scope="col">Производитель</th>
    </tr>
    <? while( $item = $goods->next() ) {?>
    <tr height="25px">
      <td>
        <?=$item['idtovar']?>
      </td>
      <td>
        <?=$item['name']?>
      </td>
      <td>
        <?=$item['articul']?>
      </td>
      <td>
        <?=$item['group']?>
      </td>
      <td>
        <?=$item['manufacturer']?>
      </td>
    </tr>
    <?}??>
  </tbody>
</table><br>
<button type="submit" name="do_last" style=" align-content: center" value="<?=@$min?>"> <<
назад </button>

```

```

        <button type="submit" name="do_next" style="align-content: center"
value="<?=@$min?>"> >> далее </button>
</form>

```



Рисунок 23 – Страница с информацией о товарах

Для регистрации и входа в личный кабинет клиента, создадим следующие страницы (рисунок 24):

Вход

Ваш E-mail

Ваш пароль

Регистрация

Ваша Фамилия

Ваше Имя

Ваш E-mail

Ваш номер телефона

Ваш пароль

Повторите ваш пароль

Рисунок 24 – Страница для регистрации и входа в личный кабинет

Код страницы для регистрации и входа в личный кабинет приведен в листинге 18.

Листинг 18 – регистрация и вход в личный кабинет

```
<?php
require "libs/db.php";
$data = $_POST;
if ( isset( $data[ 'do_login' ] ) ) {
    $errors = array();
    $user = R::findOne( 'user', 'useremail = ?', array( $data[ 'LogEmail' ] ) );
    if ( $user ) {
        if ( password_verify( $data[ 'LogPass' ], $user->userpass ) ) {
            $_SESSION[ 'logget_user' ] = $user;
            echo "<div style=\"color: green; font-size:18\">Успешный вход<br/><a
href=\"/\>на главную</a></div><hr>";
        } else {
            $errors[] = "Пароль не верный";
        }
    } else {
        $errors[] = "Пользователь не найден";
    }
    if ( !empty( $errors ) ) {
        echo "<div style=\"color: red; font-size:18\">$errors[0]</div><hr>";
    }
}
if ( isset( $data[ 'do_signap' ] ) ) {
    $errors = array();
    if ( trim( $data[ 'SurnameUser' ] ) == '' ) {
        $errors[] = 'Введите Фамилию';
    }
    if ( trim( $data[ 'NameUser' ] ) == '' ) {
        $errors[] = 'Введите Имя';
    }
    if ( trim( $data[ 'Email' ] ) == '' ) {
        $errors[] = 'Введите Email';
    }
    if ( trim( $data[ 'Phone' ] ) == '' ) {
        $errors[] = 'Введите телефон';
    }
    if ( $data[ 'Pass1' ] == '' ) {
        $errors[] = 'Введите пароль';
    }
}
```

```

if ( $data[ 'Pass2' ] != $data[ 'Pass1' ] ) {
    $errors[] = 'Пароли не совпадают';
}
if ( R::count('user', "useremail = ?",array($data['Email']))>0 ) {
    $errors[] = 'Пользователь с таким Email уже существует';
}
if ( empty( $errors ) ) {
    $user = R::dispense( 'user' );
    $user->userf = $data[ 'SurnameUser' ];
    $user->useri = $data[ 'NameUser' ];
    $user->userphone = $data[ 'Phone' ];
    $user->useremail = $data[ 'Email' ];
    $user->userpass = password_hash($data[ 'Pass1' ],PASSWORD_DEFAULT);
    $id = R::store( $user );
    echo "<div style=\"color: green; font-size:18\">Успешная регистрация</div><hr>";
} else {
    echo "<div style=\"color: red; font-size:18\">$errors[0]</div><hr>";
}
}
?>
<meta charset="utf-8">
<form action="login.php" method="POST">
<table width="100%" height="450px" border="0">
<tbody>
<tr>
<td style="border-right-width:3px;border-right-style:solid;border-right-color:#2E2726;">
<head>
<title>Войти</title>
</head>
<font color="#2E2726" face="Constantia, Lucida Bright, DejaVu Serif, Georgia, serif"
size="+3"><strong>Вход </strong> </font><br>

<p>
<font color="#2E2726" face="Constantia, Lucida Bright, DejaVu Serif, Georgia,
serif" size="" +1><strong> Ваш E-mail</strong> </font><br>
</p>
<input type="email" name="LogEmail" style="width: 200px">
<p>
<font color="#2E2726" face="Constantia, Lucida Bright, DejaVu Serif, Georgia,
serif" size="" +1><strong> Ваш пароль</strong> </font><br>
</p>
<input type="password" name="LogPass" style="width: 200px">

```



```

        <p><button type="submit" name="do_login" style="align-content: center">Войти</button>
    </p>
</td>
<td>
    <font color="#2E2726" face="Constantia, Lucida Bright, DejaVu Serif, Georgia, serif"
size="+3"><strong>Регистрация</strong> </font><br>
    <p>
        <strong> Ваша Фамилия</strong><br>
        <input type="text" name="SurnameUser" style="width: 200px" value="<?php echo
@$data['SurnameUser']; ?>">
    </p>
    <p>
        <strong> Ваше Имя</strong><br>
        <input type="text" name="NameUser" style="width: 200px" value="<?php echo
@$data['NameUser']; ?>">
    </p>
    <p>
        <strong> Ваш E-mail</strong><br>
        <input type="email" name="Email" style="width: 200px" value="<?php echo
@$data['Email']; ?>">
    </p>
    <p>
        <strong> Ваш номер телефона </strong><br>
        <input type="text" name="Phone" style="width: 200px" value="<?php echo
@$data['Phone']; ?>">
    </p>
    <p>
        <strong> Ваш пароль</strong><br>
        <input type="password" name="Pass1" style="width: 200px" value="<?php echo
@$data['Pass1']; ?>">
    </p>
    <p>
        <strong> Повторите ваш пароль</strong><br>
        <input type="password" name="Pass2" style="width: 200px">
    </p>
    <p>
        <button type="submit" name="do_signap" style="width: 150px">Зарегистрироваться
</button>
    </p>
</td>
</tr>
</tbody>

```

</table>

</form>

Создадим «Веб-кабинет» для реализации необходимых требований. На рисунке 25 можно посмотреть внешний вид личного кабинета пользователя.

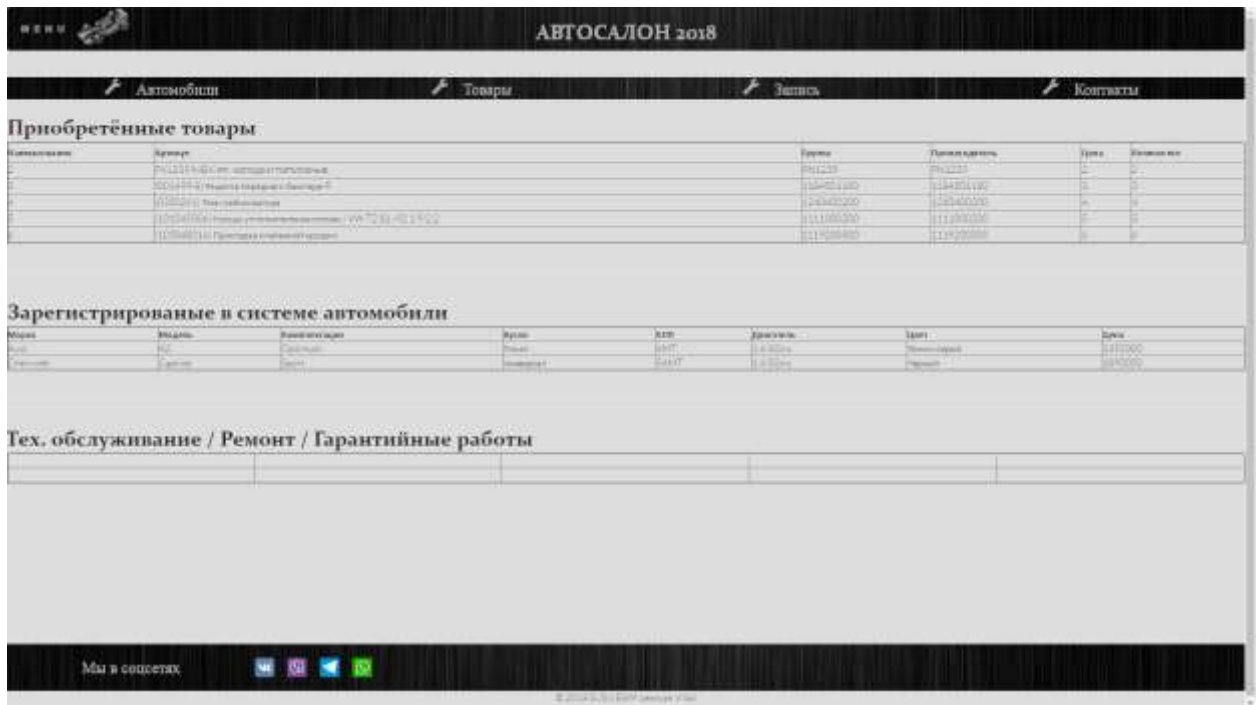


Рисунок 25 – Личный кабинет клиента

Рассмотрим подачу online заявки на прохождение ТО (рисунок 26).

formula/online.php

Онлайн заявка

Фамилия Имя Отчество

Выберите дату

Выберите марку

Ваш E-mail

Ваш номер телефона

Комментарий

Отправить

Рисунок 26 – Форма подачи заявки на ТО через личный кабинет

Код метода подачи заявки на ТО через личный кабинет приведен в листинге 19.

Листинг 19 – регистрация и вход в личный кабинет

```
<?php
require "libs/db.php";
$data = $_POST;
if ( isset( $data[ 'do_online' ] ) ) {

    $onl = R::dispense( 'recordonline' );
    $onl->fio = $data[ 'FIO' ];
    $onl->daterecord = $data[ 'Date' ];
    $onl->idmarka = $_POST[ 'Marka' ];
    $onl->phone = $data[ 'Phone' ];
    $onl->email = $data[ 'Email' ];
    $onl->comment = $data[ 'Comm' ];
    $id = R::store( $onl );
    echo "<div style=\"color: green; font-size:18\">Успешно</div><hr>";
}
?>
```

```

<meta charset="utf-8">
<form action="online.php" method="POST" style="padding-left: 25px" id="data">
    <font color="#2E2726" face="Constantia, Lucida Bright, DejaVu Serif, Georgia, serif"
size="+3"><strong>Онлайн заявка </strong></font><br>
    <p>
        <strong> Фамилия Имя Отчество </strong><br>
        <input type="text" name="FIO" style="width: 200px" value="<?php echo
@<data['FIO']; ?>" required>
    </p>
    <p>
        <strong> Выберите дату </strong><br><input type="date" name="Date" value="<?php
echo @<data['Date']; ?>" required>
    </p>
    <p>
        <strong> Выберите марку </strong><br>
        <select name="Marka" form="data" onFocus="<?php $mar = R::findCollection(
'marka'); ?>" required>
            <option value="0"></option>
            <?while( $item = $mar->next() ) {?>
            <option value="<?=@<item['idmarka']?>">
                <?=<item['markaname']?>
            </option>
            <?} ?>
        </select>
    </p>
    <p>
        <strong> Ваш E-mail</strong><br>
        <input type="email" name="Email" style="width: 200px" value="<?php echo
@<data['Email']; ?>" required>
    </p>
    <p>
        <strong> Ваш номер телефона </strong><br>
        <input type="tel" name="Phone" style="width: 200px" value="<?php echo
@<data['Phone']; ?>" required>
    </p>
    <p>
        <strong> Комментарий </strong><br>
        <textarea name="Comm" value="<?php echo @<data['Comm']; ?>" maxlength="250">
</textarea>
    </p>
    <p>
        <button type="submit" name="do_online" style="width: 150px">Отправить </button>

```

</p>
</form>

Выводы по разделу шесть

Выбраны программные средства, разработана структура web-приложения.

7 РАЗРАБОТКА ТЕСТОВ ДЛЯ ПРИЛОЖЕНИЯ

Для проверки работоспособности и корректных вычислений напишем тесты для основных задач:

7.1 Разработка закодированного теста пользовательского интерфейса

Оформление покупки товаров (рисунок 27).

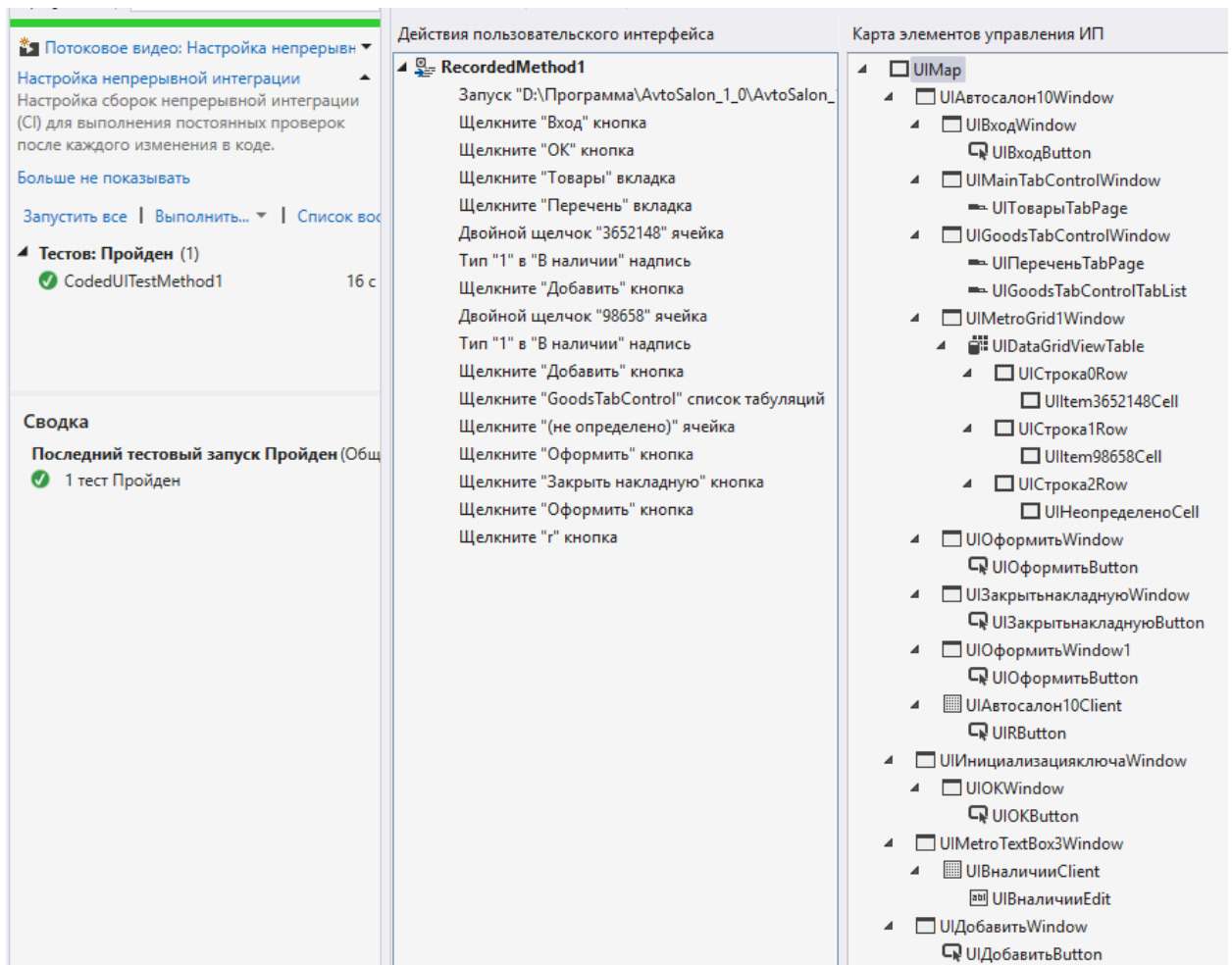


Рисунок 27 – Последовательность закодированного теста

Как мы видим тест прошёл успешно, в дальнейшем этот тест можно применять как обучающее пособие для новых сотрудников, работающих в этом приложении.

7.2 Разработка модульного теста

Код модульного теста приведен в листинге 20.

Листинг 20 – модульный тест

```
[TestClass]
public class UnitTest1
{
    public MySqlConnection avtosalon_dbConnection;
    public DataSet avtosalon_dbDataSet;
    private string DBServer = "localhost";
    private string DBName = "avtosalon_db";
    private string DBUser = "root";
    private string DBPass = "1234";
    private string constr;
    [TestMethod]
    public void TestMethod1()
    {
        constr = "Server=" + DBServer + ";"
            + "Database=" + DBName + ";"
            + "Uid=" + DBUser + ";"
            + "Pwd=" + DBPass + ";"
            + "CharSet = cp1251; ";
        avtosalon_dbConnection = new MySqlConnection(constr);
        var MyCmd = new MySqlCommand("ADD_Car", avtosalon_dbConnection);
        MyCmd.CommandType = CommandType.StoredProcedure;
        MyCmd.Parameters.AddWithValue("@idModel_new", Convert.ToInt32(1));
        MyCmd.Parameters.AddWithValue("@CarVIN_new", "1");
        MyCmd.Parameters.AddWithValue("@idColor_new", Convert.ToInt32(1));
        MyCmd.Parameters.AddWithValue("@CarDate_new", Convert.ToDateTime(DateTime.Today));
        MyCmd.Parameters.AddWithValue("@CarPrice_new", "1");
        MyCmd.Parameters.AddWithValue("@idProvider_new", Convert.ToInt32(1));
        MyCmd.Parameters.AddWithValue("@id", Convert.ToInt32(1));
        MyCmd.Parameters.AddWithValue("@idMarka_new", Convert.ToInt32(1));
        MyCmd.Parameters.AddWithValue("@Status_new", "Проверка");
        MyCmd.Parameters.AddWithValue("@DateRecord_new",
Convert.ToDateTime(DateTime.Today));
```

```
avtosalon_dbConnection.Open();  
MyCmd.ExecuteNonQuery();  
}  
}
```

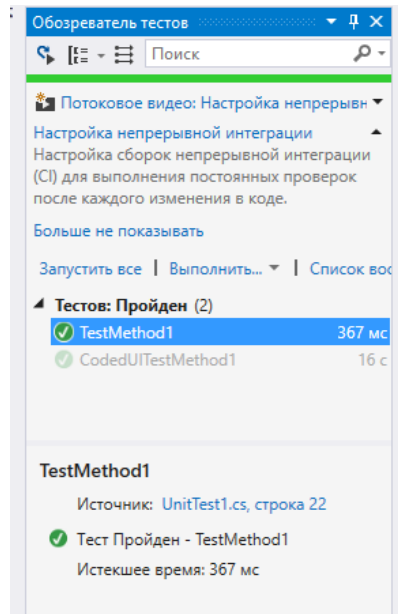


Рисунок 28 – Выполнение теста

Как мы видим (рисунок 28) тест выполнен успешно. Для демонстрации корректной работы разработанного продукта можно показать тесты всех программных модулей.

Выводы по разделу семь

Проведены тесты приложения, выпущен релиз.

ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной работы были поставлены цели и задачи, обоснована актуальность и необходимость выбранной темы. Также были проанализированы существующие информационные системы, выявлены их преимущества и недостатки.

В основном разделе работы, прежде всего, были рассмотрены выбранные средства разработки. Информационная система была разработана на языке программирования C# на платформе .NET при помощи Microsoft Visual Studio 2017, а в качестве СУБД была использована программа MariaDB.

Кроме того, была спроектирована и разработана база данных, приведена инфологическая модель данных (схема данных) и описаны используемые в программе таблицы БД.

Подводя итог проведенной работе, можно отметить, что приложение соответствует всем требованиям задания, а именно:

- Разработан модуль учета прихода и продажи автомобилей;
- Разработан модуль учета прихода и продажи товаров для автомобилей (запасные части, аксессуары, комплектующие);
- Разработан модуль учета проведения технического обслуживания автомобилей;
- Разработан модуль учета ремонтных работ;
- Разработан Web-сервис для просмотра справочного материала.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Скит, Дж. С# для профессионалов: тонкости программирования, 3-е издание, новый перевод / Дж. Скит. – М.: Вильямс, 2014.
2. Нейгел, К. С# 5.0 и платформа .NET 4.5 для профессионалов / К. Нейгел. – М.: Диалектика, 2013.
3. <https://redbeanphp.com/index.php> – RedBeanPHP is an easy to use ORM for PHP.
4. <http://php.net/manual/ru/tutorial.php> – The PHP Manual is available online in a selection of languages.
5. <http://htmlbook.ru/samhtml> – Простой учебник HTML и CSS с задачником и примерами решения задач.
6. Фримен, Э. Изучаем HTML, XHTML и CSS/ Э. Фримен, Э. Фримен. – 1-е изд. – М.: Питер, 2010.
7. Шмитт К. CSS. Рецепты программирования / К. Шмитт. – СПб.: БХВ-Петербург, 2007.
8. Рейсиг, Д. Инструменты отладки и тестирования. JavaScript. Профессиональные приёмы программирования / Д. Рейсиг; пер. с англ. Н. Вильчинский. – СПб.: Питер, 2008.