Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет им. Г.В. Плеханова»

Московский приборостроительный техникум

**Курсовой проект**

ПМ 11 «Разработка, администрирование и защита баз данных»

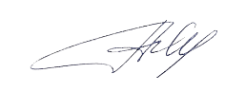
МДК 11.01 «Технология разработки и защиты баз данных»

Специальность 09.02.07 «Информационные системы и программирование»

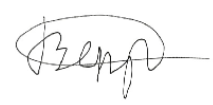
Тема: «Разработка информационной системы «Автосервис»

**Пояснительная записка**

Листов: 48



Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_ / А.А. Шимбирёв

«20» декабря 2022 г.

Исполнитель \_\_\_\_\_\_\_\_\_\_\_\_\_ / А.Н. Веприцкий

«20» декабря 2022 г.

2022

Содержание

[1. ОБЩАЯ ЧАСТЬ 5](#_Toc121948929)

[1.1. Цель разработки 5](#_Toc121948930)

[1.2. Средства разработки 5](#_Toc121948931)

[2. СПЕЦИАЛЬНАЯ ЧАСТЬ 6](#_Toc121948932)

[2.1. Постановка задачи 6](#_Toc121948933)

[2.1.1. Входные данные 6](#_Toc121948934)

[2.1.2. Выходные данные 6](#_Toc121948935)

[2.1.3. Подробные требования к проекту 6](#_Toc121948936)

[2.2. Внешняя спецификация 6](#_Toc121948937)

[2.2.1. Описание задачи 6](#_Toc121948938)

[2.2.2. Входные и выходные данные 21](#_Toc121948939)

[2.2.3. Методы 28](#_Toc121948940)

[2.2.4. Тесты 29](#_Toc121948941)

[2.2.5. Контроль целостности данных 29](#_Toc121948942)

[2.3. Проектирование 30](#_Toc121948943)

[2.3.1. Схема архитектуры программы 30](#_Toc121948944)

[2.3.3. Физическая схема данных 31](#_Toc121948945)

[2.3.5. Функциональная схема 37](#_Toc121948946)

[2.3.6. Диаграмма классов 37](#_Toc121948947)

[2.4. Результаты работы программы 38](#_Toc121948948)

[3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 40](#_Toc121948949)

[3.2. Отладка программы 40](#_Toc121948950)

[3.3. Защитное программирование 41](#_Toc121948951)

[3.4. Характеристики программы 42](#_Toc121948952)

[ЗАКЛЮЧЕНИЕ 46](#_Toc121948953)

[СПИСОК ИСПОЛЬЗОВАННЫХ МАТЕРИАЛОВ 47](#_Toc121948954)

ВВЕДЕНИЕ

В цифровую эпоху практически каждый житель планеты обладает смартфоном или персональным компьютером. Кроме того, все эти устройства подключены к сети Интернет, что позволяет обмениваться данными с беспрецедентной скоростью. Данные факторы позволяют хранить информацию в удобном формате. Также информация в цифровом виде защищена в разы лучше. Современные технологии позволяют предприятиям функционировать с повышенной эффективностью, так как работодатели и работники могут обмениваться информацией с большей скоростью.

Для того, чтобы повысить эффективность предприятия, а также чтобы сократить количество ошибок, организации прибегают к цифровизации части производственных процессов. Для этого, осуществляется наём специалистов, способных разработать программное обеспечение, удовлетворяющее потребности предприятия. Программисты разрабатывают схемы баз данных с использованием различных методологий, таких как IDEF0 и IDEF1X. Для хранения данных специалисты используют специализированные СУБД, которые позволяют разрабатывать различные объекты базы данных, такие как процедуры, таблицы и функции.

В рамках данного курсового проекта была разработана информационная система автосервиса. Программа упрощает манипулирование данными, повышает доступность информации путем создания и отображения графиков, сортировки и фильтрации данных и создания шаблонизированных отчетных документов. Кроме того, информационная система упрощает работу администратора с базой данных.

1. ОБЩАЯ ЧАСТЬ
   1. Цель разработки

Повысить эффективность предприятия путём цифровизации процессов, связанных с его работой.

* 1. Средства разработки

Для разработки информационной системы автосервиса использовались технические средства, представленные в таблице 1.

Таблица 1 - Технические средства для разработки

| № | Тип оборудования | Наименование оборудования |
| --- | --- | --- |
| 1 | 2 | 3 |
| Ноутбук Lenovo81LW | | |
| 1 | Размер экрана: | 15.6” (39.7 см) |
| 2 | Разрешение экрана: | 1920х1080 |
| 3 | Линейка процессора: | AMD Ryzen 3 |
| 4 | Количество ядер процессора: | 4 |
| 5 | Оперативная память: | 4 ГБ |
| 6 | Тип видеокарты: | Интегрированный |
| 7 | Видеокарта: | AMD Radeon Vega 3 |
| 8 | Конфигурация накопителей: | SSD |
| 9 | Общий объем всех накопителей: | 256 ГБ |

Для разработки информационной системы автосервиса использовались программные средства, представленные в таблице 2.

Таблица 2 - Программные средства

| № | Тип средства | Название средства | Назначение |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 1 | Операционная система | Microsoft Windows 10 | Организация взаимодействия программ и пользователя |
| 2 | Инструментальное средство разработки программных решений | Visual Studio 2019 | Реализация программного продукта и API |
| 3 | Утилита для взаимодействия с СУБД | pgAdmin4 | Работа с СУБД |
| 4 | Программа для проектирования и документирования баз данных | ERwin Data Modeler | Создание физической и логической моделей базы данных |
| 5 | Пакет офисных программ | MS Office 2016 | Реализация технической документации |
| 6 | Средство создания IDEF0 | AllFusion Process Modeller r7 | Моделирование бизнес -процессов в соответствии с методологией IDEF0 |

1. СПЕЦИАЛЬНАЯ ЧАСТЬ
   1. Постановка задачи

Разработать информационную систему автосервиса.

* + 1. Входные данные

Входными данными информационной системы автосервиса, являются данные, вводиме пользователем при работе с информационной системой.

Входными данными являются такие данные, как информация о клиентах, автомобилях, контрактах, заказах, услугах, договорах, поставщиках, складах, накладных и расходниках.

* + 1. Выходные данные

Выходными данными является информация, отображаемая на оконных элементах, выводимая при помощи графиков и выгружаемая по шаблону.

* + 1. Подробные требования к проекту

Спроектировать и разработать базу данных для информационной системы автосервиса.

Спроектировать и разработать программное обеспечение для информационной системы автосервиса для операционных систем Windows 8.1 и выше. В программе должно быть реализовано добавление, изменение и удаление данных (в том числе и логическое). Кроме того, должно быть доступно отображение данных с помощью графиков, а также экспорт данных по шаблону. Также должно быть реализовано разграничение прав доступа и шифрование личных данных пользователей. Кроме того, требуется реализовать функции авторизации и регистрации новых пользователей.

* 1. Внешняя спецификация
     1. Описание задачи

Разработка информационной системы автосервиса для упрощения работы с информацией и повышения эффективности предприятия.

Информационная система состоит из базы данных, хранящей данные для взаимодействия и программного решения, непосредственного взаимодействующего с базой данных. База данных должна быть спроектирована и реализована в соответствии с третьей нормальной формой. Программное решение также должно предоставлять функциональные возможности для взаимодействия с базой данных и наглядного отображения информации.

На рисунке 1 представлена диаграмма бизнес-процессов «AS-IS» предприятия «Автосервис».

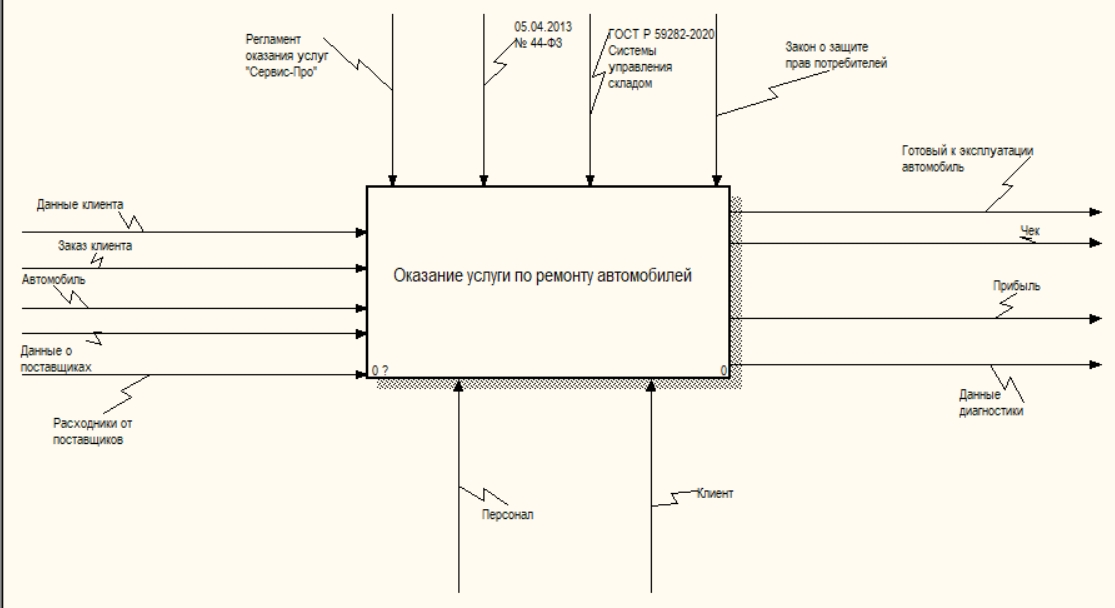


Рисунок 1 - Нулевой уровень декомпозиции "AS-IS"

В качестве входных данных выступают данные клиента, заказ клиента, данные об автомобилях, данные о поставщиках и расходниках.

В качестве управляющих связей выступают различные законы РФ, такие как 149-ФЗ, 44-ФЗ, ГОСТ Р 59282-2020, а также регламент оказания услуг организации.

В качестве механизмов выступают персонал, клиент.

В качестве выходных данных выступают данные об обслуженном автомобиле, диагностике, прибыли, а также чек.

На рисунке 2 представлен 1 уровень декомпозиции бизнес-процесса «AS-IS».

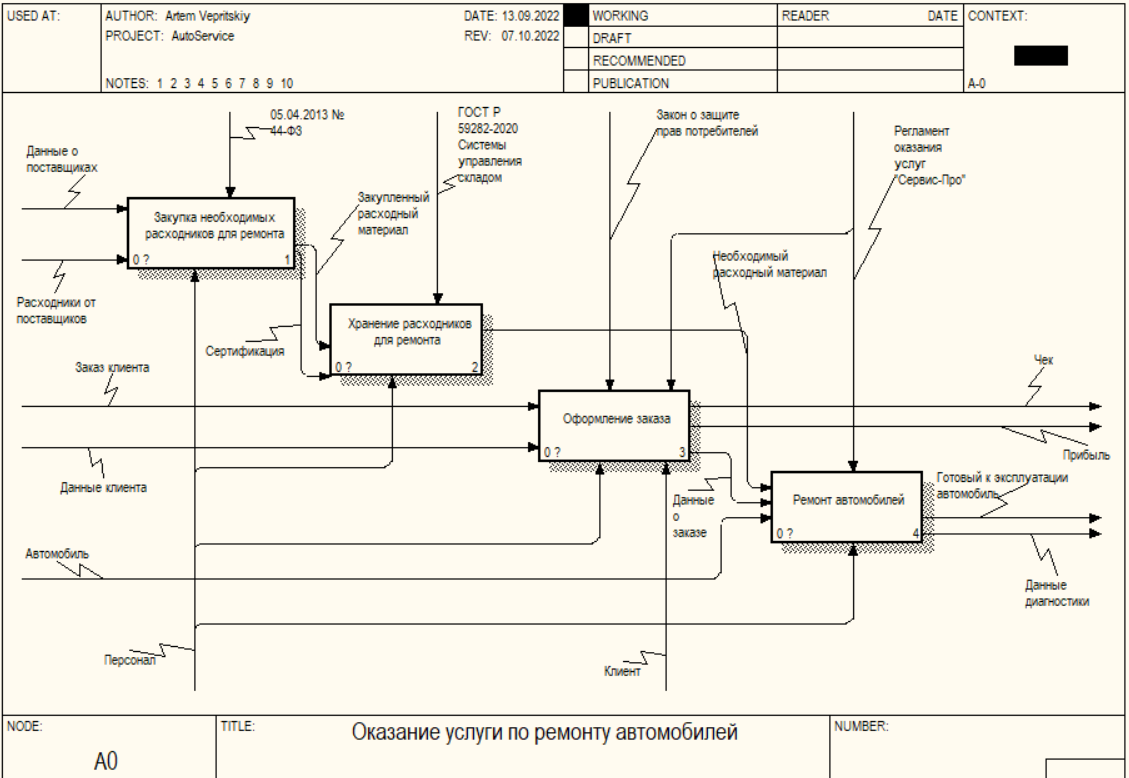


Рисунок 2 - Первый уровень декомпозиции "AS-IS"

На первом уровне декомпозиции представлены детальные процессы предприятия.

Процесс «Закупка необходимых расходников для ремонта» осуществляет закупку расходников для тех.обслуживания.

Процесс «Хранение расходников для ремонта» осуществляет хранение расходников.

Процесс «Оформление заказа» позволяет клиентам оформлять заказы, а сотрудникам отдела продаж принимать их.

Процесс «Ремонт автомобилей» осуществляет ремонт автомобилей.

На рисунке 3 представлен 2 уровень декомпозиции процесса «Закупка необходимых расходников для ремонта» «AS-IS».

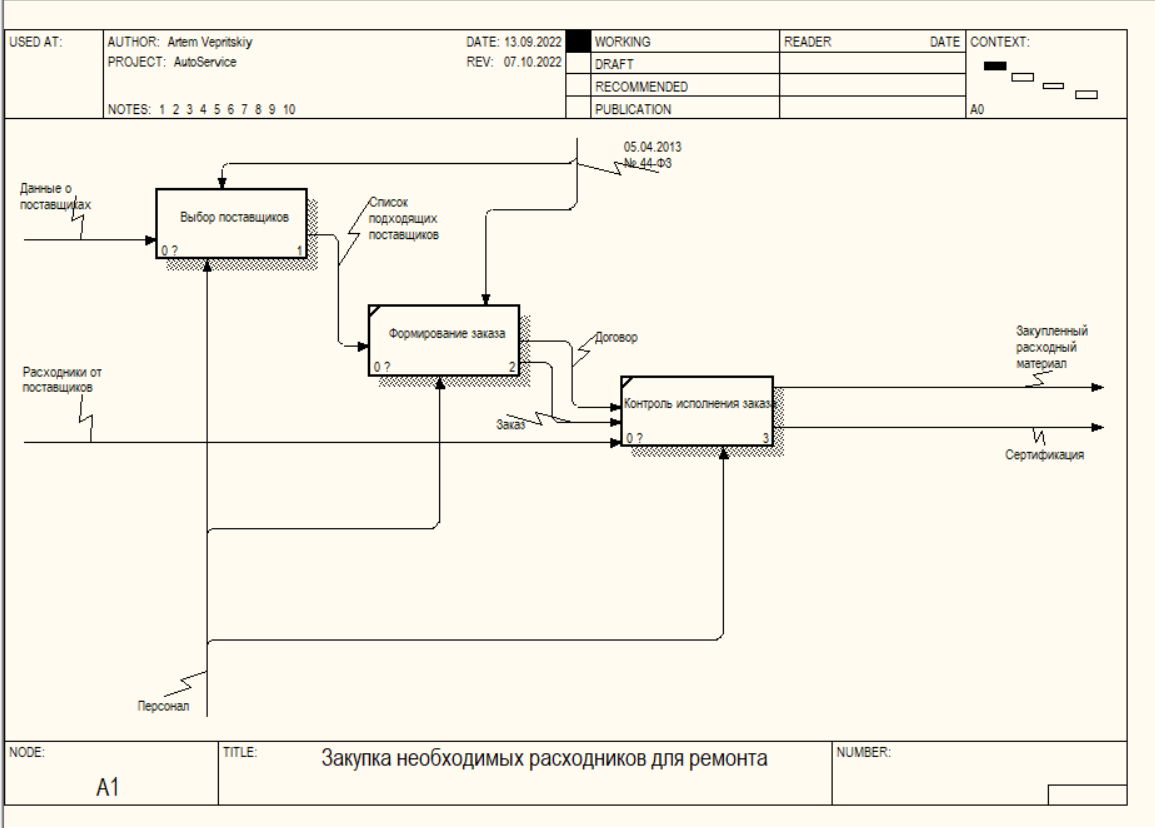


Рисунок 3 - 2 уровень декомпозиции процесса "Закупка необходимых расходников для ремонта"

Подпроцесс «Выбор поставщика» отвечает за поиск поставщика для закупки.

Подпроцесс «Формирование чека» отвечает за создание чека после успешной покупки расходников.

Подпроцесс «Контроль исполнения заказа» отвечает за успешную доставку расходников на предприятие и устранение проблем при транспортировке.

На рисунке 4 представлен 2 уровень декомпозиции процесса «Хранение расходников для ремонта» «AS-IS».

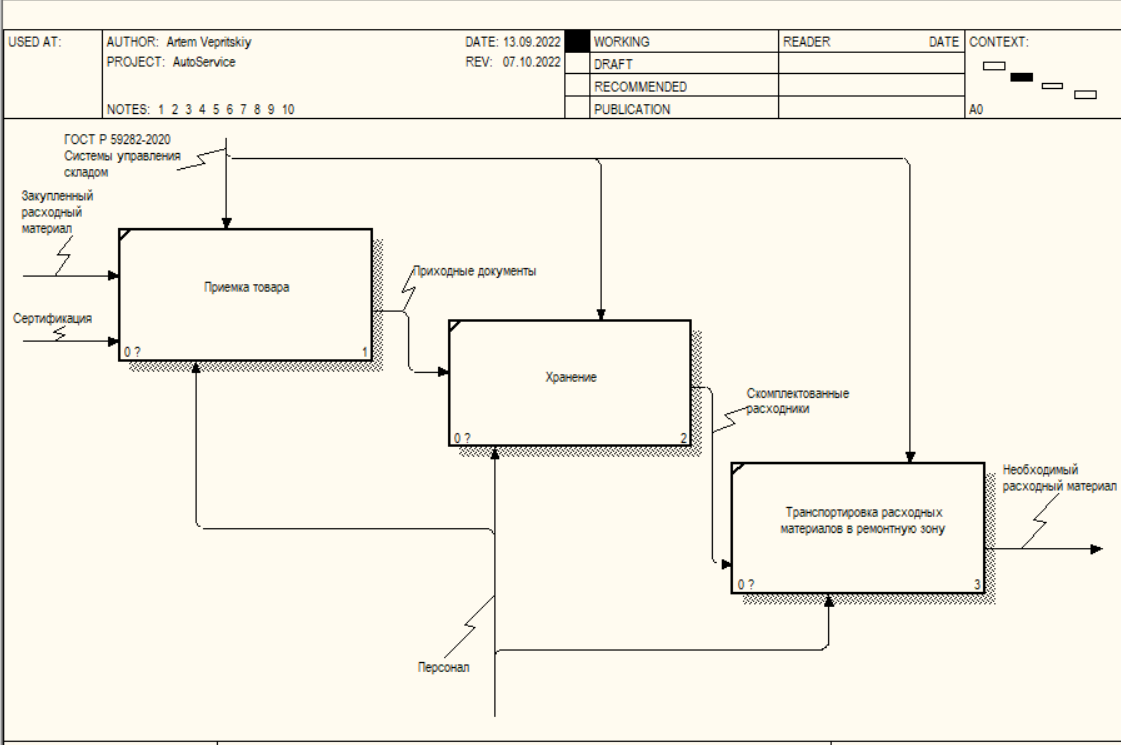


Рисунок 4 - 2 уровень декомпозиции процесса "Хранение расходников для ремонта"

Подпроцесс «Приемка товара» отвечает за передачу товара на склад после доставки.

Подпроцесс «Хранение» отвечает за учёт и хранение расходников на складе.

Подпроцесс «Транспортировка расходных материалов в ремонтную зону» отвечает за передачу необходимых расходных материалов в ремонтную зону.

На рисунке 5 представлен 2 уровень декомпозиции процесса «Оформление заказа» «AS-IS».

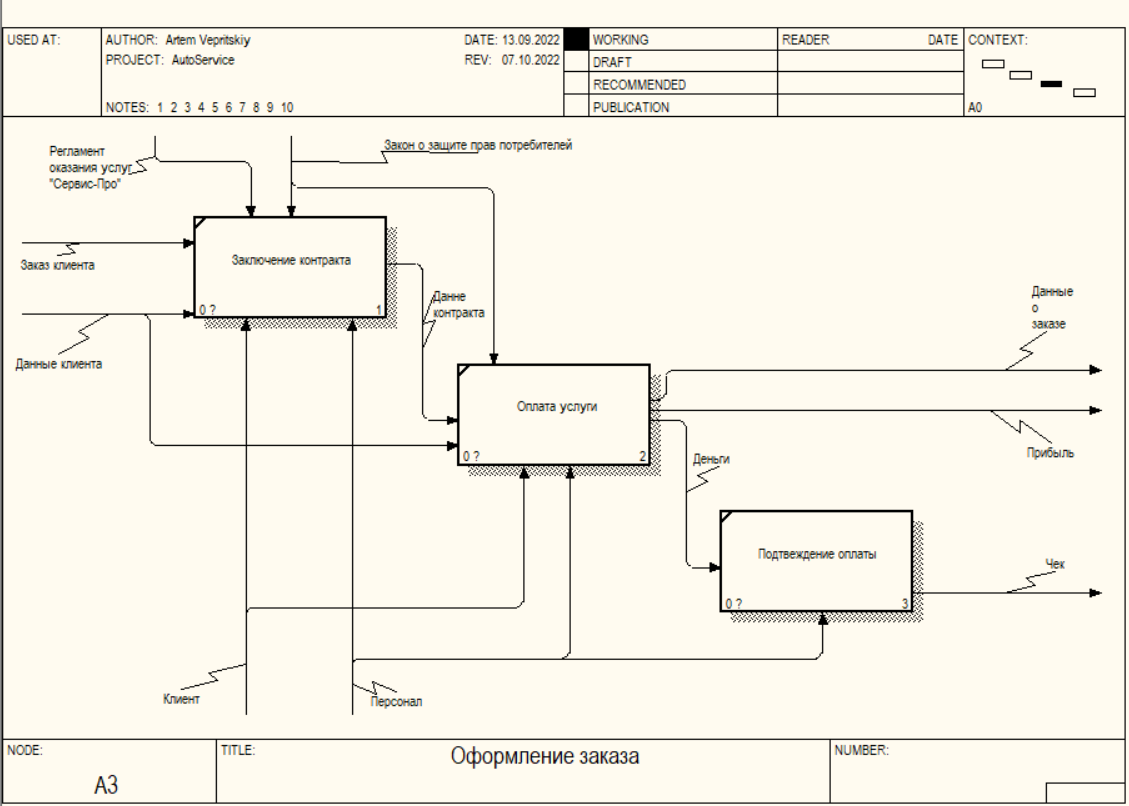


Рисунок 5 - 2 уровень декомпозиции процесса «Оформление заказа» «AS-IS».

Подпроцесс «Заключение контракта» позволяет предприятию заключать договор на предоставление услуги с клиентом.

Подпроцесс «Оплата услуг» обеспечивает получение предприятем оплаты за услугу.

Подпроцесс «Подтверждение оплаты» производит подтверждение оплаты за услугу.

На рисунке 6 представлен 2 уровень декомпозиции процесса «Ремонт автомобилей» «AS-IS».

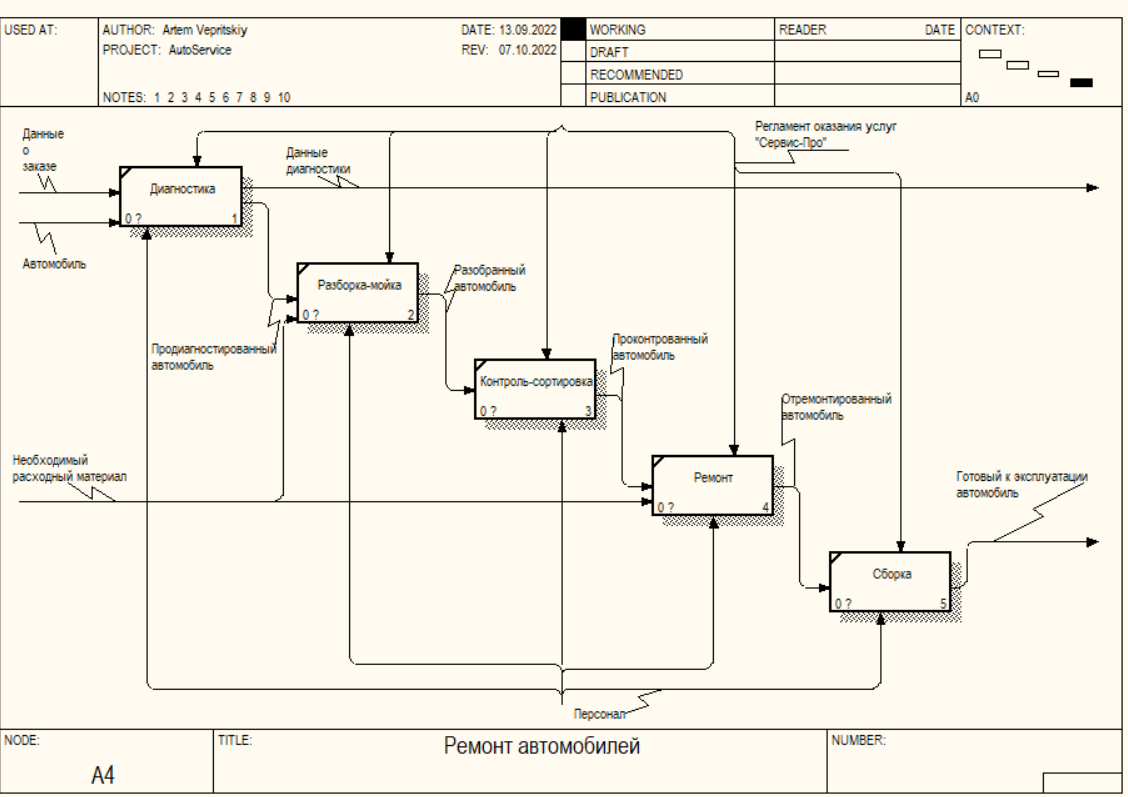


Рисунок 6 - 2 уровень декомпозиции процесса «Ремонт автомобилей» «AS-IS»

Подпроцесс «Диагностика» отвечает за выявление неисправностей в работе автомобиля.

Подпроцесс «Разборка-мойка» отвечает за демонтаж деталей и их очистку.

Подпроцесс «Контроль сортировка» отвечает за выявление поврежденных деталей.

Подпроцесс «Ремонт» отвечает за восстановление поврежденных деталей.

Подпроцесс «Сборка» отвечает за установку деталей автомобиля и проверку его работоспособности.

На рисунке 7 представлена диаграмма бизнес-процессов «TO-BE» предприятия «Автосервис».

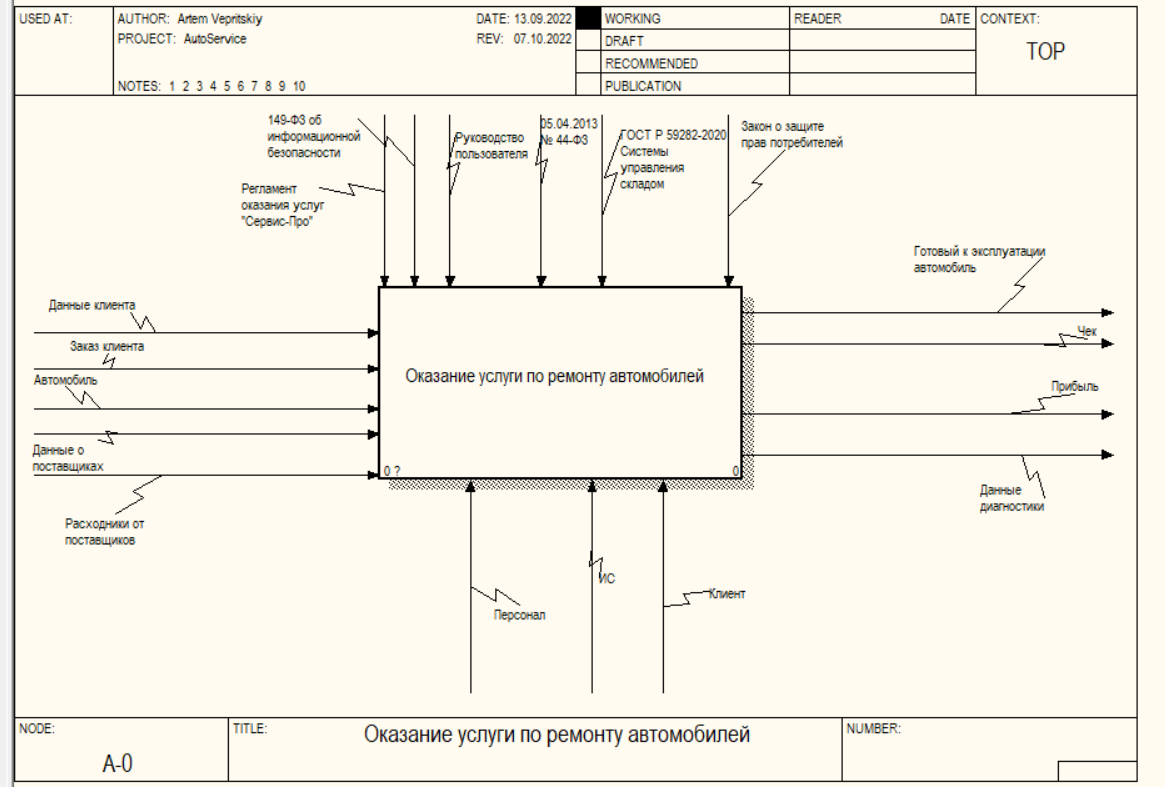


Рисунок 7 - Диаграмма бизнес-процессов «TO-BE» предприятия «Автосервис»

На диаграмме TO-BE помимо ранее упомянутых данных и механизмов применяется информационная система. Это необходимо для повышения эффективности предприятия. Кроме того, пользователю будет предоставлено руководство пользователя для работы с информационной системой.

На рисунке 8 представлен 1 уровень декомпозиции бизнес-процесса TO-BE.

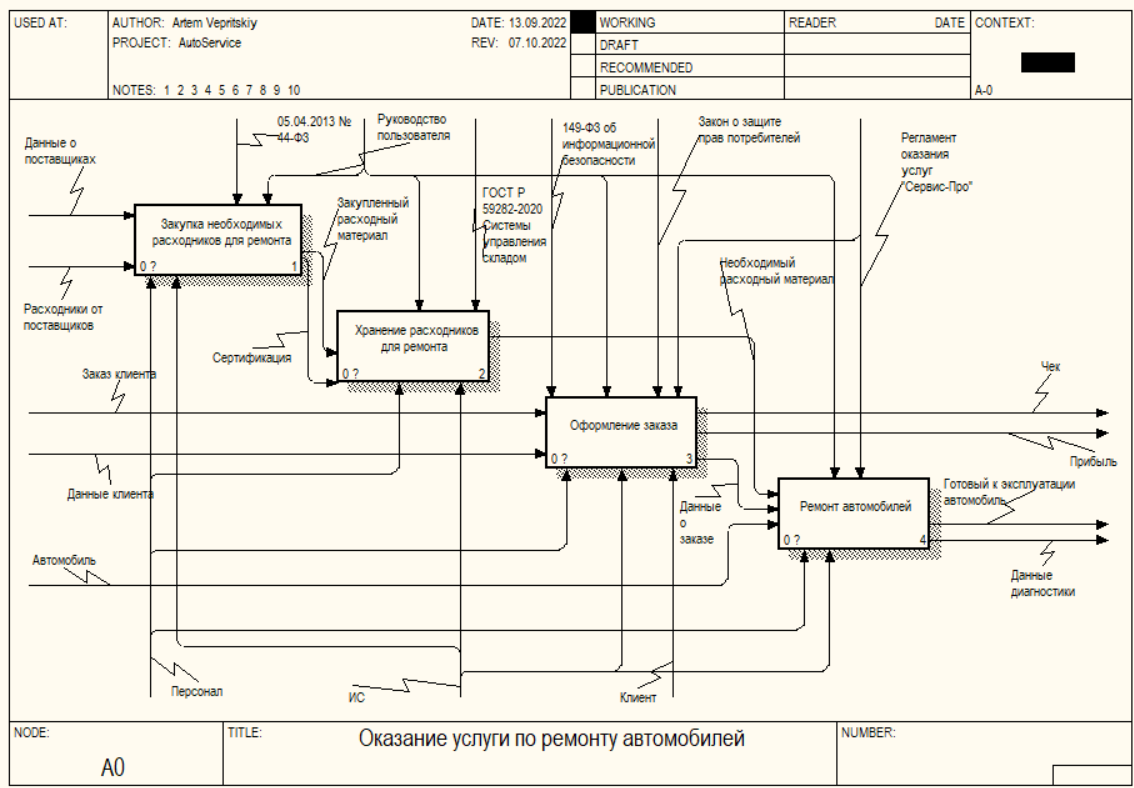


Рисунок 8 - 1 уровень декомпозиции бизнес-процесса TO-BE

В подпроцессы организации внедряется информационная система.

Процесс «Закупка необходимых расходников для ремонта» позволяет вносить пользователям данные о расходниках и поставщиках.

Процесс «Хранение расходников для ремонта» даёт пользователям возможность вносить данные о складах.

Процесс «Оформление заказа» позволяет клиентам оформлять заказы, а сотрудникам отдела продаж взаимодействовать с заказами клиентов.

Процесс «Ремонт автомобилей» даёт сотрудникам ремонтного отдела возможность вносить данные о ремонтируемых автомобилях.

На рисунке 9 представлен 2 уровень декомпозиции процесса «Закупка необходимых расходников для ремонта» «TO-BE».

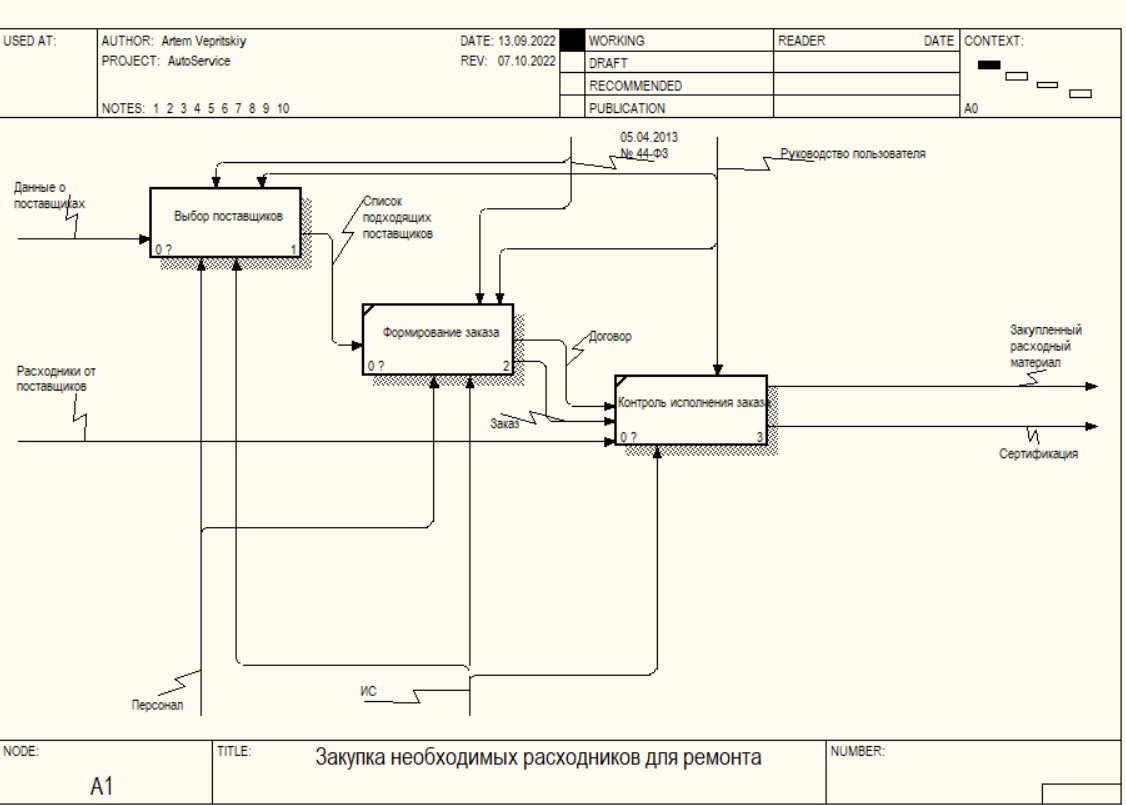


Рисунок 9 - 2 уровень декомпозиции процесса «Закупка необходимых расходников для ремонта» «TO-BE».

Как указано на схеме, в текущие подпроцессы внедряется информационная система, что повысит удобство хранения и использования информации о поставщиках и заказах.

На рисунке 10 представлен 2 уровень декомпозиции процесса «Хранение расходников для ремонта» «TO-BE».

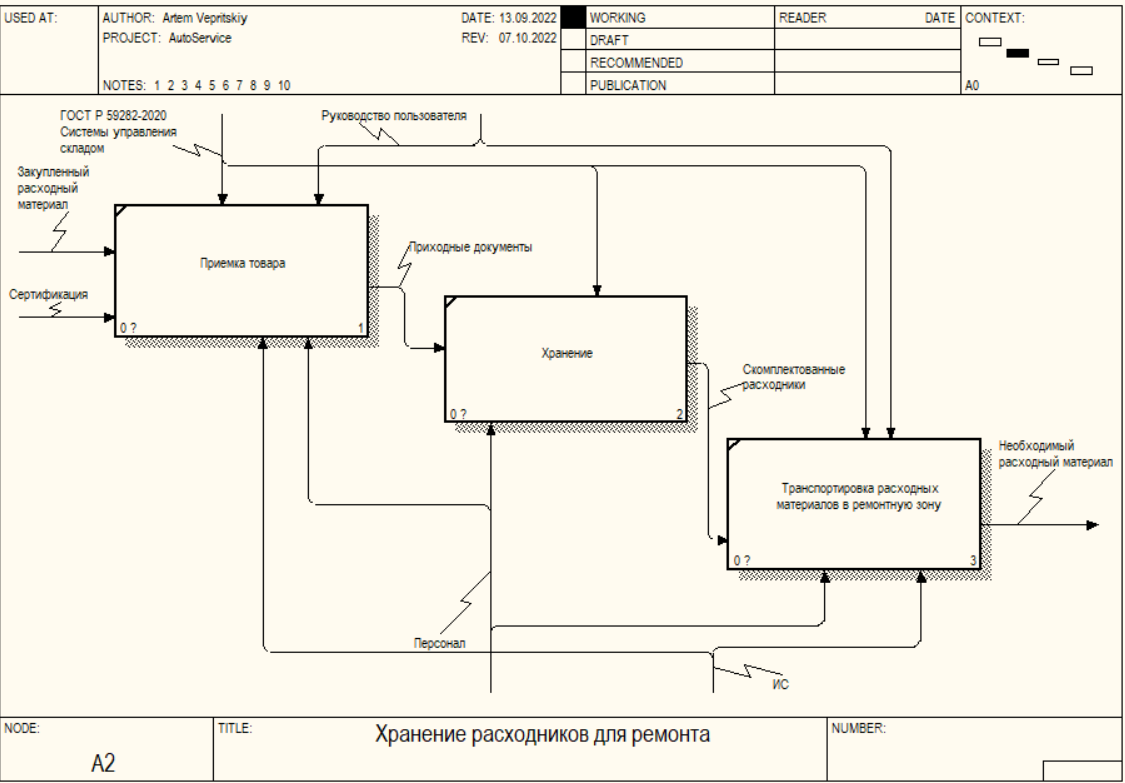


Рисунок 10 - 2 уровень декомпозиции процесса «Хранение расходников для ремонта» «TO-BE»

На данном уровне информационная система позволит автоматизировать функции приема и транспортировки расходников, что позволит сократить штат сотрудников и повысить скорость работы подсистемы.

На рисунке 11 представлен 2 уровень декомпозиции процесса «Оформление заказа» «TO-BE».

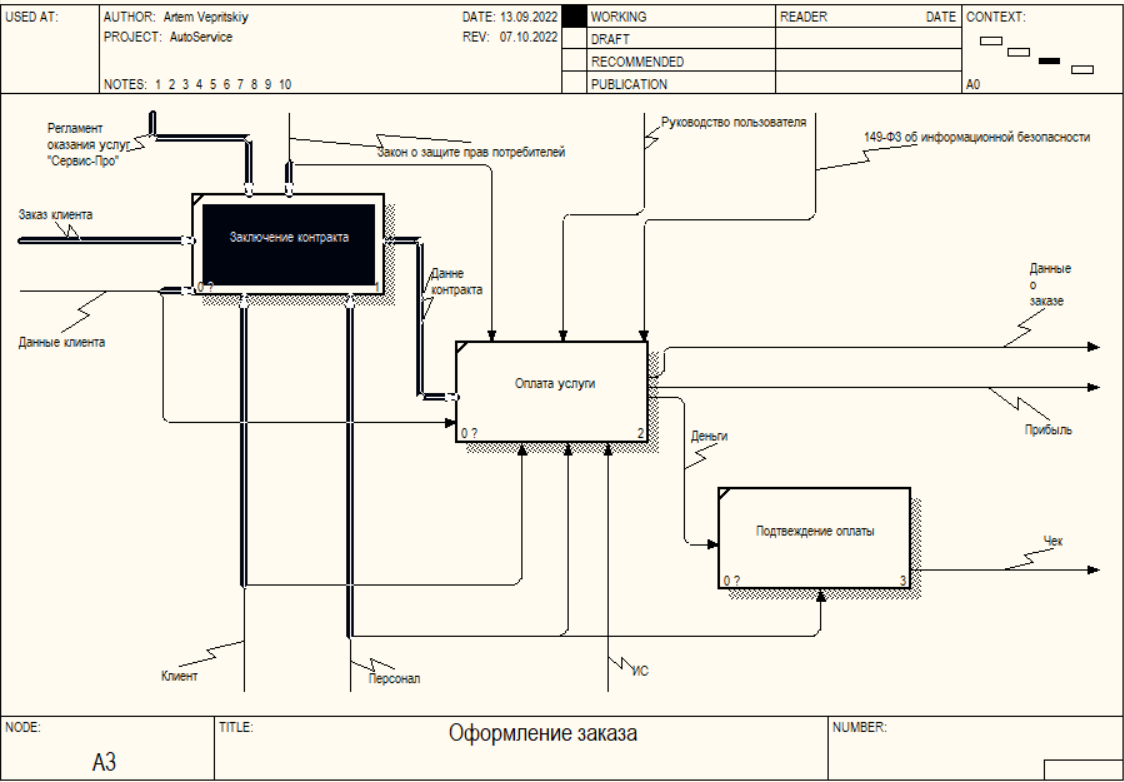


Рисунок 11 - 2 уровень декомпозиции процесса «Оформление заказа» «TO-BE»

Как указано на схеме, информационная система внедряется во все процессы текущей подсистемы. Внедрение информационной системы в данный подпроцесс повысит эффективность работы сотрудников с клиентами, а также существенно повысит уровень обслуживания клиентов.

На рисунке 12 представлен 2 уровень декомпозиции процесса «Ремонт автомобилей» «TO-BE».

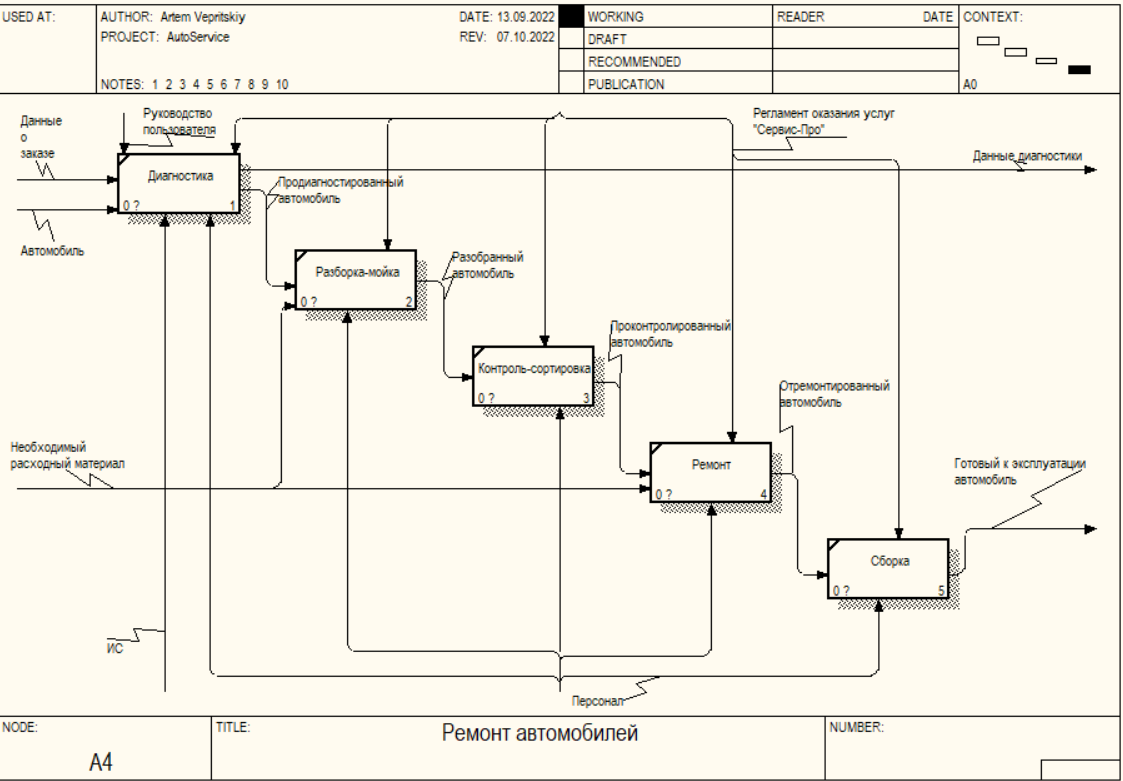


Рисунок 12 - 2 уровень декомпозиции процесса «Ремонт автомобилей» «TO-BE».

В данной подсистеме информационная система внедряется только в подпроцесс «Диагностика». Она позволит хранить данные о диагностике в удобном формате и увеличит скорость передачи данных сотрудникам ремонтного цеха.

Разрабатываемая информационная система будет внедрена в основные процессы предприятия. Будут автоматизированы функции закупки и хранения расходников, приема и оформления заказов клиентов, а также диагностика автомобиля. Кроме того, будет разработано руководство пользователя для оказания помощи при работе с проектом. Эти факторы повысят удобство хранения информации и её защищенность.

На рисунке 13 представлена диаграмма прецедентов для информационной системы автосервиса.

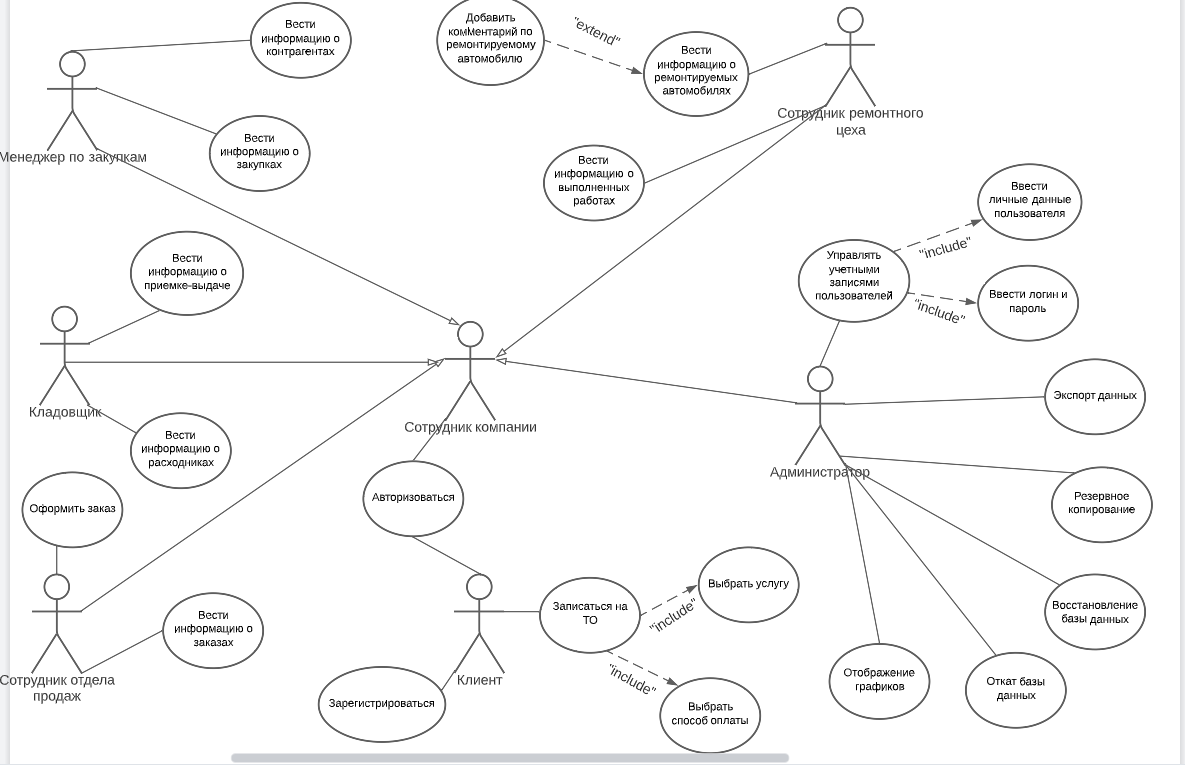


Рисунок 13 - Диаграмма прецедентов

Пользователи должны обладать следующими функциями:

1. Администратор:
   1. Добавление, изменение, удаление и логическое удаление данных о сотрудниках
   2. Фильтрация сотрудников
   3. Экспорт данных по шаблону
   4. Резервное копирование
   5. Восстановление базы данных
   6. Откат базы данных
   7. Отображение графиков
2. Сотрудник отдела закупок:
   1. Добавление, изменение и удаление данных о расходниках, заказах на поставку и договорах
3. Сотрудник отдела продаж:
   1. Добавление, изменение и удаление данных о контрактах, услугах и заказах клиентов
4. Сотрудник складского отдела:
   1. Добавление, изменение и удаление данных о складах, ячейках, поставщиках и накладных
5. Сотрудник ремонтного цеха:
   1. Добавление, изменение и удаление данных о диагностике и статусе автомобиля
      1. Входные и выходные данные

В таблице 3 представлены входные данные информационной системы автосервиса.

Таблица 3 - Входные данные

| № | Наименование | Ограничения | Тип | Описание | Формат ввода |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
| Окно авторизации | | | | | |
| 1 | Адрес электронной почты | Соответствует формату почты, уникален, [0-9][A-Z][a-z][@.], не пустое | String | Ввод адреса эл.почты пользователя для входа в систему | Поле ввода |
| 2 | Пароль | Более 7 символов, хотя бы одну заглавную букву[A-Z], хотя бы одну прописную букву[a-z], хотя бы одну цифру [0-9], хотя бы один спецсимвол[!\"#$%&'()\*+,-./::<=>?@[\\]:\_{|}] , не более 30 символов | String | Ввод пароля пользователя для входа в систему | Поле ввода |
| Окно регистрации пользователя | | | | | |
| 3 | Адрес электронной почты | Соответствует формату почты, уникален, [0-9][A-Z][a-z][@.], не пустое | String | Ввод почты при регистрации | Поле ввода |
| 4 | Пароль | Более 7 символов, хотя бы одну заглавную букву[A-Z], хотя бы одну прописную букву[a-z], хотя бы одну цифру [0-9], хотя бы один спецсимвол[!\"#$%&'()\*+,-./::<=>?@[\\]:\_{|}] , не более 30 символов | String | Ввод пароля при регистрации | Поле ввода |
| 5 | Фамилия | Не пустое, [А-Я], [а-я] | String | Ввод фамилии при регистрации | Поле ввода |
| 6 | Имя | Не пустое, [А-Я], [а-я] | String | Ввод имени при регистрации | Поле ввода |
| 7 | Отчество | Не пустое, [А-Я], [а-я] | String | Ввод отчества при регистрации | Поле ввода |
| 8 | Серия паспорта | Не пустое, 4 символа, [0-9] | String | Ввод серии паспорта | Поле ввода |
| 9 | Номер паспорта | Не пустое, 6 символов, [0-9] | String | Ввод номера паспорта | Поле ввода |
| 10 | Номер телефона | Не пустое, 10 символов, [0-9] | String | Ввод номера телефона | Поле ввода |
| Окно восстановления доступа к аккаунту | | | | | |
| 11 | Адрес электронной почты | Соответствует формату почты, уникален, [0-9][A-Z][a-z][@.], не пустое | String | Ввод почты при восстановлении доступа | Поле ввода |
| 12 | Код восстановления | [0-9], равен 6 символам, не пустое | String | Ввод кода восстановления пароля при восстановлении доступа | Поле ввода |
| Окно отдела закупок | | | | | |
| 13 | Номер заказа на поставку | Не более 35 символов, не пустое[А-Я][а-я][A-Z][a-z][0-9] | String | Ввод номера заказа на поставку | Поле ввода |
| 14 | Дата заказа на поставку | Формат даты, не раньше текущей даты, не пустое | Date | Ввод даты заказа на поставку | Поле выбора даты |
| 15 | Количество | Не более 7 символов, не пустое, [0-9] | int | Ввод количества товара | Поле ввода |
| 16 | Итого | Не более 9 символов, не пустое, [0-9][.] | double | Ввод итоговой суммы заказа на поставку | Поле ввода |
| 17 | Артикул товара | Не более 35 символов, не пустое, [А-Я][а-я][A-Z][a-z][0-9] | String | Ввод артикула товара | Поле ввода |
| 18 | Юридическое наименование | Не более 30 символов, не пустое, [А-Я][а-я][A-Z][a-z][0-9] | String | Ввод юр. наименования поставщика | Поле ввода |
| 19 | ИНН поставщика | Не более 10 символов, [0-9], не пустое | String | Ввод ИНН поставщика | Поле ввода |
| 20 | Расчётный счёт | Не более 20 символов, [0-9], не пустое | String | Ввод расчётного счета поставщика | Поле ввода |
| 21 | Дата контракта на закупку | Формат даты, не раньше текущей даты, не пустое | Date | Ввод даты контракта | Поле выбора даты |
| 22 | Номер заказа на поставку | Не предусмотрено, [А-Я][а-я][A-Z][a-z][0-9] | String | Выбор заказа на поставку | Поле выбора |
| 23 | ИНН поставщика | Не предусмотрено | String | Выбор поставщика | Поле выбора |
| 24 | ФИО сотрудника | Не предусмотрено | String | Выбор сотрудника | Поле выбора |
| 25 | Номер контракта на поставку | Не более 35 символов, не пустое, [А-Я][а-я][A-Z][a-z][0-9] | String | Ввод номера контракта | Поле ввода |
| 26 | Серийный номер расходника | Не более 35 символов, не пустое, [А-Я][а-я][A-Z][a-z][0-9] | String | Ввод серийного номера расходника | Поле ввода |
| 27 | Номер контракта на поставку | Не предусмотрено | String | Выбор номера контракта | Поле выбора |
| 28 | Название расходника | Не более 35 символов, не пустое, [А-Я][а-я][A-Z][a-z][0-9] | String | Ввод названия расходника | Поле ввода |
| 29 | Вес расходника | Не более 9 символов, не пустое | String | Ввод веса расходника | Поле ввода |
| 30 | Цвет расходника | Не более 20 символов, не пустое[А-Я][а-я][A-Z][a-z][0-9] | String | Ввод цвета расходника | Поле ввода |
| 31 | Материал расходника | Не более 35 символов, не пустое[А-Я][а-я][A-Z][a-z][0-9] | String | Ввод материала расходника | Поле ввода |
| 32 | Сертификат расходника | Не более 35 символов, не пустое[А-Я][а-я][A-Z][a-z][0-9] | String | Ввод сертификата | Поле ввода |
| Окно отдела склада | | | | | |
| 33 | Название склада | Не более 30 символов, не пустое[А-Я][а-я][A-Z][a-z][0-9] | String | Ввод названия склада | Поле ввода |
| 34 | Адрес склада | Не более 60 символов, не пустое[А-Я][а-я][A-Z][a-z][0-9] | String | Ввод адреса склада | Поле ввода |
| 35 | Номер ячейки | Не более 5 символов, не пустое[А-Я][а-я][A-Z][a-z][0-9] | String | Ввод номера ячейки | Поле ввода |
| 36 | Номер накладной | Не более 35 символов, не пустое[А-Я][а-я][A-Z][a-z][0-9] | String | Ввод номера накладной | Поле ввода |
| 37 | Серийный номер расходника | Не предусмотрено | String | Выбор номера расходника | Поле выбора |
| 38 | Номер ячейки | Не предусмотрено | String | Выбор номера ячейки | Поле выбора |
| 39 | Сотрудник | Не предусмотрено | String | Выбор сотрудника | Поле выбора |
| 40 | Дата накладной | Формат даты, не раньше текущей даты, не пустое | Date | Ввод даты накладной | Поле выбора даты |
| 41 | Количество поступившего товара | Не более 8 символов, не пустое, [0-9] | int | Ввод количества поступившего товара | Поле ввода |
| Окно отдела продаж | | | | | |
| 42 | Номер заказа | Не более 35 символов, не пустое[А-Я][а-я][A-Z][a-z][0-9] | String | Ввод номера заказа | Поле ввода |
| 43 | Дата заказа | Формат даты, не пустое | Date | Ввод даты заказа | Поле ввода даты |
| 44 | Машина | Не более 6 символов, не пустое[А-Я] [0-9] | String | Ввод номера машины | Поле ввода |
| 45 | Сумма заказа | Не пустое, [0-9][.] | double | Ввод суммы заказа | Поле ввода |
| 46 | Название услуги | Не более 35 символов, не пустое[А-Я][а-я][A-Z][a-z][0-9] | String | Ввод названия услуги | Поле ввода |
| 47 | Цена | Не более 30 символов, [0-9][.] | double | Ввод цены услуги | Поле ввода |
| 48 | Описание | Не более 500 символов[А-Я][а-я][A-Z][a-z][0-9] | String | Ввод описания услуги | Поле ввода |
| Окно ремонтного отдела | | | | | |
| 49 | Уникальный номер диагностики | Не пустое, не более 30 символов[А-Я][а-я][A-Z][a-z][0-9] | String | Ввод уникального номера диагностики | Поле ввода |
| 50 | Дата диагностики | Не пустое, формат даты, не раньше текущей даты | Date | Ввод даты диагностики | Поле ввода |
| 51 | Результат диагностики | Не пустое, не более 30 символов[А-Я][а-я][A-Z][a-z][0-9] | String | Ввод результатов диагностики | Поле ввода |
| 52 | Описание | Не пустое, не более 500 символов[А-Я][а-я][A-Z][a-z][0-9] | String | Ввод описания диагностики | Поле ввода |
| Окно администратора | | | | | |
| 53 | Адрес электронной почты | Не пустое, формат электронной почты, не более 50 символов | String | Ввод адреса электронной почты | Поле ввода |
| 54 | Пароль | Более 7 символов, хотя бы одну заглавную букву[A-Z], хотя бы одну прописную букву[a-z], хотя бы одну цифру [0-9], хотя бы один спецсимвол[!\"#$%&'()\*+,-./::<=>?@[\\]:\_{|}], не более 30 символов | String | Выбор поставщика при редактировании или добавлении | Поле выбора |
| 55 | Фамилия | Не пустое, не более 30 символов[А-Я][а-я] | String | Ввод фамилии | Поле ввода |
| 56 | Имя | Не пустое, не более 30 символов[А-Я][а-я] | String | Ввод имени | Поле ввода |
| 57 | Отчество | Не пустое, не более 30 символов[А-Я][а-я] | String | Ввод отчества | Поле ввода |
| 58 | Должность | Не предусмотрено | String | Выбор должности | Поле выбора |
| Окно клиента | | | | | |
| 59 | Номер машины | Не более 6 символов, не пустое[А-Я][а-я][A-Z][a-z][0-9] | String | Ввод номера машины | Поле ввода |
| 60 | Модель машины | Не более 50 символов, не пустое[А-Я][а-я][A-Z][a-z][0-9] | String | Ввод модели машины | Поле ввода |
| 61 | Марка машины | Не более 30 символов, не пустое[А-Я][а-я][A-Z][a-z][0-9] | String | Ввод марки машины | Поле ввода |
| 62 | Код VIN | 17 символов, не пустое | String | Ввод VIN автомобиля | Поле ввода |
| 63 | Код СТС | 10 символов, не пустое | String | Ввод СТС | Поле ввода |
| Окно оформления заказа | | | | | |
| 64 | Флажок услуги | Не предусмотрено | Bool | Выбор услуги | Поле флажка |
| 65 | Дата заказа | Формат даты, не пустое, не раньше текущей даты | Date | Ввод даты заказа | Поле выбора даты |
| 66 | Автомобиль | Не предусмотрено | String | Выбор автомобиля для обслуживания | Поле выбора |

Выходные данные для информационной системы автосервиса представлены в таблице 4.

Таблица 4 - Выходные данные

| Поле/Объект | Тип данных | Ограничение |
| --- | --- | --- |
| [1](https://tdocs.su/51443) | 2 | 3 |
| Создание, изменение, удаление и фильтр пользователей | | |
| Data | DataGrid | Ограничен количеством выводимых записей |
| Добавление, изменение и удаление заказов | | |
| Data | DataGrid | Ограничен количеством выводимых записей |
| Итого | Text | Не ограничен |
| Просмотр текущих заказов | | |
| Data | DataGrid | Ограничен количеством выводимых записей |
| Выбранные услуги | ListBox | Не ограничен |
| Статус | ListBox | Не ограничен |
| Создание, изменение, удаление и автомобилей | | |
| Data | DataGrid | Ограничен количеством выводимых записей |
| Создание, изменение, удаление заказов на поставку | | |
| Data | DataGrid | Ограничен количеством выводимых записей |
| Создание, изменение, удаление поставщиков | | |
| Data | DataGrid | Ограничен количеством выводимых записей |
| Создание, изменение, удаление контрактов на закупку | | |
| Data | DataGrid | Ограничен количеством выводимых записей |
| Создание, изменение, удаление расходников | | |
| Data | DataGrid | Ограничен количеством выводимых записей |
| Добавление, изменение и удаление контрактов на обслуживание | | |
| Data | DataGrid | Ограничен количеством выводимых записей |
| Добавление, изменение и удаление диагностики | | |
| Data | DataGrid | Ограничен количеством выводимых записей |
| Добавление, изменение и удаление заказов | | |
| Data | DataGrid | Ограничен количеством выводимых записей |
| Добавление, изменение и удаление контрактов | | |
| Data | DataGrid | Ограничен количеством выводимых записей |
| Добавление, изменение и удаление услуг | | |
| Data | DataGrid | Ограничен количеством выводимых записей |
| Добавление, изменение и удаление складов | | |
| Data | DataGrid | Ограничен количеством выводимых записей |
| Добавление, изменение и удаление ячеек | | |
| Data | DataGrid | Ограничен количеством выводимых записей |
| Добавление, изменение и удаление накладных | | |
| Data | DataGrid | Ограничен количеством выводимых записей |
| Просмотр графика товаров | | |
| Год регистрации | Data | Ограничен размером окна |
| Количество пользователей | Series | Ограничен размером окна |

* + 1. Методы

Для проектирования реляционной базы данных использовались методологии IDEF0 и IDEF1X. Для построения диаграммы бизнес – процессов использовалась программа AllFusion Process Modeller r7. Для построения логической и физической моделей базы данных была использована ERwin Data Modeler. Для хранения данных использовалась postgreSQL.

Для создания приложения информационной системы магазина одежды использовалась программа Visual Studio 2019 вместе с WPF на NETFramework 4.7.2 и языком разметки XAML.

WPF (Windows Presentation Foundation) – это аналог WinForms, система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем

При разработке программы использовались принципы объектно-ориентированного программирования, где у каждого окна есть соответствующий ему класс. Для реализации интерфейса был использован язык разметки XAML, где в тег Frame помещались все дочерние объекты, что обеспечивает меньшую нагрузку на программу и улучшает пользовательский опыт работы с программой.

Также было разработано приложение, которое реализует API (Application Programming Interface) для информационной системы и может взаимодействовать с базой данных. Для реализации API был использован паттерн MVC.

* + 1. Тесты

Для информационной системы автосервиса был составлен и исполнен список тестовых испытаний. Также было проведено тестирование методами белого и черного ящика. Данные о тестировании представлены в «Приложение В. Сценарий и результаты тестовых испытаний».

* + 1. Контроль целостности данных

В таблице 5 представлены аномалии, которые могут возникнуть при работе с информационной системой.

Таблица 5 – Контроль целостности данных

| № п/п | Ситуация | Аномалия | Реакция программы | Описание |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Добавление/  Изменение данных в каждом из окон | Не заполнены или некорректно заполнены данные одного из обязательных для заполнения полей | Появляется диалоговое окно с сообщением «Данные заполнены некорректно» | Информация должна быть корректно заполнена. |
| 2 | Регистрация/ изменение данных профиля пользователя/  Добавление/ изменение данных о сотруднике | Введена почта которая уже существуют в системе | Появляется диалоговое окно с сообщением «Такая запись уже существует» | Необходимо поменять логин или почту, которые уже хранятся в базе данных |
| 3 | Добавление/ изменение данных в каждом из окон | Введены полностью повторяющиеся данные, которые уже есть в системе | Появляется диалоговое окно с сообщением «Такая запись уже существует» | Необходимо ввести данные, которые не повторяются с данными, которые хранятся в базе данных |
| 4 | Удаление записи | Попытка удалить данные, используемые в другой таблице | Уведомление о невозможности удалить данные | Необходимо удалить данные, которые зависят от текущих данных. |

* 1. Проектирование
     1. Схема архитектуры программы

В проекте была использована клиент-серверная архитектура. Для доступа к данным приложение подключается к базе данных с помощью строки подключения. Использованная схема архитектуры проекта приведена на Рисунке 14.

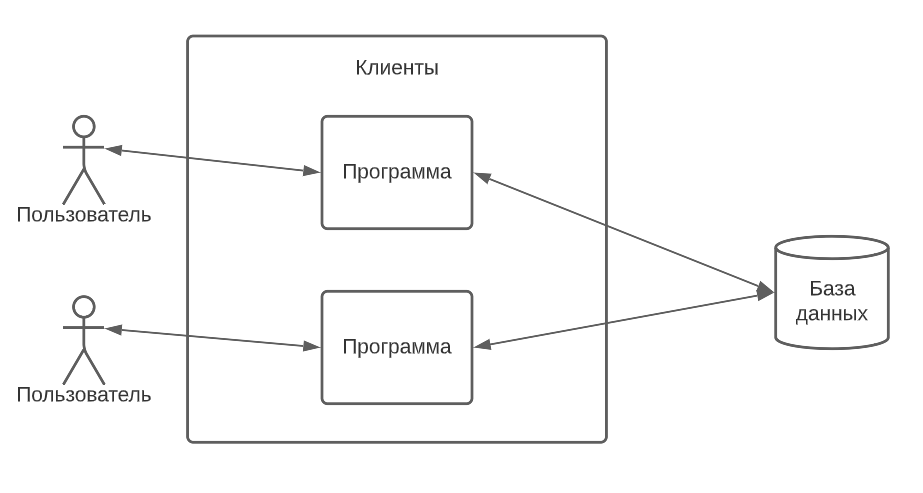


Рисунок 14 - Архитектура проекта

* + 1. Логическая схема данных

На рисунке 15 представлена логическая схема базы данных проекта.

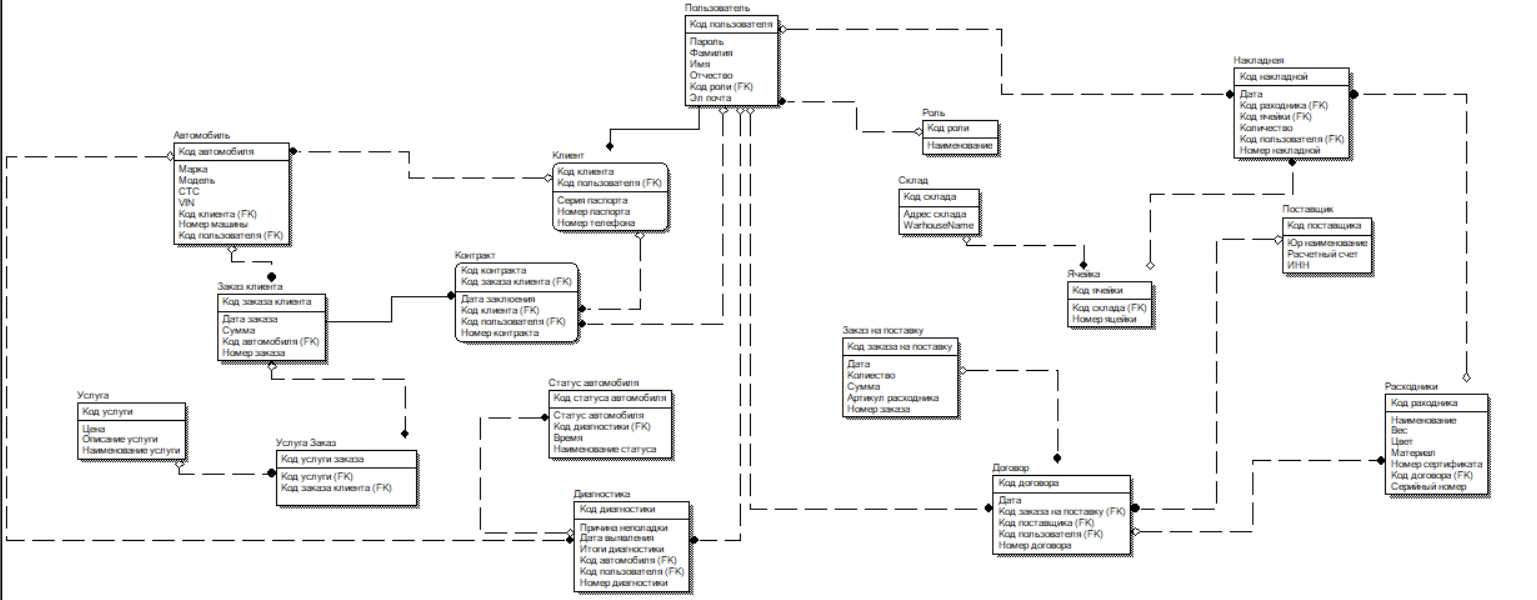


Рисунок 15 -Логическая схема БД

* + 1. Физическая схема данных

На рисунке 16 представлена физическая схема базы данных.

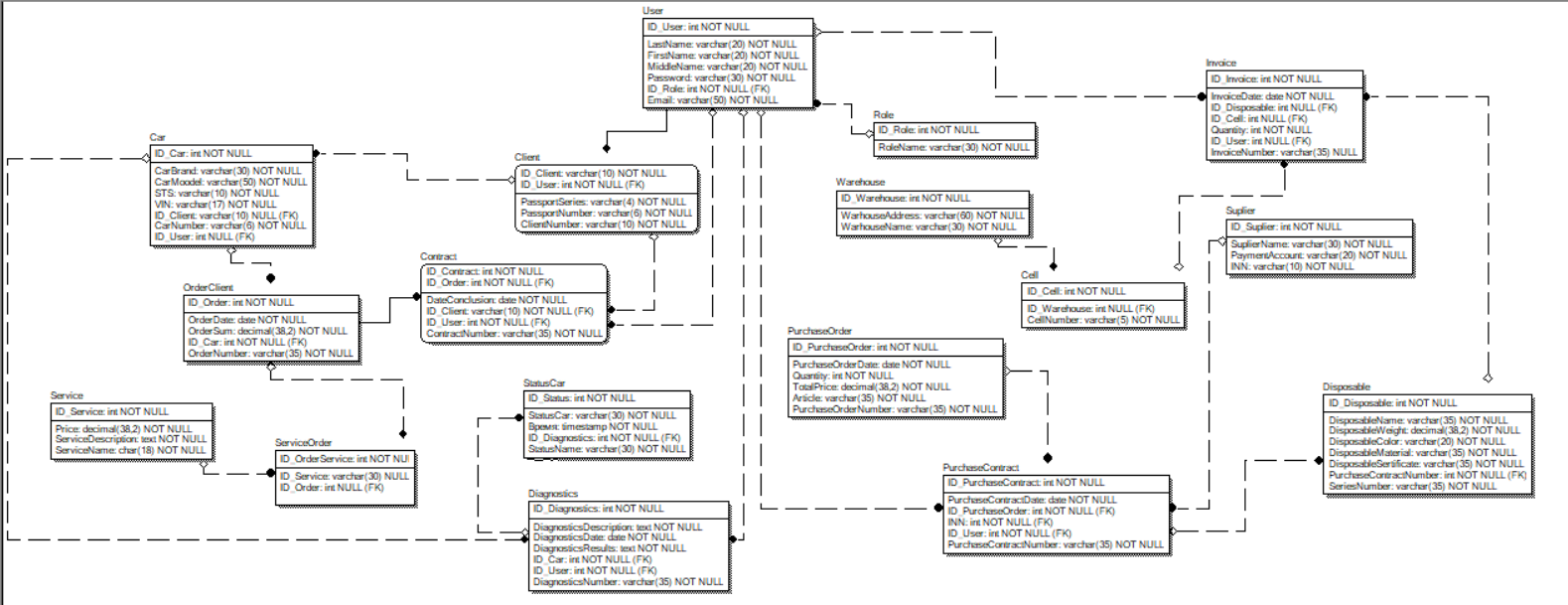


Рисунок 16 - Физическая схема БД

При проектировании базы данных использовались такие типы связей, как один к одному, один ко многим и многие ко многим.

В таблице 6 представлен словарь данных для разработанной базы данных.

Таблица 6 - Словарь данных

| Ключ | Наименование | Тип данных | Обязательность заполнения | Описание |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| **Role** | | | | |
| PK | ID\_Role | INT | Not null | Уникальный идентификатор с атрибутом auto increment |
|  | RoleName | Varchar(30) | Not null | Название роли |
| **User** | | | | |
| PK | ID\_User | INT | Not null | Уникальный идентификатор с атрибутом auto increment |
|  | LastName | Varchar(20) | Not null | Фамилия |
|  | FirstName | Varchar(20) | Not null | Имя |
|  | MiddleName | Varchar(20) | Not null | Отчество |
|  | Email | Varchar(50) | Not null | Почта |
|  | Password | Varchar(35) | Not null | Пароль |
| FK | ID\_Role | INT | Not null | Должность |
| **Client** | | | | |
| PK | ID\_Client | INT | Not null | Уникальный идентификатор с атрибутом auto increment |
|  | PassportSeries | Varchar(4) | Not null | Серия паспорта |
|  | PassportNumber | Varchar(6) | Not null | Номер паспорта |
|  | ClientNumber | Varchar(10) | Not null | Номер телефона |
| FK | ID\_User | INT | Not null | Пользователь |
| **Car** | | | | |
| PK | ID\_Car | INT | Not null | Уникальный идентификатор с атрибутом auto increment |
|  | CarBrand | Varchar(30) | Not null | Марка |
|  | CarMoodel | Varchar(50) | Not null | Модель |
|  | STS | Varchar(10) | Not null | СТС |
|  | VIN | Varchar(17) | Not null | VIN |
|  | CarNumber | Varchar(6) | Not null | Номер машины |
| FK | ID\_Client | INT | Not null | Клиент |
| FK | ID\_User | INT | Not null | Пользователь |
| **OrderClient** | | | | |
| PK | ID\_Order | INT | Not null | Уникальный идентификатор с атрибутом auto increment |
|  | OrderDate | Date | Not null | Дата заказа |
|  | OrderSum | Decimal(38,2) | Not null | Сумма |
|  | OrderNumber | Varchar(35) | Not null | Номер заказа |
| FK | ID\_Car | INT | Not null | Машина |
| **Contract** | | | | |
| PK | ID\_Contract | INT | Not null | Уникальный идентификатор с атрибутом auto increment |
|  | DateConclusion | Date | Not null | Дата заключения |
|  | ContractNumber | Varchar(35) | Not null | Номер контракта |
| FK | ID\_Order | INT | Not null | Заказ |
| FK | ID\_User | INT | Not null | Пользователь |
| FK | ID\_Client | INT | Not null | Клиент |
| **Service** | | | | |
| PK | ID\_Service | INT | Not null | Уникальный идентификатор с атрибутом auto increment |
|  | Price | Decimal(38,2) | Not null | Цена |
|  | ServiceDescription | Text | Not null | Описание |
|  | ServiceName | Char(18) | Not null | Название |
| **ServiceOrder** | | | | |
| PK | ID\_OrderService | INT | Not null | Уникальный идентификатор с атрибутом auto increment |
| FK | ID\_Service | Int | Null | Услуга |
| FK | ID\_Order | Int | Null | Заказ |
| **Diagnostics** | | | | |
| PK | ID\_Diagnostics | INT | Not null | Уникальный идентификатор с атрибутом auto increment |
|  | DiagnosticsDescription | Text | Not null | Описание |
|  | DiagnosticsDate | Date | Not null | Дата диагностики |
|  | DiagnosticsResults | Text | Not null | Результаты |
|  | DiagnosticsNumber | Varchar(35) | Not null | Номер диагностики |
| FK | ID\_Car | Int | Not null | Машина |
| FK | ID\_User | Int | Not null | Пользователь |
| **StatusCar** | | | | |
| PK | ID\_Status | INT | Not null | Уникальный идентификатор с атрибутом auto increment |
|  | StatusCar | Varchar(30) | Not null | Статус |
|  | StatusTime | Timestamp | Not null | Время |
|  | StatusName | Varchar(30) | Not null | Название статуса |
| FK | ID\_Diagnostics | Int | Not null | Диагностика |
| **Warehouse** | | | | |
| PK | ID\_Warehouse | INT | Not Null | Уникальный идентификатор с атрибутом auto increment |
|  | WarehouseAddress | Varchar(60) | Not null | Адрес |
|  | WarehouseName | Varchar(30) | Not null | название |
| **Cell** | | | | |
| PK | ID\_Cell | INT | Not Null | Уникальный идентификатор с атрибутом auto increment |
|  | CellNumber | Varchar(5) | Null | Номер ячейки |
| FK | ID\_Warehouse | INT | Null | Склад |
| **PurchaseOrder** | | | | |
| PK | ID\_PurchaseOrder | INT | Not Null | Уникальный идентификатор с атрибутом auto increment |
|  | PurchaseOrderDate | Date | Not Null | Дата закупки |
|  | Quantity | INT | Not Null | Количество расходников |
|  | TotalPrice | Decimal(38,2) | Not Null | Итого |
|  | Article | Varchar(35) | Not Null | АРтикул |
|  | PurchaseOrderNumber | Varchar(35) | Not null | Номер закупки |
| **Suplier** | | | | |
| PK | ID\_Suplier | INT | Not Null | Уникальный идентификатор с атрибутом auto increment |
|  | SuplierName | Varchar(30) | Not null | Наименование |
|  | PaymentAccount | Varchar(20) | Not null | Расчётный счет |
|  | INN | Varchar(10) | Not null | ИНН |
| PurchaseContract | | | | |
| PK | ID\_PurchaseContract | INT | Not Null | Уникальный идентификатор с атрибутом auto increment |
|  | PurchaseContractDate | Date | Not null | Дата контракта |
|  | PurchaseContractNumber | Varchar(35) | Not null | Номер контракта |
| FK | ID\_PurchaseOrder | Int | Not null | Закупка |
| FK | INN | Int | Not null | Поставщик |
| FK | ID\_User | Int | Not null | Пользователь |
| **Disposable** | | | | |
| PK | ID\_Disposable | INT | Not Null | Уникальный идентификатор с атрибутом auto increment |
|  | DisposableName | Varchar(35) | Not null | Название |
|  | DisposableWeight | Decimal(38,2) | Not null | Вес |
|  | DisposableColor | Varchar(20) | Not null | Цвет |
|  | DisposableMaterial | Varchar(35) | Not null | Материал |
|  | DisposableSertificate | Varchar(35) | Not null | Сертификат |
| FK | PurchaseContractNumber | Int | Not null | Номер контракта |
|  | SeriesNumber | Varchar(35) | Not null | Серийный номер |
| Invoice | | | | |
| PK | ID\_Invoice | INT | Not Null | Уникальный идентификатор с атрибутом auto increment |
|  | InvoiceDate | Date | Not null | Дата накладной |
| FK | ID\_Disposable | Int | Null | Расходник |
| FK | ID\_Cell | Int | Null | Ячейка |
|  | Quantity | Int | Not null | Количество |
| FK | ID\_User | Int | Null | Пользователь |
|  | InvoiceNumber | Varchar(35) | Null | Номер накладной |

* + 1. Структурная схема программы

На рисунке 17 представлена структурная схема программы.

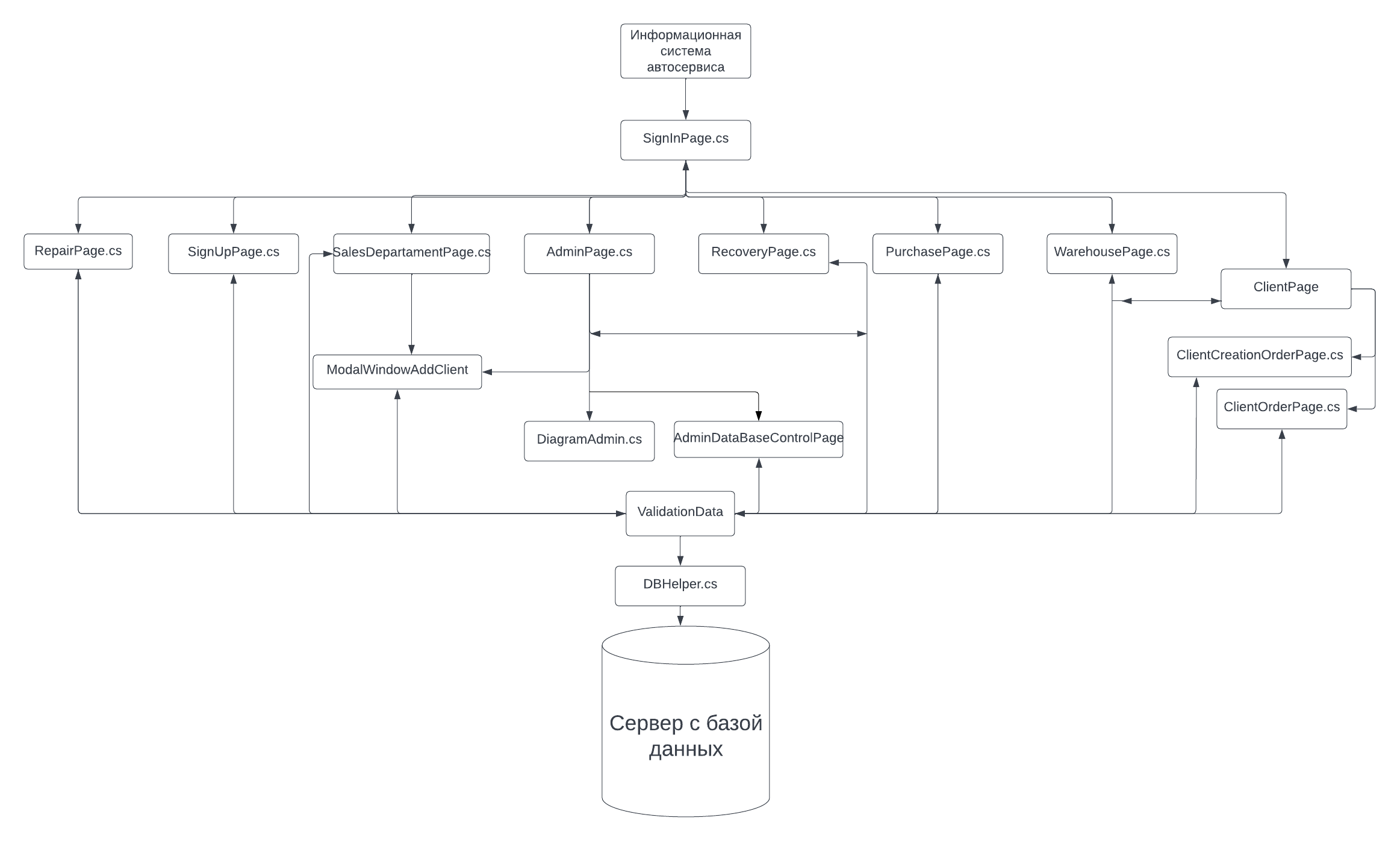


Рисунок 17 - Структурная схема

Таблица 7 - Модули

| № | Название модуля | Описание модуля |
| --- | --- | --- |
| 1 | 2 | 3 |
| 1 | AdminDataBaseControlPage.xaml.cs | Создание резервных копий и восстановление данных |
| 2 | AdminPage.xaml.cs | Управление пользователями |
| 3 | ClientOrderCreationPage.xaml.cs | Оформление заказа клиентом |
| 4 | ClientOrderPage.xaml.cs | Отображение оформленных заказов клиентом |
| 5 | ClientPage.xaml.cs | Управление машинами клиента |
| 6 | DBHelper.cs | Формирование запросов в БД |
| 7 | DiagramAdmin.xaml.cs | Формирование графиков данных |
| 8 | EmailHelper.cs | Работа с почтовым клиентом |
| 9 | FrameManager.cs | Маршрутизация приложения |
| 10 | MainWindow.xaml.cs | Инициализация главного окна |
| 11 | ModalWindowAddClient.xaml.cs | Заполнение дополнительных данных |
| 12 | PurchasePage.xaml.cs | Управление данными связанными с поставкой |
| 13 | RecoveryPage.xaml.cs | Восстановление пароля |
| 14 | RepairPage.xaml.cs | Управление данными связанными с ремонтом |
| 15 | SalesDepartmentPage.xaml.cs | Управление данными связанными с продажами |
| 16 | SignInPage.xaml.cs | Авторизация |
| 17 | SignUpPage.xaml.cs | Регистрация |
| 18 | ValidationData.cs | Валидация данных |
| 19 | WarehousePage.xaml.cs | Управление данными связанными со складским учетом |

* + 1. Функциональная схема

На Рисунке 18 представлена функциональная схема информационной системы магазина одежды. На данной схеме графически продемонстрированы все доступные функции каждого из окон.

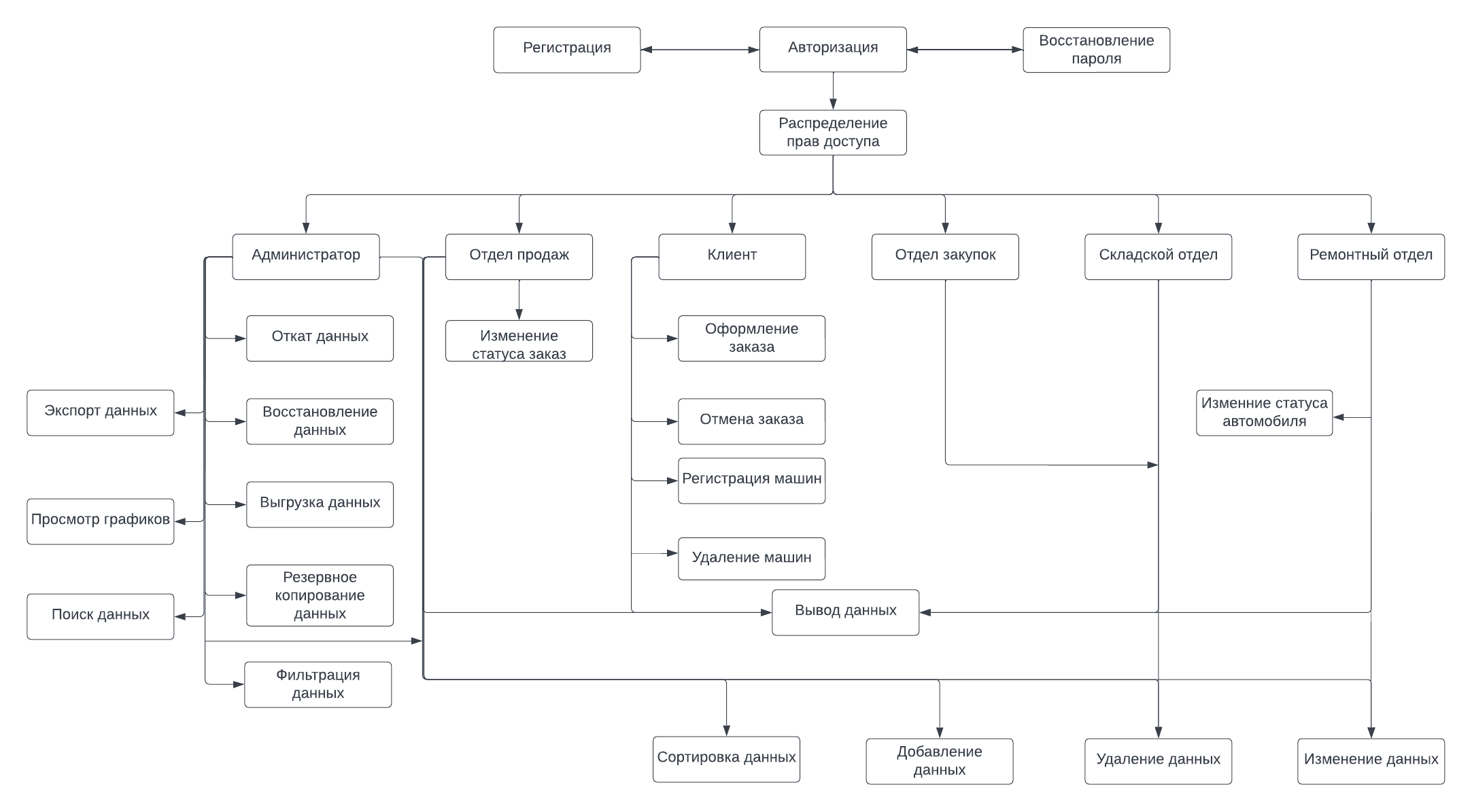


Рисунок 18 - Функциональная схема

* + 1. Диаграмма классов

На рисунке 19 представлена диаграмма классов разработанного проекта.

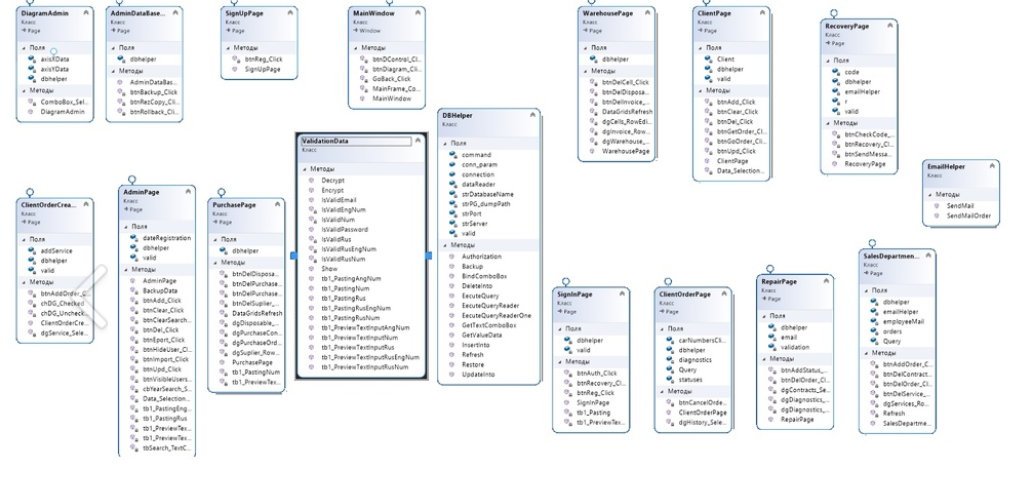


Рисунок 19 - Диаграмма классов

* + 1. Схема пользовательского интерфейса

На рисунке 19 представлена схема пользовательского интерфейса. На ней отражены окна и переходы между окнами.

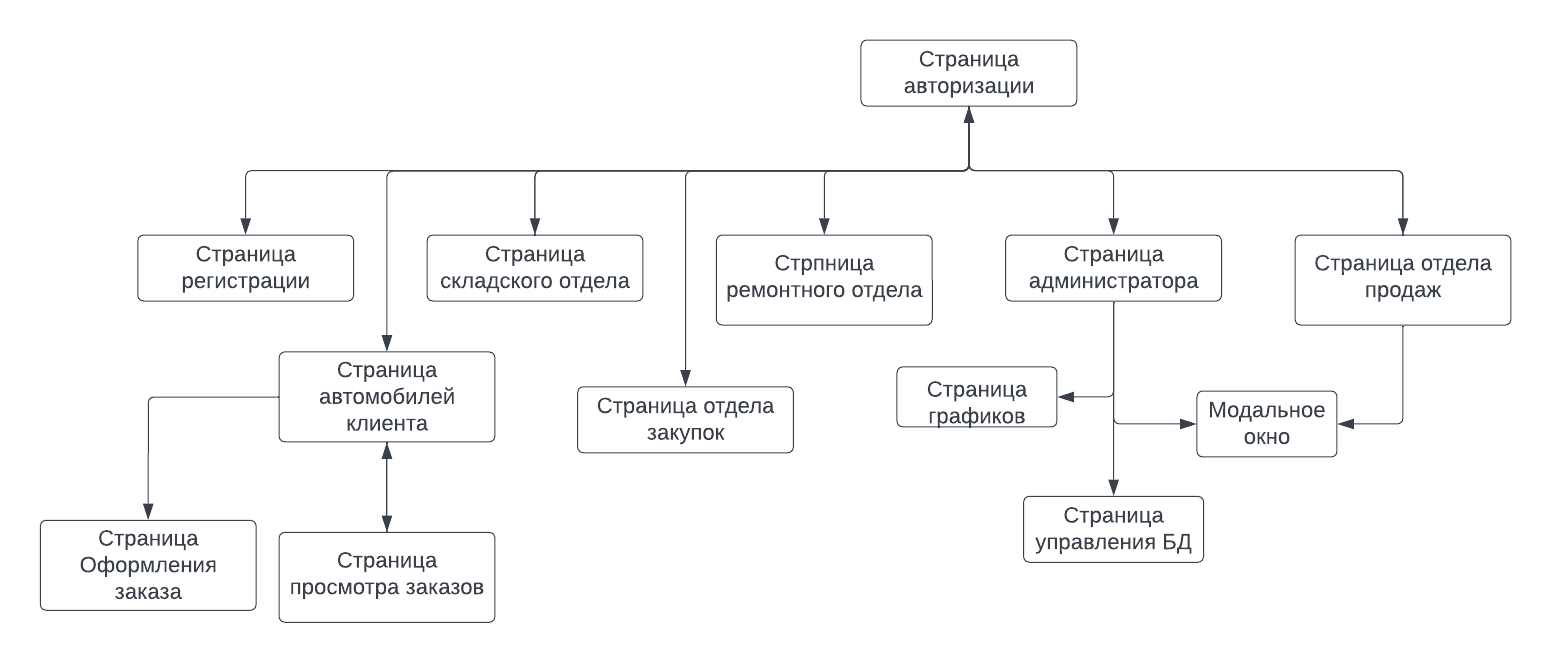


Рисунок 20 - Схема пользовательского интерфейса

* 1. Результаты работы программы

В ходе выполнения поставленной задачи было разработано программное решение состоящее из базы данных и клиентского приложения информационной системы автосервиса.

На рисунках, представленных ниже отражены результаты работы программы. Показаны окна авторизации, где пользователи могут получить доступ к программе, и окно автомобилей клиента, в котором он может добавлять, изменять или удалять данные об автомобилях.

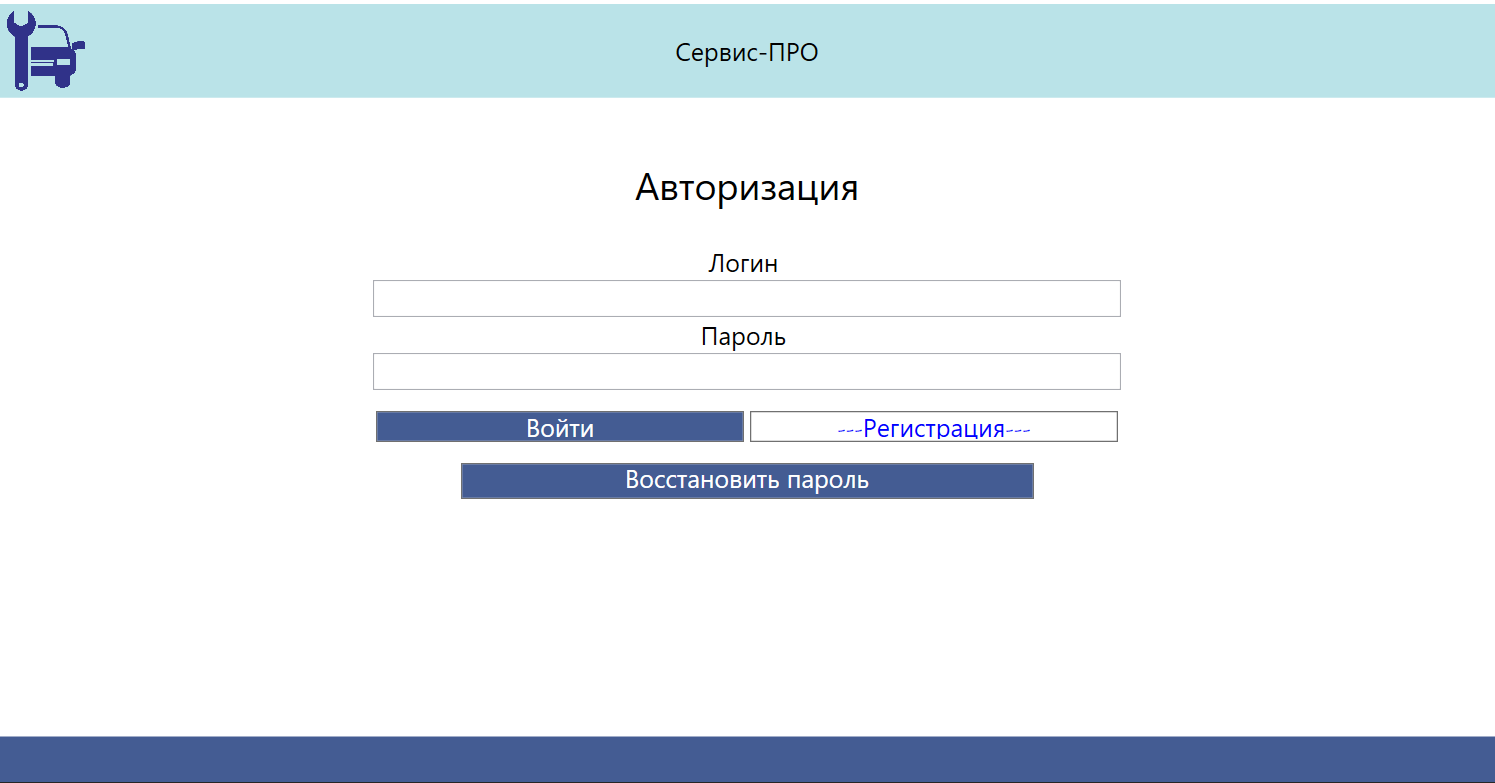


Рисунок 21 - Авторизация

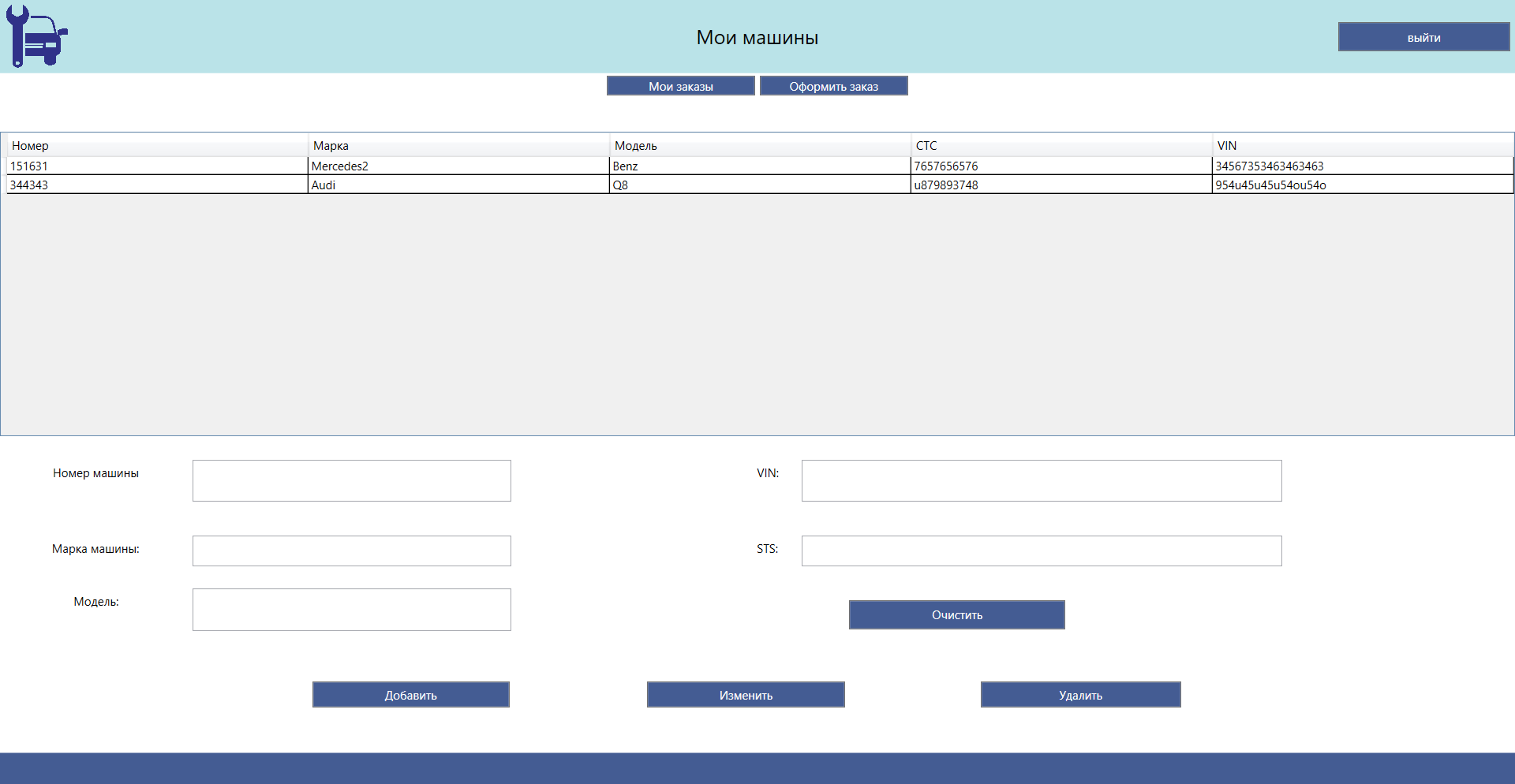


Рисунок 22 - Страница клиента

1. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ
   1. Инструментальные средства разработки

Для разработки программного решения использовался язык C#. Данный язык отличается своей универсальностью и отлично подходит для реализации бизнес решений.

Для реализации интерфейса использовался встроенный в NET.Framework язык разметки XAML, который позволяет с помощью тегов и вызываемых свойств у различных объектов реализовывать красивый и функциональный интерфейс для программного обеспечения.

Для работы с базой данных использовалась СУБД PgAdmin 4, которая позволяет администрировать и разрабатывать базы данных. Для разработки реляционной базы данных использовался язык PostgreSQL.

Для написания исходного (программного) кода использовалась среда разработки Microsoft Visual Studio 2019. Платформа обладает обширным функционалом, таким как проведение тестирования и профилировка производительности. Кроме того, среда позволяет подключать библиотеки для расширения набора инструментов.

* 1. Отладка программы

Для отладки проекта был использован встроенный инструментарий среды разработки. Были использованы такие инструменты, как точка остановы и профилировщик производительности. Точка остановы использовалась при перехвате ошибок, а профилировщик производительности отображал потребление ресурсов персонального компьютера при работе программы. Это необходимо для выявления избыточного потребления ресурсов и оптимизации программного продукта. На рисунке ниже представлен пример его работы.

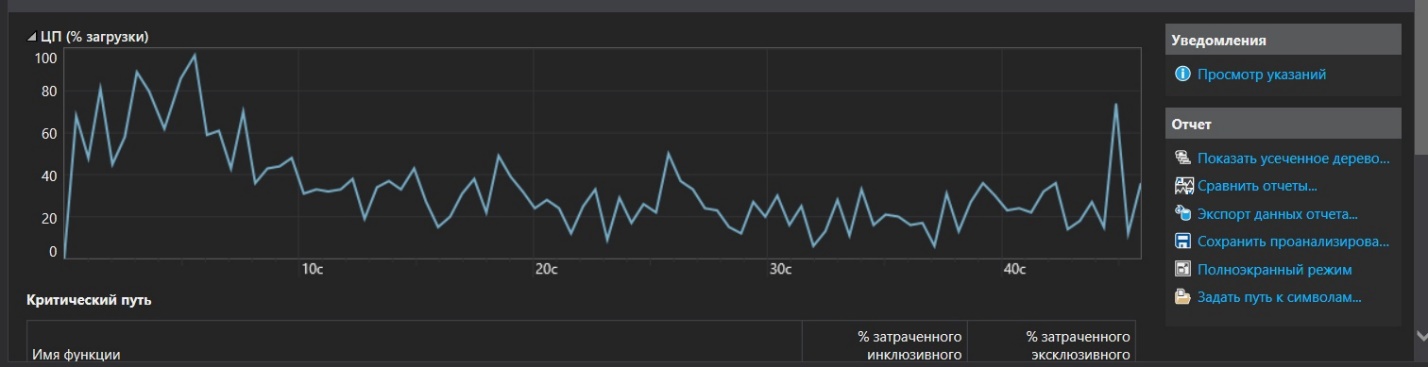


Рисунок 23 - Профилировщик

В процессе разработки программного решения часто возникали различные ошибки. Пример самой распространенной ошибки представлен на рисунке ниже.

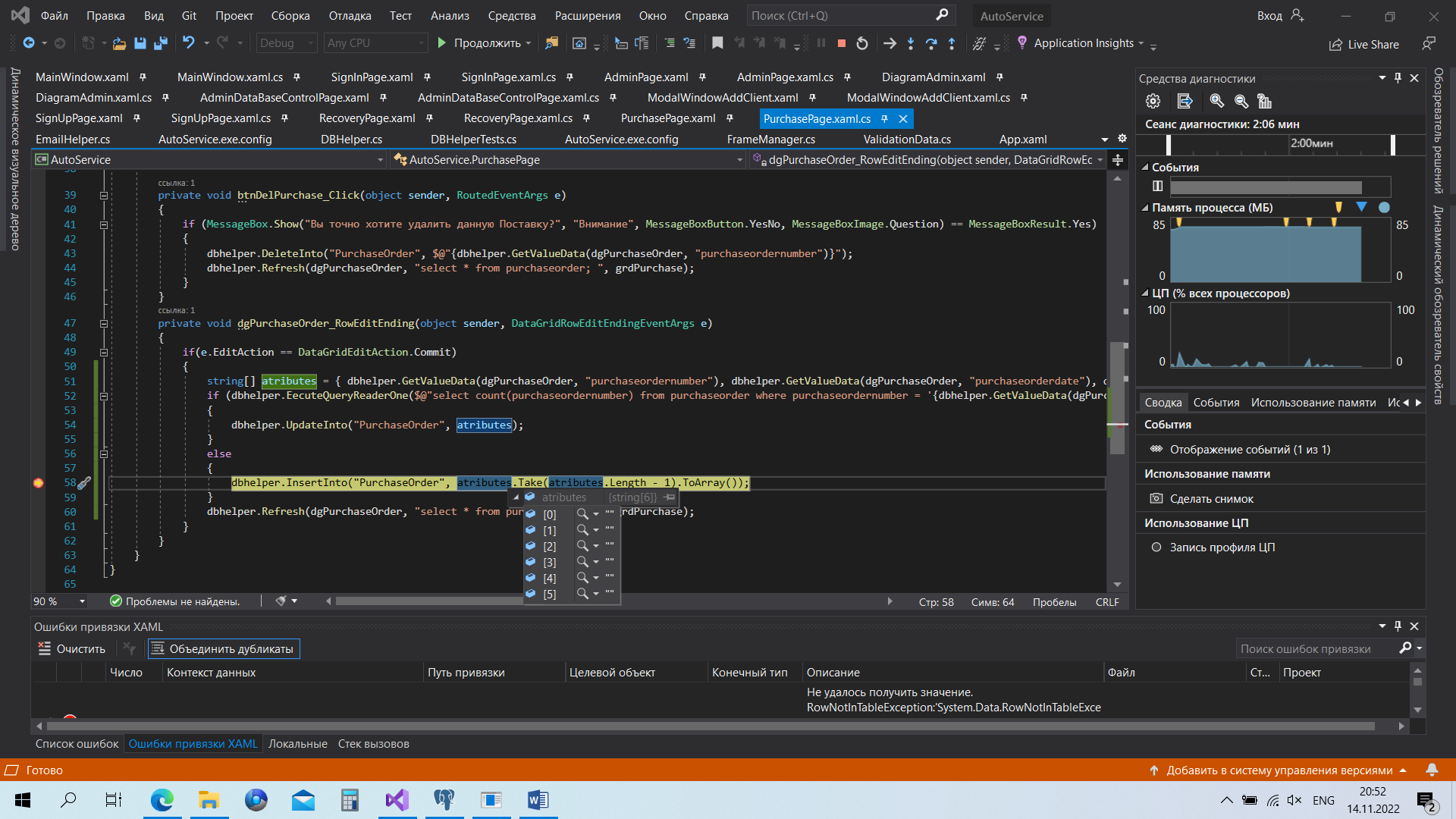


Рисунок 24 - Ошибка

* 1. Защитное программирование

Во избежание ввода некорректных данных в базу данных был создан специальный валидационный класс.

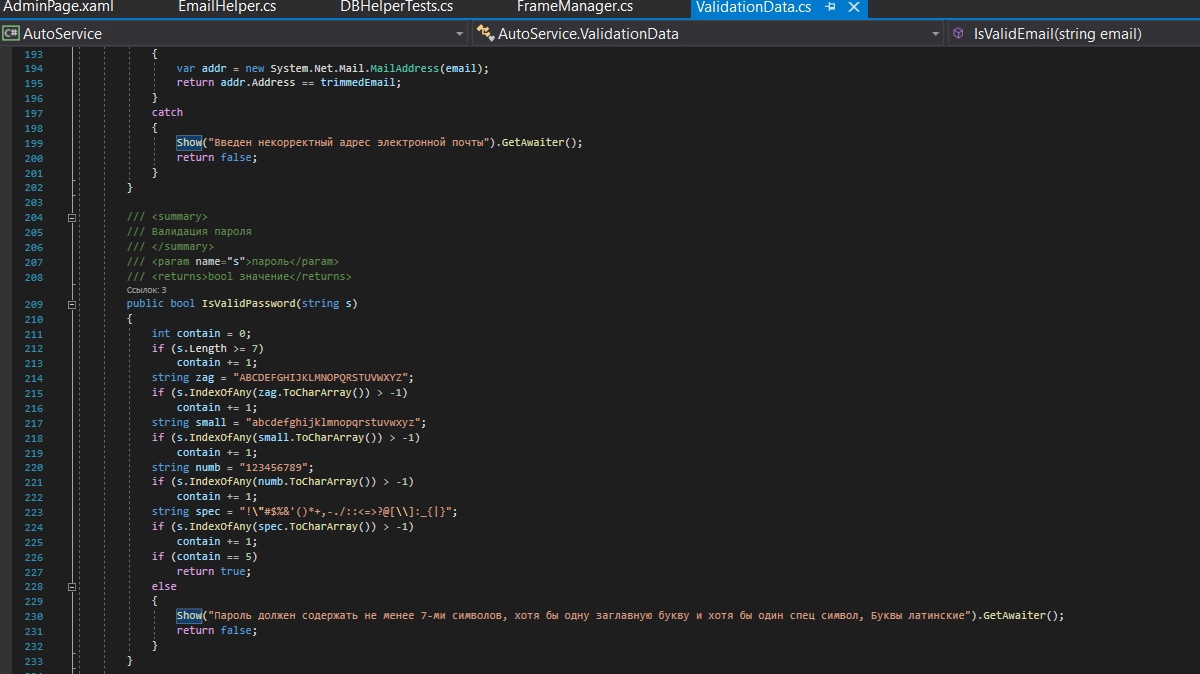


Рисунок 25 - Валидация

* 1. Характеристики программы

Информационная система автосервиса является компьютерной программой, которая может запускаться на операционной системы Windows 8.1 и выше. Основным критерием для запуска программы является нахождение в одной локальной сети с компьютером, на котором развернут сервер с базой данных.

Таблица 9 - Модули

| № | Название модуля | Описание модуля | Размер модуля в Кб | Кол-во строк |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 1 | AdminDataBaseControlPage.xaml | Отображение страницы управления БД | 1.6 kB | 33 |
| 2 | AdminDataBaseControlPage.xaml.cs | Создание резервных копий и восстановление данных | 2.2 kB | 68 |
| 3 | AdminPage.xaml | Отображение страницы администратора | 5.5 kB | 86 |
| 4 | AdminPage.xaml.cs | Управление пользователями | 12.0 kB | 210 |
| 5 | ClientOrderCreationPage.xaml | Отображение страницы оформления заказа клиентом | 4.9 kB | 71 |
| 6 | ClientOrderCreationPage.xaml.cs | Оформление заказа клиентом | 3.7 kB | 84 |
| 7 | ClientOrderPage.xaml | Отображения страницы заказов клиента | 3.0 kB | 61 |
| 8 | ClientOrderPage.xaml.cs | Отображение оформленных заказов клиентом | 4.7 kB | 99 |
| 9 | ClientPage.xaml | Отображение страницы автомобилей клиента | 4.7 kB | 73 |
| 10 | ClientPage.xaml.cs | Управление машинами клиента | 4.1 kB | 108 |
| 11 | DBHelper.cs | Формирование запросов в БД | 13.9 kB | 404 |
| 12 | DiagramAdmin.xaml | Отображение графика статистики пользователей | 3.2 kB | 69 |
| 13 | DiagramAdmin.xaml.cs | Формирование графиков данных | 4.2 kB | 85 |
| 14 | EmailHelper.cs | Работа с почтовым клиентом | 3.0 kB | 75 |
| 15 | FrameManager.cs | Маршрутизация приложения | 264 B | 14 |
| 16 | MainWindow.xaml | Главное окно | 4.0 kB | 60 |
| 17 | MainWindow.xaml.cs | Инициализация главного окна | 3.4 kB | 95 |
| 18 | ModalWindowAddClient.xaml | Модальное окно для дополнительных данных | 2.0 kB | 40 |
| 19 | ModalWindowAddClient.xaml.cs | Заполнение дополнительных данных | 2.7 kB | 80 |
| 20 | PurchasePage.xaml | Отображение страницы отдела закупок | 14.2 kB | 263 |
| 21 | PurchasePage.xaml.cs | Управление данными связанными с поставкой | 8.6 kB | 162 |
| 22 | RecoveryPage.xaml | Отображение страницы восстановления пароля | 3.7 kB | 58 |
| 23 | RecoveryPage.xaml.cs | Восстановление пароля | 2.6 kB | 76 |
| 24 | RepairPage.xaml | Отображение страницы ремонтного отдела | 5.4 kB | 92 |
| 25 | RepairPage.xaml.cs | Управление данными связанными с ремонтом | 5.4 kB | 105 |
| 26 | SalesDepartmentPage.xaml | Отображение страницы ремонтного отдела | 7.0 kB | 128 |
| 27 | SalesDepartmentPage.xaml.cs | Управление данными связанными с продажами | 5.5 kB | 108 |
| 28 | SignInPage.xaml | Отображение страницы авторизации | 2.8 kB | 42 |
| 29 | SignInPage.xaml.cs | Авторизация | 3.2 kB | 84 |
| 30 | SignUpPage.xaml | Отображение страницы регистрации | 4.7 kB | 73 |
| 31 | SignUpPage.xaml.cs | Регистрация | 2.0 kB | 53 |
| 32 | ValidationData.cs | Валидация данных | 8.7 kB | 242 |
| 33 | WarehousePage.xaml | Отображение страницы кладовщика | 6.7 kB | 125 |
| 34 | WarehousePage.xaml.cs | Управление данными связанными со складским учетом | 5.7 kB | 117 |

ЗАКЛЮЧЕНИЕ

В ходе проделанной работы удалось реализовать информационную систему автосервиса, соответствующую всем заявленным требованиям. Также в ходе работы удалось существенно попвысить навыки работы с языком C# и postgreSQL. Информационную систему можно развивать путем добавления новых функций и возможностей.

В ходе анализа требований были выявлены цель разработки и функции и возможности, которые должны быть предоставлены программным изделием.

При проектировании программы была разработана диаграммы бизнес-процессов предприятия до внедрения и после внедрения. Они были разработаны с использованием методологии IDEF0. Кроме того, были разработаны функциональная, структурная и интерфейсная схемы программы.

При разработке были использованы определенные при анализе технические и инструментальные средства разработки, а также определены методы разработки, после чего были полностью реализованы клиентское приложение и скрипт базы данных.

Также была разработана схема тестирования. Кроме того, программное изделие было протестировано методами белого и черного ящиков. Также был разработан сценарий тестовых испытаний. Разработанное программное изделие успешно прошло тестирование.

Был создан инсталяционный пакет, содержащий программное изделие и разработано руководство пользователя для помощи в работе с информационной системой.

СПИСОК ИСПОЛЬЗОВАННЫХ МАТЕРИАЛОВ

1. C# API-[Электронный ресурс]-Режим доступа: https://learn.microsoft.com/ru-ru/aspnet/core/tutorials/first-web-api?view=aspnetcore-6.0&tabs=visual-studio (Дата обращения: 10.10.2022)
2. REST API – [Электронный ресурс] – а Режим доступа:  [https://habr.com/ru/post/531106/r](https://assetstore.unity.com/packages/2d/gui/icons/trollnest-free-ui-buttons-140934#publisher) (Дата обращения: 10.10.2022)
3. Создание простейшего API-[Электронный ресурс]-Режим доступа: https://metanit.com/sharp/aspnet6/2.11.php (Дата обращения: 10.10.2022)
4. How to use API with C#-[Электронный ресурс]-Режим доступа: https://rapidapi.com/blog/how-to-use-an-api-with-c-sharp/ (Дата обращения: 10.10.2022)
5. Data Export-[Электронный ресурс]-Режим доступа: https://stackoverflow.com/questions/8207869/how-to-export-datatable-to-excel (Дата обращения: 20.10.2022)
6. Excel Export-[Электронный ресурс]-Режим доступа:  [https://blog.conholdate.com/total/export-data-to-excel-in-csharp/l](https://www.internet-technologies.ru/articles/sozdaem-8-bitnuyu-pikselnuyu-grafiku-s-pomoschyu-photoshop-chast-1.html) (Дата обращения: 20.10.2022)
7. Data Export-[Электронный ресурс]-Режим доступа: https://c-sharp.pro/%D1%8D%D0%BA%D1%81%D0%BF%D0%BE%D1%80%D1%82-%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85-%D0%B2-microsoft-excel-netcore-%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82/ (Дата обращения: 20.10.2022)
8. Youtube -[Электронный ресурс]-Режим доступа: https://www.youtube.com/watch?v=JiEKqLdvnyY (Дата обращения: 20.10.2022)
9. ГОСТ 19404- 79 ЕСПД. Пояснительная записка. ПЕРЕИЗДАНИЕ Января 2010г.
10. ГОСТ 7.80-2000 СИБИД. Библиографическая запись. Заголовок. Общие требования и правила составления.
11. ГОСТ Р 7.0.5-2008 БИБЛИОГРАФИЧЕСКАЯ ССЫЛКА. Общие требования и правила составления.
12. ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов.
13. ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных документов.
14. ГОСТ 19.105-78 ЕСПД. Общие требования к программным документам.
15. [Стиллмен Э., Грин Дж.](https://www.chitai-gorod.ru/books/authors/stillmen_e_grin_dzh/) Изучаем C#. 3-е издание. Питер СПб 2014, 816с.
16. [Роберт Мартин,](https://www.litres.ru/robert-s-martin/) [Мика Мартин.](https://www.litres.ru/mika-martin/) Принципы, паттерны и методики гибкой разработки на языке C#. 715с.
17. Адитья Бхаргава. Грокаем алгоритмы. Издательский дом «Питер», 290с.
18. Электронная свободная энциклопедия [Электронный ресурс] URL: [https://ru.wikipedia.org/wiki/Microsoft\_Visual\_Studio](https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio%20)
19. Электронная свободная энциклопедия [Электронный ресурс] URL: [https://ru.wikipedia.org/wiki/Microsoft\_Word](https://ru.wikipedia.org/wiki/Microsoft_Word%20)