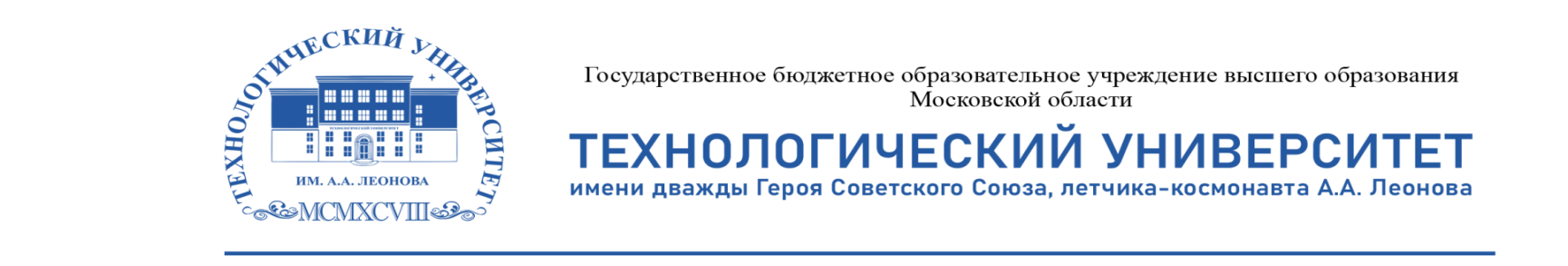
****

Колледж космического машиностроения и технологии

**Курсовой проект**

**Разработка системы классов для приложения «Автосалон»**

Пояснительная записка

КП.09.02.03.22.019ПЗ

Обучающийся группы П2-19 Ухорский Д. Н.

Руководитель курсового проекта Гусятинер Л. Б.

Результат защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Королев, 2022 г.

**ОГЛАВЛЕНИЕ**

[**1. Введение 4**](#_Toc107772412)

[**2. Теоретическая часть 5**](#_Toc107772413)

[**2.1. Описание предметной области 5**](#_Toc107772414)

[**2.2. Описание существующих разработок 7**](#_Toc107772415)

[**3. Теоретическая часть 13**](#_Toc107772416)

[**3.1. Построение диаграммы прецедентов 13**](#_Toc107772417)

[**3.2. Выбор инструментов 14**](#_Toc107772418)

[**3.2.1. Язык программирования 14**](#_Toc107772419)

[**3.2.2. Среда разработки 15**](#_Toc107772420)

[**3.2.3. База данных 15**](#_Toc107772421)

[**3.2.4. Графический интерфейс 15**](#_Toc107772422)

[**3.3. Проектирование сценария 16**](#_Toc107772423)

[**3.4. Построение диаграммы классов 17**](#_Toc107772424)

[**3.5. Описание главного модуля 21**](#_Toc107772425)

[**3.6. Описание спецификаций к модулям 22**](#_Toc107772426)

[**3.7. Описание модулей 23**](#_Toc107772427)

[**3.8. Описание текстовых наборов модулей 24**](#_Toc107772428)

[**3.9. Описание применения средств откладки 25**](#_Toc107772429)

[**4. Эксплуатационная часть 26**](#_Toc107772430)

[**4.1. Руководство оператора 26**](#_Toc107772431)

[**4.1.1. Назначение программы 26**](#_Toc107772432)

[**4.1.2. Условия запуска программы 26**](#_Toc107772433)

[**4.1.3. Выполнение программы 26**](#_Toc107772434)

[**4.1.4. Сообщение оператору 27**](#_Toc107772435)

[**4.2. To-Do лист 28**](#_Toc107772436)

[**5. Заключение 29**](#_Toc107772437)

[**6. Список использованной литературы и интернет-ресурсов 30**](#_Toc107772438)

[**7. Приложения 31**](#_Toc107772439)

[**7.1. Приложение 1 31**](#_Toc107772440)

[**7.2. Приложение 2 31**](#_Toc107772441)

[**7.3. Приложение 3 32**](#_Toc107772442)

[**7.4. Приложение 4 32**](#_Toc107772443)

[**7.5. Приложение 5 33**](#_Toc107772444)

[**7.6. Приложение 6 33**](#_Toc107772445)

[**7.7. Приложение 7 34**](#_Toc107772446)

[**7.8. Приложение 8 34**](#_Toc107772447)

[**7.9. Приложение 9 35**](#_Toc107772448)

[**7.10. Приложение 10 35**](#_Toc107772449)

[**7.11. Приложение 11 36**](#_Toc107772450)

# **Введение**

Данный курсовой проект заключается в разработке системы классов для приложения «Автосалон». Разработка приложения будет происходить на языке программирования Python и фреймворка PyQt.

В первой части будут рассмотрены предметная область и существующие продукты по данной теме.

Во второй части будут описаны инструменты разработки программы и модули программы.

В третьей части описаны взаимодействие пользователя с программой и сообщения оператору.

В заключительной части будут приведены общие выводы по проекту.

# **Теоретическая часть**

## **Описание предметной области**

**Настольное приложение** – это программа, которая работает на компьютере пользователя. Она функционирует на стороне клиента и запускается в виде обыкновенного исполняемого файла.

Настольное приложение реализует следующие возможности:

1. Работа приложения как с подключением, так и без подключения к интернету;
2. Приложение запускается с загрузкой данных из памяти устройства, на котором установлено, не требует каждый раз загрузки параметров из сети;
3. Максимальная эффективность работы за счет возможности многопроцессорных систем;

**Автосалон –** это магазин по продаже новых или подержанный автомобилей. Здесь можно выбрать и купить машину самостоятельно или с помощью сотрудников салона. Если авто есть в наличии, можно сразу оформить его покупку и забрать из салона. Можно также оформить заказ на машину, если ее нет в наличии, и забрать при поступлении в автосалон.

Кроме продажи автомобилей, салоны часто предлагают сопутствующие услуги:

* Кредитование;
* Страхование машин;
* Обмен автомобилей;
* Выкуп авто.

Главная миссия, стоящая перед автосалоном – продажа автомобилей, привоз и продажа автозапчастей индивидуально под заказ, сервисное обслуживание, расширение рынка сбыта продукции и укрепление завоеванных позиций посредством расширения ассортимента товаров и услуг.

Программы для автосалонов нужны не только для их удобного администрирования, но и возможность управлять несколькими филиалами сразу, регулировать движение продукции, создавать отчеты, осуществлять информирование клиентской базы и выполнять множество других важных функций.

Потенциальные пользователи приложения “Автосалон” – это работники автосалонов, т. е. бухгалтерия, отдел менеджмента, отдел маркетинга.

Для работников приложение «Автосалон» выступает удобной формой для просмотра данных и работы над ними, что позволяет упростить процесс работы.

## **Описание существующих разработок**

На сегодняшний день существует целый ряд систем для работы с данными для автосалонов, но большинство из них существуют в виде сайтов.

1. Система «AltairCRM» [1] - уникальная платформа, разработанная с учетом специфики бизнеса по продаже автомобилей.

В системе «AltairCRM» вы можете:

* Реклама (рисунок 1) – получать неограниченное количество источников трафика (1). Возможность быстрой активации и деактивации источников (2). Встроенный алгоритм проверки доступности источников (3). Возможность быстрого подключения телефонии (4). Возможность подключения большого количества бренд-площадок к одному источнику (5);
* Клиенты – гибка система статусов обращения. Удобство работы с большим количеством обращений за счёт сортировки по датам, статусам и т. д.;
* Кол-центр – учет рабочего времени каждого сотрудника. Удобная навигация при работе с обращениями. Фиксация в истории. Смена статусов обращений;
* Автосалон (рисунок 2) – менеджер автосалона фиксирует факт посещения автосалона и сопровождает клиентов на этапе рассмотрения и одобрения заявок на покупку авто (1). Оператор и менеджер автосалона используют единый интерфейс (2). Менеджер автосалона осуществляет резервирование необходимых автомобилей на складе (3).
* Склад – принятие на баланс средств, доступных для резерва и продажи автомобилей. Движение машин по складу, полный список характеристик авто.

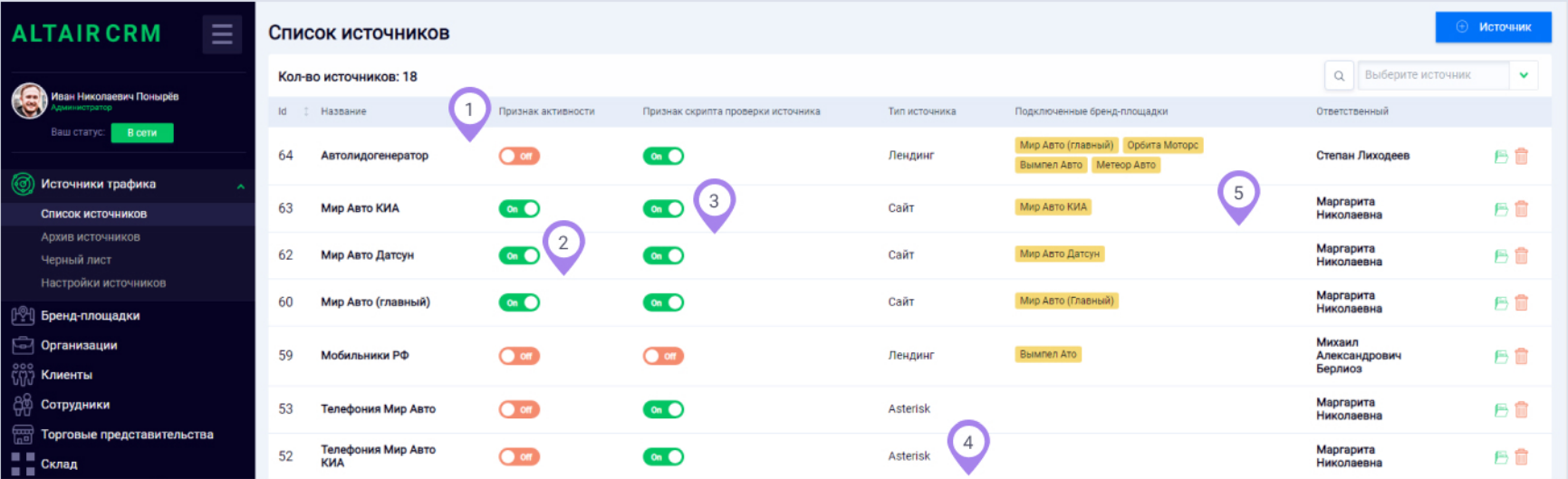


Рис.1. Реклама

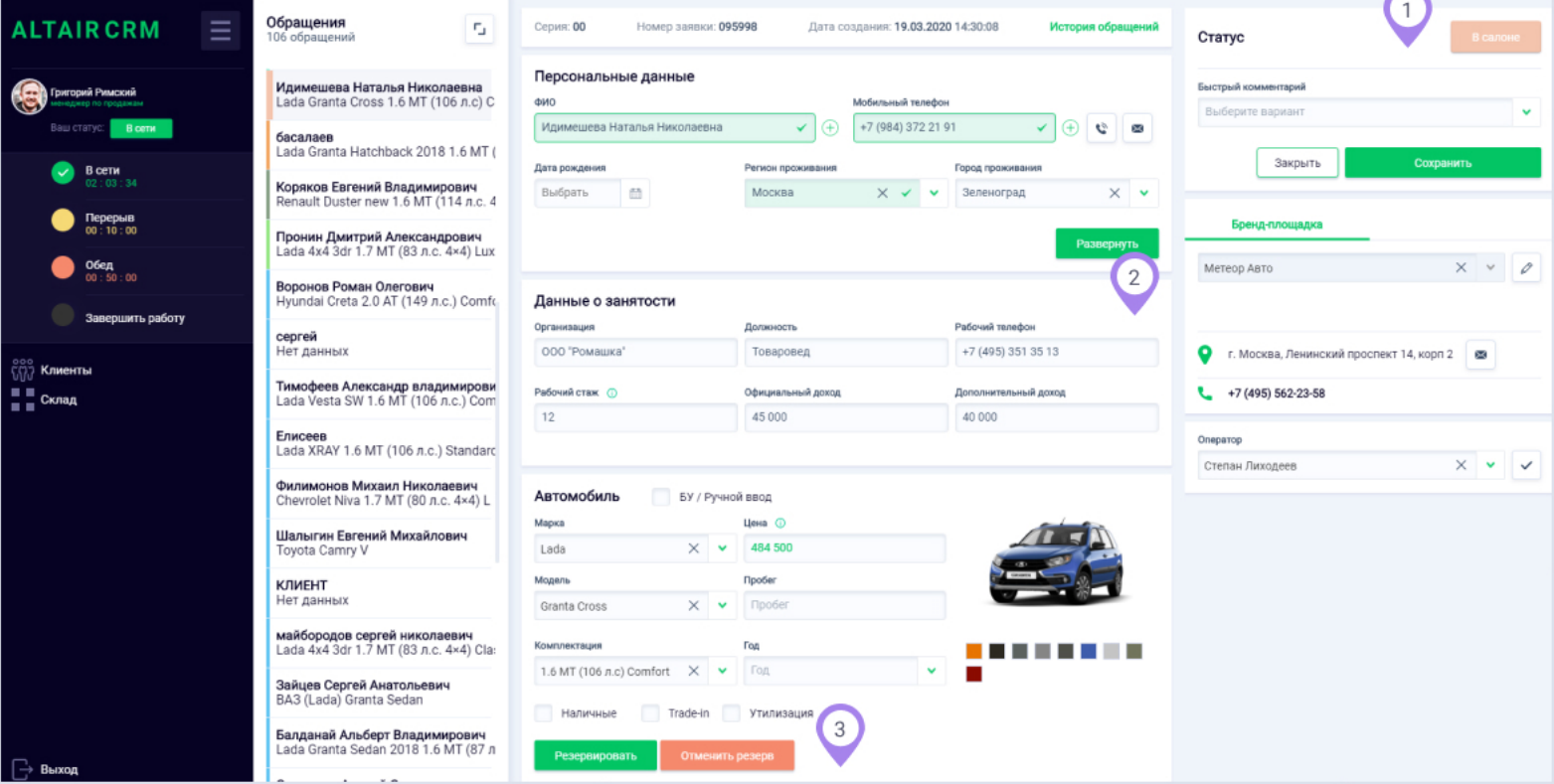


Рис.2. Автосалон

1. Система «Splus» [2] (рисунок 3) – одно из лучших систем для автосервиса ориентированная больше на клиентскую базу автосалона, автоматически напомнит клиентам о наличии невыполненных рекомендаций и ненавязчиво предложит актуализировать калькуляцию.

Преимущества использования системы «Splus»:

* Ориентированность как на покупателей, так и на удобство сотрудников;
* Кроссплатформенность (рисунок 4);
* Удобный и понятный интерфейс,

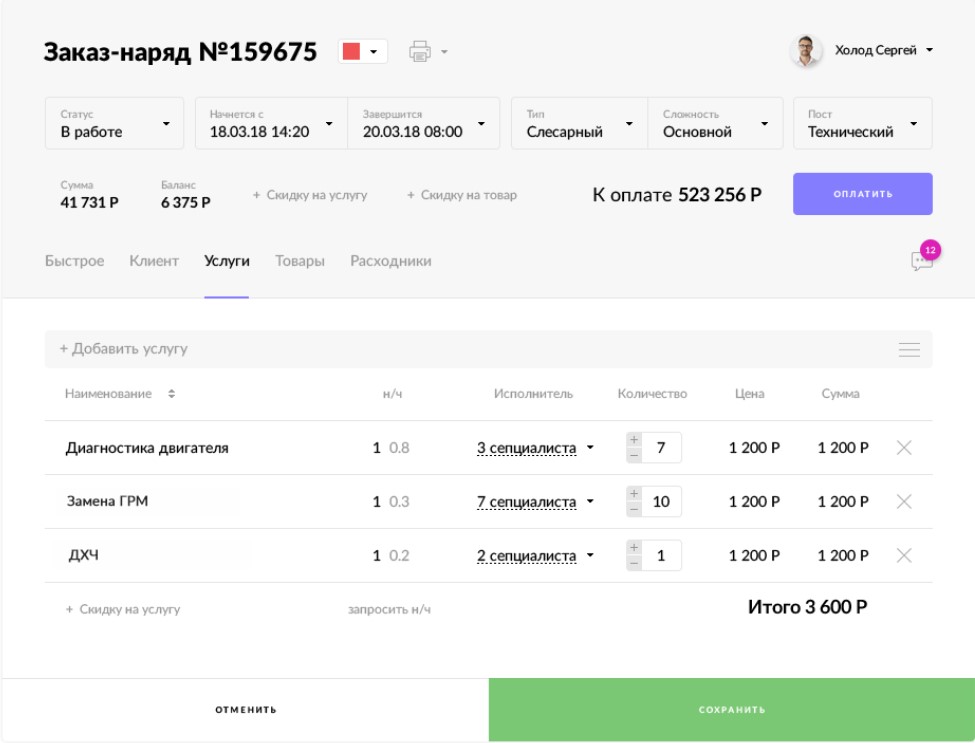


Рис.3. Основная страница «Splus»

Изображение выглядит как текст

Автоматически созданное описание

Рис.4. Кроссплатформенность «Splus»

1. Система «EasyWeek» [3] - современная платформа, которая упрощает и ускоряет путь покупателя к продавцу. При помощи сервиса EasyWeek автосалоны могут автоматизировать процессы продаж, маркетинг, управления персоналом, обслуживание клиентов и, в итоге, продавать больше автомобилей.

Преимущества использования системы «EasyWeek»:

* Учет каждого клиента (рисунок 5) – личная карточка покупателя последовательно наполняется важной информацией: контакты, заявки, покупки, программа лояльности. Сведения надежно хранятся в базе, позволяя использовать данные для анализа аудитории и подготовки рекламных стратегий.
* Генерация лидов – мгновенно реагировать на новых потенциальных клиентов с помощью таких функций, как автоответчик или рассылка.
* Улучшенная связь между автосалоном и клиентом – позволяет организовать онлайн-запись на консультацию в автосалон. Клиентам достаточно воспользоваться формой на сайте, чтобы выбрать нужно время. Не потребуется дополнительных телефонных звонков. Покупатель будет знать, что в назначенное время его будет ждать конкретный консультант. При помощи онлайн-записи автосалон сможет управлять клиентскими потоками. Отсутствие очередей и персональное внимание позволят гарантировать качественный сервис.
* Прозрачность и отчетность – с программой для автосалонов проследить за каждой сделкой можно от начала до конца, тем самым найти способы оптимизировать продажи. Директор и собственник смогут анализировать работу автосалона через мобильное приложение: загружать отчеты, координировать финансовые потоки.
* Мотивация команды автосалона – предоставляет возможность просмотра сколько автомобилей продается, и кто в команде производит эти продажи. Система генерирует отчеты по каждому сотруднику, а также команде в целом. С программой для автосалонов легко отслеживать производительность не только в рамках одной точки продаж, но и филиалов сети автосалонов.
* Управление задачами (рисунок 6) – позволяет вам управлять всеми встречами и задачами, связанными со сделкой, в одном месте. Сотрудники отдела продаж могут отслеживать статус сделок, над которыми они работают, а менеджеры по продажам могут просматривать статус каждой сделки в магазине на панели инструментов. Управление автосалоном с помощью ПО держит вашу команду в курсе каждой сделки, гарантируя, что последующие действия и информация о потенциальных клиентах никогда не пропадут.

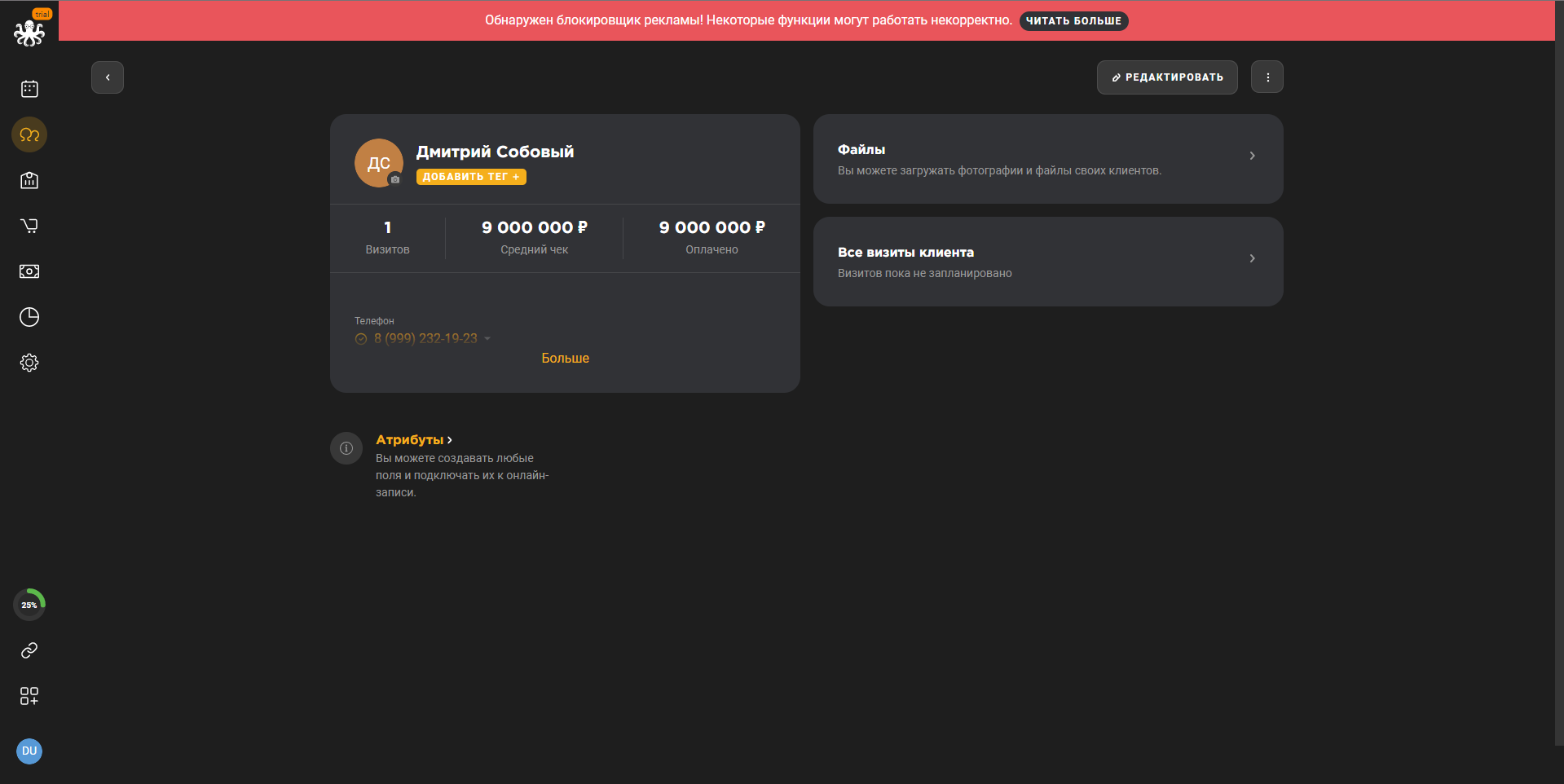


Рис.5. Учет каждого клиента

Изображение выглядит как текст, монитор, снимок экрана, экран

Автоматически созданное описание

Рис.6. Управление задачами

# **Теоретическая часть**

## **Построение диаграммы прецедентов**

Данный раздел содержит диаграмму прецедентов (рисунок 7), на которой отображается визуализация различных типов ролей в системе и то, как эти роли взаимодействуют с системой.

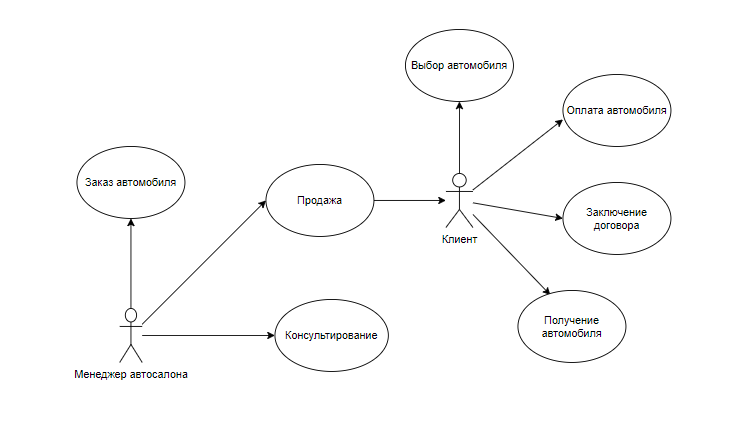


Рис. 7. Диаграмма прецедентов

## **Выбор инструментов**

### **Язык программирования**

Для написания проекта использовался Python [4] – высокоуровневый язык программирования, к преимуществам которого относят высокую производительность программных решений и структурированный, хорошо читаемый код. Синтаксис максимально облегчен, ядро имеет удобную структуру, а широкий перечень встроенных библиотек позволяет применять внушительный набор полезных функций и возможностей.

Python поддерживает несколько моделей программирования: структурное, объекта-ориентированное, функциональное. В языке присутствует динамическая типизация, автоматическое управление памятью, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных.

Институт инженеров электротехники и электроники – IEEE [8] также составил свой рейтинг наиболее популярных языков программирования на 2022 год. По их мнению, лидером также является Python, который подтверждает свой титул уже 4 год подряд. Пятерка фаворитов выглядит так же, как и в рейтинге 2021 года – Python, Java, C, C++ и JavaScript соответственно.

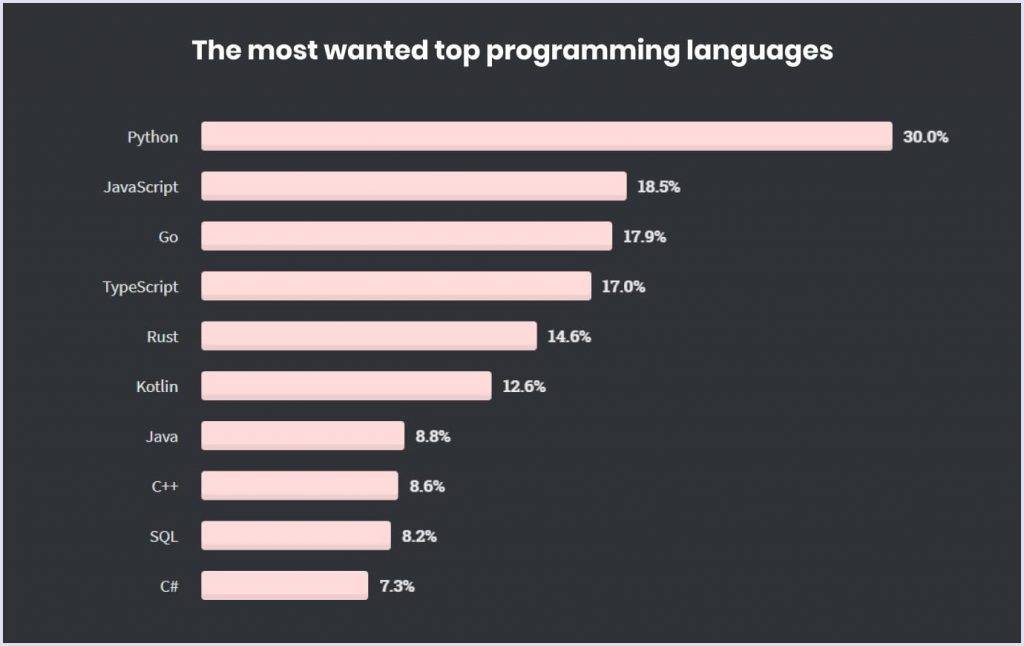


Рис.8. Таблица «Популярные языки программирования»

### **Среда разработки**

PyCharm [5] – интегрированная кроссплатформенная среда разработки для языка программирования Python предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django. Разработка IDE ведется компанией JetBrains.

PyCharm предоставляется в двух вариантах: платный (PyCharm Professional Edition) и бесплатный (PyCharm Community Edition).

### **База данных**

Уже довольно много лет большинство приложений и сайтов хранят информацию или данные. Для этого они используют базу данных – это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. Обычно они управляются системой управления базами данных (СУБД).

В качестве базы данных в проекте была выбрана SQLite [6] – это быстрая и легкая встраиваемая одно файловая СУБД, написанная на языке С, которая не имеет сервера и позволяет хранить всю базу локально на одном компьютере. Для работы SQLite не нужны сторонние библиотеки или службы.

### **Графический интерфейс**

В качестве реализации графического интерфейса в проекте использовался PyQt [7] – это библиотека Python для создания приложения с графическим интерфейсом с помощью инструментария Qt. Библиотека предоставляет из себя набор модулей, содержащих большое количество классов и функций. Разработка ведётся компанией Riverbank Computing.

## **Проектирование сценария**

Данный раздел содержит сценарий проекта (рисунок 9) – определение сочетаний условий, которые приводят к определенным результатам.

При запуске приложения открывается главное окно, в котором предоставляется возможность выбора таблицы, добавить элемент, изменить элемент, удалить элемент, вывести в Excel или JSON.

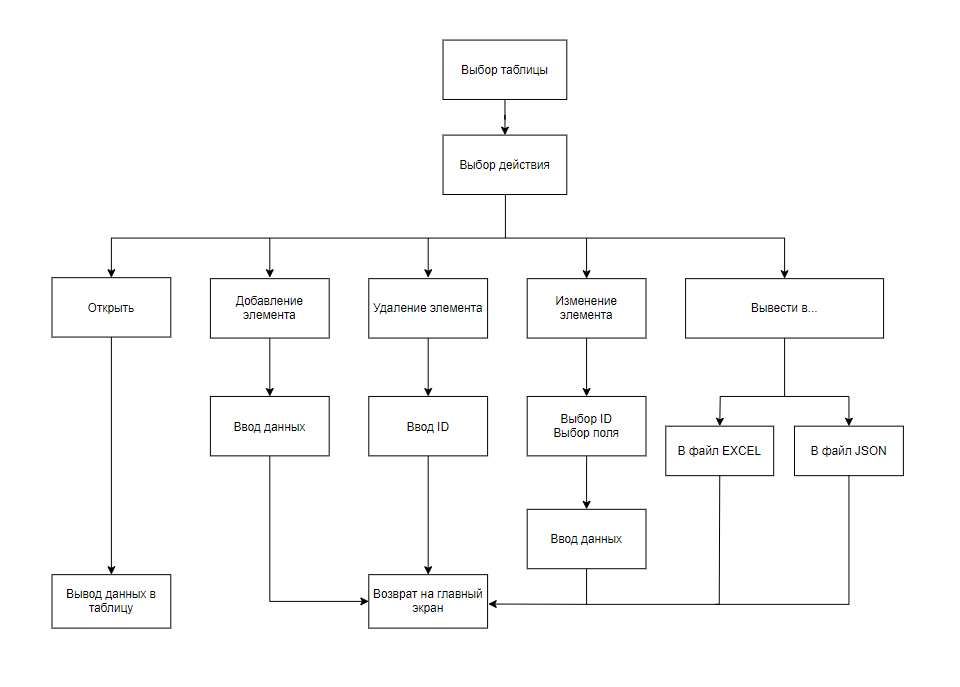


Рис.9. Сценарий проекта.

## **Построение диаграммы классов**

Данный раздел содержит диаграммы классов, их виды, а также связь между ними.

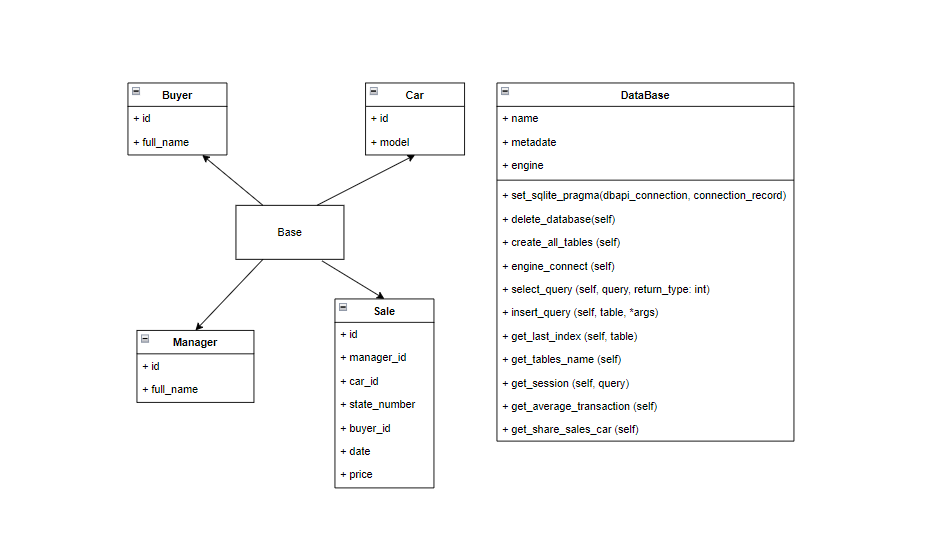


Рис.10. Диаграмма классов раздел «База данных».

* Тип «База данных» (рисунок 10) полностью отвечает за создание и получение данных из базы, который включает в себя два модуля:
* Модуль описание моделей базы «models» содержит описание всех моделей таблиц. Под описанием понимается: наименование таблиц, полей и настройки полей;
* Модуль связь с базой данных «database» содержит реализацию работы с базой данных: подключение к базе, создание таблиц, запросы.

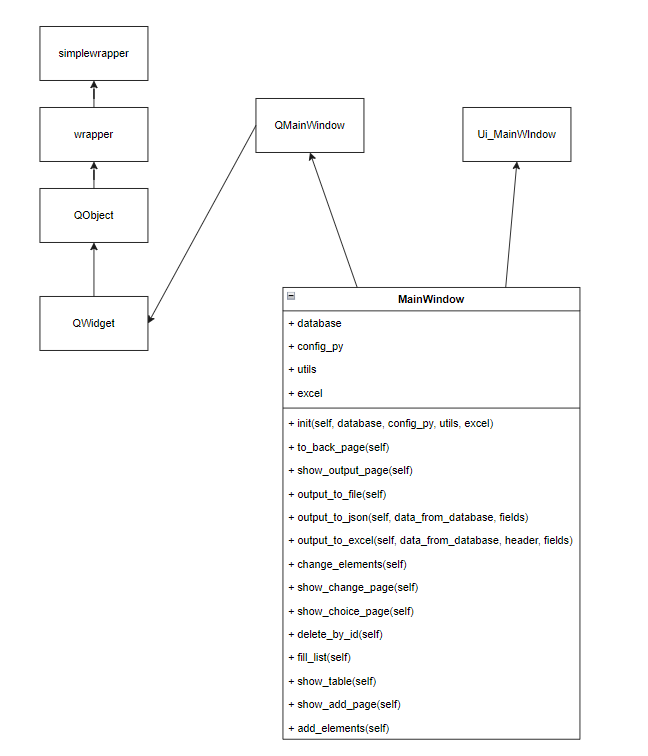


Рис.11. Диаграмма классов раздел «Графический интерфейс».

* Тип «Графический интерфейс» (рисунок 11) полностью отвечает за визуализацию приложения и включает в себя один модуль:
* Модуль работы главного окна «main\_connection». Он инициализирует окна и страницы приложения. Реализует логику взаимодействия пользователя с интерфейсом.

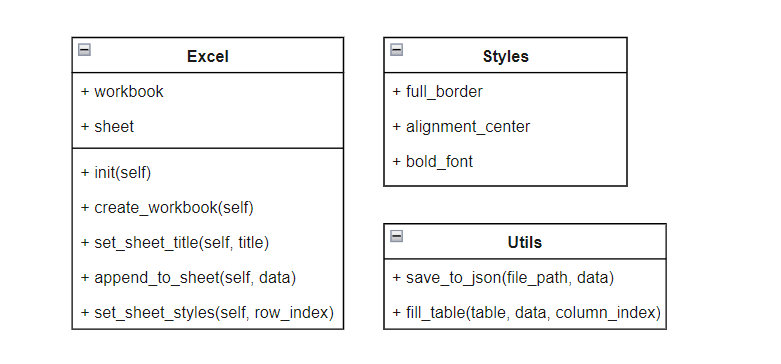


Рис.12. Диаграмма классов раздел «Утилиты».

* Тип «Вспомогательные» (рисунок 12) полностью отвечает за общие вспомогательные методы:
* Модуль «excel» отвечает за работу с одноименной программой;
* Модуль «styles\_excel» отвечает за стили для формирования текста в «MS Excel»;
* Модуль «utils» отвечает за методы, в равной степени используемые во всей программе в целом: вывод в файл c расширением json и т. д.

Изображение выглядит как текст

Автоматически созданное описание

Рис.13. Диаграмма классов раздел «Конфигурационные файлы».

* Тип «Константные файлы» (рисунок 13) включает в себя один модуль:
* Модуль «config» хранит списки и словари, необходимые в программе.

## **Описание главного модуля**

Главный модель (листинг 1) состоит из функции main. Данная функция включает в себя объекты необходимые для главного окна и запускает графический интерфейс.

Листинг 1. Главный модуль

import sys  
  
from PyQt5 import QtWidgets  
  
from config.config import Config  
from database.database import DataBase  
from gui.connection.main\_connection import MainWindow  
from utils.excel import Excel  
from utils.utils import Utils  
  
  
def main():  
 config = Config()  
 utils = Utils()  
 excel = Excel()  
  
 database = DataBase("car\_showroom")  
 database.create\_all\_tables()  
  
 app = QtWidgets.QApplication(sys.argv)  
 window = MainWindow(database, config, utils, excel)  
 window.show()  
 sys.exit(app.exec\_())  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

## **Описание спецификаций к модулям**

В проекте разработано множество модулей программы, оформленных в виде отдельных файлов.

Реализованные в проекте модули можно разделить на типы:

* База данных (рисунок 10):
* Описание моделей данных (models);
* Связь с базой данных (database);
* Графический интерфейс (рисунок 11):
* Работа главного окна (main\_connection);
* Вспомогательные (рисунок 12):
* Работа с выводом в Excel (excel);
* Работа со стилем в Excel (styles\_excel);
* Работа с сохранением в JSON (utils).
* Константные файлы (рисунок 13):
* Хранение списков и словарей (config);

Взаимодействие между модулями можно увидеть по диаграмме классов п.3.4 «Построение диаграммы классов».

## **Описание модулей**

* Тип «База данных» (рисунок 10) полностью отвечает за создание и получение данных из базы, который включает в себя два модуля:
* Модуль описание моделей базы «models» содержит описание всех моделей таблиц. Под описанием понимается: наименование таблиц, полей и настройки полей;
* Модуль связь с базой данных «database» содержит реализацию работы с базой данных: подключение к базе, создание таблиц, запросы.
* Тип «Графический интерфейс» (рисунок 11) полностью отвечает за визуализацию приложения и включает в себя один модуль:
* Модуль работы главного окна «main\_connection». Он инициализирует окна и страницы приложения. Реализует логику взаимодействия пользователя с интерфейсом.
* Тип «Вспомогательные» (рисунок 12) полностью отвечает за общие вспомогательные методы:
* Модуль «excel» отвечает за работу с одноименной программой;
* Модуль «styles\_excel» отвечает за стили для формирования текста в «MS Excel»;
* Модуль «utils» отвечает за методы, в равной степени используемые во всей программе в целом: вывод в файл c расширением json и т. д.
* Тип «Константные файлы» (рисунок 13) включает в себя один модуль:
* Модуль «config» хранит списки и словари, необходимые в программе.

## **Описание текстовых наборов модулей**

В проекте присутствует обработка ошибок, чтобы сообщить пользователю о проблеме и огородить его от сбоев программы и случайному завершению работы.

* Проверка на открытый файл (листинг 2).

В случае перезаписывания в открытый файл excel, отобразится уведомление о соответствующей ошибке в отдельном окне (приложение 11).

Листинг 2. Проверка на открытый файл.

try:  
 self.excel.workbook.save(filename=file\_path)  
 self.stackedWidget.setCurrentIndex(0)  
except PermissionError:  
 QMessageBox.warning(self, "ОШИБКА", "Закройте выбранный файл")

## **Описание применения средств откладки**

При написании проекта использовалось огромное количество раз встроенное средство откладки в IDE «PyCharm». В результате чего было исправлено большое количество ошибок различных видов.

При запуске программы без использования средства откладки в нештатных ситуациях не позволяет определить и диагностировать проблему: текст ошибки, стек вызовов с именами функций и номерами строк. Данную проблему решал запуск программы в режиме отладки, где повторное действие приводило к той же ошибке, но среда разработки при завершении программы показывала саму ошибку и ее суть.

# **Эксплуатационная часть**

## **Руководство оператора**

### **Назначение программы**

Разработка настольного приложения для сокращения времени просмотра базы данных содержащие сведение о грузах и размещении на стеллажах.

Дополнительные возможности программы:

* Добавление новых элементов;
* Изменение элементов;
* Удаление элементов;
* Вывод в Excel или JSON:
* Процент наполнения зала по сеансам;
* Самый популярный жанр.

### **Условия запуска программы**

Запуск программы с помощью файла с расширением .exe:

1. Стабильно работающее устройство на OC Windows 10.

Запуск программы с помощью исходного кода:

1. Стабильно работающее устройство на OC Windows 10;
2. Python версии 3.10 и выше;
3. Установленные библиотеки, указанные в файле «requirements.txt»

### **Выполнение программы**

При запуске программы появляется окно «Основное» (приложение 1), с помощью которого можно: просмотреть любую таблицу в базе данных (приложение 2), добавить элемент в таблицу, изменить элемент в таблице, удалить элемент в таблице и вывести в Excel или JSON файл.

Страница «Добавить элемент» (приложение 3, 4, 5, 6) зависит от открытой таблицы, позволяет добавить новый элемент в базу данных.

Страница «Изменить элемент 1» (приложение 7) зависит от открытой таблицы, позволяет выбрать ID и поле, которые мы хотим изменить.

Страница «Изменить элемент 2» (приложение 8) показывает изменяемый ID и изменяемый поле, позволяет ввести новое значение.

Страница «Удалить элемент» (приложение 9) зависит от открытой таблицы, отображает ID, которые можно удалить.

Страница «Вывести в...» (приложение 10), предоставляется выбор вывода: процент наполнения зала по сеансам, самый популярный жанр и таблицу, в файл Excel или JSON.

### **Сообщение оператору**

В случае вывода в открытый файл Excel появится соответствующее сообщение об ошибке (приложение 11).

## **To-Do лист**

Данный проект находится на стадии разработки. Планируется исправление множества ошибок и добавления нового функционала:

1. Просмотр сразу нескольких таблиц;
2. Исправление ошибок с масштабированием;
3. Исправления мелких ошибок;
4. Добавление проверок при «Добавление элемента»;
5. Пересмотр дизайна основного окна.

# **Заключение**

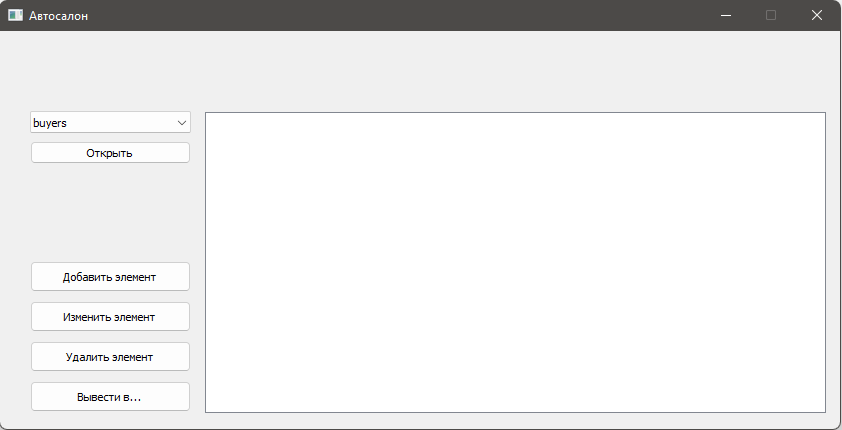
Данный проект предоставляет аналог существующих разработок управлением автоматизации автосалона. Он предоставляет минимальный, но основной функционал для работы персонала в автосалоне. После окончательной доработки приложения он может быть внедрен в работу автосалонов для улучшения удобства сотрудников при работе с данными.

# **Список использованной литературы и интернет-ресурсов**

1. Система «Altair CRM»: <https://altaircrm.ru/>
2. Система «Splus»: <https://www.splus.ru/>
3. Система «EasyWeek»: <https://easyweek.ru/biz/>
4. Язык программирования «Python»: <https://www.python.org/>
5. Среда разработки IDE «PyCharm»: <https://www.jetbrains.com/pycharm/>
6. База данных «SQLite»: <https://www.sqlite.org/index.html>
7. Графическая библиотека PyQt: <https://www.riverbankcomputing.com/software/pyqt/>
8. IEEE: <https://www.opennet.ru>
9. Github: <https://github.com/SkonSfc44/course-project>

# **Приложения**

## Приложение 1

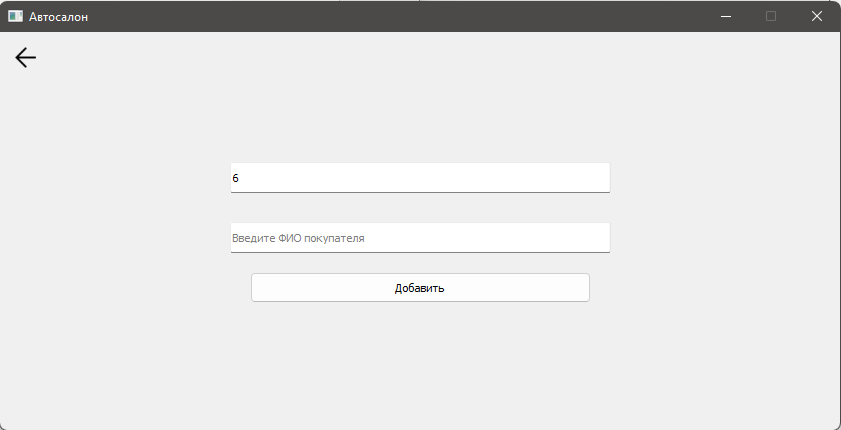


## Приложение 2

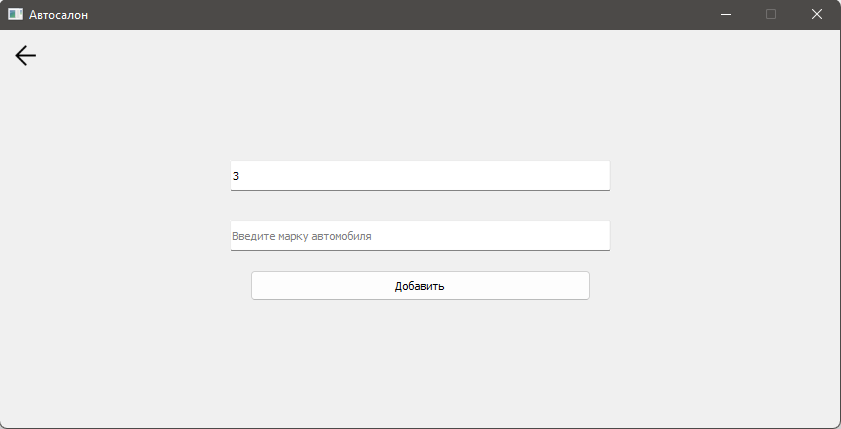
Изображение выглядит как текст

Автоматически созданное описание

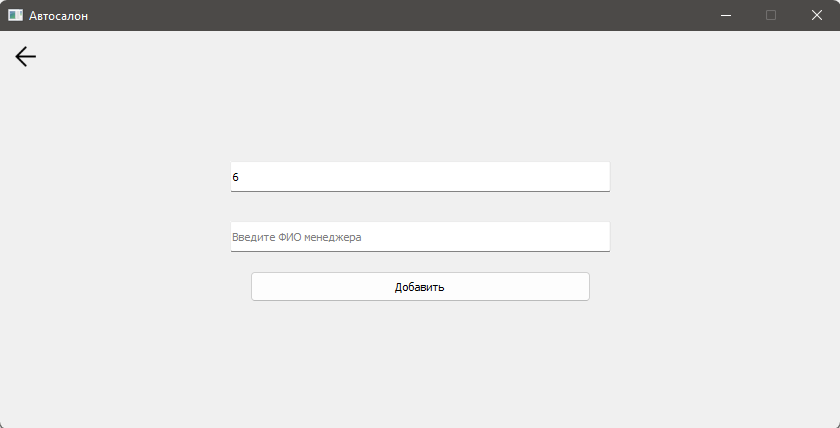
## Приложение 3



## Приложение 4



## Приложение 5

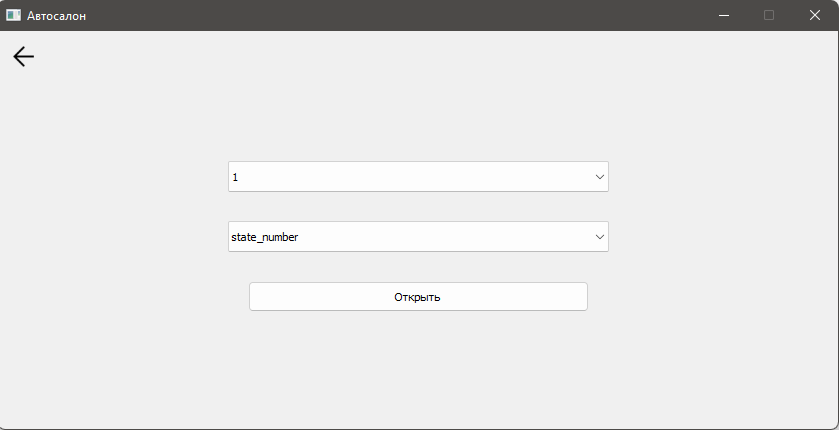


## Приложение 6

Изображение выглядит как текст

Автоматически созданное описание

## Приложение 7

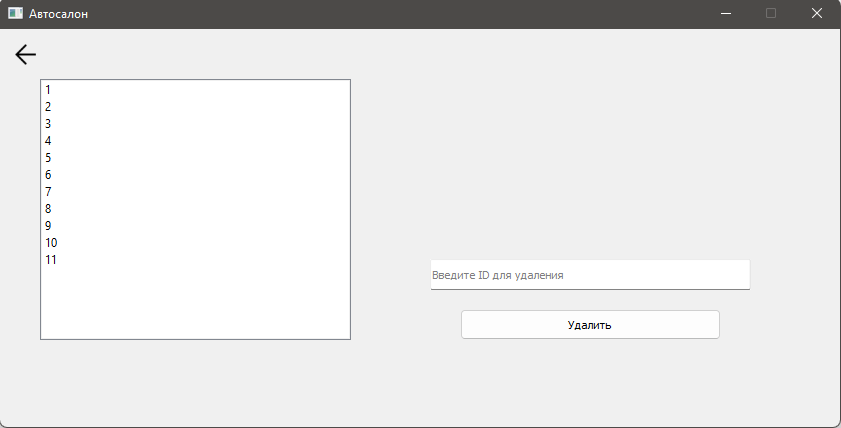


## Приложение 8

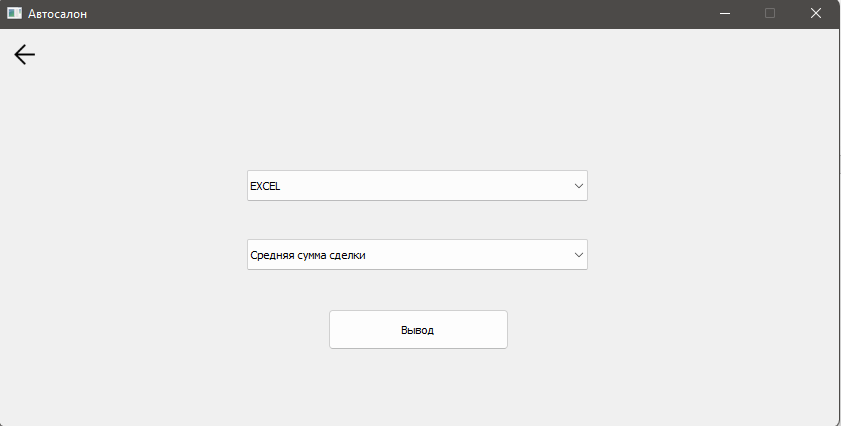
Изображение выглядит как текст

Автоматически созданное описание

## Приложение 9



## Приложение 10



## Приложение 11

