**Пояснительная записка к проекту**

**Приложение “Система контроля бизнеса”**

**Автор: Рычков Егор Андреевич**

**Краткое описание**

CRM система – часть системы контроля бизнеса.

Справка: CRM система (Customer Relationship Management) – система для взаимодействия с клиентами, хранения и обработки информации о них.

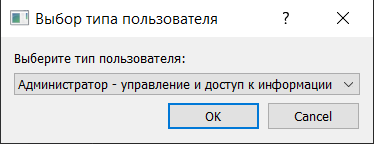
Идея: реализовать приложение с интерфейсом клиента и интерфейсом администратора. Клиент вводит информацию о себе и отправляет запросы, которые сохраняются в базу данных. Администратор может просматривать информацию, применяя разные фильтры, и отвечать клиенту. Настраивать информацию о бизнесе

**Реализация:**

Программа состоит из двух основных классов, класса второй формы и класса объекта, который нужен для отображения данных. Есть класс исключения WrongClientInf.

Два основных класса — это класс графики и класс обработки событий.

При загрузке приложения система спрашивает под каким пользователем зайти в приложение (клиент/администратор)



**Интерфейс клиента**:

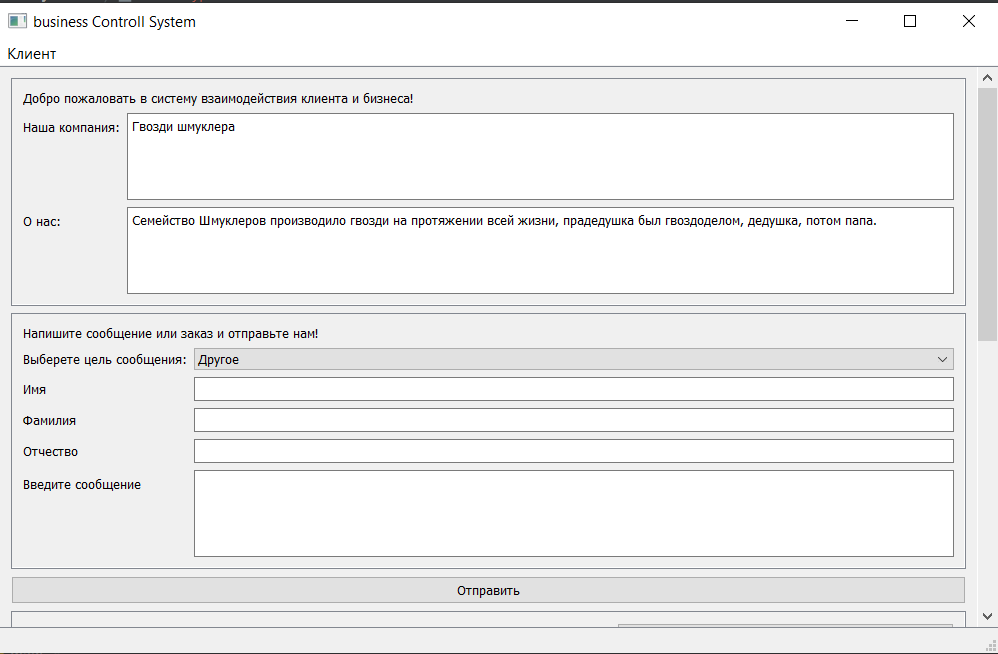
QMenuBar() клиента (меню вверху экрана):

1. Клиент – тип пользователя, который щас загружен
   1. Смена типа пользователя
   2. Выход из приложения, с сохранением изменений в config file

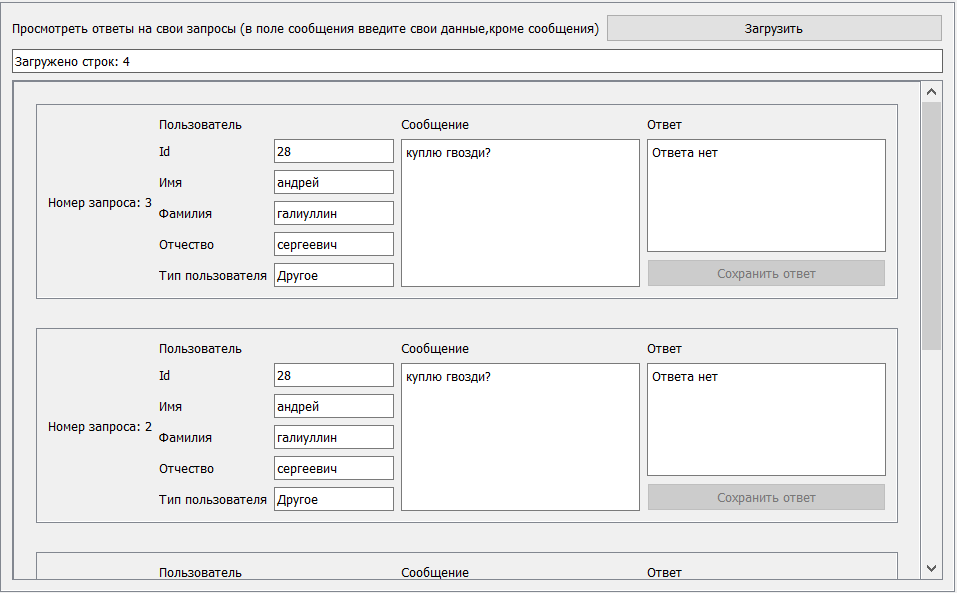
Основной интерфейс:

Интерфейс умеет проматываться, пункты 4 и 5 находятся ниже.

1. Название и описание компании
2. Сообщение клиента
   1. Цель сообщение – тип сообщения, из выпадающего списка нужно выбрать под каким типом сообщение будет отображаться в базе данных
   2. Имя клиента
   3. Фамилия клиента
   4. Отчество клиента
   5. Сообщение клиента
3. Кнопка “отправить” записывает сообщение в базу данных
4. Кнопка “Загрузить” загружает запросы пользователя по имени, фамилии и отчеству, записанным в пунктах 2.2;2.3;2.4.
5. Представление запросов клиента с ответами. О особенностях представления указанно ниже.









**Интерфейс администратора.**

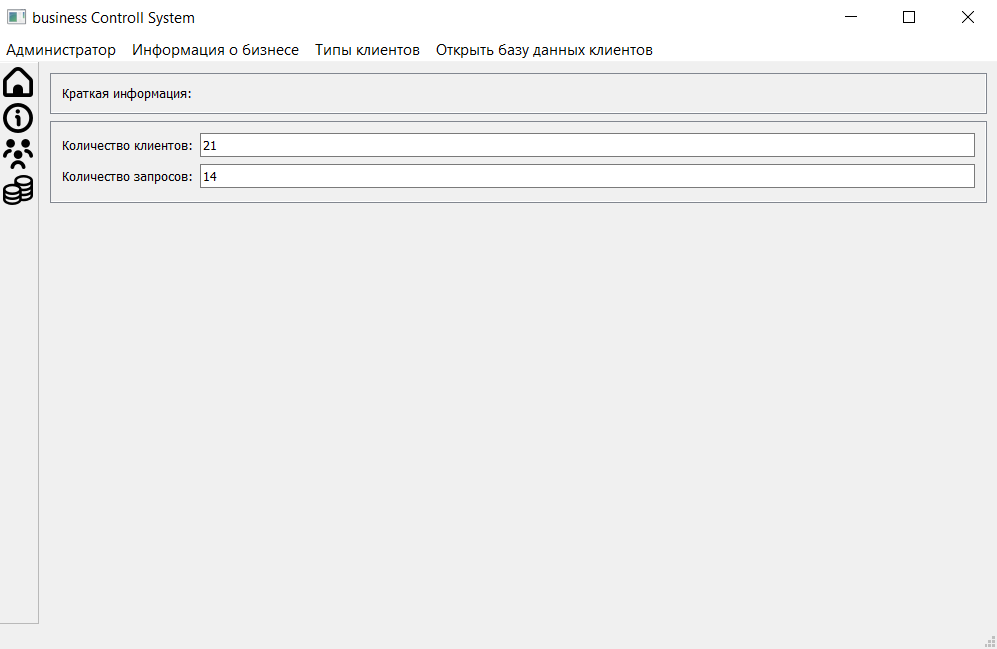
QMenuBar() администратора (меню вверху экрана):

1. Администратор – тип пользователя, который щас загружен
   1. Смена типа пользователя
   2. Выход из приложения, с сохранением изменений в config file
2. Информация о бизнесе – настройка названия и описания компании
   1. Открывает страницу информации о компании
   2. Открывает окно создания нового файла информации
3. Типы клиентов – настройка типов запросов
   1. Открывает страницу настройки типов запросов
4. Открыть базу данных – открывает базу данных о клиентах и запросах.
   1. Открывает страницу базы данных

QToolBar() администратора (находится в левой части экрана, содержит иконки страниц интерфейса, по нажатию на них открываются соответствующие страницы. Если навестись на иконку появляется подсказка с названием.)

**Первая страница** содержит краткую информацию из системы: количество клиентов и запросов в базе данных.



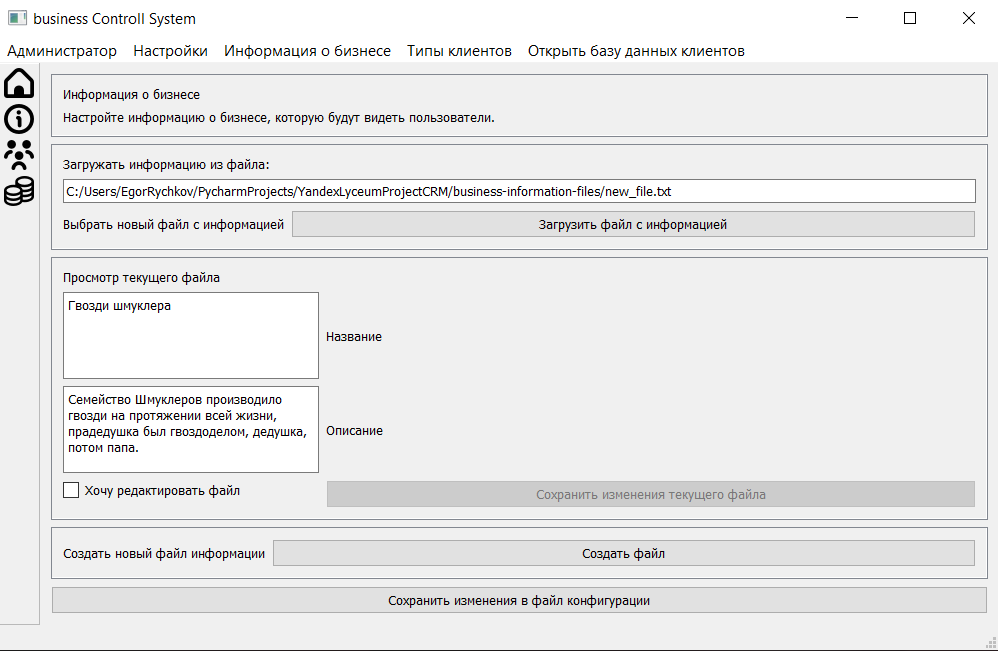




**Вторая страница – информацию о бизнесе**; настройка файла, из которого загружается информация, выбор файла, загружаемого при запуске приложения (путь до файла хранится в config.txt), создание новых файлов.

Интерфейс:

1. Настройка загружаемого файла. Отображается выбранный сейчас файл. Кнопка “Загрузить файл с информацией” открывает диалоговое окно выбора нового файла. Корректно буду загружаться только файлы созданные приложением. При открытии остальных файлов в statusBar будут выводиться ошибки.
2. Просмотр текущего файла, выводит информацию о текущем файле. При активации QCheckBox “Хочу редактировать файл” поля можно будет редактировать. Для сохранения изменений надо нажать кнопку “Сохранить изменения текущего файла”.
3. Кнопка “Создать файл” открывает новое окно, в котором можно создать новый файл информации, выбрать его название, расположение и информацию.
4. Кнопка “Сохранить изменения в файл конфигурации” сохраняет все изменения.

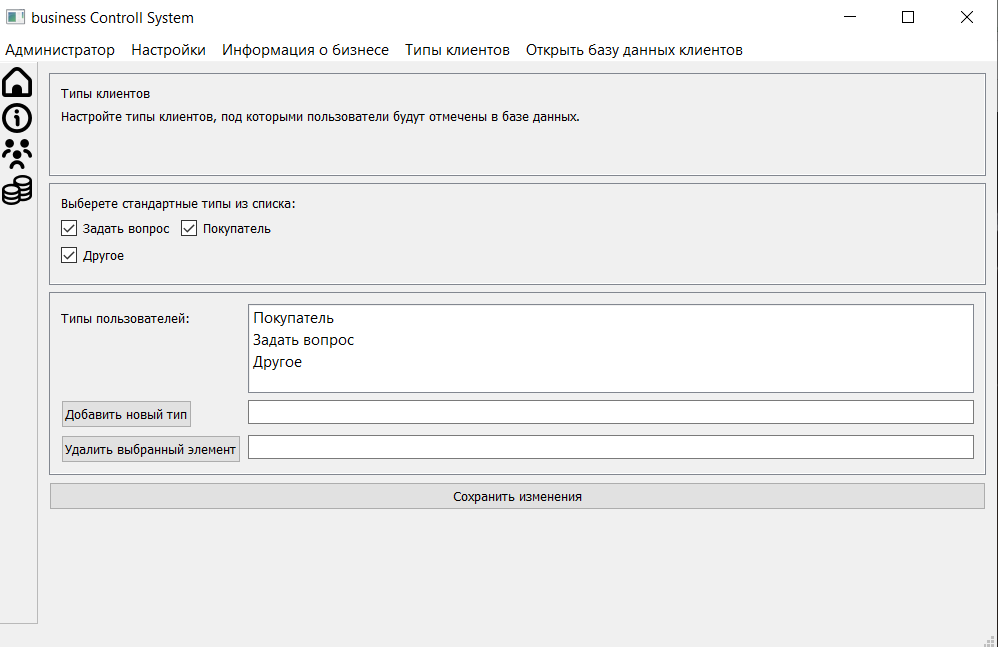




**Страница типов пользователей**. Клиент указывает основной смысл запроса. на этой странице мы можем настроить, какие типы пользователь сможет выбрать. Типы можно выбрать из стандартных или добавить свои. Типы сохранятся в файле clients-types.csv.

Интерфейс:

1. Выбор стандартных типов запросов.
2. В QListWidget() загружаются все типы из файла clients-types.csv. Кнопка “Добавить новый тип” добавляет в QListWIdget() новый тип из поля правее кнопки. При нажатии на элемент в QListWidget() он отображается в поле правее кнопки “Удалить выбранный элемент”. При нажатии на кнопку элемент будет удален из QListWidget().
3. Кнопка “Сохранить изменения” сохранит изменения в clients-types.csv.

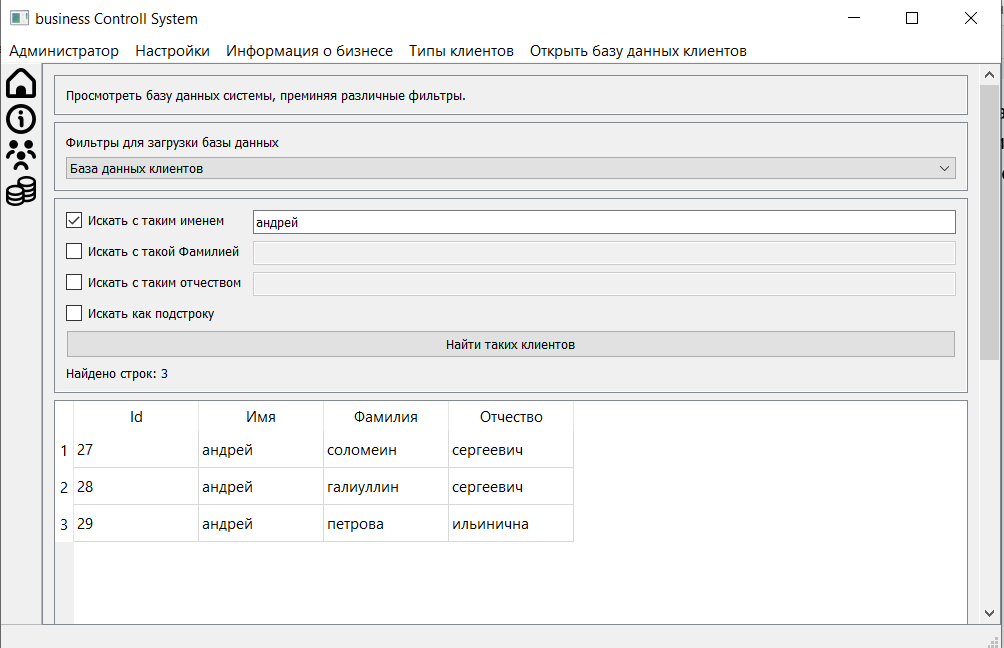
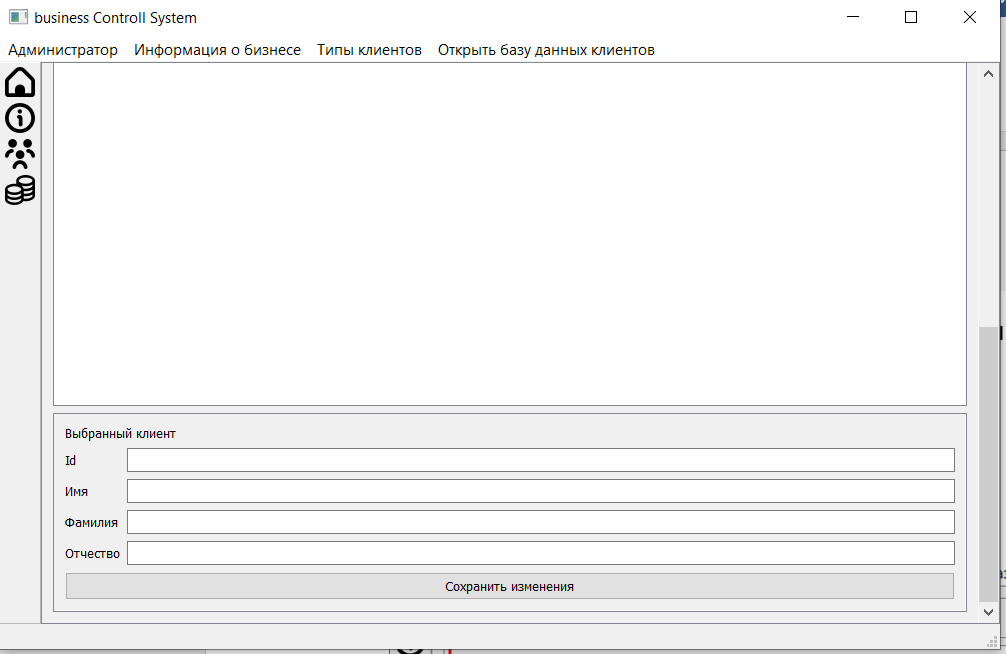




**Страница базы данных**. База данных делится на базу данных клиентов и базу данных запросов.

Интерфейс базы данных клиентов:

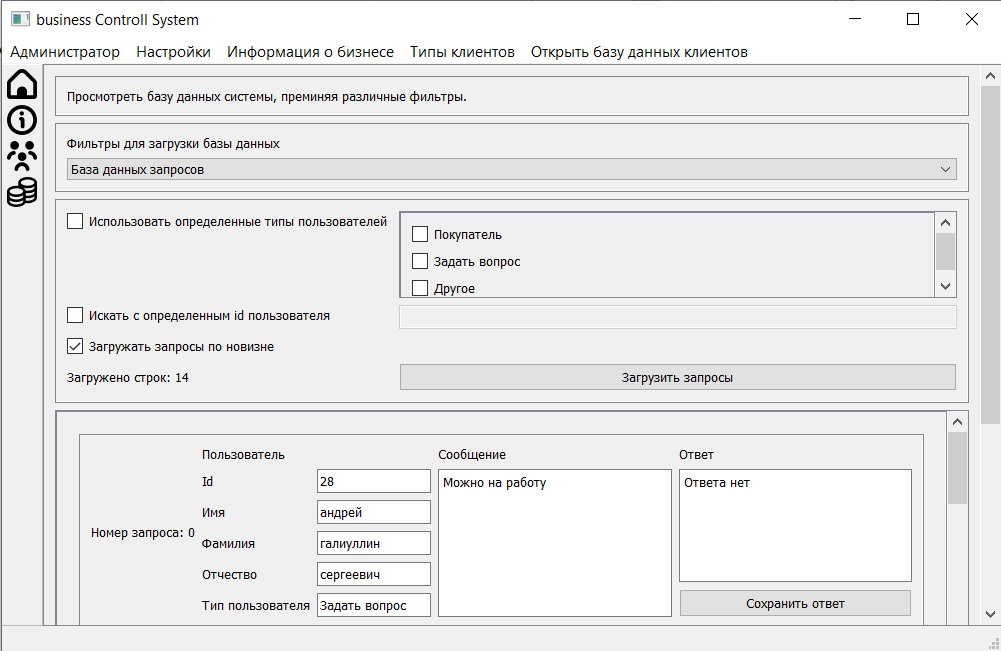
1. QComboBox() который переключает между базой данных клиентов и запросов.
2. Блок работы с базой данных. QCheckBox() активирует поле правее себя и добавляет условие к загружаемым данным.
3. Таблица с информацией из базы данных соответствующей критериям.
4. При нажатии на клиента в таблице он отображается в блоке выбранного клиента, поля ввода кроме id редактируются, при нажатии на кнопку “сохранить изменения” обновляют данный пользователя с выбранным id.



Интерфейс базы данных запросов.

1. QComboBox() который переключает между базой данных клиентов и запросов.
2. Блок с фильтрами. Checkbox() “Использовать определенные типы пользователей” добавляет условиt по типу запроса из списка с CheckBox правее.
3. Таблица запросов. Может проматываться. Поле “Ответ” редактируется. Кнопка “Сохранить ответ” сохраняет ответ к этому запросу в базе данных. По умолчанию ответ “Ответа нет”.





**Особенности приложения**

Интерфейс состоит из рамок – Qframe() — на них устанавливается layout QformLayout(), в который добавляются необходимые элементы. Все QFrame() добавляются в QVBoxLayout(); он устанавливается на central\_widget, к которому применяется SetCentralWidget(). Так интерфейс изменяется, исходя из размеров приложения. Если необходимо проматывать интерфейс, то добавляется QScrollArea(). Каждая страница – отдельный виджет QmainWidget(); виджеты добавляются в QStackedWidget() при инициализации интерфейса и переключаются с помощью функций QStackedWidget().

**Где хранятся типы запросов.**

Для сообщения пользователь указывает информацию о себе, а также тип запроса. Тип запроса – основной смысл сообщения. Он нужен для удобной сортировки дынных. Эти типы хранятся в clients-types.csv.

**Особенности представления запросов клиента в интерфейсе**.

Такое представление встречается в интерфейсе клиента пункт 5 и администратор -> база данных -> база данных запросов -> пункт 3.

Каждый запрос – запрос класса Request; он хранит информацию о клиенте, сообщение и ответ на сообщение. Класс Request наследуется от QMainWindow.

Кнопка «Сохранить ответ» и поле ввода ответа отключены, они используются в интерфейсе администратора.

По данным клиента загружаются все запросы. По информации одного запроса создается класс Request, все классы загружаются в QVBoxLayout(). QVboxLayout() связывается с виджетом QFrame(), который создает рамку для структуризации интерфейса, QFrame() связывается с виджетом QScrollArea(), что позволяет проматывать содержимое QVBoxLayout(); так создается вид таблицы, в котором хранятся запросы клиента с возможностью их проматывания.

**Как система сообщает о ошибках и подтверждениях корректности действий.**

Для этого используется QStatusBar() он отображает сообщения системы. Если красный, то это ошибка или что-то не настроено. Зеленый – действие совершено корректно. При запуске приложения с пустой базой данных, необходимо сразу сохранить типы запросов в странице администратора “Открыть настройки типов клиентов”, а также сохранить название и описания бизнеса в странице администратора “Открыть информацию о бизнесе”. Иначе при заходе о клиентах statusbar буде сообщать о ошибках загрузки информации.

**База данных(sqlite).**

База данных состоит из основной таблицы, которая хранит id пользователя, id типа запроса, запрос и ответ. Id пользователя берется из второй таблицы в которой хранятся все пользователи, id – внешний ключ. Если при добавлении запроса такой пользователь уже встречался, то его id берется из таблицы пользователей, если такой пользователь не встречался, то он добавляется в таблицу пользователей. Тип запроса – id в таблице типов запросов. В таблице запросов по id хранится текст запроса, текст запроса уникален.

**Запуск приложения**

Для запуска приложения необходимо запустить main.py; при запуске необходимо наличие папки icons, clientDB.sqlite, clients-types.csv, fileeditor.py, initclass.py, myExeptions.py, requestclass.py. Файл config.txt будет создан автоматически. Для запуска нужны библиотеки sys, PyQt5, csv, os, sqlite3.

**Что можно доработать.**

Сделать список товаров, которые можно настраивать; при запросе клиент сможет просмотреть этот список. Добавить калькулятор цены в интерфейсе клиентов, который будет считать цену заказа клиента.

У администратора сделать графики продаж и выручки.