**Мануал системного администратора**

Оглавление

[Введение 3](#_Toc130850320)

[Компоненты и зависимости 3](#_Toc130850321)

[GUI настройка (не работает!!!) 4](#_Toc130850322)

[Настройка через конфигурационный файл 5](#_Toc130850323)

[Создание приложения Yandex Disk 8](#_Toc130850324)

[Программная структура 9](#_Toc130850325)

[Админ панель 11](#_Toc130850326)

[Виды профилей 11](#_Toc130850327)

[GUI управление уровнем доступа 12](#_Toc130850328)

[Управление уровнем доступа через файловый доступ 13](#_Toc130850329)

[Описание исполняемых фалов и модулей 15](#_Toc130850330)

[Общее 15](#_Toc130850331)

[Раздел UI 15](#_Toc130850332)

[Файлы бизнес-логики 16](#_Toc130850333)

[Описание модулей 17](#_Toc130850334)

# Введение

Данный мануал предназначен для программиста/системного администратора.

Он хранит в себе информацию о базовых функциональных возможностях, возможных способов настройки или управления программным продуктом.

Данный манул не является полно описывающим все нюансы. Он даёт только примерное понимание устройства и структуры ПО.

Также не рассматривается описание используемых библиотек. Для получения подробности по «третьим» библиотекам обращаться к официальной документации и/или разработчикам интересующей библиотеки.

Также данный мануал не уделяет внимания на знания и образованность человека, подразумевается, что читающий имеет опыт программирования и/или работы.

# Компоненты и зависимости

Для работы данного ПО необходима зависимость интерпретатора ЯП Python минимальной версии 3.8.

Разрабатывалось данное ПО на версии Python 3.11.2, но также активно проверялось на версии 3.8.1.

Интерпретаторы языка можно найти в корневой паке программы в папке Data. Папка Data содержит все архивы необходимые для работы данного ПО. **НО**, при возможности, желательно устанавливать все необходимые библиотеки через пакетный менеджер PIP.

**Используемые библиотеки:**

* **PySide6 (**[PySide6 · PyPI](https://pypi.org/project/PySide6/)**)**
* **Peewee (**[peewee · PyPI](https://pypi.org/project/peewee/)**)**
* **Docxtpl (**[docxtpl · PyPI](https://pypi.org/project/docxtpl/)**)**
* **docx2pdf (**[pdf2docx · PyPI](https://pypi.org/project/pdf2docx/)**)**
* **yadisk (**[yadisk · PyPI](https://pypi.org/project/yadisk/" \l "description)**)**

# GUI настройка (не работает!!!)

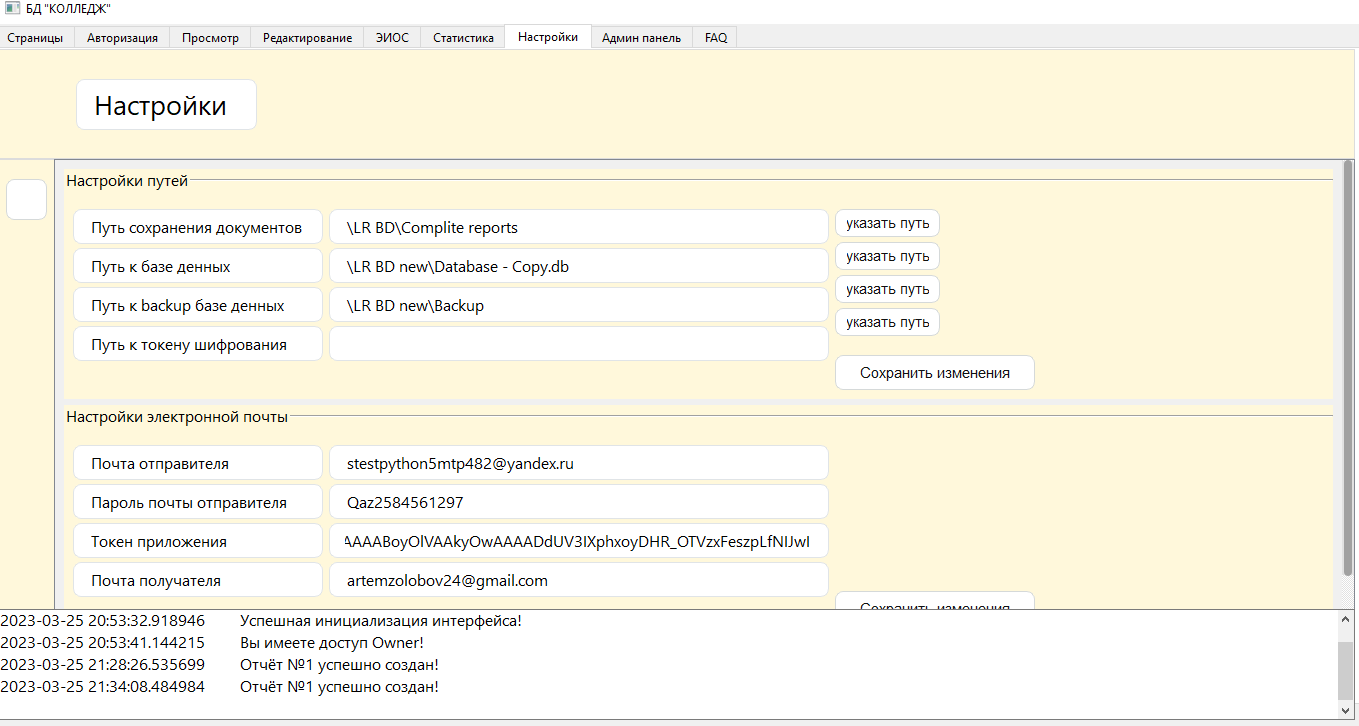
Настройку приложения можно провести с использованием GUI интерфейса.

GUI настройка позволяет провести изменение или просмотр параметров приложения, изменить их или вернуть к стандартному состоянию.

Данный раздел находится прямо в приложении в разделе «Настройки».

Все группы настроек и сами настройки, их визуальный вид и количество зависит от программиста.

Данный раздел выглядит следующим образом:



Раздел «Настройки» имеет две группы настроек: «Настройки путей» и «Настройки электронной почты».

Все параметры подразделов имеют «говорящее» название, поэтому проблем с их пониманием быть недолжно.

**\*Стоит помнить, что, применяя изменения, процесс сохранения новых настроек несёт с собой не просто переписывание изменённого поля, а всего раздела полностью.**

**\*Для работы с данным разделом рекомендуется использовать профиль со статусом «Owner»**

Описание доступных параметров настроек:

* Путь сохранения документов – параметр хранит стандартный путь сохранения документов(отчётов), создаваемых приложением;
* Путь к базе денных – параметр хранит путь к файлу базы данных;
* Путь к backup базе денных – параметр хранит путь, по которому хранится и происходит бэкап файлы базы данных (процедура бэкапа происходит каждый день);
* Путь к токену шифрования – путь к файлу, который хранит ключ шифрование/дешифрования файлов приложения;
* Почта отправителя – параметр хранит своим аргументом email отправителя; (необходимо для осуществления функции отправки отчётов на email)
* Пароль почты отправителя – параметр хранит своим аргументом пароль от почты отправителя (необходимо для осуществления функции отправки отчётов на email);
* Токен приложения – параметр хранит своим аргументом токен доступа к боту Yandex, через который происходит взаимодействие с Yandex disk (необходимо для осуществления функции отправки отчётов на email)
* Почта получателя – параметр хранит своим аргументом почту получателя (необходимо для осуществления функции отправки отчётов на email)

# Настройка через конфигурационный файл

Настройку приложения также можно провести через конфигурационные файлы. Все конфигурационные файлы имеют расширение .ini.

**\*Все параметры работы приложение берёт из конфигурационных файлов, поэтому при удалении, изменении, перемещении или других действий рекомендуется сохранять дубликаты данных файлов или не изменять их вовсе!**

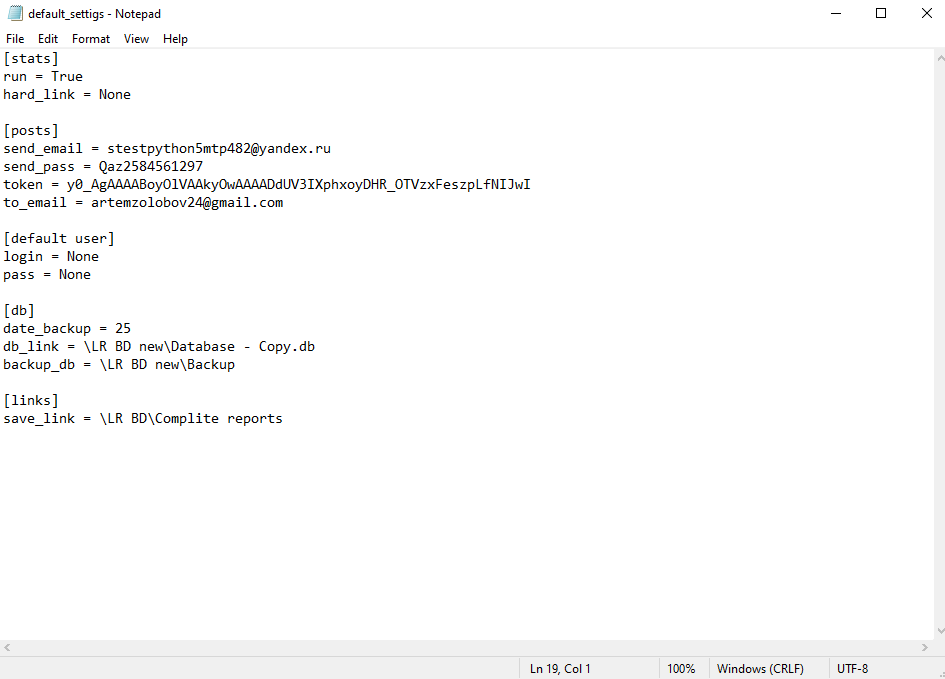
Главный конфигурационный файл расположен поп пути

D:\LR BD new\Settings\ default\_settigs.ini

Он является основным является основным настроечным файлом.

Внутренняя структура данного файла имеет следующий вид:

* stats
* posts
* default user
* db
* links

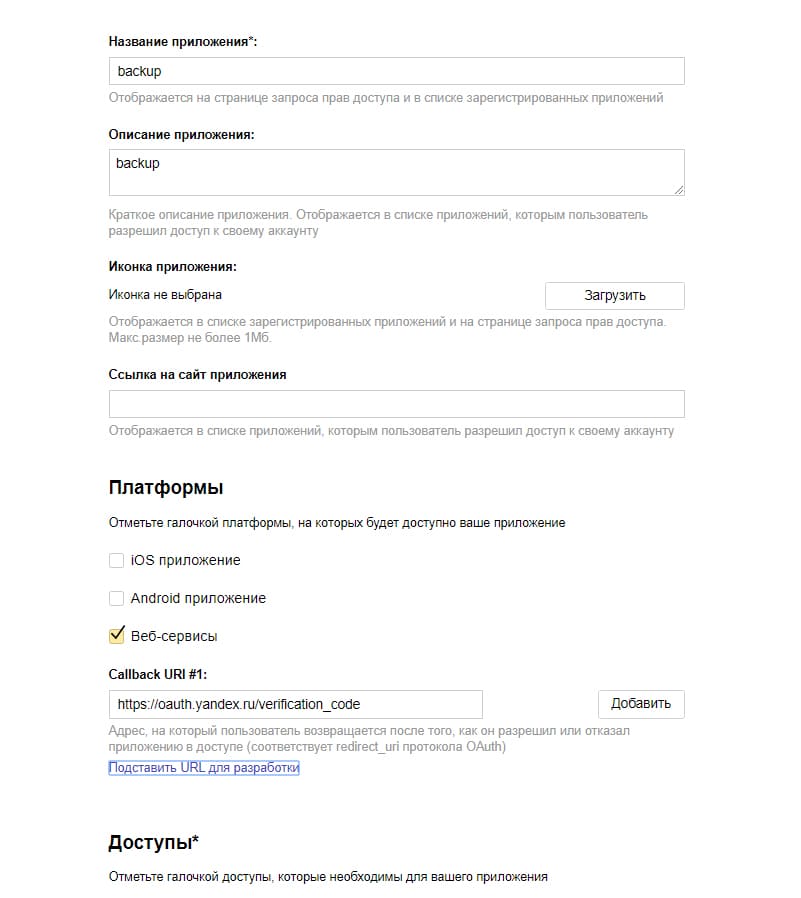


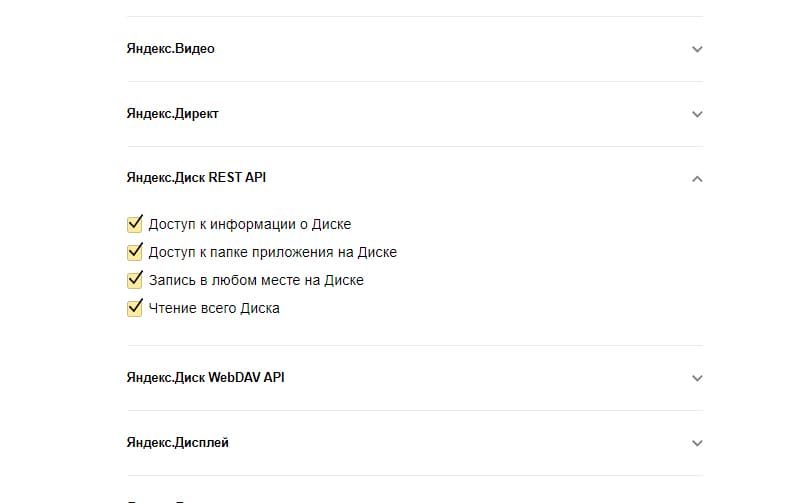
Рассмотрим кратко все функции каждого раздела:

1. stats
   1. run – принимает два значения (True, False). Данный параметр отвечает за возможность запуска и работы приложения;
   2. hard\_link – хранит в себе значение MAC-адреса диска для идентификации «неповторимого» пользователя. По умолчанию используется None.
2. [posts]
   1. send\_email – хранит в себе адрес электронной почты, с которой происходит отправка сообщений;
   2. send\_pass – хранит в себе пароль от электронной почты, с которой производится отправка сообщений;
   3. token – хранит в себе токен приложения, которое отвечает за работу с сервисом Yandex disk (подробнее в разделе; «Создание приложения Yandex Disk»)
   4. to\_email – хранит в себе адрес электронной почты, на которую происходит отправка сообщений;
3. default user
   1. login – хранит в себе логин, сохранённый при опции «Запомнить меня» (по умолчанию хранит значение None)
   2. pass – хранит в себе пароль, сохранённый при опции «Запомнить меня» (по умолчанию хранит значение None)
4. db
   1. date\_backup – хранит в себе актуальный номер дня в месяце. Используется для функции создания и поддержания бэкап файла базы данных;
   2. db\_link – хранит в себе путь, по которому расположен файл базы данных;
   3. backup\_db – хранит в себе путь, по которому расположен файл бэкапа базы данных;
5. links
   1. save\_link – хранит в себе путь сохранения создаваемых документов (отчётов).

# Создание приложения Yandex Disk

Для начала получим токен, который необходим для работы с API Диском. Для этого перейдем по [ссылке](https://passport.yandex.ru/auth?retpath=https%3A%2F%2Foauth.yandex.ru%2F&noreturn=1&origin=oauth) и зарегистрируем наше приложение, назовём его к примеру, backup, и дадим ему полное разрешение для работы с Я.Диском.



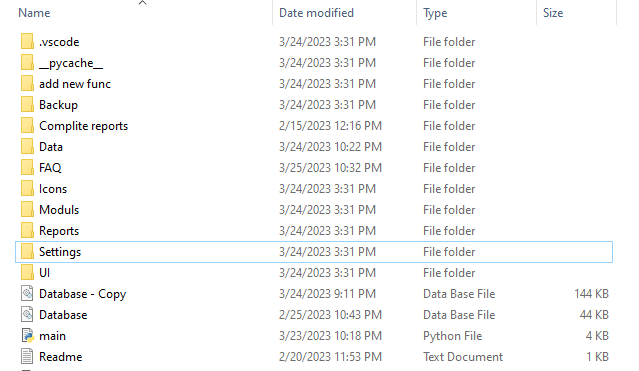


Теперь нужно просто перейти по этой ниже ссылке, заменив ID приложения на свой (ID вы получили после регистрации приложения), и токен готов.

[**https://oauth.yandex.ru/authorize?response\_type=token&client\_id=<id\_приложения**](https://oauth.yandex.ru/authorize?response_type=token&client_id=%3cid_приложения)**>**

# Программная структура

Корневая папка приложения имеет следующий вид:



Данный раздел кратко рассмотрит значение каждого элемента структуры данного приложения для более быстрого ввода в устройство программы нового пользователя.

Программная структура:

1. backup – директория, которая хранит в себе файл-бэкап базы данных, который обновляется каждые сутки;
2. complite reports – директория, которая хранит в себе (по стандарту) все создаваемые документы (отчёты) самим приложением в момент его работы;
3. data – директория, которая хранит в себе все необходимые установочные версии языка, библиотек и других дополнений, необходимых для корректной работы приложения;
4. FAQ – директория, которая хранит в себе мануалы пользователя и системного администратора, а также ресурсы для раздела FAQ в самом приложении;
5. Icons – директория, которая хранит в себе изображения иконок (кнопки, логотипы, изображения);
6. Moduls –директория, которая хранит в себе все внутренние библиотеки, которые отвечают за бизнес логику всего приложения или частных его элементов;
7. Reports – директория, которая хранит в себе шаблоны всех используемых отчётов, которые используются для создания уже заполненных форм отчётов;
8. Settings – директория, которая хранит все конфигурационные файлы, используемые самим приложением;
9. UI – директория, которая хранит все UI элементы, которые используются программой в момент своей работы;
10. Database – файл базы данных;
11. Main – исполняемый файл, через который происходит запуск самого приложения;
12. Readme – файл-инструкция по установке всего необходимого для работы приложения, а также инструкция первоначального запуска.

# Админ панель

## Виды профилей

В программе реализован принцип доступа посредством профилей. Каждый профиль имеет свой профиль доступ, в соответствии с которым и даётся определённый функционал.

Существует 3 статуса учётной записи:

* Limited
* Full
* Owner

**Limited**

Данный уровень доступа является начальным и даёт доступ только к базовым возможностям, таким как просмотр (раздел «Просмотр»), а также доступ к справке (раздел «FAQ»).

Данный профиль задумывался для создания ограниченного доступа, который подразумевает только просмотр таблиц базы данных.

**Full**

Данный уровень доступа является полным и включает в себя полный функционал по работе с приложением.

Владелец такого доступа получает следующий функционал: просмотр (раздел «Просмотр»), возможность редактирования (раздел «Редактирование»), удаление, добавление, редактирование записей БД, доступ к электронной среде (раздел «ЭИОС»), доступ к разделу статистика (раздел «Статистика»), доступ к настройкам (раздел «Настройки»).

Данный профиль подразумевался для рядового работника, которому необходим полный функционал.

**Owner**

Данный статус профиля даёт полный доступ ко всем разделам приложения.

Профилями с данным статусом рекомендуется иметь только программисту и/или системному администратору.

В дополнение к данному статусу становится доступен раздел «Админ панель».

Данный раздел позволяет осуществлять управление над всеми профилями для приложения.

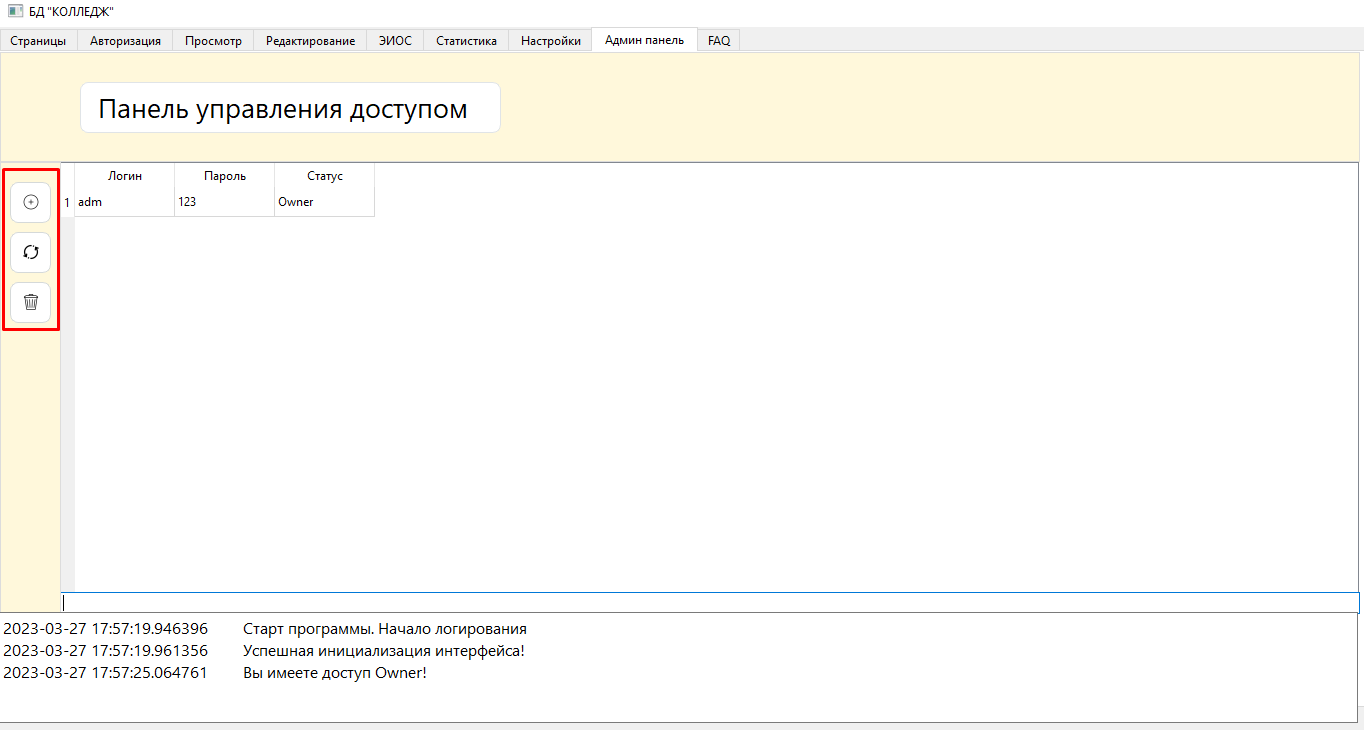
**\*Будьте осторожны в момент работы с данным разделом.**

## GUI управление уровнем доступа

В программе имеется специальный графический раздел, через который можно получить доступ ко всем учетным записям приложения.

Он позволяет выполнять добавление, удаление и обновление имеющихся/планируемых записей.

Это позволяют сделать кнопки на боковой левой панели:



**\*По умолчанию приложение не имеет профилей кроме начального со статусом «Owner» для возможной настройки и создания профиля. Его данные: логин – admin, пароль – 123.**

**\*После настройки приложения рекомендуется или удалить начальную учётную запись или произвести изменение её пароля для безопасности.**

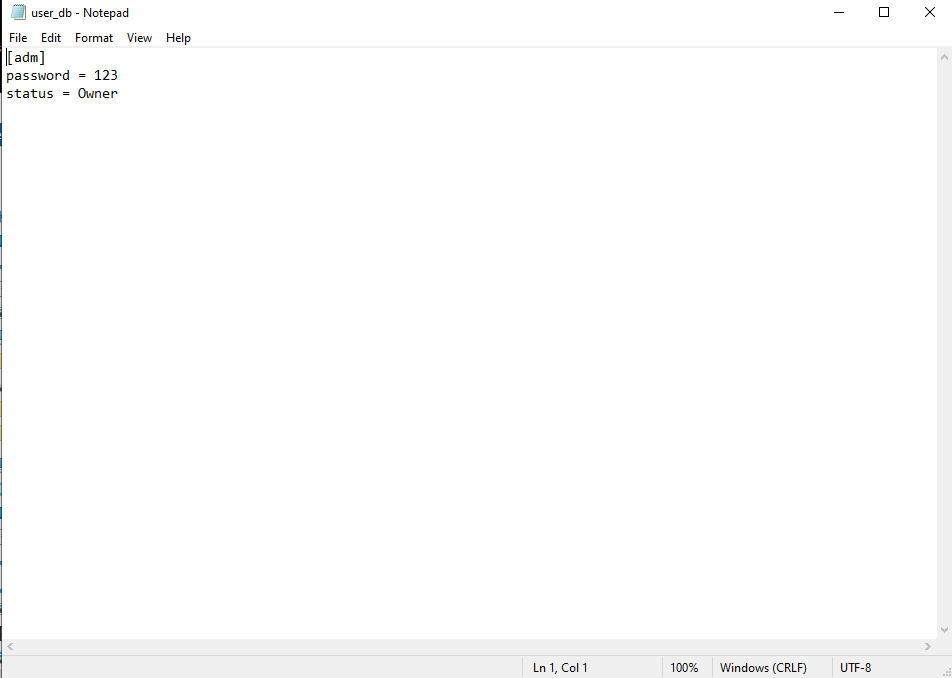
## Управление уровнем доступа через файловый доступ

Помимо GUI настройки можно создавать, менять, удалять профили доступа и через работу с файлом напрямую.

Данный файл можно найти по пути:

\LR BD new\Settings\ user\_db.ini

Данный файл хранит в себе записи всех существующий аккаунтов. Он имеет следующий вид:



Для того, чтобы обновить запись необходимо поменять какое-то значение после равно.

**\*Трогать значение перед равно категорически запрещается! Иначе работать ничего не будет!**

Схема такого файла, следующая (для добавления новых записей):

[“name”]

password = “password”

status = “status”

где “name” – имя профиля (логин), которое будет использоваться для авторизации, “password” – пароль от профиля, который будет использоваться для авторизации, “status” – статус доступа профиля, который определяет доступ в приложении (Limited, Full, Owner).

**\*Поля, заключенные в кавычки (например “name”), являются полями, в место которых должны находится данные. ДАННЫЕ ДОЛЖНЫ БЫТЬ ЗАПИСАНЫ БЕЗ КОВЫЧЕК!!!**

# Описание исполняемых фалов и модулей

## Общее

Все модули находятся по пути:

\LR BD new\Moduls

Данный репозиторий хранит в себе все файлы, которые отвечают за бизнес-логику.

Данный раздел рассмотрит краткое погружение в структуру их устройства с описанием функций и методов класса (глобальных).

## Раздел UI

Данный раздел находится по пути:

\LR BD new\UI

Он хранит в себе всю презентационную часть приложения.

Главным UI-файлом является файл ui.py. Он является основой всего приложения!!!

**\*Данный репозиторий хранит сразу два типа файлов:**

* **.ui**
* **.py**

**.ui является базовым фалом графической библиотеки Pyside (PyQt). Файлы с данным расширением имеют базовое расширение XML.**

**.py являются конвертированной версией файлов .ui, с которыми и работает сам python.**

Все UI-файлы в данном репозитории имеют в своём названии «ключи». Рассмотрим их:

* add – все файлы с данным ключом имеют отношение к UI частям, отвечающим за добавление новых записей в таблицы (например: add\_arch\_ui.py);
* change – все файлы с данным ключом имеют отношение к UI частям, отвечающим за редактирование записей (например: change\_group\_ui.py);
* find – все файлы с данным ключом имеют отношение к UI частям, отвечающим за поиск какой-то информации посредством «Поиск» (**!!!Не быстрый поиск**) (например: find\_info\_ui.py)**;**
* more\_info – все файлы с данным ключом имеют отношение к UI частям, отвечающим за получение более подробной информации конкретной записи (например: more\_info\_ui.py).

## Файлы бизнес-логики

В данном разделе будут рассмотрены все глобальные функции и методы классов всех файлов бизнес-логики.

Все файлы бизнес-логики находятся по пути:

\LR BD new\Moduls

Описание модулей приложения:

* auth.py – отвечает за логику авторизации, разблокировки доступа к соответствующим разделам приложения;
* change\_page2.py – отвечает за отрисовку раздела приложения «Редактирование». Также отвечает за логику работы других функций данного раздела (отрисовка таблиц, редактирование, удаление, добавление записей);
* create\_start\_db.py – отвечает за создание «базового каркаса», который необходим для проверки и отладки приложения (используется, если не был найден основной файл БД);
* FAQ\_page.py – отвечает за раздел приложения «FAQ». Занимается отрисовкой данных;
* firts\_start.py – отвечает за «первичную» инициализацию интерфейса, наполнение некоторых полей и так далее…;
* load\_yadisk.py – отвечает за работу с Yandex Disk (загрузка файлов на него);
* models.py – хранит в себе все модели таблиц БД. **Нужен для работы ORM Peewee!!!**;
* report.py – отвечает за создание отчётов и обработку дополнительных функций в момент создания отчётов;
* send\_email.py – отвечает за отправку документов, файлов, архивов на адрес электронной почты;
* settings.py – отвечает за работу и инициализацию раздела приложения «Настройки». Работает с конфигурационным файлами;
* show\_page1.py – отвечает за работу раздела приложения «Просмотр»;
* stats\_page.py - отвечает за работу раздела приложения «Статистика»;
* work\_ini.py – отвечает за «коннект» с конфигурационными файлами для получения нужных данных.

## Описание модулей

* **auth.py**

Данный модуль отвечает за вкладку приложения «Админ панель». Используется для настройки, создания и редактирования профилей доступа.

Состоит из следующих методов:

* + show\_table – рисует таблицу, которая отображает профили;
  + select\_page – вызывается в момент выбора раздела в приложении и вызывает метод «show\_table»;
  + cheak – отвечает за проверку статуса профиля и разблокирует функционал приложения;
  + open\_file – читает конфигурационный файл с профилями доступа и загружает их в память;
  + delete\_section – срабатывает в момент удаления профиля доступа из списка. Отвечает за перезапись файла с профилями;
  + write\_file – отвечает за дополнение данных в файл профилей;
  + change\_data – отвечает за перезапись файла с профилями;
  + append\_file – отвечает за создание нового профиля доступа для приложения;
  + change\_file – отвечает за обновление данных в таблице;
  + delete\_file – отвечает за обработку удаления записи в таблице.
* **change\_page2.py**

Отвечает за обработку бизнес-логики раздела «Редактирование». Данный раздел отвечает за создание отчётов, а также поддерживает полноценное взаимодействие с таблицами БД.

Состоит из следующих методов:

* + create\_rep – отвечает за создание отчётов;
  + delete\_data – отвечает за процесс удаления данных из таблиц;
  + msg\_error – отвечает за отрисовку оконных ошибок/предупреждений;
  + change\_data – отвечает за изменение данных в таблице БД;
  + add\_new\_data – добавление новых уникальных данных;
  + dynamic\_search – отвечает за функцию «быстрого поиска»;
  + search – отвечает за поиск по точным данным в таблице;
  + find\_info – отвечает за отрисовку окна поиска и передачи информации поиска;
  + show\_more\_info – отвечает за формирование данных для таблицы «Подробная информация»;
  + more\_info – отвечает за отрисовку окна «Подробная информация»
  + clear\_com – отвечает за очистку comboBox-ов;
  + press – отвечает за обработку изменения comboBox-ов в таблицах;
  + create\_combobox – отвечает за создание динамических comboBox-ов;
  + show\_table – отвечает за отрисовку таблиц на главном экране.
* **create\_start\_db.py**

Отвечает за создание тестовой БД для тестирования настроек и работоспособности приложения.

Имеет ряд данных, которые используются для заполнения.

Модуль также срабатывает в момент отсутствия основанного файла БД.

Включает в себя следующие функции:

* **create\_start\_db –** отвечает за инициализацию таблиц и их наполнение.
* **FAQ\_page.py**

Отвечает за работу раздела «FAQ».

Отвечает за отрисовку справочника, его наполнения, отрисовки нужных разделов.

Состоит из следующих методов;

* change\_func **–** отвечает за отрисовку определённого раздела справочника;
* initilize – отвечает за формирование раздела навигации справочника.
* **firts\_start.py**

Отвечает за первичную инициализацию интерфейса всего приложения, установку первичных данных, данных списков, картинок.

Состоит из следующих методов:

* set\_visible – отвечает за первичное скрытие/отображение дополнительных полей, comboBox-ов;
* setup\_icon – отвечает за установку иконок на кнопки активности;
* setup\_combobox – отвечает за наполнение и инициализацию comboBox-ов;
* set\_tool\_tips – отвечает за установку «подсказок» для различных полей, кнопок и прочего.
* **load\_yadisk.py**

Отвечает за загрузку документов, файлов необходимых для работы приложения на Yandex Disk. Используется в момент, когда отчётов становится много и необходимо произвести отправку архива, содержащего все отчёты.

Состоит из следующих функций:

* start\_load – отвечает за загрузку, получение ссылки на загруженный файл и возврат ссылки на файл основной программе.
* **models.py**

Хранит в себе модели таблиц БД, необходимых для работы ORM Peewee.

* **report.py**

Отвечает за формирование отчётов, обработки дополнительных функций (отправка на почту, конвертация в PDF), а также за процесс сохранения файлов по выбранным путям.

Состоит из следующих функций:

* more\_func – отвечает за обработку дополнительных функций формирования отчётов;
* create\_link – отвечает за процесс создания директорий, определения их путей, а также предоставления данных путей для дальнейшего сохранения отчётов;
* debt\_group – отвечает за формирование отчёта «За должности по группе (оплата)»;
* student\_list\_of\_group – отвечает за формирование отчёта «Наполнение группы»;
* student\_card – отвечает за формирование отчёта «Карточка студента»;
* about\_student – отвечает за формирование отчёта «Справка об обучении студента»;
* change\_group – отвечает за формирование отчёта «Перевод студента в другую группу»;
* append\_rep – отвечает за формирование отчёта «Зачисление студента»;
* delete\_rep – отвечает за формирование отчёта «Отчисление студента»;
* **send\_email.py**

Отвечает за процесс отправки документов, отчётов на электронный адрес.

Состоит из следующих функций:

* get\_file – отвечает за формирование нужного вложения для письма файла (архива);
* start – отвечает за сам процесс формирования письма, прикрепления файла и отправку письма.
* **settings.py**

Отвечает за логику раздела «Настройки» приложения. Отвечает за инициализацию интерфейса, установки значений, а также за процесс backup-а базы данных.

Состоит из следующих методов:

* initialize – отвечает за инициализацию интерфейсов раздела;
* create\_backup – отвечает за процесс создания backup версии БД.
* **show\_page1.py**

Отвечает за работу раздела «Просмотр». Отвечает за инициализацию элементов интерфейса, отрисовки таблиц, фильтров отображения информации, а также получения подробной информации и функции поиска.

Состоит из следующих методов:

* dynamic\_search – отвечает за функцию «Горячий поиск»;
* search – отвечает за выделение искомого значения в таблице приложения;
* find\_info – отвечает за отрисовку окна поиска, получения искомых данных;
* show\_more\_info – отвечает за отрисовку окна «Подробная информация»;
* more\_info – отвечает за формирование и инициализацию данных для функции «Подробная информация»;
* clear\_com – отвечает за очистку comboBox-ов;
* show\_table – отвечает за отрисовку таблиц БД.
* **stats\_page.py**

Отвечает за бизнес-логику работы раздела «Статистика».

Отвечает за создание подгрупп, наполнение статистики, получение данных, а также построение графиков и диаграмм.

Состоит из следующих методов:

* initialize – отвечает за первичную инициализацию раздела «Статистика»;
* initialize\_groups – отвечает за отрисовку и наполнение данными раздел «Статистика групп»;
* stats\_groups – отвечает за отрисовку графиков раздела «Статистика групп»;
* initialize\_specs – отвечает за отрисовку и наполнение данными раздел «Статистика специальностей»;
* stats\_specs – отвечает за отрисовку графиков раздела «Статистика специальностей»;
* initialize\_income – отвечает за отрисовку и наполнение данными раздел «Доходы»;
* income – отвечает за отрисовку графиков раздела «Доходы»;
* **work\_ini.py**

Отвечает за «чтение» данных для других частей приложения из конфигурационных файлов.