**Департамент образования и науки города Москвы**

**Государственное бюджетное общеобразовательное учреждение**

**города Москвы «Образовательный центр «Протон»**

**Помощник составления учебного расписания**

Федяев Александр Андреевич

ученик 10-Т класса

ГБОУ Образовательный центр «Протон»

Руководитель:

учитель информатики

ГБОУ Образовательный центр «Протон»

Федоров Кирилл Евгеньевич

**Москва, 2022**

**Оглавление**

[Введение 2](#_Toc95071323)

[Актуальность работы 3](#_Toc95071324)

[Обоснование выбора темы 3](#_Toc95071325)

[Цель и задачи работы 3](#_Toc95071326)

[Методика выполнения работы 3](#_Toc95071327)

[Анализ существующих решений 3](#_Toc95071328)

[Создание визуальной части проекта (frontend) 4](#_Toc95071329)

[Создание базы данных для хранения нашей информации 4](#_Toc95071330)

[Создание функционала нашей программы (backend) 4](#_Toc95071331)

[Место и сроки выполнения работы 4](#_Toc95071332)

[Результаты 4](#_Toc95071333)

[Описание завершённого продукта 4](#_Toc95071334)

[Список использованной литературы 4](#_Toc95071335)

# **Введение**

## **Актуальность работы**

На данный момент многие работники учебных заведений составляют расписание совершенно не автоматизировано, что делает их работу более долгой и сложной.

## **Обоснование выбора темы**

Мы понимаем, как сложно и долго работники школ и других образовательных учреждений составляют учебное расписание, учебные планы и выгрузки информации. Поэтому было решено сделать проект на данную тему, чтобы облегчить им работу.

# **Цель и задачи работы**

*Цель:*

Создание независимого приложения с красивым дизайном и интегрированным функционалом составления учебного расписания с последующей выгрузкой нагрузки учителей, чтобы максимально автоматизировать работу при составлении учебного расписания.

*Задачи:*

1. Проанализировать существующие решения автоматизированного составления учебного расписания.
2. Составление плана работы и установка сроков.
3. Создание визуальной части программы-приложения.
4. Создание базы данных для хранения информации.
5. Создание внутреннего функционала программы.

# **Методика выполнения работы**

## **Анализ существующих решений**

На данный момент существует совсем небольшое количество подобных программ. Но если посмотреть на отзывы данных программ, то можно понять, что практически все работают некорректно или не имеют нужное количество необходимого функционала.

Были рассмотрены такое программы и сайты как TIMETABLE, uchportal, canva и составительрасписания.рф.

## **Создание визуальной части проекта (frontend)**

Дизайн программы был создан на языке python при помощи библиотеки PyQt5, главной целью было сделать максимально удобный и интуитивно понятный интерфейс, поэтому мы создали главное меню, которое связывает все остальные окна.

# **Создание базы данных для хранения нашей информации**

База данных нашей программы-приложения была создана с помощью программы SQLiteStudio на языке программирования MySQL. База данных четко структурирована и разделена на множество отдельных таблиц для удобства.

# **Создание функционала нашей программы (backend)**

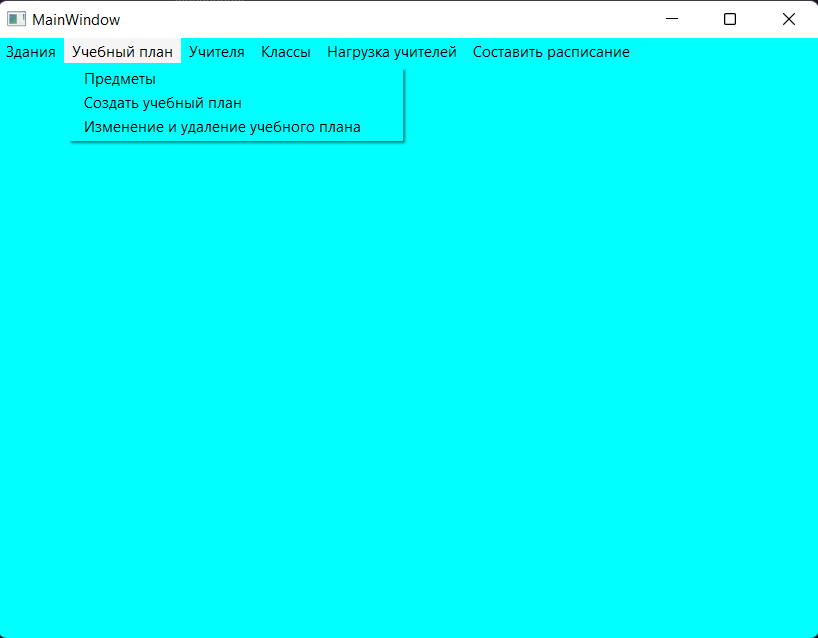
Самая важная и сложная часть проекта. Как упоминалось ранее, проект написан на языке программирования python. Для каждого окна приложение был уделен один файл, которые связываются в главном файле программы.

**Соединение визуальной части с нашим функционалом**.

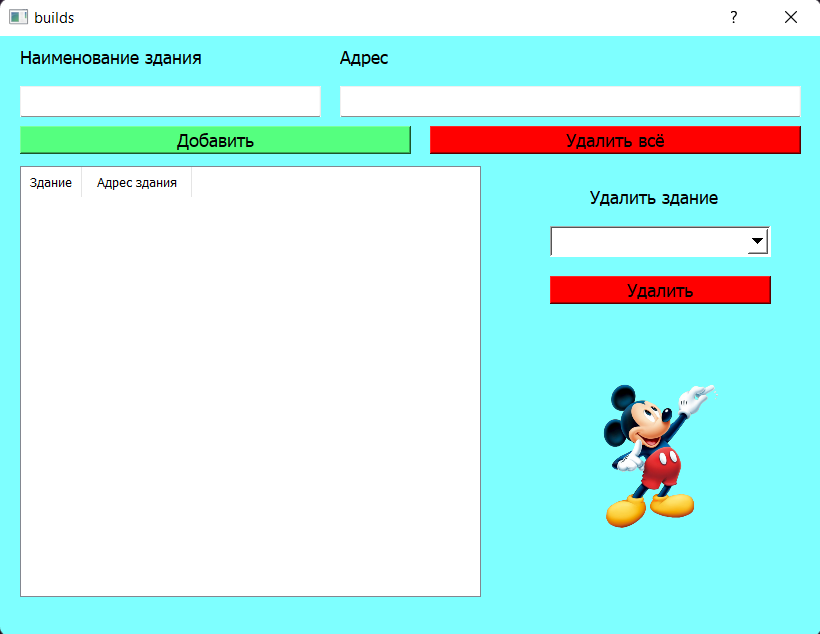
Так как соблюдена четкая структурированность программы, данный этап не вызвал тяжелых усилий, файлы с визуальной частью и базой данных портируются в нужный нам файл и инициализируются в соответствующем классе программы.

**Архитектура.**

Файл main\_window\_with\_menu представляет собой главный экран программы. Так же этот файл связывает все остальные окна программы. Данный файл занимает 294 сток кода.



Файл MyBuildings представляет собой окно, которое отвечает за добавление, редактирование и удаление зданий.



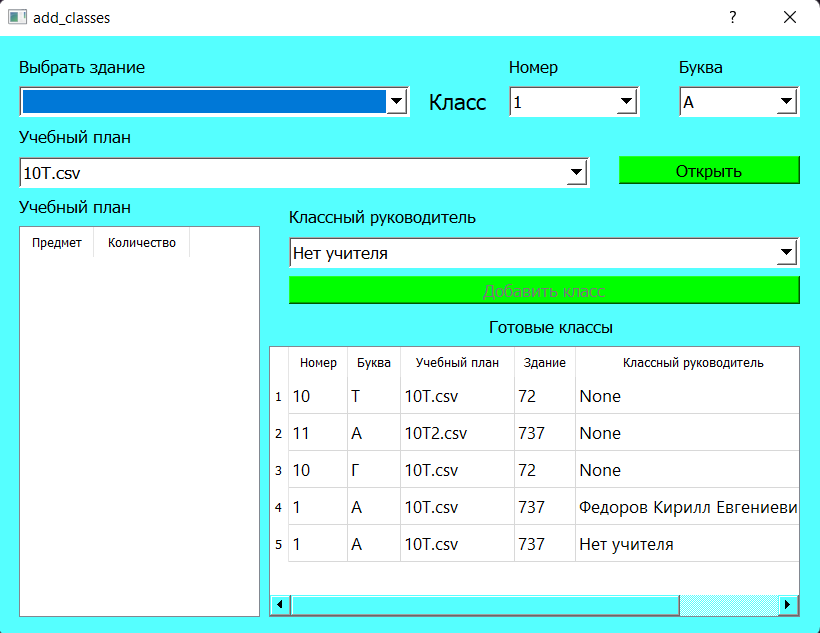
Файл MyLessons представляет собой окно, которое отвечает за добавление и удаление учебных предметов.

Файл School\_plan представляет собой окно, которое отвечает за создание учебных планов, после создания учебного файла он сохраняется в csv файл.

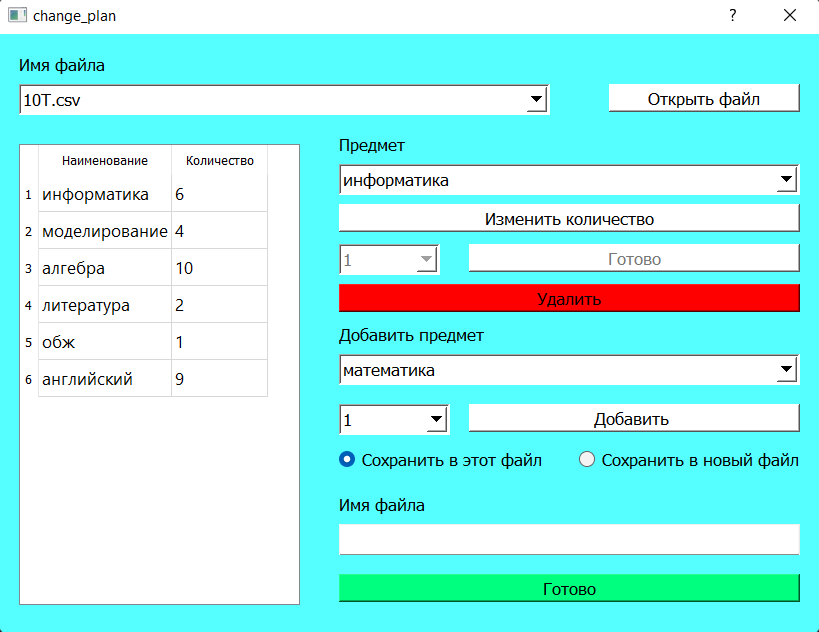
Файл change\_plans представляет собой окно, которое отвечает за редактирование учебного плана, а нем можно добавлять предметы, изменять их количество и удалять. После редактирования можно сохранить изменения либо в этот же файл, либо в новый.

Файл del\_teachers представляет собой окно, которое позволяет просматривать и удалять учителей, если была совершена ошибка.

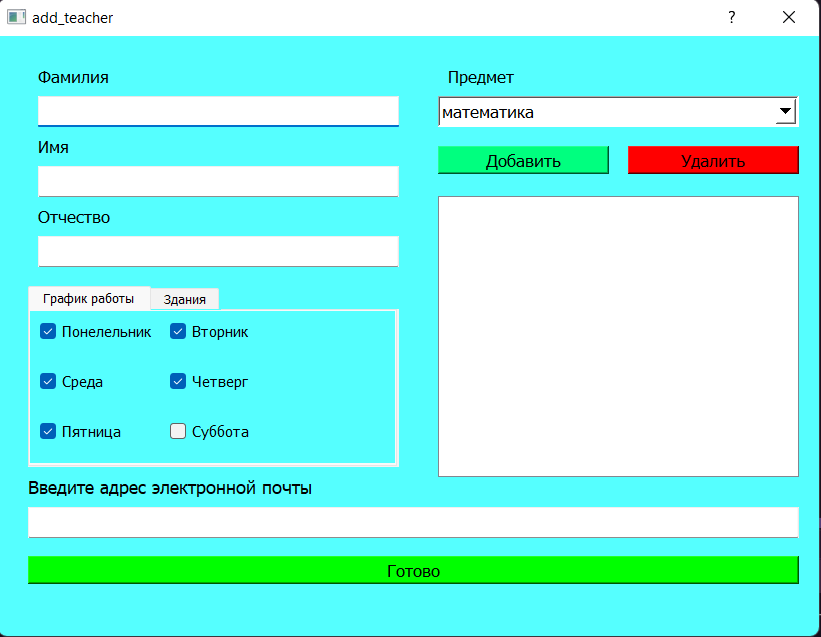
Файл Classes представляет собой окно, которое позволяет создавать классы и выбирать им учебный план.



Файл del\_classes представляет собой окно, которое позволяет удалять классы.



Файл new\_teachers представляет собой окно, которое отвечает за добавление учителей. В нем можно указать предмет, которое он ведет, здание, где он работает и его график работы

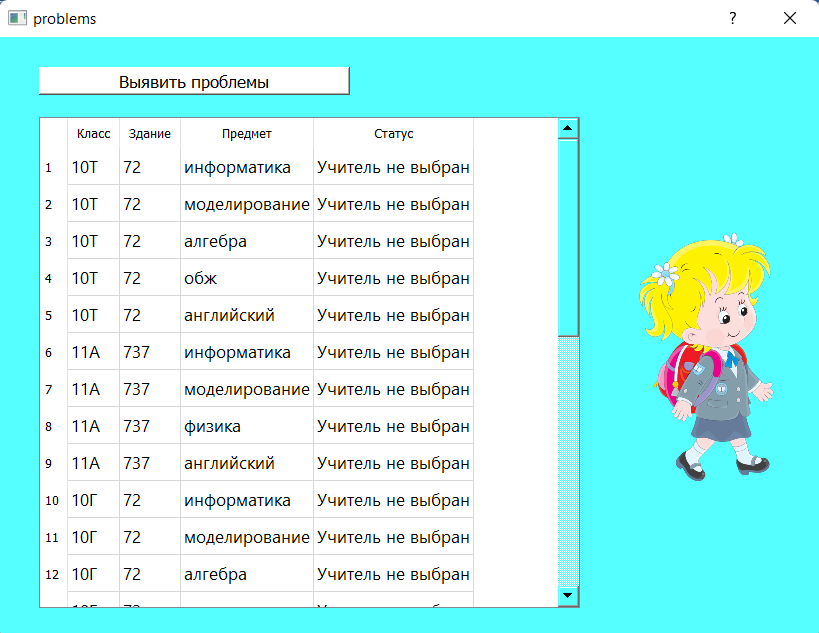


Файл create\_workload представляет собой окно, в котором вы можете создавать нагрузку учителей.

Файл definite\_teacher представляет собой окно, которое позволяет просматривать и редактировать нагрузку конкретного учителя.

Файл class\_teachers представляет собой окно, которое позволяет просматривать и редактировать нагрузку класса.

Файл workload\_problems представляет собой окно, которое показывает нам незаполненные пробелы.



Файл export\_workload представляет собой окно, которое позволяет нам экспортировать нагрузку учителей в файл word.

Общее количество кода – более 1500 строк.

## **Место и сроки выполнения работы**

Работа выполнялась с середины октября 2021 года до февраля 2022 года на базе ГБОУ Образовательный центр «Протон».

# **Результаты**

На данный приложение практически полностью готово и функционирует. Дизайн программы красивый и интуитивно понятный, соблюдена строгая структура программы, база данных корректно работает. Также создан exe файл для удобного запуска программы.

Ссылка на проект: <https://github.com/saaasssska/helper_by_school>

# **Описание завершённого продукта**

Данный проект представляет собой приложение с графическим интерфейсом. Оно состоит из главного меню и окон-виджетов. Все данные хранятся в базе данных и разделены на логические части. Были реализованы функции сбора информации о учителях, зданиях и классах, функция рассылки нагрузки учителей, и функция генерации расписания.

# **Список использованной литературы**

Python. PyQt5: [Qt for Python — Qt for Python](https://doc.qt.io/qtforpython/) (дата обращения 12 октября 2021)

Python: [Our Documentation | Python.org](https://www.python.org/doc/) (дата обращения 12 октября 2021)