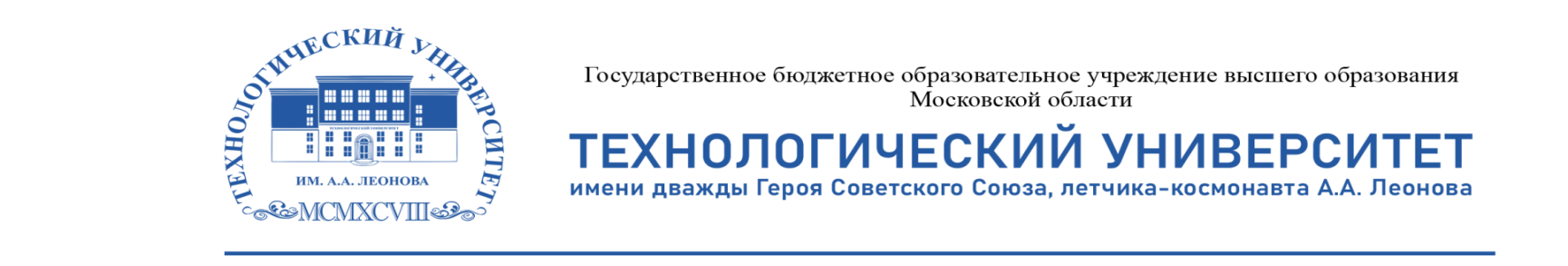
****

Колледж космического машиностроения и технологии

**Курсовой проект**

**Разработка модуля графического интерфейса для настольного приложения «Журнал успеваемости»**

Пояснительная записка

КП.09.02.03.22.20ПЗ

Обучающийся группы П2-19 Собовый Д. В.

Руководитель курсового проекта Гусятинер Л. Б.

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

Королев, 2022 г.

**Оглавление**

[**1.** **Введение** 4](#_Toc106716023)

[**2.** **Теоретическая часть** 5](#_Toc106716024)

[**2.1.** **Описание предметной области** 5](#_Toc106716025)

[**2.2.** **Описание существующих разработок** 7](#_Toc106716026)

[**3.** **Проектная часть** 12](#_Toc106716027)

[**3.1.** **Построение диаграммы прецедентов** 12](#_Toc106716028)

[**3.2.** **Выбор инструментов** 13](#_Toc106716029)

[**3.2.1.** **Язык программирования** 13](#_Toc106716030)

[**3.2.2.** **Среда разработки** 13](#_Toc106716031)

[**3.2.3.** **База данных** 14](#_Toc106716032)

[**3.2.4.** **Графический интерфейс** 14](#_Toc106716033)

[**3.3.** **Проектирование сценария** 15](#_Toc106716034)

[**3.4.** **Построение диаграммы классов** 16](#_Toc106716035)

[**3.5.** **Описание главного модуля** 19](#_Toc106716036)

[**3.6.** **Описание спецификаций к модулям** 20](#_Toc106716037)

[**3.7.** **Описание модулей** 22](#_Toc106716038)

[**3.8.** **Описание тестовых наборов модулей** 26](#_Toc106716039)

[**3.9.** **Описание применения средств отладки** 27](#_Toc106716040)

[**4.** **Эксплуатационная часть** 28](#_Toc106716041)

[**4.1.** **Руководство оператора** 28](#_Toc106716042)

[**4.1.1.** **Назначение программы** 28](#_Toc106716043)

[**4.1.2.** **Условия выполнения программы** 28](#_Toc106716044)

[**4.1.3.** **Выполнение программы** 29](#_Toc106716045)

[**4.1.4.** **Сообщение оператору** 30](#_Toc106716046)

[**4.2.** **To-Do лист** 31](#_Toc106716047)

[**5.** **Заключение** 32](#_Toc106716048)

[**6.** **Список использованной литературы и интернет-ресурсов** 33](#_Toc106716049)

[**7.** **Приложения** 34](#_Toc106716050)

[7.1. Приложение 1 34](#_Toc106716051)

[7.2. Приложение 2 35](#_Toc106716052)

[7.3. Приложение 3 35](#_Toc106716053)

[7.4. Приложение 4 36](#_Toc106716054)

[7.5. Приложение 5 36](#_Toc106716055)

[7.6. Приложение 6 36](#_Toc106716056)

[7.7. Приложение 7 37](#_Toc106716057)

[7.8. Приложение 8 37](#_Toc106716058)

[7.9. Приложение 9 37](#_Toc106716059)

[7.10. Приложение 10 38](#_Toc106716060)

[7.11. Приложение 11 38](#_Toc106716061)

[7.12. Приложение 12 38](#_Toc106716062)

[7.13. Приложение 13 39](#_Toc106716063)

[7.14. Приложение 14 39](#_Toc106716064)

[7.15. Приложение 15 39](#_Toc106716065)

[7.16. Приложение 16 39](#_Toc106716066)

[7.17. Приложение 17 40](#_Toc106716067)

# **Введение**

В последнее время большую популярность в качестве средства предоставления учебной информации приобретают портальные системы. Они позволяют сочетать информационные и образовательные функции в одно целое.

Образовательные порталы дают не только представление о деятельности и структуре учебного заведения, но и организуют пространство для учащихся, в котором содержится информация об успеваемости учеников или их групп в целом, о расписании занятий и т. п.

Образовательные порталы используются в целях обеспечения доступа к научной и образовательной информации, имеющейся в отдельных учебных заведениях.

Целью работы является разработка приложения для ОС Windows, которое будет предоставлять минимальный, но один из основных функционалов образовательного портала – отражение журналов успеваемости любой группы, состоящей в объединении учебных заведений: Технологический университет, ККМТ и ТТД.

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

Во второй части будут описаны инструменты разработки и модули программы.

В третьей части будет отражено взаимодействие пользователя с программой.

# **Теоретическая часть**

## **Описание предметной области**

Настольное приложение – это программа, для работы которой необходима ОС стационарного компьютера. Она устанавливается в систему через специальный установщик и использует для работы ресурсы компьютера.

Настольное приложение реализует следующие возможности:

* возможность работы как с подключением, так и без подключения к Интернету;
* запуск приложения происходит с помощью загрузки данных из памяти устройства, на котором установлено приложение, т. е. не требуется загрузка данных из сети;
* максимальная эффективность достигается за счет возможности работы с памятью, файловой системой, привязки локальной базы данных и т. п.

Образовательный портал – это сайт, обеспечивающий поддержку учебного процесса, информационное наполнение которого посвящено образовательной тематике конкретного учебного заведения.

Сайт разработан таким образом, чтобы информация могла быть доступна только ограниченному кругу лиц, т. е. студентам, относящимся к учебному заведению.

Для работы с порталом необходим только компьютер с доступом в Интернет.

Образовательный портал реализует следующие возможности:

* ведение электронных журналов;
* получение и сдача домашних заданий;
* возможность общения с пользователями портала;
* самостоятельное формирование личного портфолио;
* ведение учебных трансляций и т. п.

Потенциальные пользователи образовательного портала – это студенты и преподаватели.

Для студентов общеобразовательный портал выступает удобной формой для просмотра расписания занятий и выданных домашних заданий, вдобавок предоставляет возможность просматривать свою статистику и рейтинги по группе. Он призван улучшать мотивацию учащегося к повышению успеваемости, вызвать интерес к процессу обучения.

Возможности системы «Образовательный портал»:

* Расписание занятий.Учебное расписание доступно студентам и преподавателям. В нем по дням недели отражена следующая информация:
* время проведения занятия;
* ФИО преподавателя;
* название дисциплины;
* место проведения занятия.
* Журнал успеваемости. Каждому студенту доступны для просмотра все выставленные ему оценки по всем предметам.
* Дистанционное обеспечение. Образовательный портал реализует функцию домашних заданий, что позволяет преподавателям выдавать сами задания, а студентам – отправлять результаты.
* подобие социальной сети. Портал содержит в себе все основные функции социальных сетей:
* персональные страницы каждого пользователя, содержащие личные и корпоративные данные;
* возможность отправки сообщений любому пользователю и т. п.

## **Описание существующих разработок**

На сегодняшний день существует целый ряд систем для автоматизации учебного процесса, но не каждая из них соответствует требованиям того или иного пользователя. Рассмотрим наиболее известные и распространенные программные решения.

1. Система «1С:Образование» [[1](#_Список_использованной_литературы)] – комплексная система автоматизации учебного процесса, доступ к которой предоставляется локально или удаленно. Удаленный доступ позволяет не тратить время на установку самой системы, ее администрирование и эксплуатацию.

К особенностям продукта можно отнести:

* цифровая библиотека. Одна из самых востребованных форм использования программы в учебных заведениях – работа с образовательными ресурсами и учебными пособиями. Эту возможность предоставляет цифровая библиотека, ресурсами которой можно воспользоваться на любом устройстве, подключенному к Интернету (рисунок 1);
* создание учебных материалов. Программа позволяет создавать учебные материалы: иллюстрированные тексты, ресурсы на основе рисунков, электронных таблиц и презентаций;
* задания и тесты. В программе можно создавать различные по назначению задания с автоматической проверкой – обучающие, тренажерные, лабораторные и практические, контрольные, творческие. Они позволяют организовывать процесс обучения и систематически контролировать его результаты, выявлять индивидуальные сложности в обучении и помогать в их преодолении;
* контроль и анализ результатов учебной деятельности. Электронный журнал – основной инструмент для контроля и анализа результатов учебной деятельности. Каждая оценка может быть описана типами урока и учебной деятельности, за который она выставлена. Весовые коэффициенты позволяют учесть вклад текущих оценок студента в итоговую оценку за учебный период. На основе выставленных оценок формируются отчеты для преподавателя – текущая успеваемость и качество знаний, средний балл учащегося по различным видам учебной деятельности (рисунок 2).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1. Цифровая библиотека в системе «1С:Образование»

**Изображение выглядит как стол

Автоматически созданное описание**

Рисунок 2. Журнал успеваемости в системе «1С:Образование»

1. Программный комплекс «Планы СПО» [[3](#_Список_использованной_литературы)] – информационная система для комплексного автоматизированного планирования учебного процесса в учреждениях среднего профессионального образования.

Система включает в себя модули, предназначенные для разработки учебных планов СПО и их проверки на соответствие основным требованиям Государственного образовательного стандарта СПО. Пример разработанного плана изображен на рисунке 3.

К дополнительным возможностям модулей можно отнести:

* автоматический расчет сводных данных по бюджету времени;
* возможность автоматического расчета самостоятельной учебной нагрузки;
* проверки кратности часов учебным неделям, орфографии содержимого учебного плана СПО и корректности ввода информации;
* автоматический перенос информации из одного рабочего учебного плана в другой;
* печать рабочего учебного плана или отдельных его разделов.

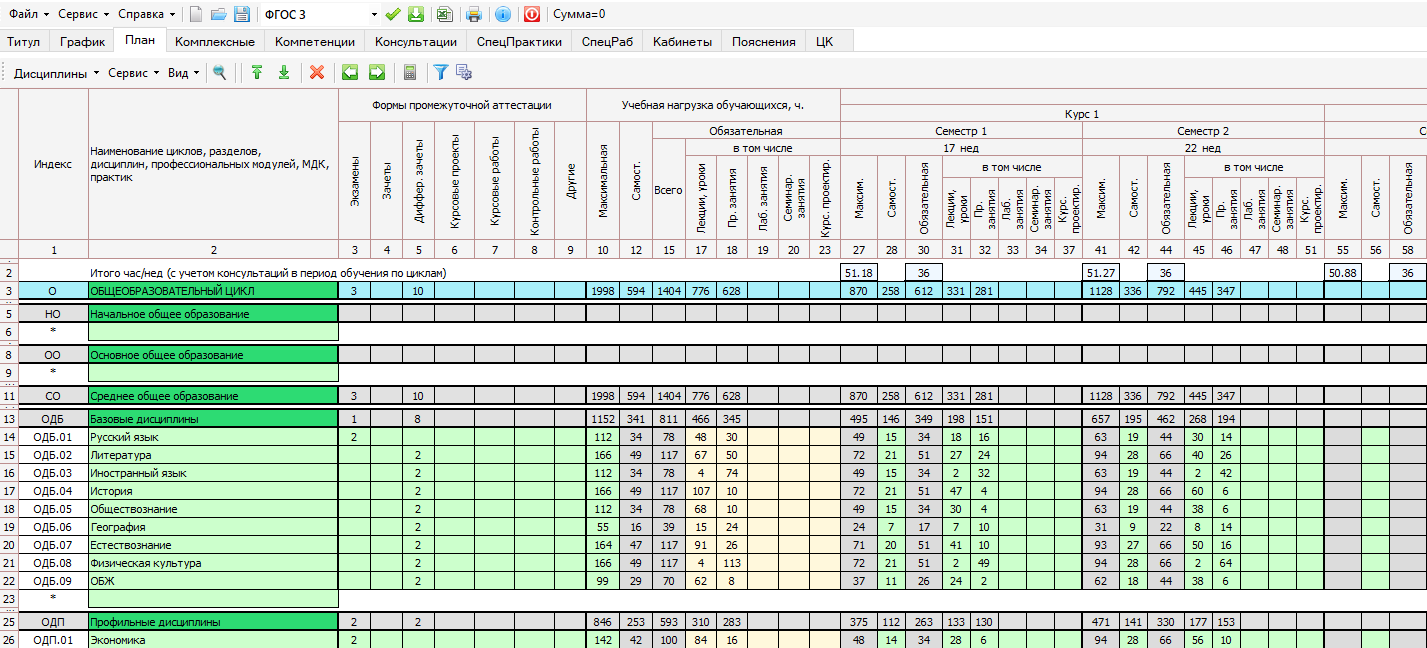


Рисунок 3. Учебный план в программе «Планы СПО»

1. Moodle [[4](#_Список_использованной_литературы)] – интегрированная система для создания персонализированной среды обучения, загрузка которой производится на собственный веб-сервер. Она является одной из самых популярных систем электронного обучения в мире.

Гибкость настройки – изменение дизайна и расширение функциональных возможностей – достигается за счет модулей, которые разрабатывают участники сообщества Moodle.

Основные возможности Moodle заключаются в:

* создание онлайн-курсов. Текстовые документы, презентации и видео можно объединить в обучающий курс, который будет доступен всем ученикам или отдельному классу (рисунок 4);
* тестирование. Доступны различные типы тестов, которые система автоматически проверяет, показывает допущенные ошибки и указывает набранный балл;
* форум и комментарии. Чтобы связаться с преподавателем, ученики могут оставлять комментарии под курсами или заводить беседы на встроенном форуме (рисунок 5);
* мобильное обучение. У сервиса есть мобильное приложение, которое позволяет проходить курсы и решать тесты с мобильного устройства;
* статистика по обучению. Система отслеживает успеваемость учеников и составляет отчёты для преподавателей.

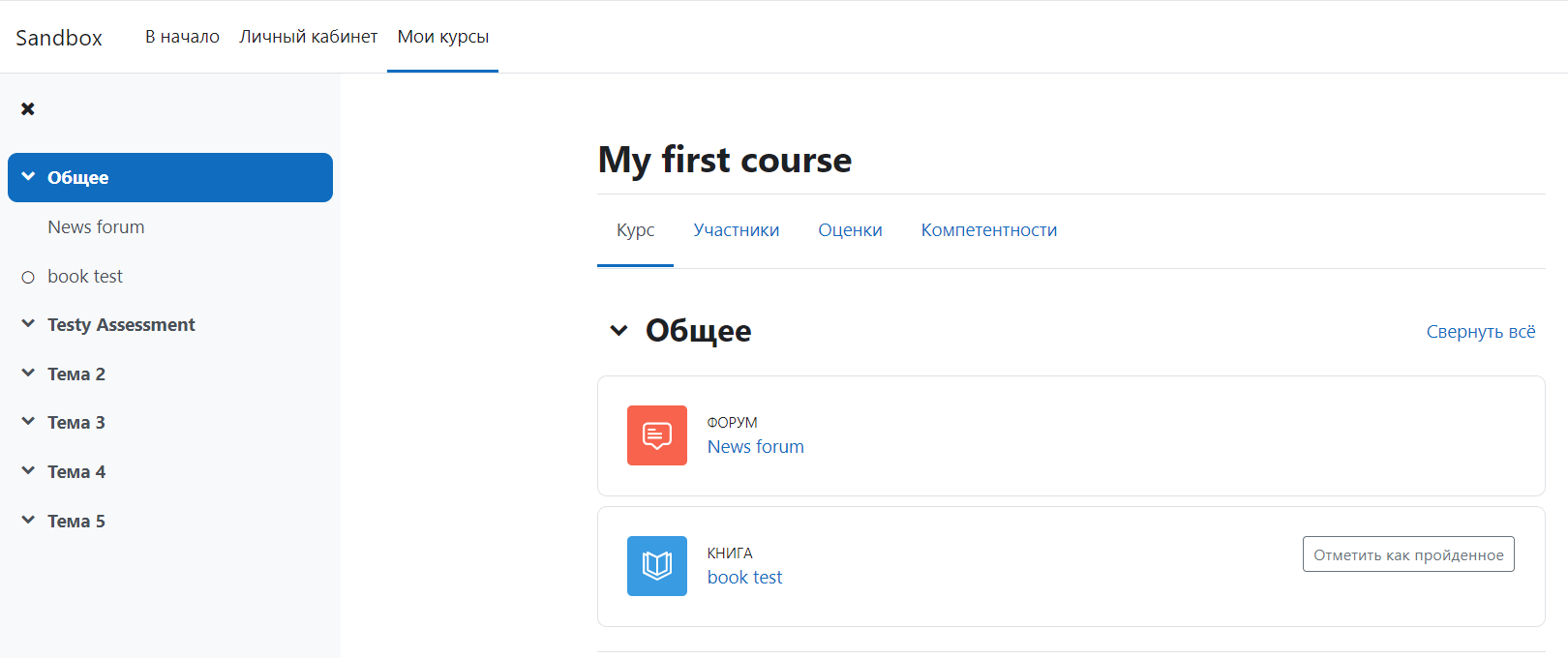


Рисунок 4. Созданный онлайн-курс в системе «Moodle»

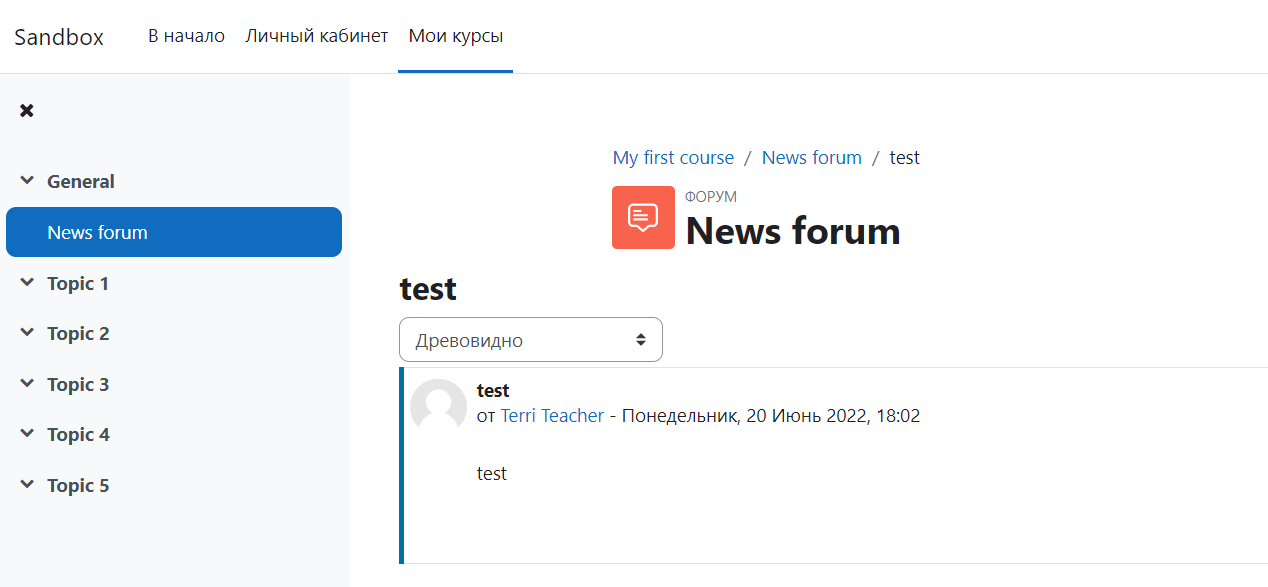


Рисунок 5. Форум онлайн-курса в системе «Moodle»

# **Проектная часть**

## **Построение диаграммы прецедентов**

Раздел содержит диаграмму прецедентов (рисунок 6), на которой изображаются отношения между актерами (действующими лицами) и вариантами использования (прецедентами).

Результатом правильно выстроенной логики должно быть разделение прав пользователей в зависимости от их роли в структуре учебного заведения. По согласованию с руководителем проекта было принято решение о реализации единого типа пользователя, который наделен всеми возможным правами.

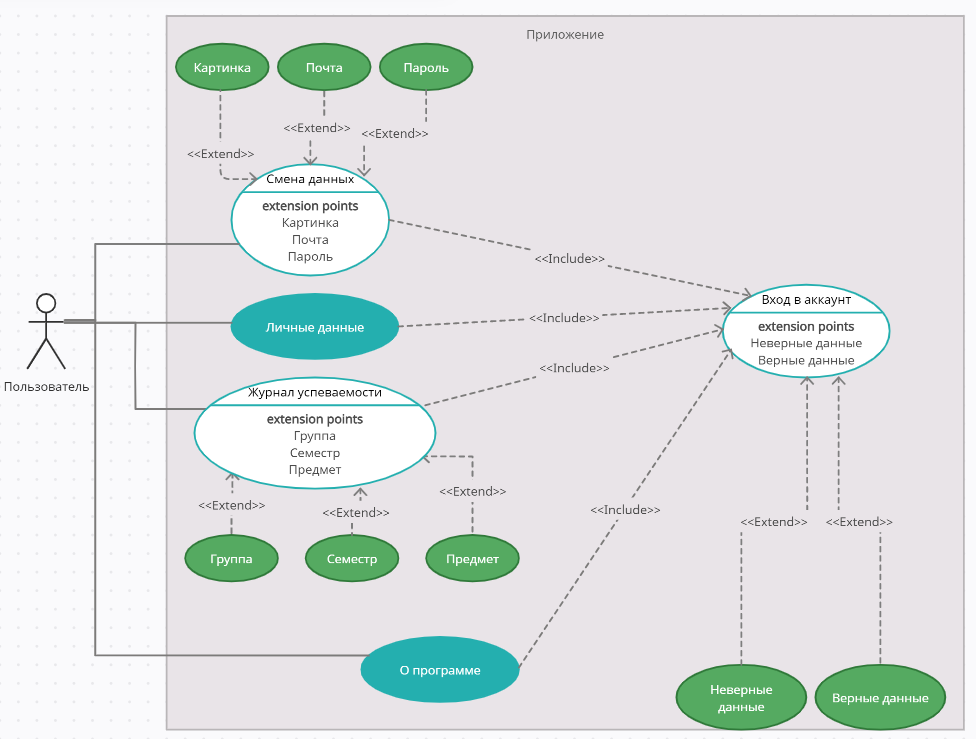
****

Рисунок 6. Диаграмма прецедентов

## **Выбор инструментов**

### **Язык программирования**

Весь функционал реализован на Python [[6](#_Список_использованной_литературы)] – высокоуровневом языке программирования общего назначения с динамической типизацией и автоматическим управлением памятью. Язык является полностью объектно-ориентированным в том плане, что всё является объектами. Он имеет простой и понятный синтаксис, что предполагает легкость изучения.

Возможность решения широкого круга задач способствовала выбору в его пользу. Python можно с легкостью расширить для взаимодействия с другими программными системами или встроить в программы в качестве компонента.

По данным индекса TIOBE [[7](#_Список_использованной_литературы)] на июнь 2022 года, Python является самым популярным языком программирования, оставив позади себя Java и семейство языков С (рисунок 7).

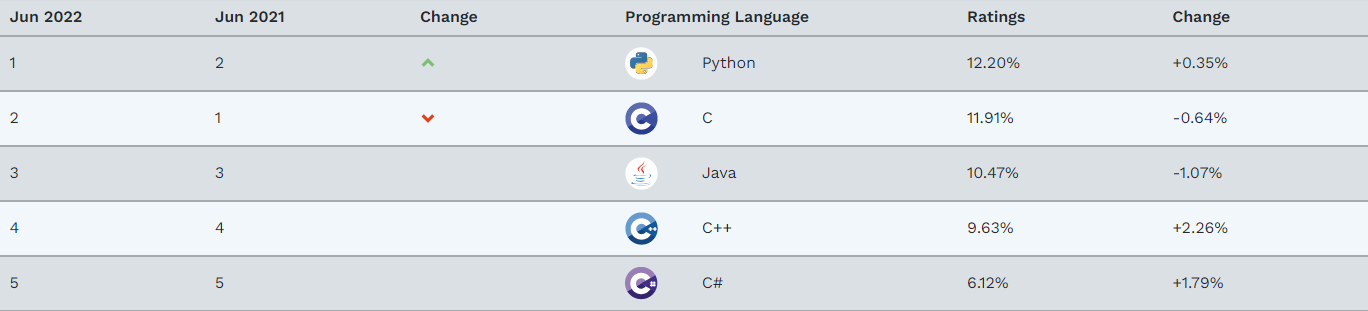


Рисунок 7. Таблица «Популярные языки программирования»

### **Среда разработки**

Правильно подобранный инструмент может значительно повысить производительность и помочь поддерживать рабочий процесс проекта.

PyCharm [[8](#_Список_использованной_литературы)] – интегрированная среда разработки для языка программирования Python. Она включает в себя средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и многое другое. Разработка IDE ведется компанией JetBrains [[13](#_Список_использованной_литературы)].

PyCharm распространяется в двух вариантах: платном (PyCharm Professional Edition) и бесплатном (PyCharm Community Edition).

### **База данных**

Почти каждое современное приложение нуждается в хранении данных. Хранение осуществляется в базах данных – отдельных независимых программах, которые не входят в состав языка программирования, позволяют хранить и обрабатывать информацию в структурированном виде.

В качестве базы данных была выбрана SQLite [[9](#_Список_использованной_литературы)] – быстрая и легко встраиваемая СУБД, написанная на языке программирования C, которая не имеет сервера и позволяет хранить всю базу локально на одном устройстве.

### **Графический интерфейс**

В проекте присутствует графический интерфейс, реализованный с помощью PyQt [[11](#_Список_использованной_литературы)] – набора расширений графического фреймворка Qt [[12](#_Список_использованной_литературы)] для языка программирования Python. Является одним из самых мощных и популярных кроссплатформенных библиотек GUI. Разработка ведется компанией RiverBank Computing Ltd [[14](#_Список_использованной_литературы)].

Библиотека представляет из себя набор модулей, содержащих большое количество классов и функций.

## **Проектирование сценария**

Раздел содержит сценарий проекта – определенное сочетание условий, которые приводят к определенным результатам (рисунок 8).

Первичный запуск приложения сопровождает появление окна авторизации, с помощью которого можно войти в свою учетную запись или восстановить пароль от нее.

После успешной авторизации пользователю открывается весь функционал, а именно:

1. Окно «Профиль» – возможность просмотра данных о себе, смены фотографии профиля, пароля и почты;
2. Окно «Журнал» – поэтапный выбор данных приводит к открытию журнала со всеми оценками студентов выбранной группы по выбранному предмету.
3. Окно «О программе» – краткая информация о программе, ее создателях и версии самой программы.

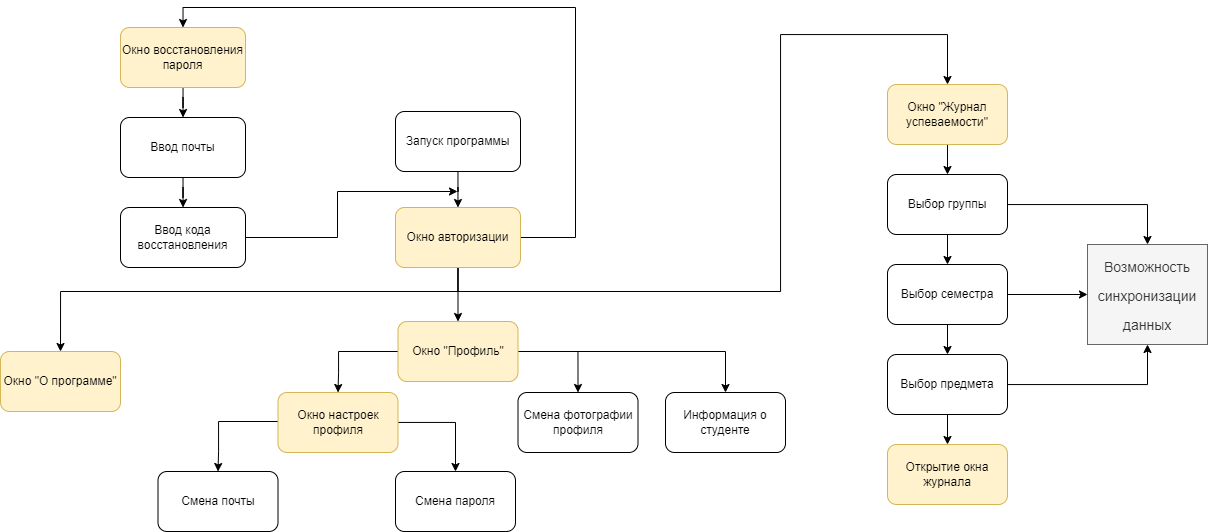


Рисунок 8. Сценарий проекта

## **Построение диаграммы классов**

Раздел содержит диаграммы классов, на которых изображается внутренняя структура программы в виде классов и связей между ними.

Подробное описание разделов можно найти в пунктах «[Описание спецификаций к модулям](#_Описание_спецификаций_к)» (3.6) и «[Описание модулей](#_Описание_модулей)» (3.7).

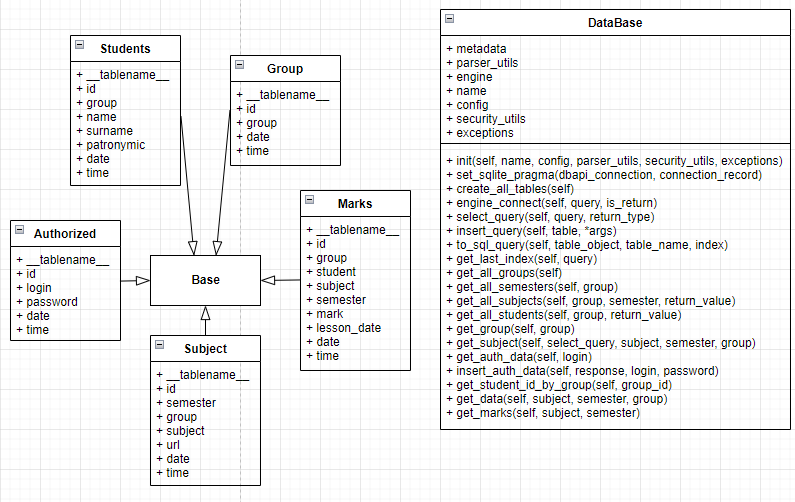


Рисунок 9. Диаграмма классов раздела «База данных»

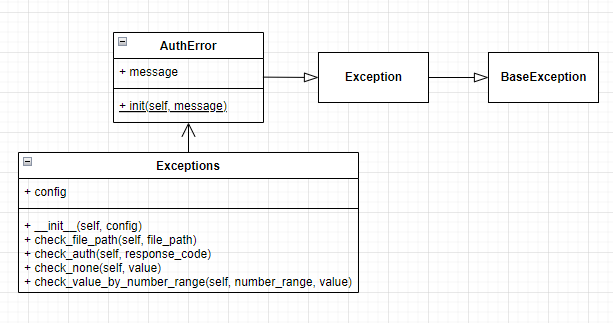


Рисунок 10. Диаграмма классов раздела «Исключения»

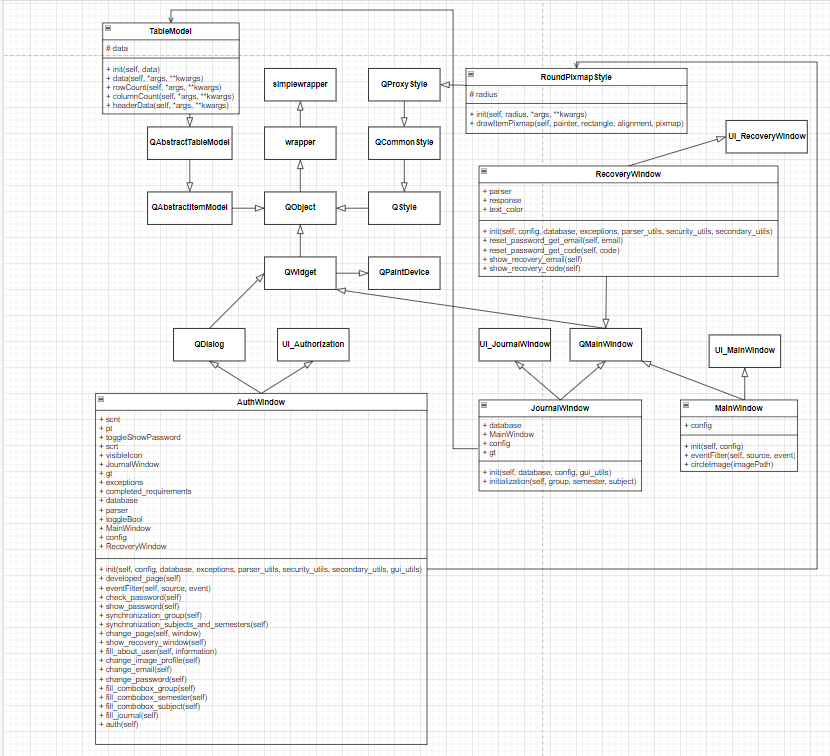


Рисунок 11. Диаграмма классов раздела «Графический интерфейс»

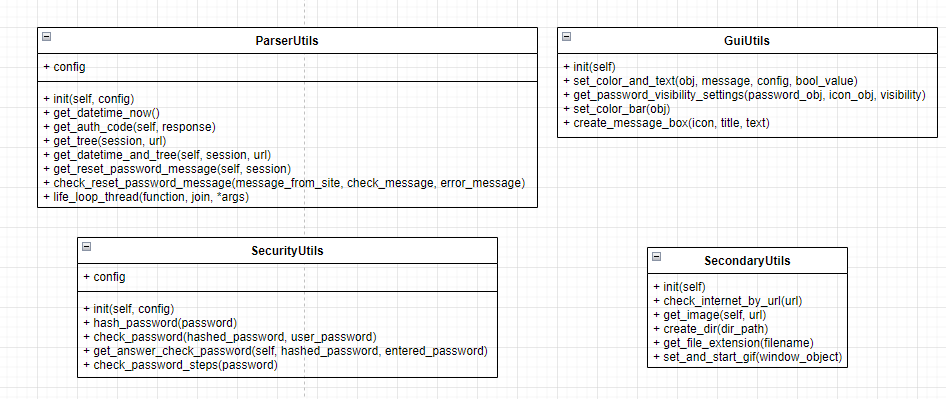


Рисунок 12. Диаграмма классов раздела «Вспомогательные модули»

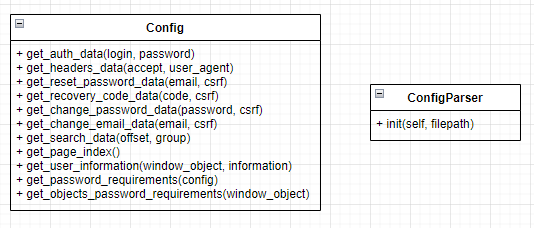
****

Рисунок 13. Диаграмма классов раздела «Константные файлы»

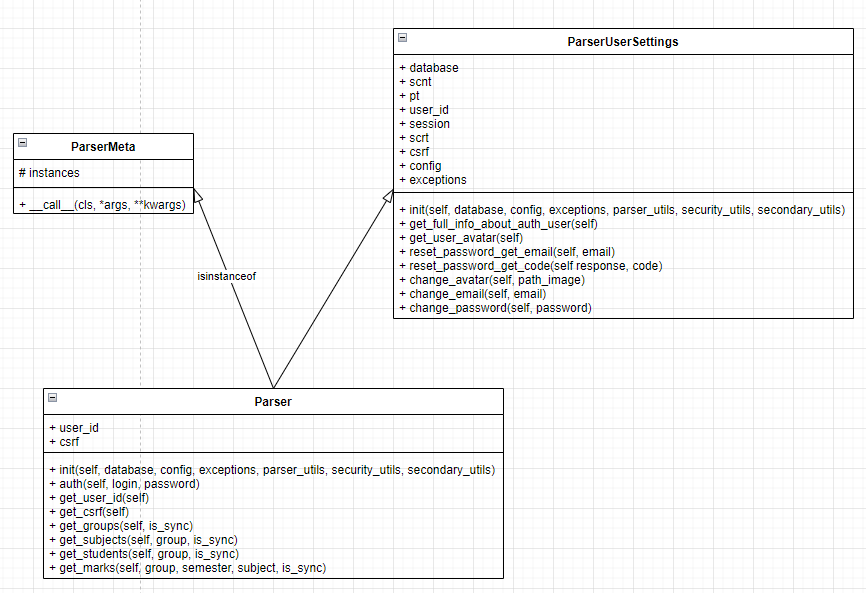
****

Рисунок 14. Диаграмма классов раздела «Сборщик данных»

## **Описание главного модуля**

Главный модуль (листинг 1) является стартовой точкой программы. В проекте он инициализирует объекты необходимых для окна авторизации классов и запускает графический интерфейс:

* ConfigParser – считывание файлов с расширением .ini;
* \*Utils – вспомогательные методы, разделенные по их назначению в различных модулях;
* Exceptions – работа с исключениями;
* DataBase – работа с базой данных;
* AuthWindow – класс, отвечающий за стартовое окно авторизации.

|  |
| --- |
| **import** **sys**  **from** **PyQt5** **import** QtWidgets  **from** **config.config\_parser** **import** ConfigParser  **from** **database.database** **import** DataBase  **from** **exceptions** **import** Exceptions  **from** **gui.connection.auth\_connection** **import** AuthWindow  **from** **utils** **import** ParserUtils, SecurityUtils, SecondaryUtils  **from** **utils.gui\_utils** **import** GuiUtils  **def** **main**():  config = ConfigParser("config/config.ini")  parser\_utils = ParserUtils(config)  security\_utils = SecurityUtils(config)  secondary\_utils = SecondaryUtils()  gui\_utils = GuiUtils()  exceptions = Exceptions(config)  database = DataBase("database.sqlite3", config, parser\_utils, security\_utils, exceptions)  database.create\_all\_tables()  app = QtWidgets.QApplication(sys.argv)  window = AuthWindow(config, database, exceptions, parser\_utils, security\_utils, secondary\_utils, gui\_utils)  window.show()  sys.exit(app.exec\_())  **if** \_\_name\_\_ == "\_\_main\_\_":  main() |

Листинг 1. Главный модуль

## **Описание спецификаций к модулям**

В программе присутствует множество модулей – фрагментов программы, оформленных в виде отдельных файлов, – что позволяет упростить ее тестирование и обнаружение ошибок.

Модули предоставляют удобную для многократного использования функциональность в виде набора функций, классов, констант.

Реализованные в программе модули можно разделить на несколько типов по зоне ответственности:

* Исключения (рисунок 10):
* реализация собственных исключений и необходимых в программе проверок (exceptions).
* Константные файлы (рисунок 13):
* хранение списков и словарей (config);
* считывание файлов с расширением .ini (config\_parser).
* База данных (рисунок 9):
* связь с базой данных (database);
* описание моделей базы данных (models).
* Графический интерфейс (рисунок 11):
* работа окна «Журнал» (journal\_connection);
* работа окна «Восстановление доступа» (recovery\_connection);
* работа окна «Авторизация», обработка событий основного окна (auth\_connection);
* установка иконок, закругление фотографий, обработка некоторых событий в основном окне (main\_connection);
* реализация круглой GIF-анимации (roundpixmapstyle);
* реализация собственной модели таблицы (tablemodel).
* Сборщик данных (рисунок 14):
* работа с данными пользователя (parser\_user\_settings);
* работа с учебными данными (parser);
* реализация паттерна «Одиночка» (parser\_meta).
* Вспомогательные модули (рисунок 12):
* работа с графическим интерфейсом (gui\_utils);
* работа со сборщиком данных (parser\_utils);
* работа во многих частях программы (secondary\_utils);
* хеширование паролей, их проверка на соответствие (security\_utils).

Подробное содержание каждого модуля можно наглядно рассмотреть в разделе «[Построение диаграммы классов](#_Построение_диаграммы_классов)» (3.4).

## **Описание модулей**

1. Модуль «Исключения» (рисунок 10) необходим для хранения собственно реализованных исключений и необходимых в программе проверок, например, существование указанного пути в файловой системе (листинг 2) или вхождение числа в переданный набор чисел (листинг 3).

**def** **check\_file\_path**(self, file\_path: str):

"""Метод проверки существования файла

Args:

file\_path: Путь до файла

Raises:

FileNotFoundError: Если файла не существует

"""

**if** **not** os.path.exists(file\_path):

**raise** FileNotFoundError(self.config.file\_not\_found\_error)

Листинг 2. Проверка существования указанного пути в файловой системе

**def** **check\_value\_by\_number\_range**(self, number\_range: tuple, value: int):

"""Метод проверки вхождения значения в интервал

Args:

number\_range: Интервал значений

value: Значение для проверки

Raises:

ValueError: Если значение value не входит в number\_range

"""

**if** value **not** **in** number\_range:

**raise** **ValueError**(self.config.value\_error.format(possible\_values=number\_range))

Листинг 3. Проверка на вхождение числа в переданный набор чисел

1. Тип «Константные файлы» (рисунок 13) состоит из двух модулей:

* модуль «config» хранит списки и словари, необходимые в программе. В большую часть словарей подставляются переданные в методы значения;
* модуль «config\_parser» считывает файлы с расширением .ini, которые хранят в себе строковые константы (листинг 4).

**from** **configparser** **import** SafeConfigParser

section\_names = 'urls', 'codes', 'messages', 'errors', 'values', 'paths', 'headers', 'stylesheets'

**class** **ConfigParser**:

**def** **\_\_init\_\_**(self, filepath):

parser = SafeConfigParser()

found = parser.read(filepath, encoding="utf-8")

**if** **not** found:

**raise** FileNotFoundError("No config file found")

**for** name **in** section\_names:

self.\_\_dict\_\_.update(parser.items(name))

Листинг 4. Модуль «config\_parser», реализующий считывание файлов с расширением .ini

1. Тип «База данных» (рисунок 9) состоит из двух модулей:

* модуль «database» содержит в себе реализацию работы с базой данных: подключение к базе данных, создание таблиц, запросы;
* модуль «models» описывает все модели таблиц, использующихся в программе. Под описыванием понимается название таблицы и ее поля со всеми настройками.

1. Графический интерфейс (рисунок 11) является одним из самых трудоемких типов, который включает в себя:

* модуль «journal\_connection» инициализирует страницу «Журнал», вносит необходимые данные в таблицу;
* модуль «recovery\_connection» – полная работа окна «Восстановление доступа»: поэтапный ввод данных для успешного восстановления доступа к аккаунту с сопровождающими проверками;
* модуль «auth\_connection» инициализирует окно «Авторизация», обрабатывает почти все события главного окна, является триггером для переключения между остальными окнами;
* модуль «main\_connection» используется для инициализации основного окна программы. Вдобавок устанавливаются необходимые иконки и реализован метод закругления фотографий;
* модуль «roundpixmapstyle» хранит в себе переопределение уже существующего метода. Новая реализация дает возможность закругления GIF-анимации ([приложение 1](#_Приложение_1));
* в модуле «tablemodel» реализована собственная модель таблицы для отображения журнала успеваемости. Это позволяет ускорить процесс сборки самой таблицы ([приложение 2](#_Приложение_2)).

1. Сборщик данных (рисунок 14) полностью взаимодействует с учебным порталом. Его архитектура разделена на три модуля:

* модуль «parser\_user\_settings» отвечает за получение или изменение данных аккаунта пользователя. Изменения пароля, электронной почты или картинки профиля происходят благодаря этому модулю;
* модуль «parser» содержит в себе сборщики учебных данных: групп, студентов, предметов и оценок;
* модуль «parser\_meta» реализует паттерн «Одиночка» – паттерн проектирования, который гарантирует, что у класса есть только один экземпляр, и предоставляет к нему глобальную точку доступа ([приложение 3](#_Приложение_3)).

1. Вспомогательные модули (рисунок 12) делятся по типу их принадлежности:

* модуль «gui\_utils» отвечает за изменение стилистики некоторых виджетов, отображение информационных окон;
* parser\_utils – методы, преимущественно используемые в сборщике данных: получение текущих даты и времени, проверка попытки аутентификации т. д;
* secondary\_utils – методы, в равной степени используемые во всей программе в целом: создание директории, получение расширения файла и т. д;
* модуль «security\_utils» призван быть «безопасным» модулем. Он хеширует пароли, проверяет их на соответствие требованиям.

## **Описание тестовых наборов модулей**

В языке программирования Python исключения являются типом данных, с помощью которых извлекается информация об ошибке в программе. Если исключение не обрабатывается, то приложение досрочно останавливает свое выполнение.

В проекте присутствует обработка ошибок, что увеличивает отказоустойчивость кода, защищая его от потенциальных сбоев, которые могут привести к преждевременному завершению работы.

Некоторые обработки ошибок, присутствующие в проекте:

1. в случае ввода и отправки ошибочных данных во время авторизации, пользователь получит уведомление о соответствующей ошибке в отдельном окне ([приложение 4](#_Приложение_4));
2. для восстановления доступа к учетной записи учебного портала необходимо ввести электронную почту, указанную в настройках аккаунта. Если почта будет указана неверно, то пользователь получит уведомление об ошибке ([приложение 5](#_Приложение_5)). Подобный итог будет и на стадии ввода кода восстановления;
3. в модуле, отвечающем за работу с базой данных, присутствует обобщенный метод для получения данных по переданному запросу. Данные могут быть возвращены как в единичном экземпляре[[1]](#footnote-1), так и последовательностью значений[[2]](#footnote-2). Возвращаемый формат регулируется специальным значением-аргументом, которое должно принимать ограниченный массив значений. Это значение ограничивается благодаря обработке исключения ([приложение 6](#_Приложение_6)).

## **Описание применения средств отладки**

В течение всего периода написания проекта встроенное средство отладки в IDE «PyCharm» использовалось огромное количество раз. Благодаря ему были исправлены ошибки различных уровней опасности, которые могли бы привести к фатальным последствиям: неверное отражение данных, алгоритмические ошибки и т. п.

В основном, средство отладки позволяло определить источник проблемы, т. к. среда разработки не всегда указывала это. Вместо конкретного источника проблемы она могла выдать набор цифр, который давал минимум информации об ошибке. Эту проблему решал запуск программы в режиме отладки – повтор действий приводил к той же ошибке, но среда разработки перед завершением программы указывала точный источник проблемы в виде модуля и номера строки, где возникает та самая ошибка. Кроме этого, в терминале среды разработки указывалась сама ошибка, ее суть.

Вся история изменений хранится в репозитории[[3]](#footnote-3), указанном в разделе «[Список использованной литературы](#_Список_использованной_литературы)» (6) в пункте 10.

# **Эксплуатационная часть**

## **Руководство оператора**

### **Назначение программы**

Основное назначение настольного приложения – упрощенное получение журналов успеваемости любой из групп, присутствующих в Технологическом университете, ККМТ и ТТД.

Дополнительными возможностями программы являются:

* Авторизация;
* Восстановление доступа к аккаунту;
* Изменение электронной почты и пароля от аккаунта:
* Изменение пароля происходит с учетом требований, разработанных создателем приложения. Эта возможность считается исправлением недоработки учебного портала.
* Изменение картинки профиля;
* Синхронизация учебных данных.

### **Условия выполнения программы**

В случае запуска программы с помощью файла с расширением .exe:

1. Устройство, стабильно работающее на поддерживаемых ОС Windows[[4]](#footnote-4).

В случае запуска исходных кодов:

1. Устройство, стабильно работающее на поддерживаемых ОС Windows.
2. Python версии 3.10 и выше;
3. Библиотеки, указанные в файле «requirements.txt».

### **Выполнение программы**

Стартовым окном программы является окно «Авторизации», с помощью которого можно войти в свою учетную запись или восстановить доступ к аккаунту ([приложение 7](#_Приложение_7)). В окне присутствует функционал отображения/скрытия вводимого пароля.

После успешной авторизации открывается основное окно программы, на котором будет отображен профиль пользователя ([приложение 8](#_Приложение_8)). На странице доступны смена картинка профиля и кнопка «Настройки», которая открывает страницу смены для электронной почты и пароля от аккаунта ([приложение 9](#_Приложение_9)). Иконка в виде знака вопроса в круге при наведении отразит требования к новому паролю. В случае их несоблюдения пароль не будет изменен.

С помощью навигационного меню в левой части программы возможна смена страниц ([приложение 10](#_Приложение_10)).

Страница «Журнал» ([приложение 11](#_Приложение_11)) позволяет открыть желаемый журнал успеваемости, выбрав группу, семестр и предмет ([приложение 12](#_Приложение_12)).

Учебные данные сохраняются в базу данных после каждого открытия журнала успеваемости. Это сделано для того, чтобы журналы открывались без каких-либо задержек во времени.

Иконки под полями выбора синхронизируют данные, если они ранее были занесены в базу данных. Поля в базе данных взаимосвязаны – это может привести к непонятной для пользователя логике работы приложения.

Таблица, отражающая выбор действия и результат его работы:

|  |  |
| --- | --- |
| **Действие** | **Результат** |
| Синхронизация групп | Удаление всех данных из базы, кроме данных об аккаунте |
| Синхронизация предметов и семестров | Удаление оценок из базы |

Таблица 1. Логика работы синхронизации данных в базе данных

Страница «О программе» разработана для предоставления краткого описания программы, ее версии и информации о ее разработчиках ([приложение 13](#_Приложение_13)).

### **Сообщение оператору**

В случае ввода ошибочных данных в форму авторизации появится соответствующее сообщения об ошибке ([приложение 4](#_Приложение_4)).

В случае ввода неверной электронной почты в форму восстановления доступа к аккаунту появится соответствующее сообщения об ошибке ([приложение 5](#_Приложение_5)).

В случае ввода неверного кода в форму восстановления доступа к аккаунту появится соответствующее сообщения об ошибке ([приложение 14](#_Приложение_14)).

В случае несоблюдения требований во время смены пароля на странице «Настройки аккаунта», он не будет изменен ([приложение 15](#_Приложение_15)). Сами требования будут отображены в случае наведения курсора на иконку вопросительного знака в круге ([приложение 16](#_Приложение_16)).

Если журнал успеваемости по какой-то причине пуст, т. е. в нем не заполнены даты проведенных занятий, то соответствующее сообщение отобразится на экране пользователя (приложение 17).

## **To-Do лист**

Проект находится на ранней (alpha) стадии разработки. Планируется исправление множества ошибок и добавление нового функционала:

* Пересмотр дизайна окна «Журнал»;
* Исправление ошибки с ограничением размера отправляемого файла для смены картинки профиля;
* Исправление ошибки, связанной со сбрасыванием полей выбора после синхронизации данных;
* Написание документации;
* Исправление ошибок с масштабированием;
* Реализация функционала преподавателя:
* Выставление оценок;
* Формирование отчетов и т. п.
* Реализация возможности смены рабочей базы данных;
* Замена SQLite на PostgreSQL;
* Добавление таблицы в базу данных, связанной с учителями;
* Реализация синхронизаций оценок и учителей;
* Реализация работы приложения без подключения к Интернету;
* Исправление остальных мелких ошибок и т. п.

# **Заключение**

Проект, в случае конечной реализации всех задумок, может быть внедрен в учебный процесс Технологического университета, ККМТ или ТТД, так как является упрощенной версией учебного портала, информация на котором, порой, бывает избыточна. Вдобавок приложение исправляет некоторые ошибки, допущенные разработчиками во время написания учебного портала.

В случае реализации формирования отчетностей и выставления оценок преподавателям будет дана возможность использования единой платформы для выполнения ежедневных обязанностей.

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

1. Система «1С:Образование»: <https://obrazovanie.1c.ru/education/>
2. Демонстрационная версия «1С:Образование»:

[https://e4demo.1c.ru/library.html#section=home](https://e4demo.1c.ru/library.html" \l "section=home)

1. Программный комплекс «Планы СПО»:

<https://www.mmis.ru/programs/planyspo>

1. Система «Moodle»: <https://moodle.org/>
2. Сайт «Diagrams» – построение диаграмм: <https://app.diagrams.net/>
3. Язык программирования «Python»: <https://www.python.org/>
4. Таблица «Популярные языки программирования»:

<https://www.tiobe.com/tiobe-index/>

1. IDE «PyCharm»: <https://www.jetbrains.com/pycharm/>
2. База данных «SQLite»: <https://www.sqlite.org/index.html>
3. Репозиторий проекта: <https://github.com/frrusi/portal-parser>
4. Графическая библиотека PyQt:

<https://www.riverbankcomputing.com/software/pyqt/>

1. Графический фреймворк Qt: <https://www.qt.io/product/framework>
2. Компания JetBrains: <https://www.jetbrains.com/>
3. Компания RiverBank Computing Ltd:

<https://www.riverbankcomputing.com/>

1. Репозиторий курсовой работы:

<https://github.com/frrusi/course-project>

# **Приложения**

## Приложение 1

**from** **PyQt5** **import** QtGui, QtWidgets

**from** **PyQt5.QtCore** **import** Qt

**class** **RoundPixmapStyle**(QtWidgets.QProxyStyle):

**def** **\_\_init\_\_**(self, radius=**10**, \*args, \*\*kwargs):

super(RoundPixmapStyle, self).\_\_init\_\_(\*args, \*\*kwargs)

self.\_radius = radius

**def** **drawItemPixmap**(self, painter, rectangle, alignment, pixmap):

painter.save()

pix = QtGui.QPixmap(pixmap.size())

pix.fill(Qt.transparent)

p = QtGui.QPainter(pix)

p.setBrush(QtGui.QBrush(pixmap))

p.setPen(Qt.NoPen)

p.drawRoundedRect(pixmap.rect(), self.\_radius, self.\_radius)

p.end()

super(RoundPixmapStyle, self).drawItemPixmap(painter, rectangle, alignment, pix)

painter.restore()

## Приложение 2

**from** **PyQt5** **import** QtCore

**from** **PyQt5.QtCore** **import** Qt

**class** **TableModel**(QtCore.QAbstractTableModel):

**def** **\_\_init\_\_**(self, data):

super(TableModel, self).\_\_init\_\_()

self.\_data = data

**def** **data**(self, \*args, \*\*kwargs):

**if** args[**1**] == Qt.DisplayRole:

value = self.\_data.iloc[args[**0**].row(), args[**0**].column()]

**return** str(value)

**elif** args[**1**] == Qt.TextAlignmentRole:

**return** Qt.AlignCenter

**def** **rowCount**(self, \*args, \*\*kwargs):

**return** self.\_data.shape[**0**]

**def** **columnCount**(self, \*args, \*\*kwargs):

**return** self.\_data.shape[**1**]

**def** **headerData**(self, \*args, \*\*kwargs):

**if** args[**2**] == Qt.DisplayRole:

**if** args[**1**] == Qt.Horizontal:

**return** str(self.\_data.columns[args[**0**]])

**if** args[**1**] == Qt.Vertical:

**return** str(self.\_data.index[args[**0**]])

## Приложение 3

"""Модуль, реализующий паттерн 'Одиночка'"""

**class** **ParserMeta**(type):

"""Класс, реализующий паттерн 'Одиночка'"""

\_instances = {}

**def** **\_\_call\_\_**(cls, \*args, \*\*kwargs):

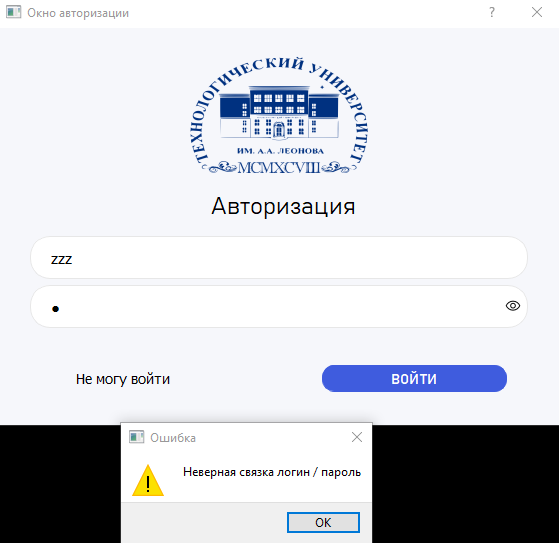
**if** cls **not** **in** cls.\_instances:

instance = super().\_\_call\_\_(\*args, \*\*kwargs)

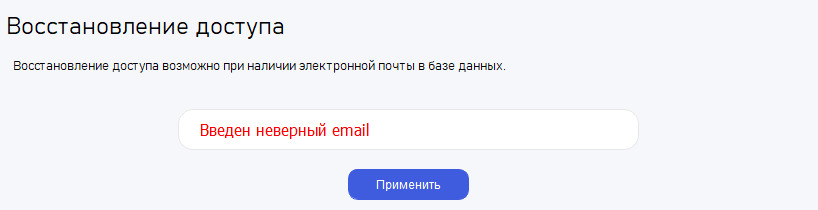
cls.\_instances[cls] = instance

**return** cls.\_instances[cls]

## Приложение 4



## Приложение 5



## Приложение 6

**def** **select\_query**(self, query, return\_type: int):

self.exceptions.check\_value\_by\_number\_range((**1**, **2**), return\_type)

**with** self.engine.connect() **as** connection:

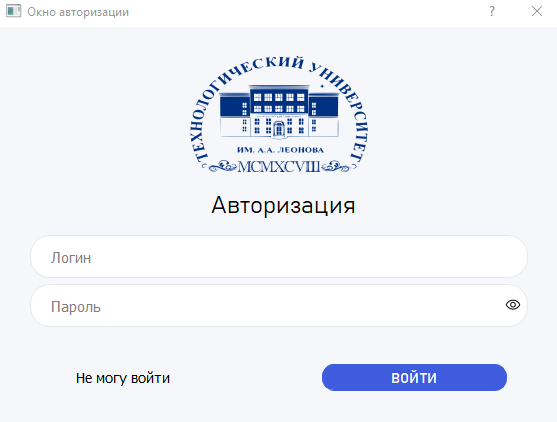
**if** return\_type == **1**:

**return** connection.execute(query).fetchall()

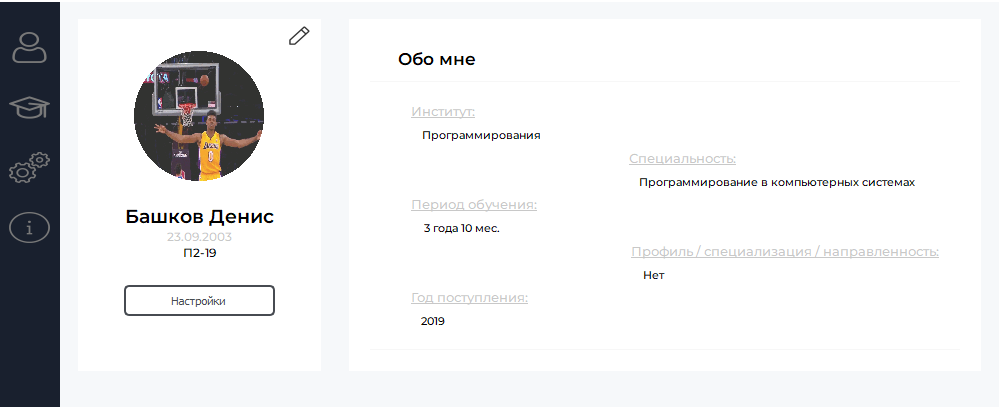
**elif** return\_type == **2**:

**return** connection.execute(query).fetchone()

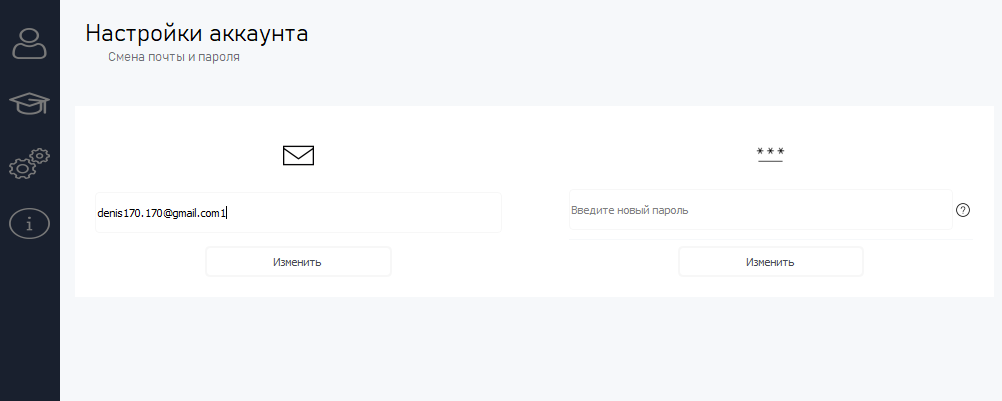
## Приложение 7



## Приложение 8



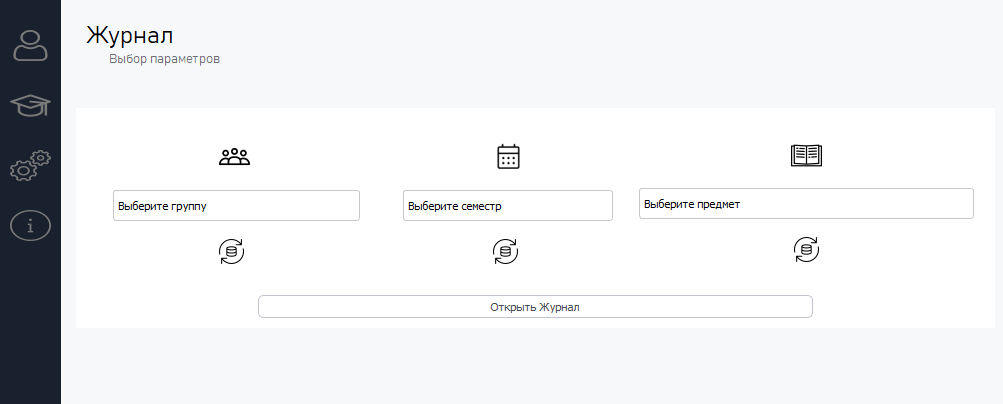
## Приложение 9



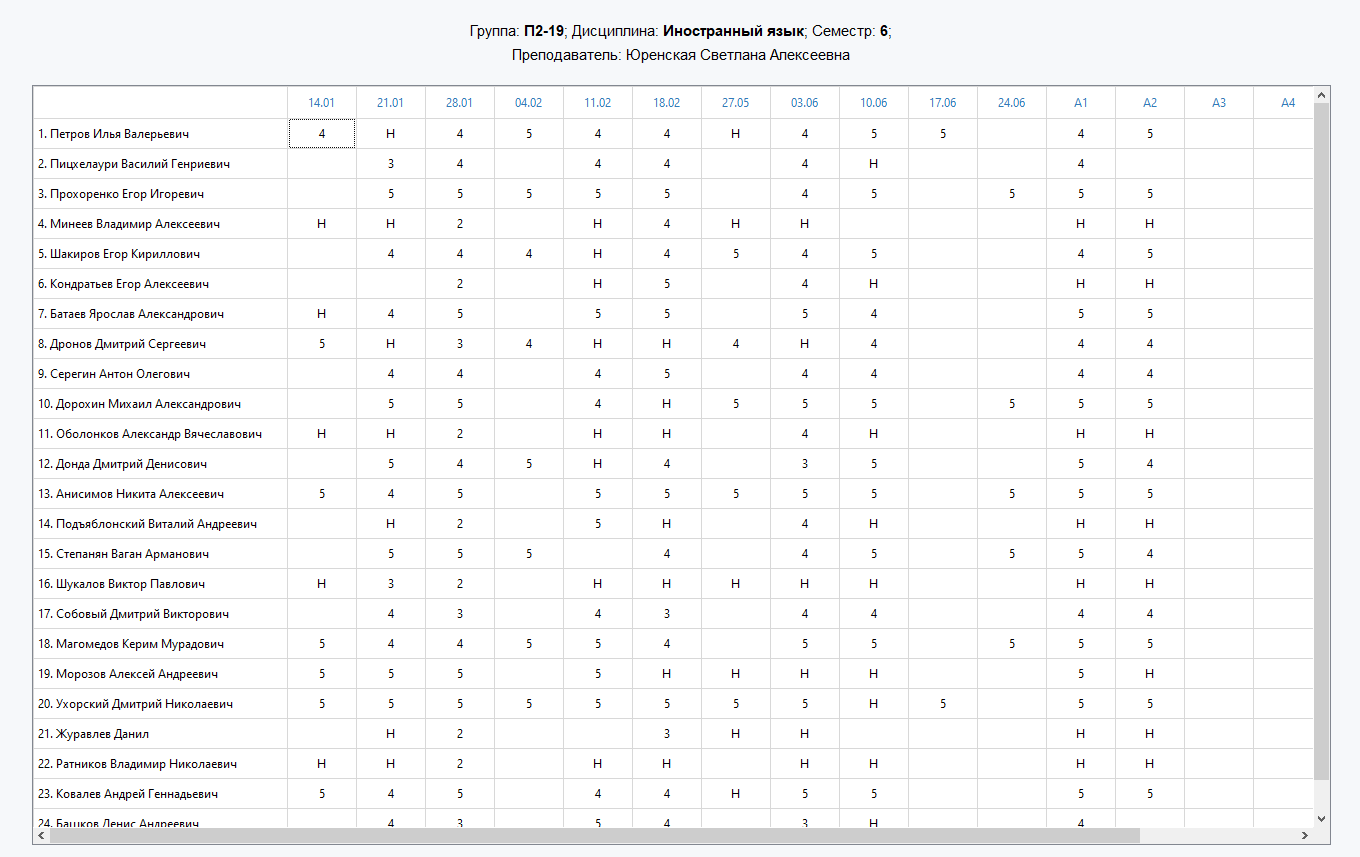
## Приложение 10



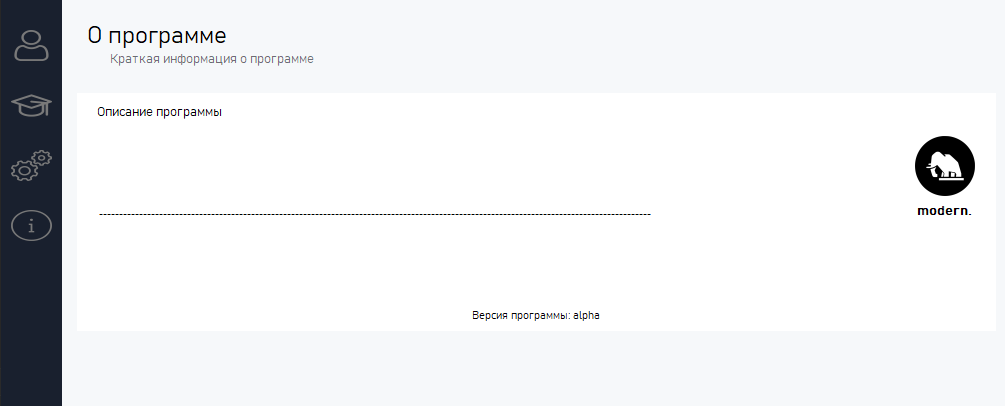
## Приложение 11



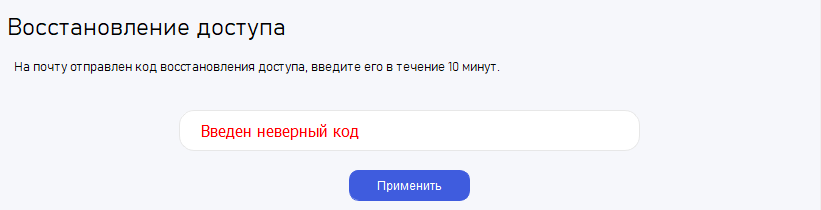
## Приложение 12



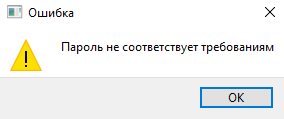
## Приложение 13



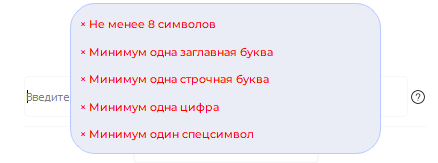
## Приложение 14



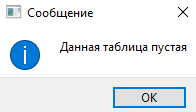
## Приложение 15



## Приложение 16



## Приложение 17



1. Первый полученный элемент [↑](#footnote-ref-1)
2. Все полученные элементы [↑](#footnote-ref-2)
3. Облачное хранилище данных [↑](#footnote-ref-3)
4. На момент написания отчета поддерживаемыми версиями являются: Windows 10, Windows 11 [↑](#footnote-ref-4)