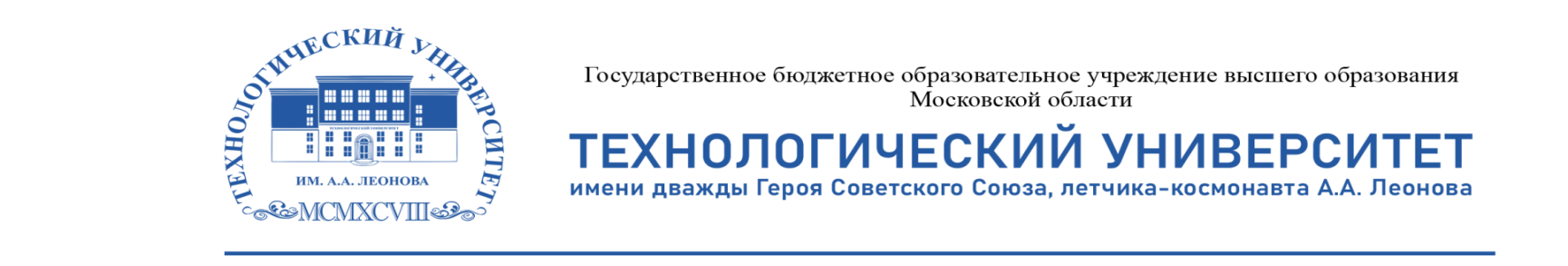
****

Колледж космического машиностроения и технологии

**Курсовой проект**

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

Пояснительная записка

КП.09.02.03.22.03ПЗ

Обучающийся группы П2-19 Башков Д. А.

Руководитель курсового проекта Гусятинер Л. Б.

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

Королев, 2022 г.

**ОГЛАВЛЕНИЕ**

[**1. Введение 4**](#_Toc107162286)

[**2. Теоретическая часть 5**](#_Toc107162287)

[**2.1. Описание предметной области 5**](#_Toc107162288)

[**2.2. Описание существующих разработок 7**](#_Toc107162289)

[**3. Проектная часть 13**](#_Toc107162290)

[**3.1. Построение диаграммы прецедентов 13**](#_Toc107162291)

[**3.2. Выбор инструментов 14**](#_Toc107162292)

[**3.2.1. Язык программирования 14**](#_Toc107162293)

[**3.2.2. Среда разработки 14**](#_Toc107162294)

[**3.2.3. База данных 14**](#_Toc107162295)

[**3.2.4. Графический интерфейс 15**](#_Toc107162296)

[**3.3. Проектирование сценария 16**](#_Toc107162297)

[**3.4. Построение диаграммы классов 17**](#_Toc107162298)

[**3.5. Описание главного модуля 20**](#_Toc107162299)

[**3.6. Описание спецификаций к модулям 21**](#_Toc107162300)

[**3.7. Описание модулей 23**](#_Toc107162301)

[**3.8. Описание тестовых наборов модулей 27**](#_Toc107162302)

[**3.9. Описание применения средств отладки 28**](#_Toc107162303)

[**4. Эксплуатационная часть 29**](#_Toc107162304)

[**4.1. Руководство оператора 29**](#_Toc107162305)

[**4.1.1. Назначение программы 29**](#_Toc107162306)

[**4.1.2. Условия запуска программы 29**](#_Toc107162307)

[**4.1.3. Выполнение программы 29**](#_Toc107162308)

[**4.1.4. Сообщение оператору 30**](#_Toc107162309)

[**4.2. To-Do лист 31**](#_Toc107162310)

[**5. Заключение 32**](#_Toc107162311)

[**6. Список использованной литературы и интернет-ресурсов 33**](#_Toc107162312)

[**7. Приложения 34**](#_Toc107162313)

[**7.1. Приложение 1 34**](#_Toc107162314)

[**7.2. Приложение 2 34**](#_Toc107162315)

[**7.3. Приложение 3 35**](#_Toc107162316)

[**7.4. Приложение 4 35**](#_Toc107162317)

[**7.5. Приложение 5 35**](#_Toc107162318)

[**7.6. Приложение 6 36**](#_Toc107162319)

[**7.7. Приложение 7 36**](#_Toc107162320)

[**7.8. Приложение 8 36**](#_Toc107162321)

[**7.9. Приложение 9 37**](#_Toc107162322)

[**7.10. Приложение 10 37**](#_Toc107162323)

[**7.11. Приложение 11 37**](#_Toc107162324)

[**7.12. Приложение 12 38**](#_Toc107162325)

[**7.13. Приложение 13 38**](#_Toc107162326)

[**7.14. Приложение 14 38**](#_Toc107162327)

[**7.15. Приложения 15 39**](#_Toc107162328)

[**7.16. Приложение 16 39**](#_Toc107162329)

# **Введение**

За последние годы наметилась устойчивая тенденция учебных заведений в создании образовательных порталов в качестве средства предоставления учебной информации.

Основной целью создания образовательного портала является организация между различными учебными заведениями: Технологический университет, ККМТ и ТТД. Что позволяет им обеспечивать доступ к научной и образовательной информации.

Данный курсовой проект заключается в разработке приложения для OC Windows, которая будет предоставлять минимальный, но самый основной функционал образовательного портала. Разработка приложения будет происходить на Python.

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, 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;

**Образовательный портал** – это информационная система, предоставляющая доступ к информационным ресурсам и услугам образовательного характера, сайт предназначена для доступа только ограниченному кругу лиц, т. е. студентам, преподавателям относящихся к учебному заведению.

Образовательный портал реализует следующие возможности:

1. Введение электронных журналов;
2. Возможность посмотреть расписание занятий;
3. Получение и сдача домашних заданий;
4. Возможность общения с пользователями портала;
5. Возможность получить доступ к учебным материалам;
6. Возможность публикаций новостных постов.

Студентам общеобразовательный портал необходим для просмотра расписаний занятий, выданных домашних заданий и успеваемости за пройденный материал.

Преподавателям общеобразовательный портал необходим для организации и построения образовательного процесса, а также для предоставления необходимой информации по обучению и успеваемости.

Описание возможностей системы «Образовательного портала**»**:

**Журнал успеваемости**

Студенты имеют доступ для просмотра выставленных оценок по всем дисциплинам.

Преподаватели в журнале имеют доступ в систему для выставления и внесение изменений оценок по своей дисциплине, а также ведение ежедневной программы обучения.

**Расписание занятий**

Расписание занятий по дисциплинам на общеобразовательном портале отображается в свободном доступе для всех преподавателей и студентов, и предоставляется следующая информация:

1. Дата проведения дисциплины;
2. Время проведения дисциплины;
3. ФИО преподавателя;
4. Название дисциплины;
5. Место проведения занятия;
6. Информация об изменении расписания.

**Дистанционное обеспечение**

Образовательный портал реализует функцию взаимодействия преподавателя и студента на расстоянии, для предоставления домашних заданий и отправление выполненного результата.

## **Описание существующих разработок**

1. Moodle [1] – бесплатная система для создания персонализированной среды обучения. Через нее вы можете дистанционно развивать и тестировать учеников со всего мира. Она одна из самых популярных систем электронного обучения. Система переведена более чем на 100 языков, в ней работают крупные университеты во всем мире.

Основные возможности Moodle:

1. **Создания онлайн-курсов.** Текстовые документы, презентации и видео можно объединить в обучающий курс, который будет доступен всем ученикам или отдельному классу.
2. **Тестирование.** В Moodle встроен редактор тестов. По умолчанию доступно 15 типов заданий. Чтобы ученики не списывали, можно ограничить время на решение теста и число попыток. Система автоматически проверяет ответы, показывает допущенные ошибки и указывает набранный балл.
3. **Форум и комментарии.** Чтобы связаться с преподавателем, задать вопрос или обсудить тему урока, ученики могут оставлять комментарии под курсами или заводить беседы на встроенном форуме (рисунок 1).
4. **База знаний.** Круглосуточно всем пользователям доступен архив учебных материалов. В любой момент ученики могут зайти в базу знаний и найти нужный доклад, видеоурок или статью.
5. **Мобильное обучение.** У сервиса есть мобильное приложение Moodle Mobile, которое позволяет проходить курсы и решать тесты с планшета или смартфона.
6. **Статистика по обучению.** Moodle отслеживает успеваемость учеников (рисунок 2) и составляет отчёты для преподавателей (рисунок 3). Например, позволяет сформировать отчет о времени и об ошибках при прохождении курса обучения.

Изображение выглядит как текст

Автоматически созданное описание

Рис. 1. Создание форума для обсуждений «Moodle»

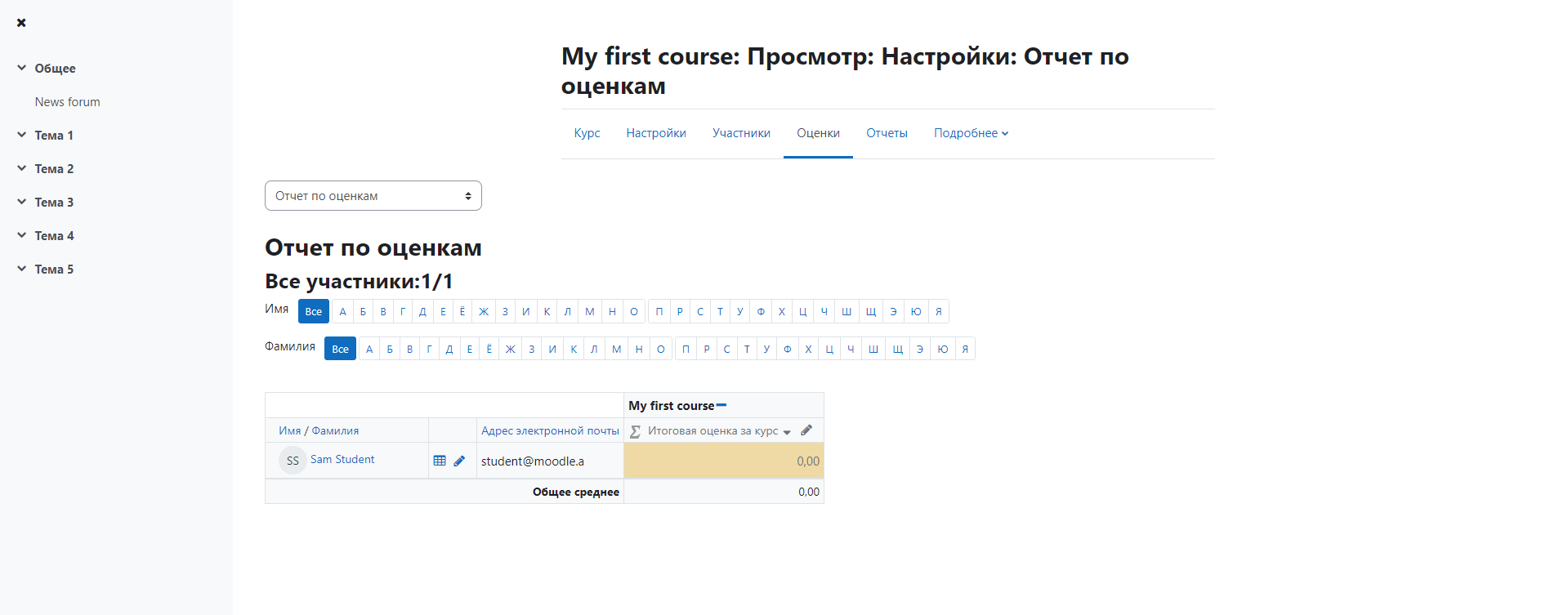


Рис. 2. Отслеживание успеваемости учеников «Moodle»

Изображение выглядит как текст

Автоматически созданное описание

Рис.3. Составление отчетов для преподавателя «Moodle»

1. Canvas [2] – система управления обучением, которая помогает организовать учебный процесс на базе новейших информационных технологий. Программный продукт предназначен для дистанционного обучения. Canvas применяется академическими и образовательными учреждениями (школами, колледжами, институтами, университетами).

Программная система Canvas реализует основные функции управления обучением: управление классом, группой, управление испытаниями (тестовые здания, зачёты), совместные работы и организации учебных материалов.

Основные возможности Canvas:

1. Создание интересного внутреннего содержания курсов, с помощью текста, фотографий, видео и управления (рисунок 4).
2. Импорт ранее разработанных учебных курсов.
3. Изменение и отслеживание преподавателем прогресса учащихся для принятия своевременных и обоснованных учебных решений (рисунок 5).
4. Организация и участие в синхронных онлайн-коммуникациях.
5. Доступ родителей/руководителей к данным о прогрессе обучения своего ребенка.

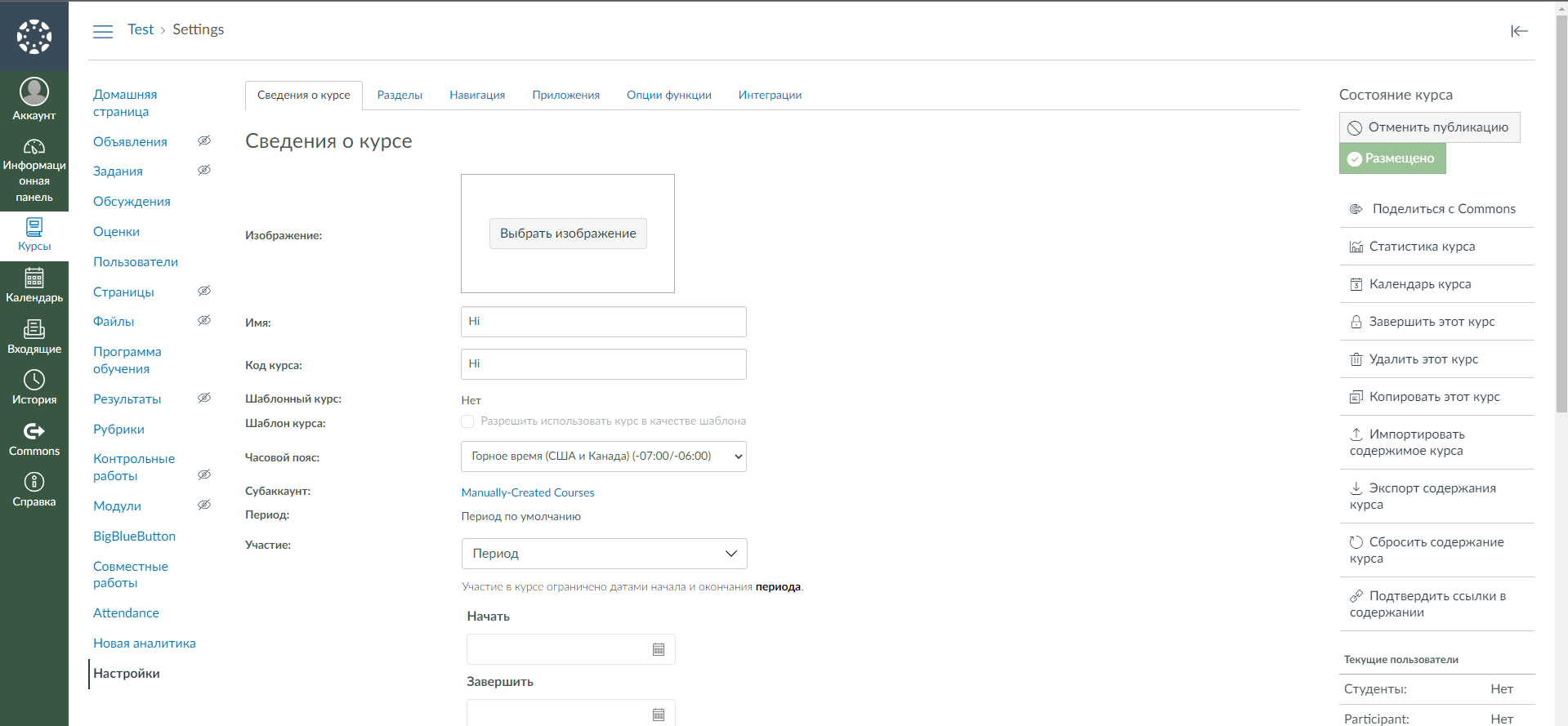


Рис.4. Создания курса «Canvas»

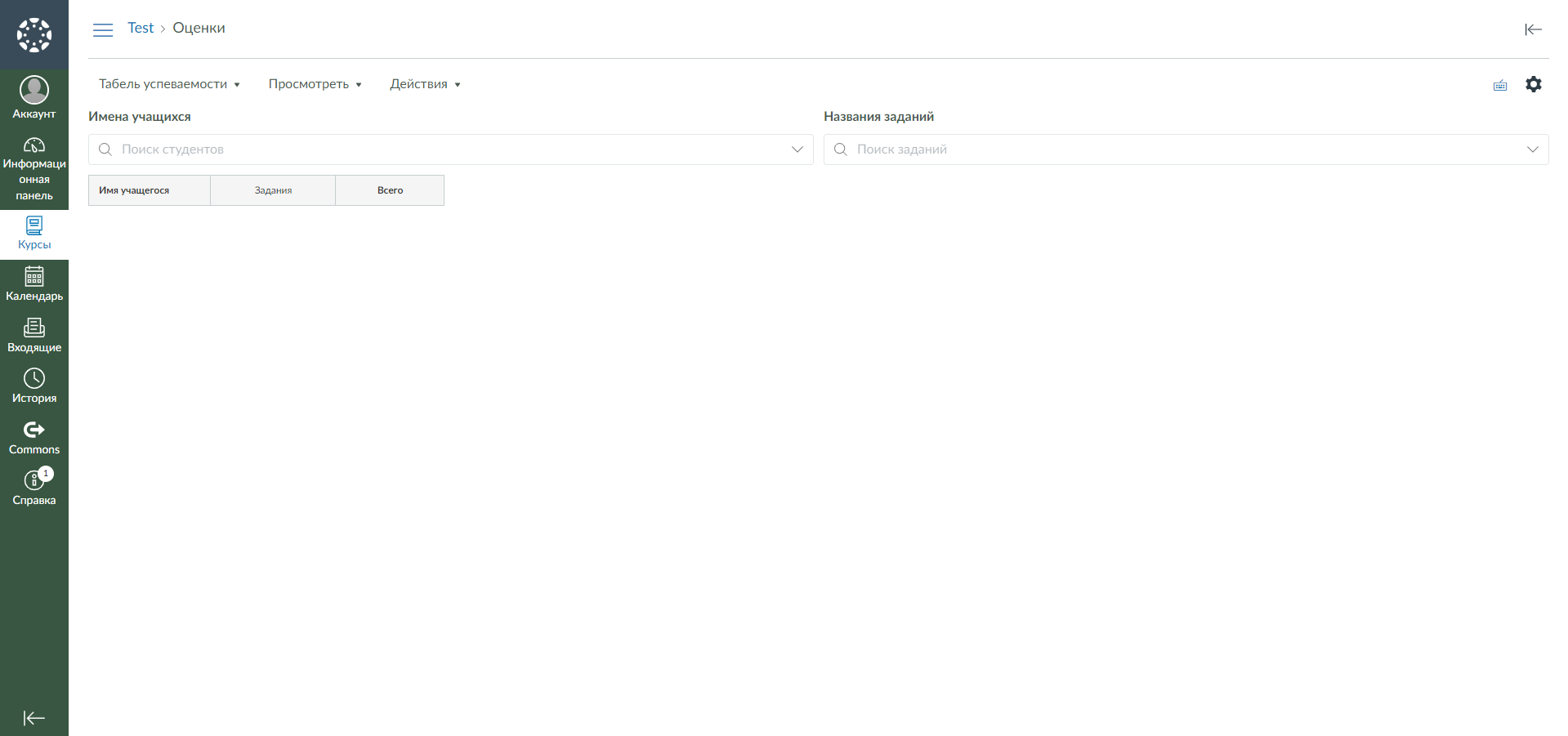


Рис.5. Отслеживание преподавателем прогресса «Canvas»

1. Система «1C:Образование» [3] – это комплексная облачная система автоматизации учебного процесса. Фирма «1С» предлагает пользователям доступ через интернет к системе электронного и дистанционного обучения «1С: Образование» без затрат на ее развертывание, администрирование и эксплуатацию в сети образовательной организации. В комплект также входит большая библиотека цифровых учебных материалов.

Основные возможности 1С Образования:

1. Цифровая библиотека. Онлайн работа с цифровыми образовательными ресурсами и учебными пособиями является одной из самых востребованных форм использования программы в школе или колледже. Такую возможность педагогам и учащимся дает цифровая библиотека. Её ресурсами можно воспользоваться на любом компьютере, ноутбуке или планшете, подключенном к сети интернета.
2. Создание учебных материалов. Педагоги и учащиеся могут сами создавать цифровые учебные материалы. Это могут быть иллюстрированные тексты, материалы на основе видеофрагментов, рисунков, электронных таблиц, презентации.
3. Задания и тесты. Редактор тестов поможет педагогу подготовить интерактивные задания с автоматической проверкой знаний учащихся по разным темам и в разных форматах. (рисунок 6).
4. Контроль и анализ результатов учебной деятельности. Основным инструментом для контроля и анализа результатов учебной деятельности является электронный журнал. Каждая выставленная оценка может быть описана типом урока и типом учебной деятельности, за которую она поставлена. На основе выставленных оценок формируются отчеты для преподавателя – текущая успеваемость и качество знаний, средний балл учащегося за различные виды учебной деятельности (рисунок 7).

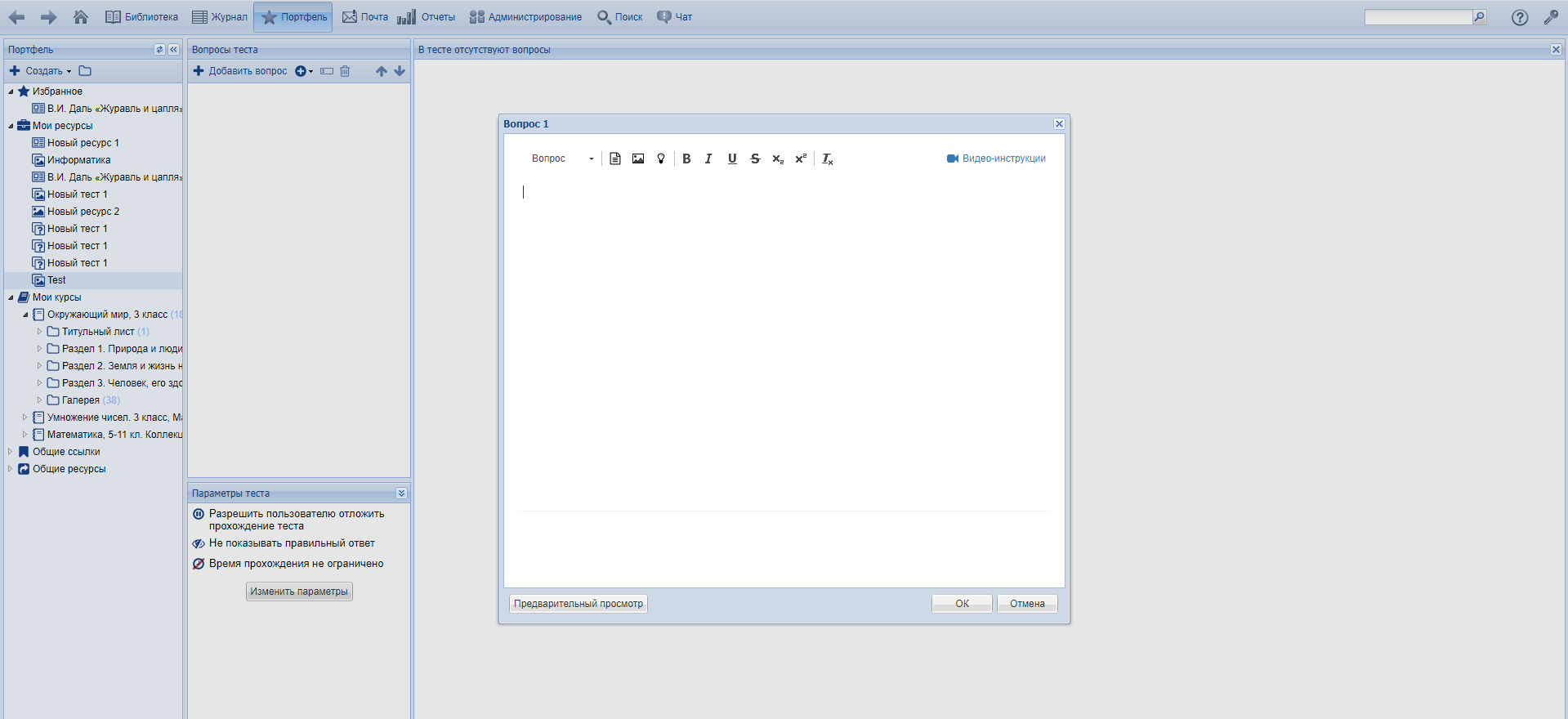


Рис.6. Создание теста «1C:Образование»

Изображение выглядит как текст

Автоматически созданное описание

Рис.7. Журнал успеваемости «1С:Образование»

# **Проектная часть**

## **Построение диаграммы прецедентов**

Данный раздел содержит диаграмму прецедентов (рисунок 8), на которой отображается визуализация различных типов ролей в системе и то, как эти роли взаимодействуют с системой.

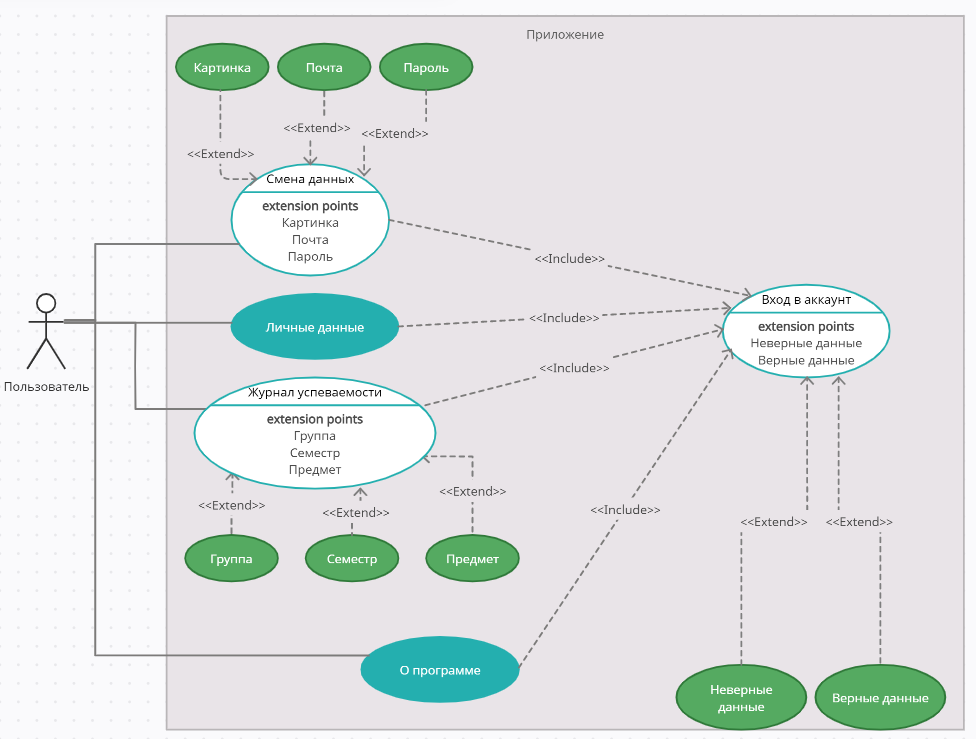
****

Рис.8. Диаграмма прецедентов. Выполнил: Собовый Дмитрий Викторович

## **Выбор инструментов**

### **Язык программирования**

Для написания проекта использовался Python [5] – 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.

### **Среда разработки**

PyCharm [6] – 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 – 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 [8] – 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 – определение сочетаний условий, которые приводят к определенным результатам (рисунок 9).

При запуске приложения открывается окно авторизации, в котором необходимо ввести логин и пароль своей учетной записи, а также есть возможность для восстановления доступа (пароль) для входа в систему.

После успешной авторизации в системе, пользователю предоставляется полный функционал приложения, а именно:

1. Окно «Профиль» – в данном разделе отображается основная информация о пользователе (Личные данные: ФИО, группа, дата рождения; корпоративные данные: институт, специальность, период обучения, специализация, год поступления.) и предоставляется возможность смены фотографии профиля, почты и пароля;
2. Окно «Журнал успеваемости» – открывается после пошагового выбора данных: группы, семестра и предмета.
3. Окно «О программе» – информация о программе, её создателях и версии самой программы.

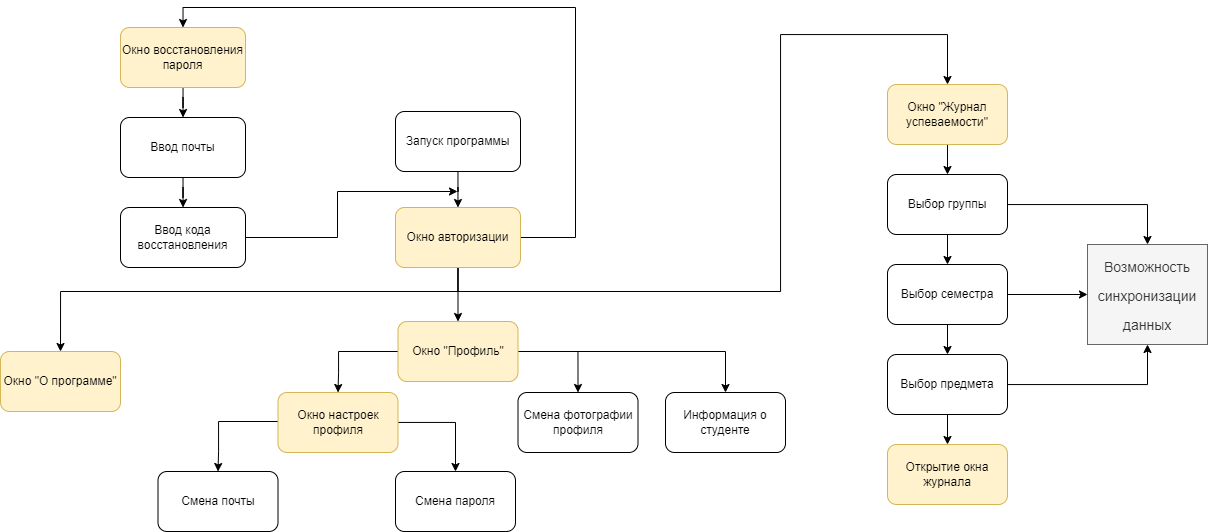
****

Рис.9. Сценарий проекта. Выполнил: Собовый Дмитрий Викторович

## **Построение диаграммы классов**

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

Подробное описание разделов предоставлены в пунктах: п.3.6 «Описание спецификаций к модулям» и п.3.7 «Описание модулей».

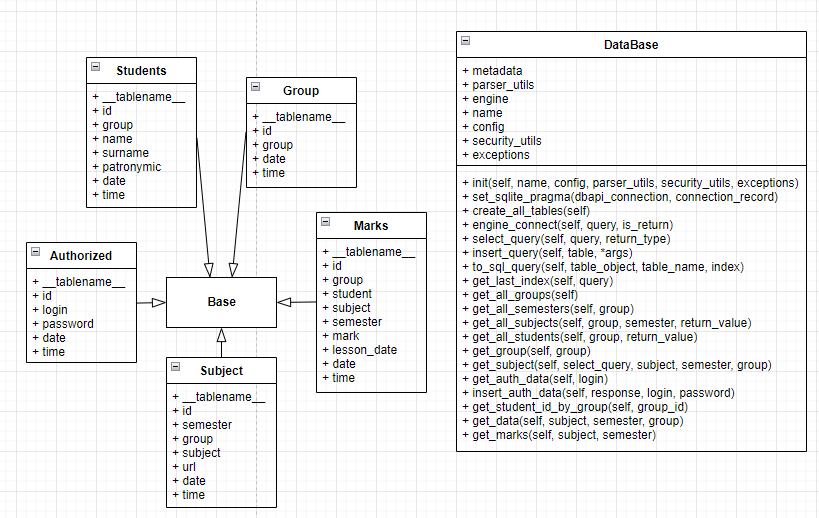


Рис.10. Диаграмма классов раздел «База данных». Выполнил: Собовый Дмитрий Викторович

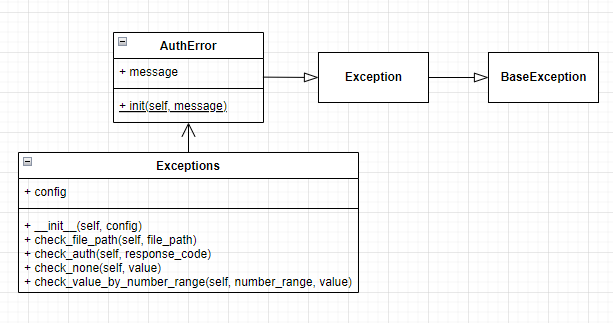


Рис.11. Диаграмма классов раздел «Исключения». Выполнил: Собовый Дмитрий Викторович

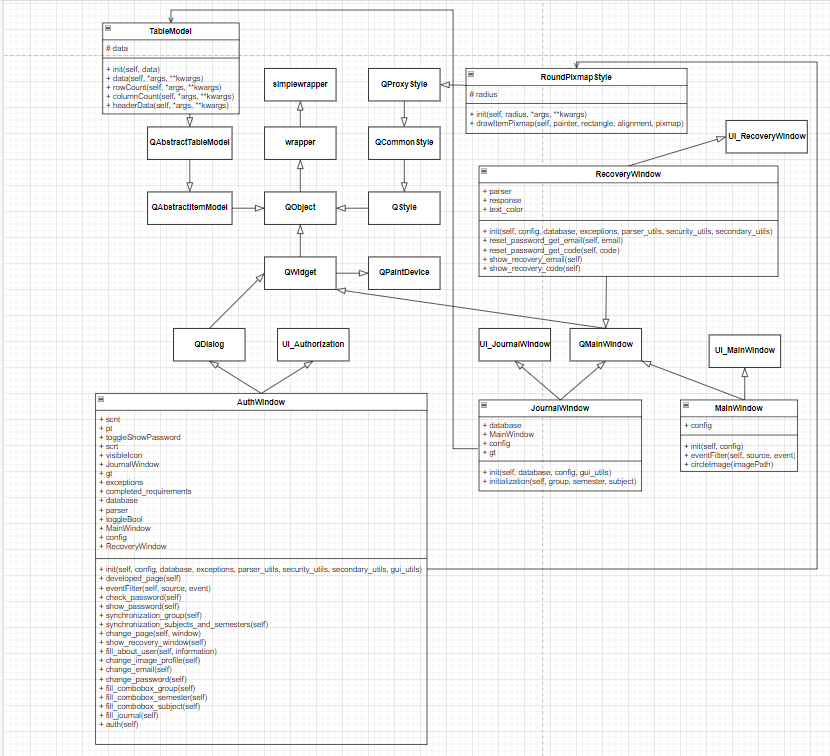


Рис.12. Диаграмма классов раздел «Графический интерфейс». Выполнил: Собовый Дмитрий Викторович

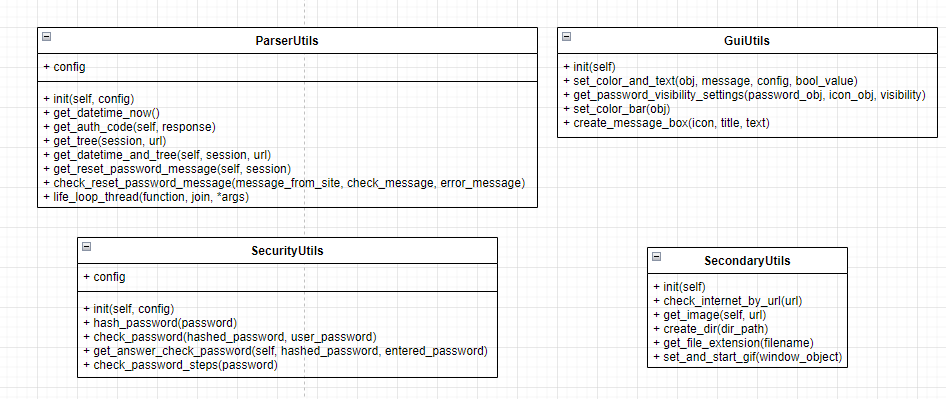


Рис.13. Диаграмма классов раздел «Утилиты». Выполнил: Собовый Дмитрий Викторович

**Изображение выглядит как текст

Автоматически созданное описание**

Рис.14. Диаграмма классов раздел «Конфигурационные файлы». Выполнил: Собовый Дмитрий Викторович

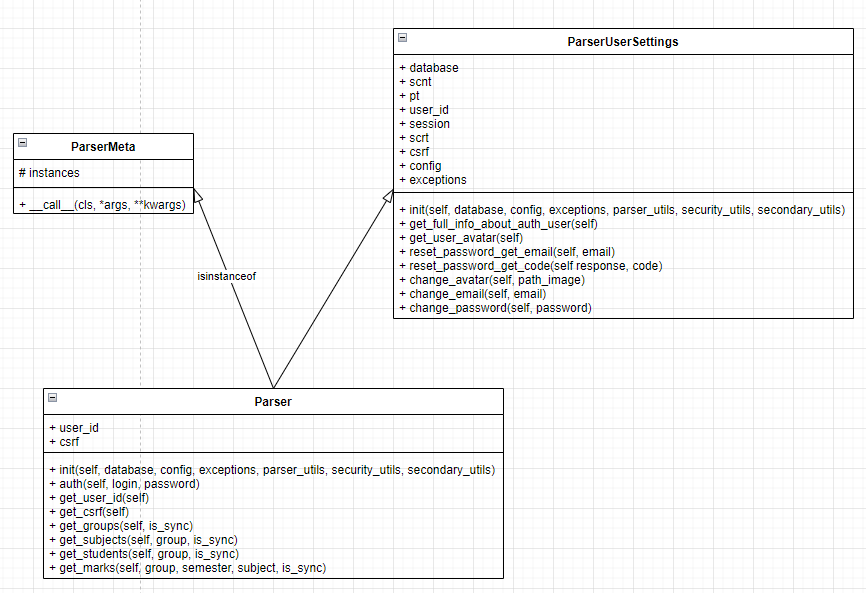
****

Рис.15. Диаграмма классов раздел «Парсер». Выполнил: Собовый Дмитрий Викторович

## **Описание главного модуля**

Главный модуль (листинг 1) состоит из функции main. Данная функция включает в себе объекты необходимых для окна авторизации классов и запускает графический интерфейс.

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

Реализованные в проекте модули можно разделить на типы:

1. Сборщик данных (рисунок 15):
   1. Работа с учебными данными (parser);
   2. Работа с данными пользователя (parser\_user\_settings);
   3. Реализация паттерна «Одиночки» (parser\_meta).
2. База данных (рисунок 10):
   1. Описание моделей данных (models);
   2. Связь с базой данных (database).
3. Графический интерфейс (рисунок 12):
   1. Работа окна «Авторизации», подключение действий основного окна (auth\_connection);
   2. Работа окна «Восстановление доступа»;
   3. Установка иконок, закругление фотографии, подключение некоторых действий в основном окне (main\_connection);
   4. Работа окна «Журнал» (journal\_connection);
   5. Реализация закругления GIF-анимации (roundpixmapstyle);
   6. Реализация собственной модели журнала в виде таблицы (tablemodel).
4. Константные файлы (рисунок 14):
   1. Хранение списков и словарей (config);
   2. Считывание файлов с расширением .ini (config\_parser).
5. Вспомогательные (рисунок 13):
   1. Работа с графическим интерфейсом (gui\_utils);
   2. Работа со сборщиком данных (parser\_utils);
   3. Работа во многих частях программы (secondary\_utils);
   4. Хеширование паролей, и их проверка на соответствие (security\_utils).
6. Исключения (рисунок 11):
   1. Реализация собственных исключений и необходимых в программе проверок (exceptions).

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

## **Описание модулей**

1. Тип «Сборщик данных» (рисунок 15) полностью отвечает за взаимодействие с учебным порталом, который включает в себя три модуля:
   1. Модуль с учебными данными «parser» содержит в себе сборщик учебных данных: группы, студенты и предметы, оценки по этим предметам;
   2. Модуль с данными пользователя «parser\_user\_settings» отвечает за получение или изменение данных аккаунта пользователя. Изменения пароля, электронной почты или фотографии в профиле;
   3. Модуль реализации паттерна «Одиночка» «parser\_meta» - структура проектирования, которая гарантирует, что у класса есть только один экземпляр, и предоставляет к нему глобальную точку доступа (приложение 3).
2. Тип «База данных» (рисунок 10) полностью отвечает за создание и получение данных из базы, который включает в себя два модуля:
   1. Модуль описания моделей базы «models» содержит описание всех моделей таблиц. Под описанием понимается: наименование таблиц, полей и настройки полей;
   2. Модуль связь с базой данных «database» содержит реализацию работы с базой данных: подключение к базе, создание таблиц, запросы.
3. Тип «Графический интерфейс» (рисунок 12) полностью отвечает за визуализацию приложения и включает в себя 6 модулей:
   1. Модуль работы окна авторизации «auth\_connection». Он содержит почти все события главного окна;
   2. Модуль работы окна восстановления доступа «recovery\_connection» содержит в себе поэтапный ввод данных для успешного восстановления доступа к аккаунту с сопровождающими проверками;
   3. Модуль, инициализирующий основное окно «main\_connection» содержит в себе установку необходимых иконок и реализацию закругления фотографии;
   4. Модуль работы окна журнала «journal\_connection» содержит в себе заполнение необходимых данных в таблицу;
   5. Модуль реализация закругления GIF-анимации «roundpixmapstyle» содержит в себе переопределенный существующий метод, позволяющий закруглить GIF-анимацию (приложение 1).
   6. В модуле «tablemodel» реализована собственная модель таблицы для отображения журнала успеваемости с целью ускорить процесс сборки самой таблицы (приложение 2).
4. Тип «Константные файлы» (рисунок 14) включает в себя два модуля:
   1. Модуль хранения списков и словарей «config» содержит в себе списки и словари, необходимые в программе.
   2. Модуль «config\_parser» считывает файлы с расширением .ini, которые хранят в себе строковые константы (листинг 2).

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 2. Модуль «config\_parser», реализующий считывание файла с расширением .ini

1. Тип «Вспомогательные» полностью отвечает за общие вспомогательные методы:
   1. Модуль «gui\_utils» отвечает за изменение стилистики некоторых виджетов, отображение информационных окон;
   2. Модуль «parser\_utils» отвечает за используемые методы в сборщике данных: получение текущей даты и времени, проверка аутентификации т. д;
   3. Модуль «secondary\_utils» отвечает за используемые методы во всей программе: создание директории, получение расширения файла и т. д;
   4. Модуль «security\_utils» отвечает за безопасность, другими словами, он хеширует пароли, проверяет их на соответствие требованиям.
2. Тип «Исключения» (рисунок 11) полностью отвечает за хранение собственных реализованных исключений и необходимых в программе проверок, например, проверка на авторизацию пользователя (листинг 3) или проверка на пустое значение (листинг 4).

def check\_auth(self, response\_code: str):  
 *"""Метод проверки успеха авторизации  
  
 Args:  
 response\_code: Код ответа от сервера  
 Raises:  
 AuthError: Если авторизация не пройдена  
 """* if response\_code != self.config.successful\_code:  
 raise AuthError(self.config.auth\_error)

Листинг 3. Проверка на авторизацию пользователя

def check\_none(self, value: Any):  
 *"""Метод проверки значения на None-значение  
  
 Args:  
 value: Переданное значение  
 Raises:  
 ValueError: Если переменная - None  
 """* if value is None:  
 raise ValueError(self.config.none\_value\_error)

Листинг 4. Проверка на пустое значение

## **Описание тестовых наборов модулей**

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

1. Проверка авторизацию пользователя.

В случае отправки некорректных данных во время авторизации, пользователь получит уведомление о соответствующей ошибке в отдельном окне (приложение 4).

1. Проверка восстановления доступа к учетной записи портала.

В случае ввода неверной электронной почты, не привязанной к учебному порталу, пользователь получит уведомление об ошибке. Такой же исход будет на стадии ввода кода восстановления (приложение 5).

## **Описание применения средств отладки**

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.

Все история изменения храниться в облачном хранилище данных, указанном в пункте 6 «Список использованной литературы» [9].

# **Эксплуатационная часть**

## **Руководство оператора**

### **Назначение программы**

Разработка настольного приложения для сокращения по времени получения журналов успеваемости любой из групп, присутствующих в Технологическом университете, ККМТ и ТТД.

Дополнительные возможности программы:

1. Авторизация;
2. Восстановление доступа к аккаунту;
3. Изменение электронной почты и пароля от аккаунта:
   1. Изменение пароля происходит с учетом требований.
4. Изменение фотографии профиля;
5. Синхронизация учебных данных.

### **Условия запуска программы**

Запуск программы с помощью файла с расширением .exe:

1. Стабильно работающее устройство на OC Windows[[1]](#footnote-1).

Запуск программы с помощью исходного кода:

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

### **Выполнение программы**

При запуске программы появляется окно «Авторизации», с помощью которого можно войти в свою учебную запись или восстановить доступ к аккаунту (приложение 7). В данном окне присутствует функционал отображения/скрытия вводимого пароля.

После успешного ввода пароля открывается окно «Основное», на котором отображается профиль пользователя (приложение 8). На странице доступна смена фотографии профиля и кнопка «Настройки», которая открывает страницу со сменой почты и пароля (приложение 9). При наведении на иконку в виде знака вопроса отображаются требования к новому паролю, при несоблюдении данных требований пароль не будет изменен (приложения 15).

Навигационное меню в левой части программы позволяет переключаться между страницами в приложении. (приложение 10).

Страница «Журнал» (приложение 11) позволяет открыть журнал успеваемости, выбрав группу, семестр и предмет (приложение 12).

Учебные данные сохраняются в базу данных после каждого нового открытого журнала успеваемости. Это уменьшает время при повторном открытии данного окна.

Иконки под полями выбора синхронизируют данные, что позволяет обновить данные и избежать устаревшей информации. Иконка под полем «Выбор группы», синхронизирует группы, т. е. удаление всех данных из базы, кроме данных об аккаунте. Иконка под полями «Выбор семестра» и «Выбор предмета», синхронизирует семестры и предметы, т. е. удаление оценок из базы.

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

### **Сообщение оператору**

В случае ввода неверных данных в окне «Авторизации» появится соответствующее сообщение об ошибке (приложение 4).

В случае ввода неверной электронной почты в окне «Восстановления доступа» появится соответствующее сообщение об ошибке (приложение 5).

В случае ввода неверного кода в окне «Восстановления доступа» появится соответствующее сообщение об ошибке (приложение 6).

В случае несоблюдения требований во время смены пароля на странице «Настройки аккаунта», пароль не будет изменен (приложение 14).

В случае если журнал успеваемости по какой-то причине не заполнен, то соответствующее сообщение отобразится на экран пользователя (приложение 17).

## **To-Do лист**

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

1. Пересмотр дизайна страницы «Журнал»;
2. Реализация работы приложения без подключения к Интернету;
3. Добавление выбора темы приложения;
4. Написание документации;
5. Исправление ошибок с масштабированием;
6. Реализация функционала преподавателя:
   1. Выставление оценок;
   2. Формирование отчетов.
7. Замена базы данных SQLite на PostgreSQL;
8. Добавление таблицы в базу данных, связанной с преподавателеми;
9. Реализация синхронизаций оценок и преподавателей;
10. Исправление ошибки, связанной со сбрасыванием полей выбора после синхронизации данных;
11. Исправление ошибки с ограничением размера отправляемого файла для смены картинки профиля;
12. Исправление остальных мелких ошибок.

# **Заключение**

Данный проект предоставляет минимальный, но самый основной функционал образовательного портала. После окончательной доработки приложения он может быть внедрен в учебный процесс Технологического университета, ККМТ и ТТД. Это позволит сократить время просмотра актуальный данных в журнале успеваемости любой из групп, а также внедрение данного проект исключит некоторые ошибки разработчиков образовательного портала.

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

1. Система «Moodle»: <https://moodle.org/>
2. Система «Canvas»: <https://www.instructure.com>
3. Система «1С:Образование»: <https://obrazovanie.1c.ru/education/>
4. Демонстрационная версия «1С:Образования»: [https://e4demo.1c.ru/library.html#section=home](https://e4demo.1c.ru/library.html%23section=home)
5. Язык программирования «Python»: <https://www.python.org/>
6. Среда разработки IDE «PyCharm»: <https://www.jetbrains.com/pycharm/>
7. База данных «SQLite»: <https://www.sqlite.org/index.html>
8. Графическая библиотека PyQt: <https://www.riverbankcomputing.com/software/pyqt/>
9. Облачное хранилище проекта: <https://github.com/frrusi/portal-parser>

# **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

`Изображение выглядит как текст

Автоматически созданное описание

## Приложение 5

Изображение выглядит как текст

Автоматически созданное описание

## Приложение 6

Изображение выглядит как текст

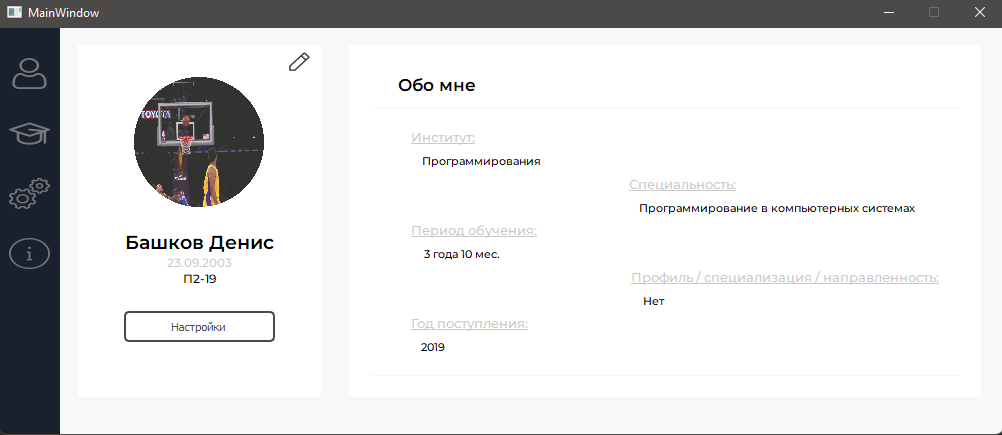
Автоматически созданное описание

## Приложение 7

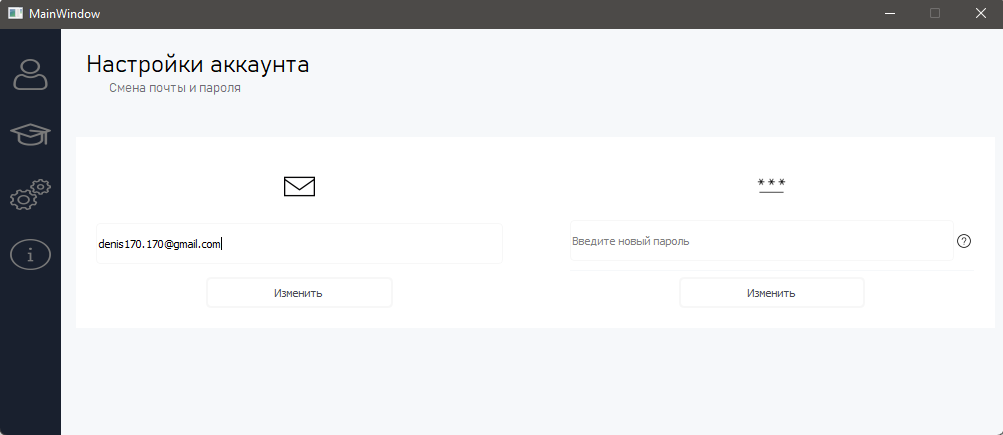
Изображение выглядит как текст

Автоматически созданное описание

## Приложение 8



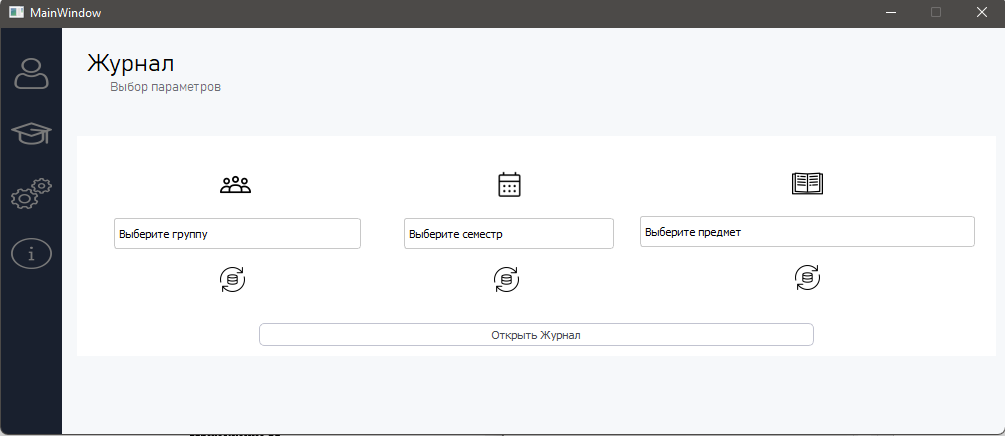
## Приложение 9



## Приложение 10



## Приложение 11



## Приложение 12

Изображение выглядит как текст, внутренний, снимок экрана

Автоматически созданное описание

## Приложение 13

Изображение выглядит как текст

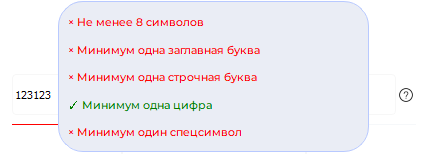
Автоматически созданное описание

## Приложение 14

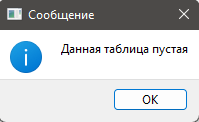
Изображение выглядит как текст

Автоматически созданное описание

## Приложения 15



## Приложение 16



1. На момент написания отчета поддерживаемыми версиями являются: Windows 10, Windows 11 [↑](#footnote-ref-1)