

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра вычислительной техники

УТВЕРЖДАЮ
Заведующий кафедрой
_____ О. В. Непомнящий
подпись
« _____ » _____ 2020 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 – «Информатика и вычислительная техника»

Приложение для мониторинга студентов

Руководитель	_____	канд. техн. наук, доцент	А. И. Постников
	подпись, дата		
Выпускник	_____		П. К. Толкачев
	подпись, дата		
Консультант	_____	ст. преподаватель	К. В. Пушкарев
	подпись, дата		
Нормоконтролер	_____		А. И. Постников
	подпись, дата		

Красноярск 2020

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ
Заведующий кафедрой
_____ О. В. Непомнящий
подпись
« ____ » _____ 20 ____ г

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы**

Студенту Толкачеву Павлу Константиновичу.

Группа: КИ16-06Б. Направление (специальность): 09.03.01
«Информатика и вычислительная техника».

Тема выпускной квалификационной работы: «Приложение для мониторинга студентов».

Утверждена приказом по университету № 6622/с от 26.05.2020.

Руководитель ВКР: А. И. Постников, канд. техн. наук, доцент, доцент каф.
вычислительной техники ИКИТ СФУ.

Исходные данные для ВКР: нет.

Перечень разделов ВКР:

1. Анализ задания на выпускную квалификационную работу.
2. Проектирование и реализация приложения.
3. Инструкции.

Перечень графического материала: не требуется.

1 Задание

Разработать приложение, позволяющее хранить и обрабатывать информацию о студентах вуза, их успеваемости, посещении занятий, научной работе, внеучебной деятельности.

2 Основные требования

Разработанное приложение должно быть кроссплатформенным для настольных систем (ОС Windows, Linux), иметь графический пользовательский интерфейс, открытый исходный код и обладать следующими возможностями:

- а) хранение и обработка информации о студентах вуза, их успеваемости, посещении занятий, научной работе, внеучебной деятельности;
- б) ввод/вывод данных через графический интерфейс, а также в машиночитаемом файловом формате (например, CSV, XML и т. п.);
- в) обмен информацией с LMS Moodle;
- г) наглядное представление информации в различных разрезах: по студентам, по времени, по дисциплине, по группе, по курсу, по семестру, по

форме контроля (зачёт, экзамен, курсовая работа/проект), по сессии (летняя/зимняя), по виду элемента учебного плана (дисциплина, практика), по виду занятия (лекция, практическое, лабораторное, экзамен и т. д.);

д) базовые возможности анализа и визуализации данных: вычисление статистических показателей, построение диаграмм;

е) создание отчётов.

Руководитель ВКР

подпись

А. И. Постников

Задание принял к исполнению

подпись

П. К. Толкачев

30 декабря 2019 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Приложение для мониторинга студентов» содержит 54 страницы текстового документа, 11 таблиц, 31 иллюстраций, 23 использованных источника.

МОНИТОРИНГ СТУДЕНТОВ, УСПЕВАЕМОСТЬ СТУДЕНТОВ, АНАЛИЗ УСПЕВАЕМОСТИ, MYSQL, PYTHON, SQLALCHEMY, ВИЗУАЛИЗАЦИЯ, БАЗА ДАННЫХ, QT, ПРИЛОЖЕНИЕ.

Целью выпускной квалификационной работы является разработка кроссплатформенного приложения, позволяющего хранить и обрабатывать информацию о студентах вуза, их успеваемости, посещении занятий, научной работе, внеучебной деятельности. Для достижения данной цели были выполнены следующие задачи: анализ задания на выпускную квалификационную работу, проектирование, реализация приложения и составление инструкций.

По результатам проведенного анализа было принято решение использовать Python, MySQL, SQLAlchemy. Разработано программное обеспечение для анализа данных успеваемости студентов, способное визуализировать данный анализ в виде диаграмм, а также сохранять отчеты.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Анализ задания на выпускную квалификационную работу	6
1.1 Обзор аналогичных программных продуктов	6
1.1.1 Precision Campus	6
1.1.2 Tableau	6
1.1.3 SEAtS	8
1.1.4 Creatrix Campus	9
1.1.5 Результаты обзора аналогичных программных продуктов	10
1.2 Интерфейс программы	12
1.3 Диаграмма прецедентов	13
1.4 Выбор инструментов	16
1.4.1 Выбор языка программирования	16
1.4.1.1 Java	17
1.4.1.2 Python	17
1.4.1.3 C++	18
1.4.1.4 Итог	18
1.4.2 Выбор системы управления базами данных	18
1.4.2.1 MySQL	19
1.4.2.2 SQLite	19
1.4.2.3 PostgreSQL	19
1.4.2.4 Итог	20
1.4.3 Выбор ORM	20
1.4.3.1 SQLAlchemy	20
1.4.3.2 Django ORM	21
1.4.3.3 Peewee ORM	21
1.4.3.4 Итог	21
1.5 Итоги анализа задания	22
2 Проектирование и реализация приложения	23
2.1 Архитектура приложения	23

2.2 Модульная структура	24
2.3 Структура базы данных.....	26
2.4 Диаграмма классов	32
2.5 Примеры интерфейса приложения	35
3 Инструкции.....	39
3.1 Инструкция пользователя	39
3.2 Инструкция разработчика	48
ЗАКЛЮЧЕНИЕ	51
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	52

ВВЕДЕНИЕ

Успеваемость по дисциплинам является показателем работы студентов и преподавателей. Руководство высших учебных заведений заинтересовано, чтобы осуществлялся непрерывный контроль за результатами учебного процесса. Для повышения успеваемости помимо контроля необходимо анализировать результаты, сравнивать результаты разных академических групп. Для снижения трудовых затрат необходимо автоматизировать процесс анализа успеваемости студентов.

Целью выпускной квалификационной работы является разработка кроссплатформенного приложения, позволяющего хранить и обрабатывать информацию о студентах вуза, их успеваемости, посещении занятий, научной работе, внеучебной деятельности.

Разработанное приложение должно быть кроссплатформенным для настольных систем (ОС Windows, Linux), иметь графический пользовательский интерфейс, открытый исходный код и обладать следующими возможностями:

- хранение и обработка информации о студентах вуза, их успеваемости, посещении занятий, научной работе, внеучебной деятельности;
- ввод/вывод данных через графический интерфейс, а также в машиночитаемом файловом формате (например, CSV, XML и т. п.);
- обмен информацией с LMS Moodle;
- наглядное представление информации в различных разрезах: по студентам, по времени, по дисциплине, по группе, по курсу, по семестру, по форме контроля (зачёт, экзамен, курсовая работа/проект), по сессии (летняя/зимняя), по виду элемента учебного плана (дисциплина, практика), по виду занятия (лекция, практическое, лабораторное, экзамен и т. д.);
- базовые возможности анализа и визуализации данных: вычисление статистических показателей, построение диаграмм;
- создание отчётов.

Для достижения цели в работе решаются следующие задачи:

- анализ задания на выпускную квалификационную работу;
- проектирование;
- реализация приложения;
- составление инструкций.

1 Анализ задания на выпускную квалификационную работу

1.1 Обзор аналогичных программных продуктов

Использование новых технологий для обучения и анализа результатов является начальным этапом для дальнейшего развития цифровой педагогики. IT-компании занимаются разработкой собственных платформ, позволяющих создавать учебные курсы преподавателям, вести учет и аналитику успеваемости студентов. Рассмотрим подробнее некоторые из них.

1.1.1 Precision Campus

Precision Campus – это цифровая платформа для высших учебных заведений, позволяющее создавать аналитику и формировать отчетность на основе данных [1]. Данная платформа обладает следующим набором функций:

- позволяет отслеживать успеваемость групп, сформированных по критериям, заданным пользователем;
- предоставляет таблицы и диаграммы, которые дают представление о процентном соотношении, поступивших и закончивших обучение;
- отображает успеваемости студента по каждой дисциплине;
- предоставляет ежегодную информацию о количестве новых студентов на каждом направлении;
- помогает оптимизировать курсы.

Данная система имеет только web-интерфейс, и не работает в offline режиме, отсутствует поддержка русского языка, отсутствует импорт данных из системы LMS Moodle, доступ к платформе предоставляется на платной основе. Годовая подписка от 249\$.

1.1.2 Tableau

Tableau – линейка программных продуктов, реализующая предсказательную аналитику [2]. В линейке программных продуктов имеется

программное обеспечение для различных операционных систем, в том числе и для мобильных, а также пользоваться сервисом можно и с помощью web-интерфейса. Данное программное обеспечение позволяет понять, как расписание и планы обучения влияют на успеваемость, как убрать студентов из зоны риска отчисления, как увеличить успеваемость студентов, а также предсказать на основе имеющихся данных дальнейшую успеваемость студента. Tableau имеет большую библиотеку учебных пособий и внутреннее сообщество, которое помогает быстро внедрить данное ПО в рабочий процесс и решить возникшие проблемы. Tableau ставит акцент на визуализации полученных данных аналитики [3].

Tableau имеет личные кабинеты для преподавателя и для студента, экранные формы приведены на рисунках 1 и 2.

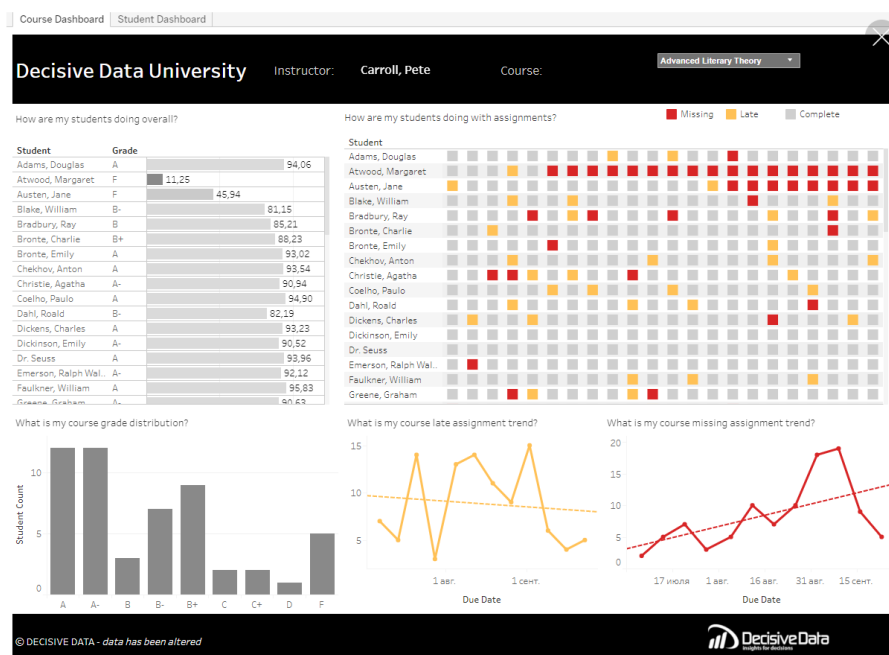


Рисунок 1 – Личный кабинет преподавателя в системе Tableau

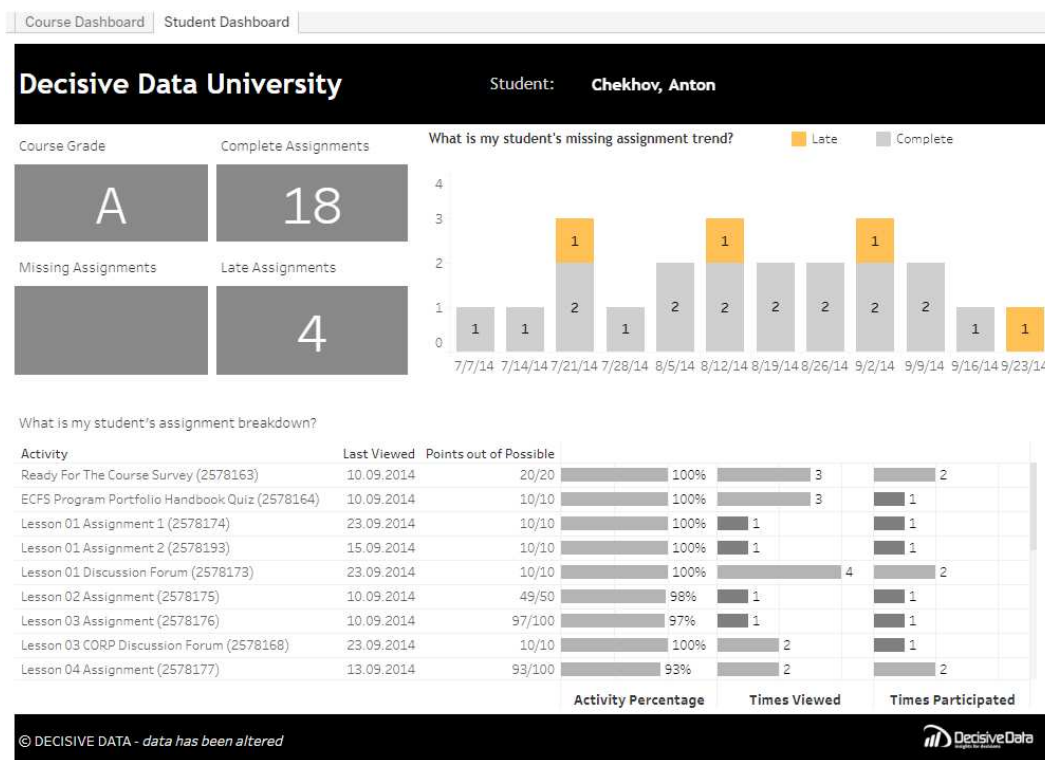


Рисунок 2 – Личный кабинет студента в системе Tableau

К недостаткам данного программного обеспечения стоит отнести отсутствие импорта данных из системы LMS Moodle, отсутствие поддержки русского языка, отсутствие системы оповещения, если успеваемость студента снизилась. Данное ПО распространяется под коммерческой лицензией и действует месячная подписка от 70\$.

1.1.3 SEAtS

SEAtS – система оценивания успеваемости учащихся использует алгоритмы машинного обучения и методы статистического моделирования для быстрого и точного прогнозирования и выявления учащихся, которым грозит риск отчисления. SEAtS отслеживает успеваемость в режиме реального времени и выявляет студентов с плохой посещаемостью и успеваемостью и делает упор на исправление данной ситуации [4]. SEAtS предлагает широкий спектр визуализаторов данных для удобной обработки данных. Однако панель инструментов может быстро переполниться данными, которые пользователям

не нужны. По этой причине каждый пользователь может выбрать, какие показатели будут отображаться на его панели мониторинга для большей эффективности.

Система позволяет преподавателям создавать и настраивать свои собственные рабочие процессы, автоматизировать коммуникации и назначать консультантов на основе определенных параметров. Для каждого студента доступен календарь с расписанием занятий, который можно сортировать по дням, неделям или месяцам. Также преподаватель может добавлять занятия для каждого студента индивидуально.

Для работы с системой необходим постоянный доступ в интернет, так как работа осуществляется через web-интерфейс или через мобильные приложения. Система поддерживает только английский язык. Также данная система является платным решением. Годовая подписка от 399\$.

1.1.4 Creatrix Campus

Creatrix Campus – система управления посещаемостью студентов [5]. В системе могут быть созданы все виды отчетов, таких как результаты работы студентов, ежедневные отчеты, отчеты об отсутствии и опоздании студента на пары, как индивидуально, так и для всего института.

Отчеты можно отслеживать в режиме реального времени, чтобы отправить напоминания и уведомления студентам, не отвечающим критериям, а также их преподавателям. Также студент может отправить информацию о болезни и других причинах отсутствия. Система позволяет вести учет и успеваемости студентов через личный кабинет преподавателя и формировать аналитику в режиме реального времени.

Система предлагает 3 типа личных кабинетов: для студента, для преподавателя и для сотрудника университета. Доступ в систему осуществляется посредством web-интерфейса, а также мобильных приложений для IOS и Android. Мобильные приложения часть информации сохраняют в кэш данных, поэтому приложение может работать в offline

режиме, но для синхронизации данных необходимо подключение к интернету. Результаты анализа процесса обучения студентов можно экспортировать в CSV. Особое внимание в системе уделено безопасности хранения данных. Creatrix Campus поддерживает обмен информации с LMS Moodle.

Система не поддерживает импорт данных, отсутствует поддержка русского языка. Система распространяется по коммерческой лицензии для университетов США и Индии. Стоимость системы определяется индивидуально исходя из количества пользователей и поддержки необходимых модулей.

1.1.5 Результаты обзора аналогичных программных продуктов

Из обзора существующих решений, представленных в пунктах 1.1.1-1.1.4, можно сделать вывод, что все рассмотренные аналоги предназначены для учета успеваемости студентов. Precision Campus, SEAtS и Creatrix Campus имеют возможность вывода информации в различных разрезах: по студентам, по времени, по дисциплине, по группе, по курсу, по семестру, по форме контроля (зачёт, экзамен, курсовая работа/проект), по сессии (летняя/зимняя), по виду элемента учебного плана (дисциплина, практика), по виду занятия (лекция, практическое, лабораторное, экзамен и т. д.). Все продукты имеют возможность вычислять статистические показатели и визуализировать результаты анализа в виде диаграмм, поэтому необходимо реализовать данные функции в разрабатываемом ПО.

Работа со всеми системами возможна через web-интерфейс, кроме того, Tableau, SEAtS и Creatrix Campus имеют и версии приложений для мобильных ОС. Creatrix Campus поддерживает обмен информации с LMS Moodle. Системы являются платными программными продуктами. Количество готовых программных продуктов на сегодняшний день ограничено, большинство вузов имеют внутренние сервисы анализа успеваемости студентов, доступ к таким системам носит корпоративный характер. Результаты сравнения продуктов представлены в таблице 1.

Таблица 1 – Сравнение аналогов

Свойство	Precision Campus	Tableau	SEAtS	Creatrix Campus
Поддержка русского языка	Нет	Нет	Нет	Нет
Возможность экспорта результатов аналитики	Да	Да	Да	Да
Анализ данных в режиме реального времени	Нет	Нет	Да	Да
Поддержка обмена информации с LMS Moodle	Нет	Нет	Нет	Да
Лицензия	Коммерческая	Коммерческая	Коммерческая	Коммерческая
Наличие кроссплатформенных приложений для настольных ОС	Нет	Нет	Нет	Нет
Открытый исходный код	Нет	Нет	Нет	Нет
Ввод и вывод данных через машиночитаемый файловый формат (CSV, XML и т. п.)	Нет	Нет	Нет	Да
Представление информации в различных разрезах (по студентам, по дисциплине, и т.д.)	Да	Да	Да	Да

Окончание таблицы 1

Свойство	Precision Campus	Tableau	SEAtS	Creatrix Campus
Настраиваемые оповещения о событиях по выбранному пользователем условию	Да	Да	Нет	Да
Вычисление статистических показателей и построение диаграмм	Да	Да	Да	Да

1.2 Интерфейс программы

На основе задания на ВКР был сделан макет интерфейса, который должен быть в конечном приложении. На рисунке 3 изображен этот макет.

	ФИО	Группа	Дисциплина	Список не сданных работ
1	Горшков Артур Викторович	432432234	Вычислительная математика	Практическая работа №3
2	Капустин Августин Пантелеймонович	432432234	Вычислительная математика	Лабораторная работа №2, Практическая работа №3
3	Горшков Артур Викторович	432432234	Теория вероятностей	Практическая работа №1, Лабораторная работа №2, Контрольная работа №3
4	Иванов Иван Иванович	432432234	Теория вероятностей	Практическая работа №1, Лабораторная работа №2, Контрольная работа №3
5	Капустин Августин Пантелеймонович	432432234	Теория вероятностей	Практическая работа №1, Лабораторная работа №2, Контрольная работа №3
6	Горшков Артур Викторович	432432234	Основы проектирования приложений	Лабораторная работа №3
7	Капустин Августин Пантелеймонович	432432234	Основы проектирования приложений	Лабораторная работа №2, Лабораторная работа №3
8	Горшков Артур Викторович	432432234	Автоматы	Лабораторная работа №1, Лабораторная работа №2, Лабораторная работа №3
9	Иванов Иван Иванович	432432234	Автоматы	Лабораторная работа №1, Лабораторная работа №2, Лабораторная работа №3
10	Капустин Августин Пантелеймонович	432432234	Автоматы	Лабораторная работа №1, Лабораторная работа №2, Лабораторная работа №3
11	Гаврилов Игорь Глебович	342234	Теория вероятностей	Лабораторная работа №1, Лабораторная работа №2, Лабораторная работа №3
12	Горбунов Елисей Иринеович	342234	Теория вероятностей	Лабораторная работа №1, Лабораторная работа №2, Лабораторная работа №3
13	Давыдов Ярослав Николаевич	342234	Теория вероятностей	Лабораторная работа №1, Лабораторная работа №2, Лабораторная работа №3
14	Тихонов Валентин Мартынович	342234	Теория вероятностей	Лабораторная работа №1, Лабораторная работа №2, Лабораторная работа №3
15	Юдин Петр Валерьевич	342234	Теория вероятностей	Лабораторная работа №1, Лабораторная работа №2, Лабораторная работа №3
16	Гаврилов Игорь Глебович	342234	Основы проектирования приложений	Лабораторная работа №1, Лабораторная работа №2, Лабораторная работа №3
17	Горбунов Елисей Иринеович	342234	Основы проектирования приложений	Лабораторная работа №1, Лабораторная работа №2, Лабораторная работа №3
18	Давыдов Ярослав Николаевич	342234	Основы проектирования приложений	Лабораторная работа №1, Лабораторная работа №2, Лабораторная работа №3
19	Тихонов Валентин Мартынович	342234	Основы проектирования приложений	Лабораторная работа №1, Лабораторная работа №2, Лабораторная работа №3
20	Юдин Петр Валерьевич	342234	Основы проектирования приложений	Лабораторная работа №1, Лабораторная работа №2, Лабораторная работа №3

Рисунок 3 – Макет интерфейса

1.3 Диаграмма прецедентов

На рисунке 4 изображена диаграмма прецедентов.

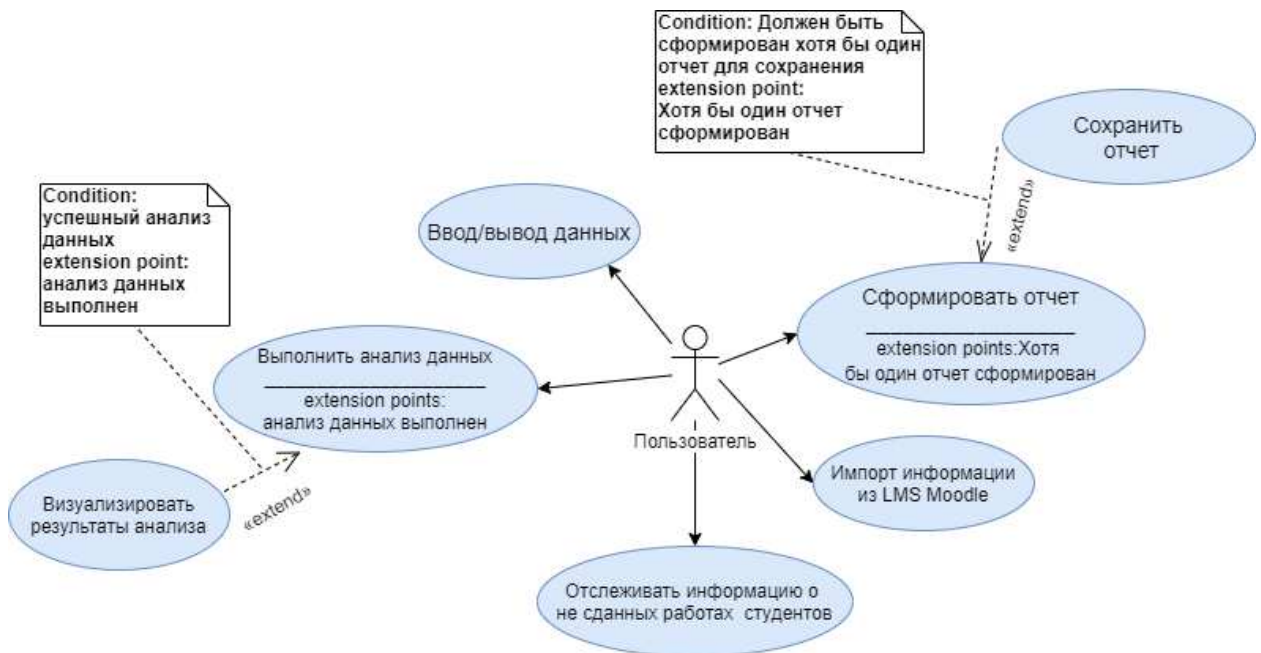


Рисунок 4 – Диаграмма прецедентов

Текстовое описание прецедентов приведено ниже.

Название прецедента: Ввод/вывод данных.

Предусловие: пользователь находится на главном окне, нажимает на кнопку «Студенты», в открывшемся окне пользователь нажимает на кнопку «Группы», в открывшемся окне пользователь нажимает на кнопку «Добавить».

Основной сценарий:

А. Данные проверяются на корректность и на наличие данных в базе данных.

В. Данные добавляются в базу данных.

Постусловие: пользователь попадает в окно «Список групп» с отображением обновленного списка групп.

Условие ввода в действие альтернативных сценариев:

Условие 1. Учебная группа с таким названием уже существует.

А. Приложение выдает сообщение о существовании студента с таким названием в базе данных.

Условие 2. Проблема со связью с базой данных.

А. Приложение выдает сообщение о проблеме связи с базой данных.

Условие 3. Данные введены некорректно.

А. Приложение выдает сообщение о некорректности введенных данных.

Название прецедента: Импорт информации из LMS Moodle.

Предусловие: пользователь находится на главном окне, нажимает на кнопку «Параметры», в открывшемся окне пользователь нажимает на кнопку «Импорт данных из LMS Moodle», в открывшемся окне пользователь вводит необходимые данные для доступа к LMS Moodle.

Основной сценарий:

А. Приложение обращается к LMS Moodle и запрашивает необходимые данные.

В. Данные проходят проверку на наличие данных в системе.

С. Данные добавляются в базу данных.

Постусловие: пользователь попадает в главное окно приложения.

Условие ввода в действие альтернативных сценариев:

Условие 1. Данные из LMS Moodle уже существуют в базе данных.

А. Приложение игнорирует эти данные.

Условие 2. Проблема со связью с базой данных.

А. Приложение выдает сообщение о проблеме связи с базой данных.

Условие 3. Проблема с доступом к LMS Moodle.

А. Приложение выдает сообщение о проблеме доступа к LMS Moodle.

Название прецедента: Отслеживать информацию о не сданных работах студентов.

Предусловие: пользователь запускает приложение.

Основной сценарий:

А. Приложение запрашивает у ОС текущую дату.

В. Приложение формирует список студентов и работ, которые просрочили срок сдачи.

С. Данные отображаются в главном окне.

Постусловие: пользователь попадает в главное окно приложения.

Название прецедента: Выполнить анализ данных.

Предусловие: пользователь, находясь в главном окне нажимает на кнопку «Аналитика» и выбирает объект анализа.

Основной сценарий:

А. Приложение анализирует критерии, заданные пользователем.

В. Приложение запрашивает необходимые данные из базы данных, производит вычисления. Полученный результат формируется.

С. Полученный результат представляется в виде таблицы.

Постусловие: пользователь попадает в окно с проанализированными данными.

Название прецедента: Визуализировать результаты анализа.

Предусловие: пользователь, находясь в окне просмотра результатов анализа, нажимает на кнопку «Отобразить диаграмму».

Основной сценарий:

А. Приложение обрабатывает результаты аналитики и строит необходимую диаграмму.

В. Приложение отображает полученную диаграмму.

Постусловие: пользователь попадает в окно с визуализированными данными.

Название прецедента: Сформировать отчет.

Предусловие: пользователь, находясь в главном окне приложения «Анализ данных», нажимает на кнопку «Сформировать отчет».

Основной сценарий:

А. Приложение получает необходимые данные для отчета.

В. Приложение формирует отчет.

С. Приложение открывает окно, где оповещает пользователя о формировании отчета.

Постусловие: система формирует отчет и делает его доступным для сохранения пользователем.

Название прецедента: Сохранить отчет.

Предусловие: пользователь, находясь в главном окне приложения, нажимает на кнопку «Отчеты», выбирает из списка нужный отчет, и нажимает на кнопку «Сохранить отчет».

Основной сценарий:

А. Приложение получает необходимые данные для отчета.

В. Приложение сохраняет отчет в PDF формате.

С. Приложение открывает окно сохранения отчета, где пользователь указывает директорию и имя файла.

Д. Файл сохраняется.

Постусловие: пользователь получает сохраненный файл.

1.4 Выбор инструментов

1.4.1 Выбор языка программирования

Исходя из задания на выпускную квалификационную работу, приложение должно работать на платформах Window и Linux, кроме того, приложение должно поддерживать графический интерфейс пользователя.

Для написания кроссплатформенного приложения выбор стоял между следующими языками и фреймворками:

- Java с библиотекой JavaFX;
- Python с библиотекой PyQt5;
- C++ с фреймворком Qt.

1.4.1.1 Java

Java — язык программирования общего назначения. Относится к объектно-ориентированным языкам программирования, к языкам с сильной типизацией. написанное на Java приложение можно запустить на любой платформе, если на ней установлена среда исполнения Java (JRE, Java Runtime Environment). Эта задача решается благодаря компиляции написанного на Java кода в байт-код. Этот формат исполняет JVM или виртуальная машина Java. JVM — часть среды исполнения Java (JRE). Виртуальная машина не зависит от платформы [6]. Графический интерфейс можно реализовывать с помощью таких библиотек, как «Standard Widget Toolkit», который использует компоненты операционной системы, тем самым повышая скорость работы, а также обладает обширной документацией. Или JavaFX, который для отрисовки использует графический конвейер, который так же позволяет увеличить скорость работы, обладает множеством различных компонентов и поддержкой стилей [7].

1.4.1.2 Python

Python — высокоуровневый язык программирования, ориентированный на повышение производительности разработчика и читаемости кода [8]. Реализовывать графический интерфейс можно с помощью tkinter (поставляется в стандартной библиотеке Python) или с помощью библиотек, поддерживающих компоненты Qt – PyQt5 или PySide2. Библиотеки PyQt5 и PySide2 не сильно отличаются в поддержке функционала, они отличаются в именовании методов, разработка библиотек ведется параллельно. Стоит отметить, что у PyQt5 больше профессиональное сообщество, а PySide2 официально поддерживают разработчики Qt.

1.4.1.3 C++

C++ — компилируемый, статически типизированный язык программирования общего назначения. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также игр [9]. Кроссплатформенность на C++ реализуется с помощью фреймворка Qt, который включает в себя все основные классы, которые могут потребоваться при разработке прикладного программного обеспечения, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных и XML [10].

1.4.1.4 Итог

Программа на Java, которая использует JVM, является медленной по сравнению с программой на C++, однако C++ является сложным языком, где высока вероятность ошибки и медленная скорость разработки. В этом плане Python является оптимальным, так как использует библиотеки написанные на C++ (таким образом достигается большая скорость вычислений) и прост в написании программ (является одним из принципов языка), синтаксис команд напоминает обычные выражения, которыми пользуются люди повседневно, поэтому ключевые слова понятны любому новичку, также нет строгой типизации, нет конструкций типа «{ }» или «begin-end» - вместо них просто отступы в теле оператора. В итоге выбор был сделан в пользу Python с библиотекой PyQt5.

1.4.2 Выбор системы управления базами данных

Так как данное приложение должно работать с большим объемом данных, выбор стоял между следующими СУБД:

- MySQL;

- SQLite;
- PostgreSQL.

1.4.2.1 MySQL

MySQL — это одна из самых популярных серверных СУБД. В ней достаточно просто разобраться и в интернете можно найти очень много различной информации. В MySQL полностью не реализованы все SQL-стандарты, но зато она предлагает широкую функциональность и обладает следующими преимуществами:

- MySQL прост в установке. Существует большое количество инструментов, включая визуальные, облегчающих работу с базой данных;
- MySQL поддерживает большую часть функциональности SQL;
- MySQL поддерживает работу с большими объёмами данных, и хорошо подходит для масштабируемых приложений [11].

1.4.2.2 SQLite

SQLite — это библиотека, встраиваемая в приложение, которое её использует. SQLite предоставляет отличный набор инструментов для более простой обработки любых видов данных [11].

1.4.2.3 PostgreSQL

PostgreSQL — это СУБД, ориентирующаяся в первую очередь на полное соответствие стандартам и расширяемость. PostgreSQL, или Postgres, пытается полностью соответствовать SQL-стандартам ANSI/ISO.

PostgreSQL отличается от других СУБД тем, что обладает объектно-ориентированным функционалом, в том числе полной поддержкой концепта ACID (Atomicity, Consistency, Isolation, Durability) [11].

1.4.2.4 Итог

Несмотря на то, что PostgreSQL и SQLite обладают рядом преимуществ, выбор пал на MySQL. MySQL легко устанавливается, имеет много сторонних инструментов, позволяющих облегчить работу. Является популярной СУБД, что означает большое сообщество разработчиков, а также является бесплатной в использовании.

1.4.3 Выбор ORM

ORM (Object-Relational Mapping) — технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, то есть ORM — прослойка между базой данных и кодом, который пишет программист, которая позволяет созданные в программе объекты записывать в базу данных или получать из нее [12]. Библиотеки ORM существуют для самых разных языков программирования. В общих чертах, технология ORM позволяет проектировать работу с данными в терминах классов, а не таблиц данных. Она позволяет преобразовывать классы в данные, пригодные для хранения в базе данных, причем схему преобразования определяет сам разработчик [13].

1.4.3.1 SQLAlchemy

SQLAlchemy – это популярный инструментарий SQL и Object Relational Mapper. Он написан на Python и предоставляет полную мощность и гибкость SQL для разработчика приложений. Это кроссплатформенное программное обеспечение с открытым исходным кодом, выпущенное по лицензии MIT.

SQLAlchemy славится своим объектно-реляционным отображением (ORM), с помощью которого классы могут быть сопоставлены с базой данных,

что позволяет с самого начала четко развязать объектную модель и схему базы данных.

Поскольку размер и производительность баз данных SQL начинают иметь значение, они ведут себя не так, как коллекции объектов. С другой стороны, когда абстракция в коллекциях объектов начинает иметь значение, они ведут себя не так, как таблицы и строки. SQLAlchemy стремится учитывать оба эти принципа [14].

1.4.3.2 Django ORM

Django ORM – часть фреймворка Django, которая позволяет очень просто делать определённое подмножество всех сайтов. Django ORM обладает теми же свойствами: более-менее типичные запросы на ней можно сделать очень легко. А вот со сложными запросами Django ORM не справится.

Django ORM нельзя использовать отдельно от Django, поэтому и сообщество ограничено пользователями Django [15].

1.4.3.3 Peewee ORM

Peewee ORM – это небольшое ORM, которое в данный момент поддерживает PostgreSQL, MySQL и SQLite. Хорошая сторона Peewee – это то, что он занимает мало места, его легко освоить, и можно приступить к работе с приложениями за несколько минут [16].

1.4.3.4 Итог

Несмотря на то, что о всех рассмотренных ORM хорошо отзываются разработчики, выбор пал на SQLAlchemy. Несмотря на большое сообщество разработчиков и удобство разработки на Django ORM, данная ORM не реализуется вне Django.

Peewee ORM, несмотря на то, что является достаточно легким в освоении, уступает SQLAlchemy, который является как и более производительным так и более функциональным.

1.5 Итоги анализа задания

На основе аналогов, сделаны выводы: не было найдено ни одного приложения, полностью удовлетворяющего требованиям задания. Задание подразумевает наличие открытого исходного кода. Так как все рассмотренные аналоги распространяются на платной основе, открытого исходного кода они предоставить не могут. Были сформулированы функциональные требования к разрабатываемому приложению в виде диаграммы прецедентов, и их описания. В качестве инструмента разработки был выбран Python 3.8.2 с библиотекой PyQt5, так как задание предполагало кроссплатформенность. А также выбор пал на MySQL и SQLAlchemy ORM.

2 Проектирование и реализация приложения

2.1 Архитектура приложения

В основе архитектуры приложения лежит паттерн проектирования Model-View (рисунок 5). Данный паттерн немного отличается от другого паттерна – Model-View-Controller (рисунок 6) отсутствием контроллера. Паттерн Model-View также рекомендуется к использованию разработчиками Qt [17].

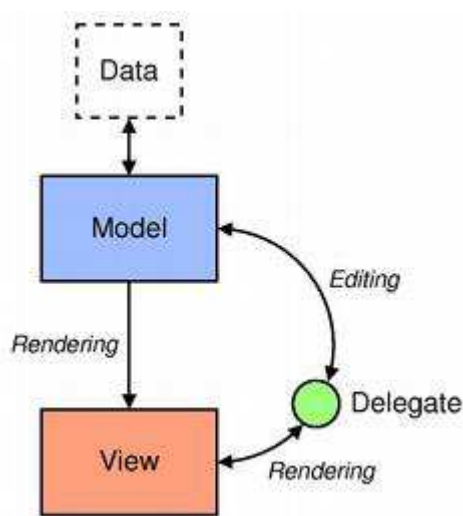


Рисунок 5 – Паттерн Model/View [18]

Модель связывается с источником данных, предоставляя интерфейс. Представление получает модельные индексы из Model, это ссылки на элементы данных. Предоставляя модельные индексы Model, View может получать элементы данных из источника данных.

В стандартных представлениях Delegate отображает элементы данных. Когда элемент редактируется, Delegate связывается с Model напрямую, используя модельные индексы [18]. Так же Delegate дает возможность для настройки редактирования (Editing) и представления (Rendering) [18].

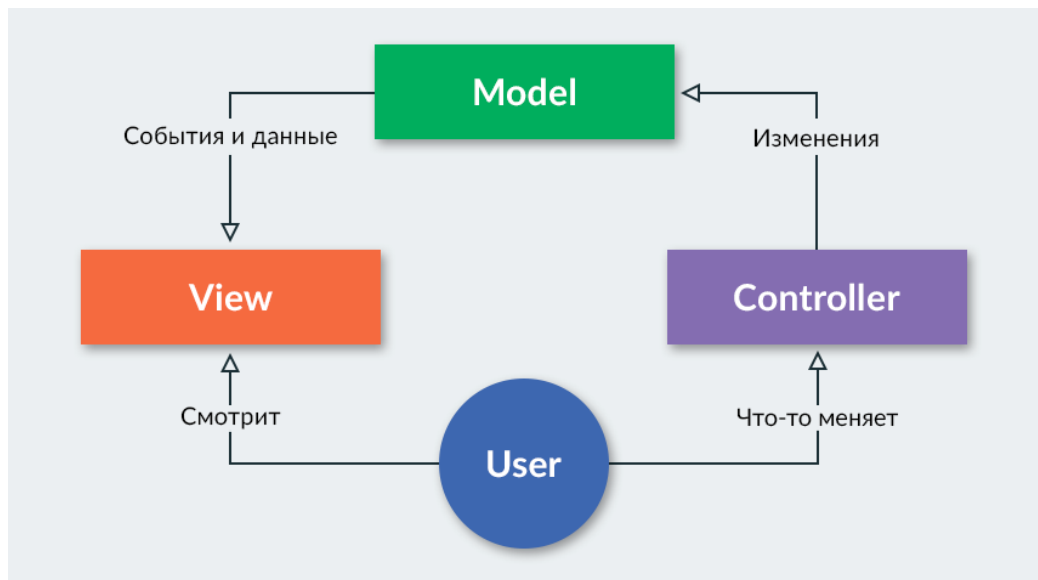


Рисунок 6 – Паттерн Model-View-Controller [19]

Model отвечает за предоставление данных и меняет свое состояние в зависимости от команд. View отображает данные модели пользователю, реагируя на изменения модели.

2.2 Модульная структура

В процессе разработки приложения были выделены следующие функциональные модули (рисунок 7):

- дисциплины;
- студенты;
- оценки;
- аналитика;
- отчеты;
- LMS Moodle;
- интерфейс.

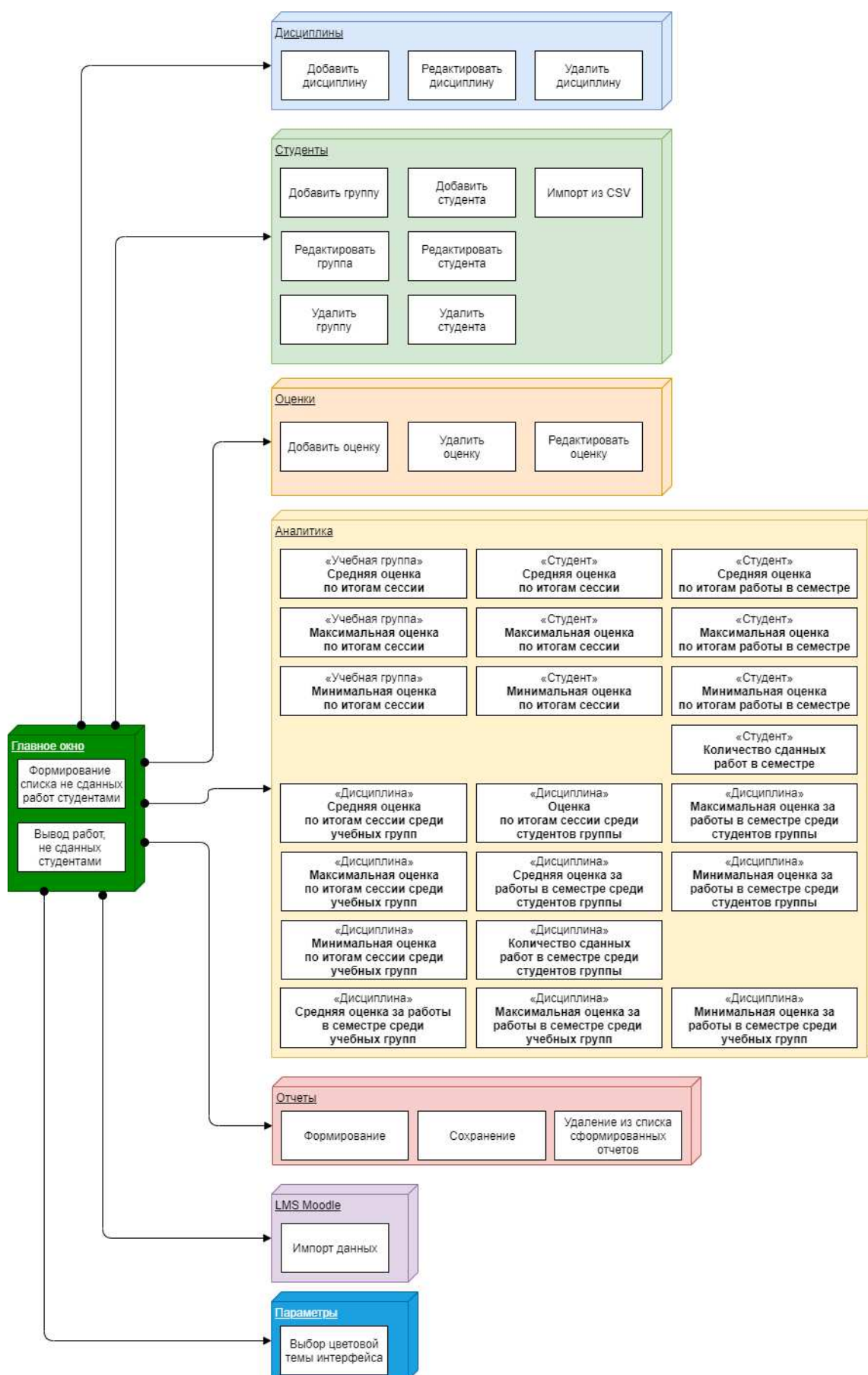


Рисунок 7 – Модульная структура приложения

2.3 Структура базы данных

В приложении используется СУБД – MySQL. Одними из главных плюсов MySQL является высокая скорость обработки информации и функциональность СУБД. Данная система представляет собой программное обеспечение с открытым кодом, и она доступна любому пользователю абсолютно бесплатно [20].

После анализа предметной области были выделены следующие сущности:

- дисциплина;
- специальность;
- группа;
- работа;
- студент;
- оценки;
- контроль;
- аттестация;
- вид сессии;
- вид аттестации.

Диаграмма сущностей и связей представлена на рисунке 8.

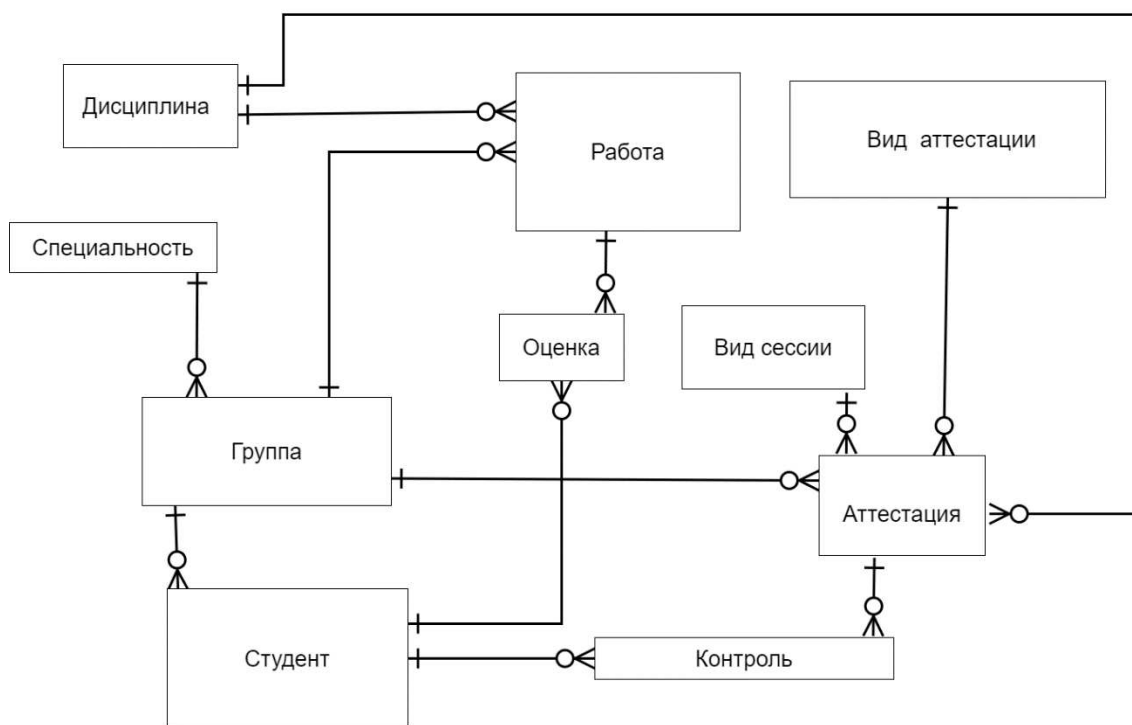


Рисунок 8 – Диаграмма сущностей и связей

Было решено создать сущность «Студент», который будет относиться к одному объекту сущности «Группа», которая в свою очередь относится к одному объекту сущности «Специальность». У каждого «Студента» будет некоторое количество «Оценок» за какую-либо «Работу». Объекты сущности «Работа» будут относиться к какому-то объекту сущности «Дисциплина» и объекту сущности «Группа». Так же у «Студента» есть зачеты и экзамены, которые он может сдать или не сдать. Следовательно, объекты сущности «Студент» относятся к нескольким объектам сущности «Контроль», объекты которой относятся к объекту «Аттестация», которая включает в себя «Вид аттестации», «Вид сессии», «Дисциплину», «Группы».

На основе диаграммы сущностей и связей была создана логическая модель базы данных (рисунок 9).

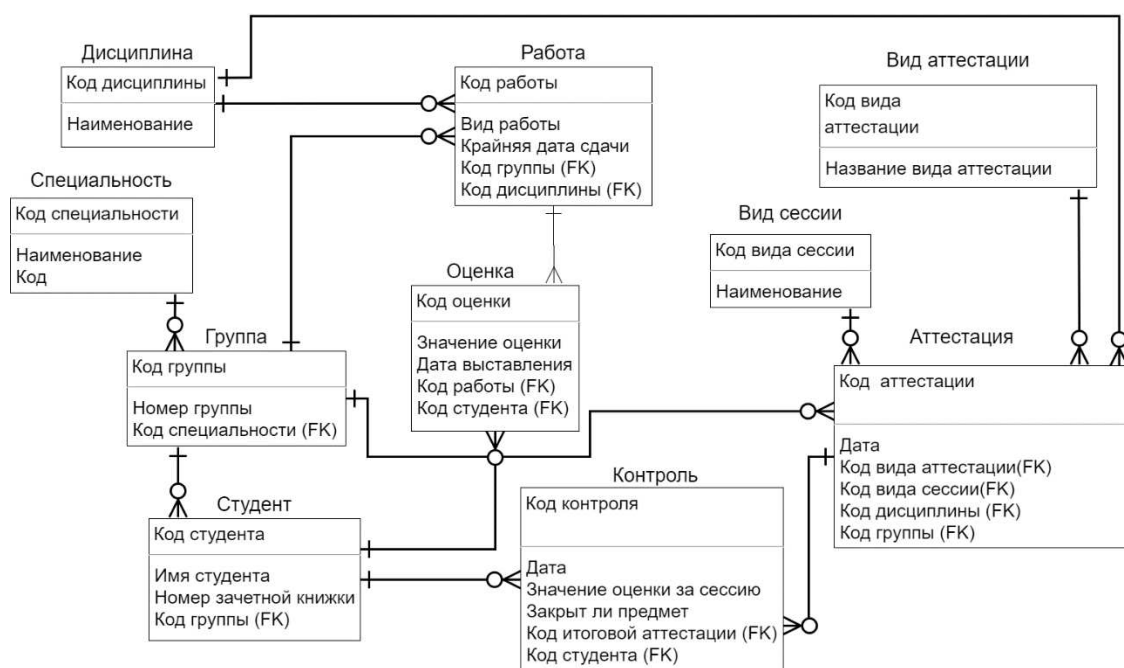


Рисунок 9 – Логическая модель базы данных

Физическая модель разработанной базы данных отображена на рисунке

10.

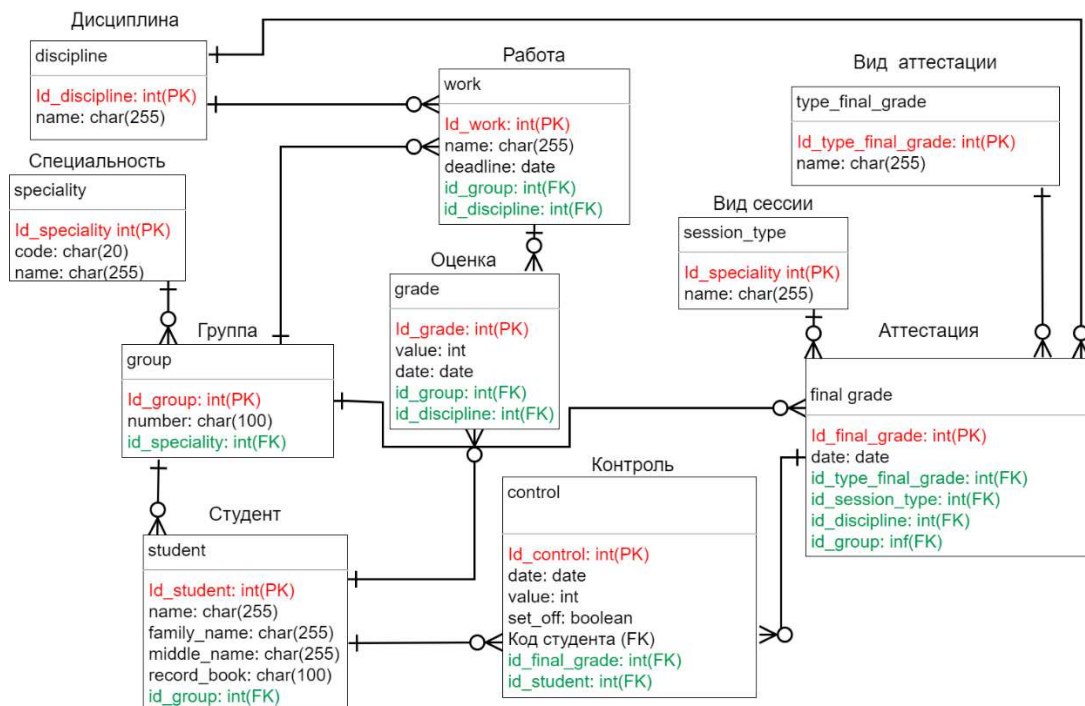


Рисунок 10 – Физическая модель базы данных

Ниже в таблицах 2-11 представлено подробное описание всех сущностей базы данных с названием полей и их назначением

Сущность `discipline` (дисциплина) содержит информацию о дисциплинах.

Таблица 2 – Описание `discipline` (дисциплина)

Поле	Тип	Назначение
<code>id_discipline</code>	<code>int</code>	первичный ключ
<code>name</code>	<code>char (255)</code>	наименование дисциплины

Сущность `specialty` (специальность) содержит информацию о специальностях.

Таблица 3 – Описание `specialty` (специальность)

Поле	Тип	Назначение
<code>id_specialty</code>	<code>int</code>	первичный ключ
<code>code</code>	<code>char (20)</code>	код специальности
<code>name</code>	<code>char (255)</code>	наименование

Сущность `group` (группа) содержит информацию об учебных группах студентов.

Таблица 4 – Описание `group` (группа)

Поле	Тип	Назначение
<code>id_group</code>	<code>int</code>	первичный ключ
<code>number</code>	<code>char (100)</code>	номер группы
<code>id_specialty</code>	<code>int</code>	внешний ключ к таблице <code>Specialty</code>

Сущность `student` (студент) содержит информацию о студенте.

Таблица 5 – Описание student (студент)

Поле	Тип	Назначение
id_student	int	первичный ключ
family_name	char (255)	фамилия
name	char (255)	имя
middle_name	char (255)	отчество
record_book	char (100)	номер зачетной книжки
id_group	int	внешний ключ к таблице Group

Сущность grade (оценка) содержит информацию о полученной оценке студента.

Таблица 6 – Описание grade (оценка)

Поле	Тип	Назначение
id_grade	int	первичный ключ
value	int	значение оценки
date	date	дата выставления
id_work	int	внешний ключ к таблице work
id_student	int	внешний ключ к таблице student

Сущность work (работа) содержит информацию о задании для работы студентов.

Таблица 7 – Описание work (работа)

Поле	Тип	Назначение
id_work	int	первичный ключ
name	char (255)	название работы
deadline	date	дата крайнего срока сдачи
id_group	int	внешний ключ к таблице group
id_discipline	int	внешний ключ к таблице discipline

Сущность session_type (вид сессии) содержит информацию о виде сессии.

Таблица 8 – Описание session_type (вид сессии)

Поле	Тип	Назначение
id_session_type	int	первичный ключ
name	char (255)	наименование сессии (Зимняя, летняя)

Сущность type_final_grade (вид аттестации) содержит информацию о виде аттестации.

Таблица 9 – Описание type_final_grade (вид аттестации)

Поле	Тип	Назначение
id_type_final_grade	int	первичный ключ
name	char (255)	наименование вида аттестации

Сущность control (контроль) содержит информацию о том сдал ли студент предмет.

Таблица 10 – Описание control (контроль)

Поле	Тип	Назначение
id_control	int	первичный ключ
date	date	дата сдачи работы
value	int	оценка, полученная за работу
set_off	boolean	флаг, сдана ли работа
id_final_grade	int	внешний ключ к таблице final_grade
id_student	int	внешний ключ к таблице student

Сущность final_grade (аттестация) содержит информацию об аттестации.

Таблица 11 – Описание final_grade (аттестация)

Поле	Тип	Назначение
id_final_grade	int	первичный ключ
date	date	дата сдачи работы
id_type_final_grade	int	внешний ключ к таблице type_final_grade
id_session_type	int	внешний ключ к таблице session_type
id_discipline	int	внешний ключ к таблице discipline
id_group	int	внешний ключ к таблице group

2.4 Диаграмма классов

Каждое окно приложения соответствует классу, и каждая таблица базы данных соответствует классу. В примере ниже (рисунок 11) рассмотрена диаграмма классов для работы со студентами. Работа с остальными данными реализована подобным образом.

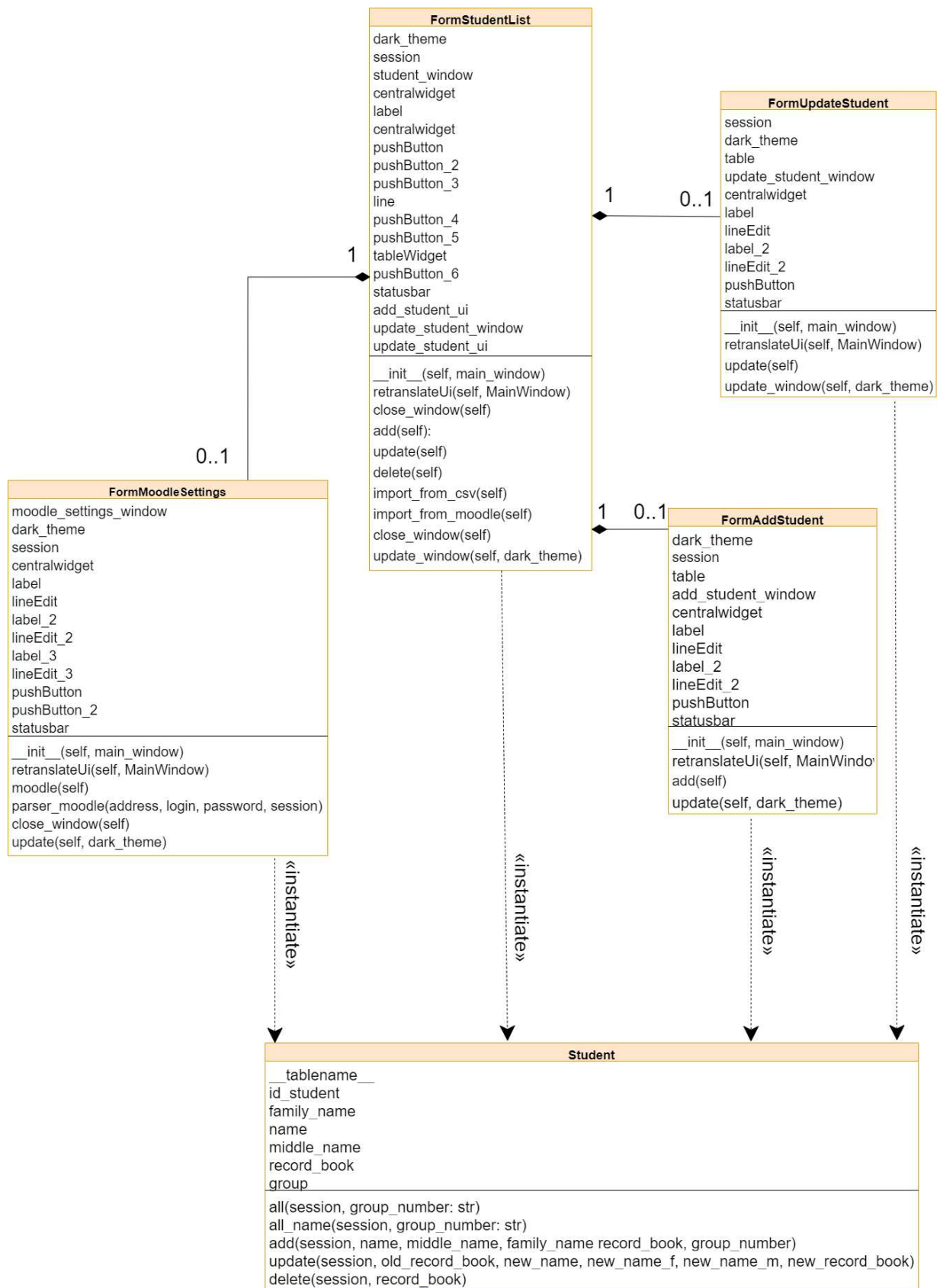


Рисунок 11 – Диаграмма классов

Классы `FormStudentList`, `FormAddStudent`, `FormUpdateStudent`, `FormMoodleSettings` реализуют графическое представление окна, следовательно, атрибуты каждого из этих классов, содержат в себе какой-либо элемент этого окна. Кнопки, текстовые поля, заголовки и так далее.

Класс `FormStudentList` реализует графическое представление окна «Список студентов». Функция `__init__(self, main_window)` отвечает за оформление окна и действия по нажатиям на кнопки. Функция `close_window(self)` отвечает за закрытие окна. Функция `add(self)` отвечает за вызов окна «Добавить студента», а `update(self)` отвечает за вызов окна «Редактирование группы». Функция `retranslateUi(self, MainWindow)` отвечает за название кнопок и надписи в окне. Функция `update(self, dark_theme)` реализует темную тему интерфейса окна. Функция `import_from_csv(self)` и `import_from_moodle(self)` реализует импорт из файла формата CSV и вызывает окно «Импорт из Moodle» соответственно.

Класс `FormAddStudent` реализует добавление студента в базу данных. Функция `__init__(self, main_window)` отвечает за оформление окна и действия по нажатиям на кнопки. Функция `add_student(self)` отвечает за добавление студента в базу данных с помощью класса «Student». Функция `retranslateUi(self, MainWindow)` отвечает за название кнопок и надписи в окне. Функция `update(self, dark_theme)` реализует темную тему интерфейса окна.

Класс `FormUpdateStudent` реализует редактирование студента в базе данных. Функция `__init__(self, main_window)` отвечает за оформление окна и действия по нажатиям на кнопки. Функция `update(self)` отвечает за изменение информации по студенту. Функция `retranslateUi(self, MainWindow)` отвечает за название кнопок и надписи в окне. Функция `update(self, dark_theme)` реализует темную тему интерфейса окна.

`FormMoodleSettings` реализует импорт из LMS Moodle. Функция `__init__(self, main_window)` отвечает за оформление окна и действия по нажатиям на кнопки. Функция `moodle(self)` реализует импорт из LMS Moodle в базу

данных. Функция `close_window(self)` отвечает за закрытие окна. Функция `update(self, dark_theme)` реализует темную тему интерфейса окна.

Класс `Student` реализует методы для работы с сущностью «студент» в базе данных, он содержит атрибуты:

- `id_student` (id студента);
- `family_name` (фамилия);
- `name` (имя);
- `middle_name` (отчество);
- `record_book` (номер зачетной книжки);
- `group` (группа).

И функции:

- `all(session, group_number: str)` (получение полного списка студентов);
- `all_name(session, group_number: str)` (получение списка имен студентов);
- `add(session, name, middle_name, family_name, record_book, group_number)` (запись информации по студенту в базу данных);
- `update(session, old_record_book, new_name, new_name_f, new_name_m, new_record_book)` (редактирование информации по студенту в базе данных);
- `delete(session, record_book)` (удаление студента из базы данных).

2.5 Примеры интерфейса приложения

Исходя из макета, был разработан следующий интерфейс приложения (представлен на рисунках 12-17).

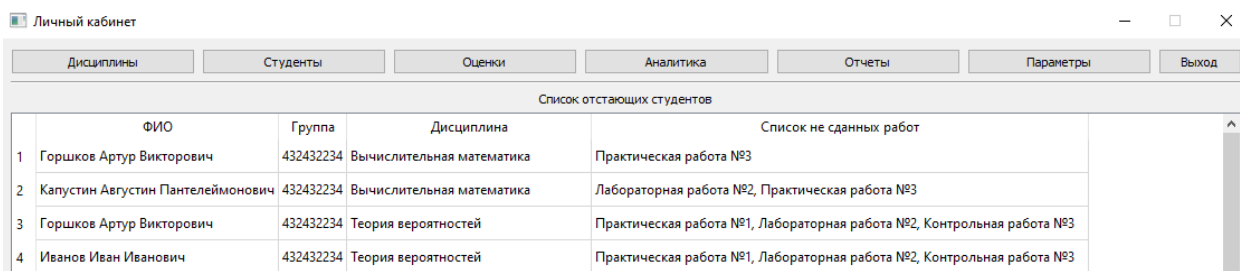


Рисунок 12 – Главное окно приложения

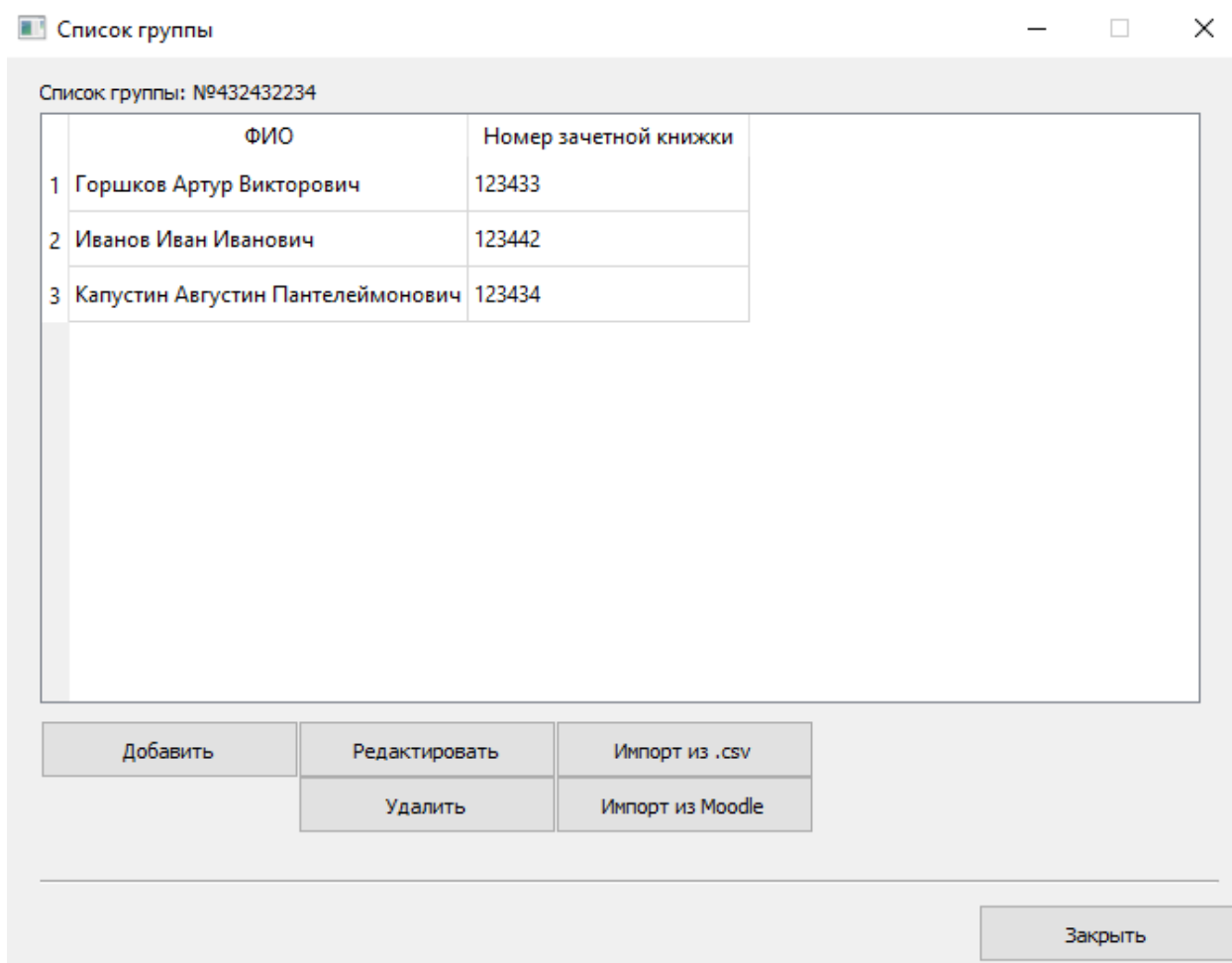
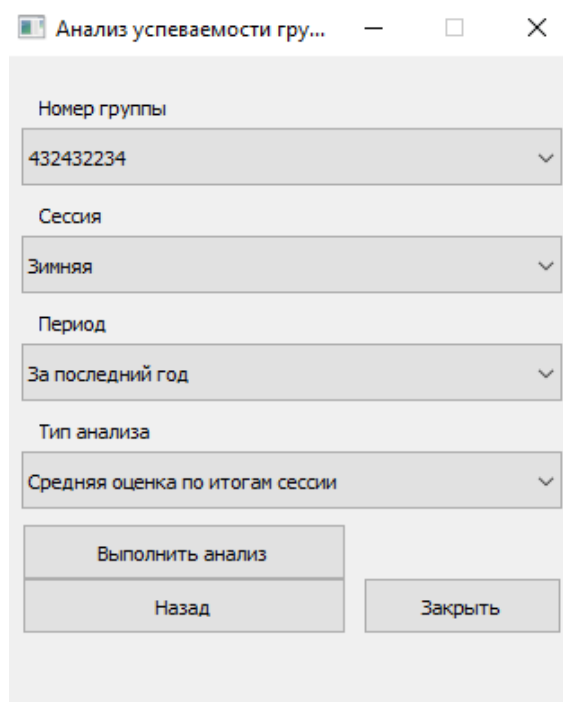


Рисунок 13 – Список студентов



Анализ успеваемости гру...

Номер группы
432432234

Сессия
Зимняя

Период
За последний год

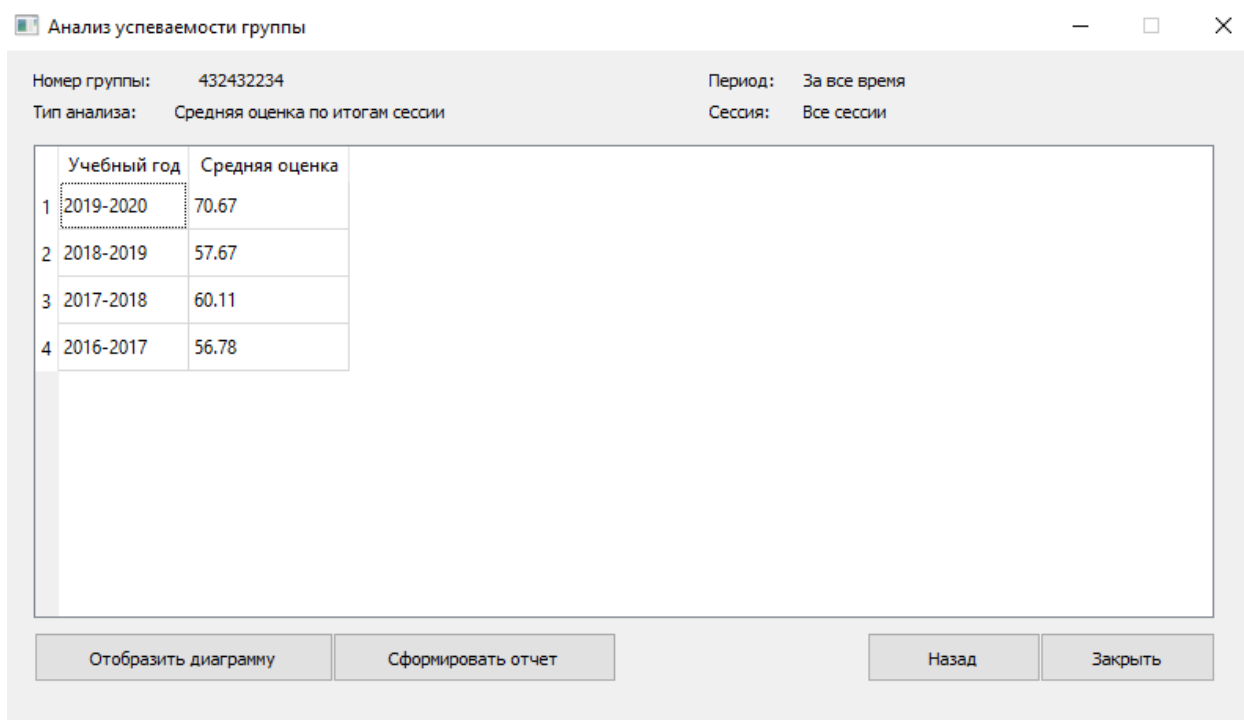
Тип анализа
Средняя оценка по итогам сессии

Выполнить анализ

Назад

Заккрыть

Рисунок 14 – Окно настройки анализа



Анализ успеваемости группы

Номер группы: 432432234 Период: За все время
Тип анализа: Средняя оценка по итогам сессии Сессия: Все сессии

	Учебный год	Средняя оценка
1	2019-2020	70.67
2	2018-2019	57.67
3	2017-2018	60.11
4	2016-2017	56.78

Отобразить диаграмму Сформировать отчет Назад Заккрыть

Рисунок 15 – Окно анализа успеваемости

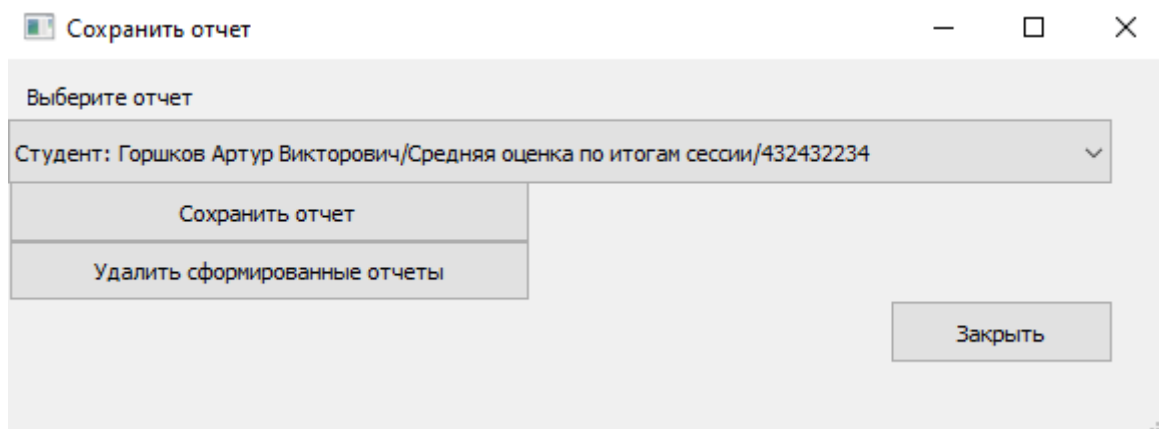


Рисунок 16 – Окно формирования отчета

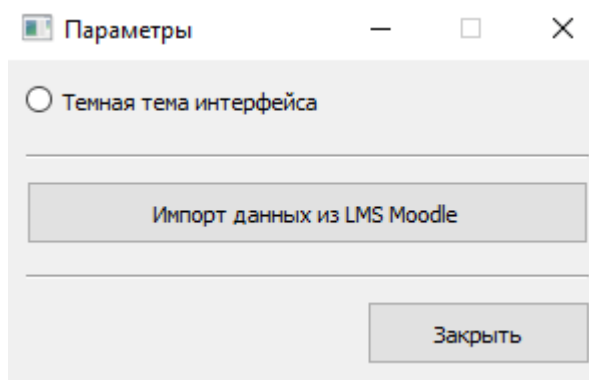


Рисунок 17 – Окно параметров

3 Инструкции

3.1 Инструкция пользователя

При запуске программы мы попадаем на главное окно, на котором отображается список отстающих студентов. На этом окне, пользователь сможет посмотреть, какие работы не сдал студент, группу, дисциплину. Далее, для работы с приложением, необходимо добавить студентов, группы. Это можно сделать вручную в приложении, можно через файл формата CSV или сделать импорт из Moodle. Для добавления студентов или группы вручную необходимо нажать на кнопку «Студенты», далее выбрать кнопку «Просмотреть список группы» или «Группы», как показано на рисунке 18.

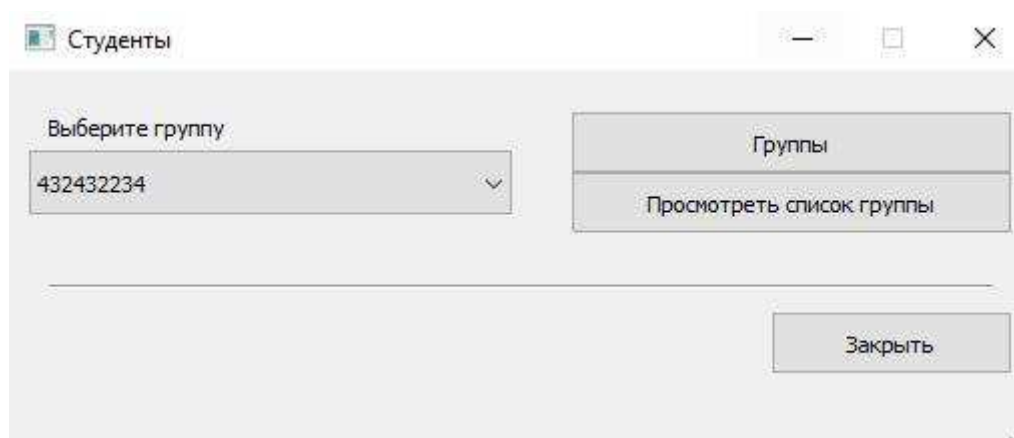


Рисунок 18 – Окно «Студенты»

Далее рассмотрим на примере добавления студента, переходим по кнопке «Посмотреть список группы» в следующее окно, которое показано на рисунке 19.

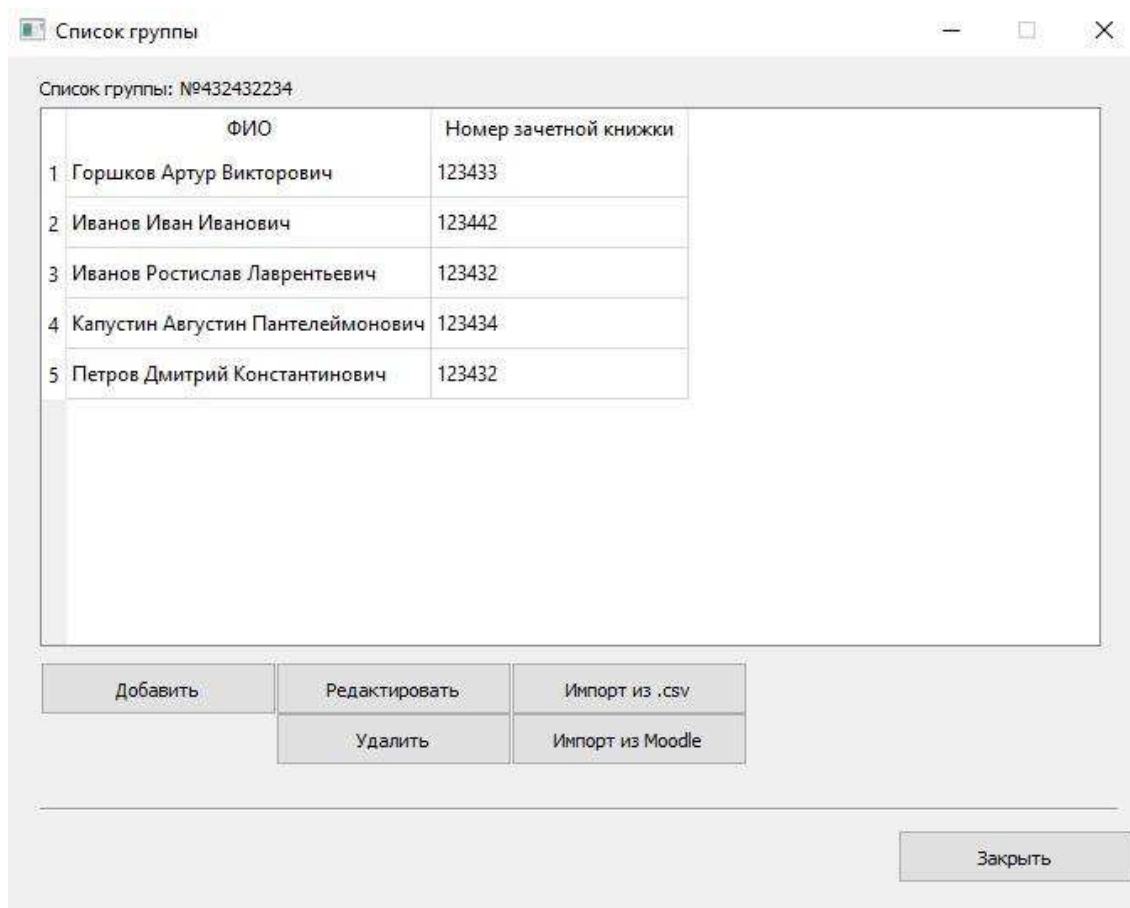


Рисунок 19 – Список группы

Тут пользователь уже сам выбирает, каким образом ему добавить студента в зависимости от выбранной кнопки. Также пользователь может удалить или отредактировать информацию.

Для того, чтобы совершить импорт из Moodle, пользователю необходимо зайти в окно «Параметры», которое показано на рисунке 17, далее нажать на кнопку «Импорт данных из LMS Moodle», ввести данные для авторизации и нажать кнопку «Синхронизировать данные» как показано на рисунке 20.

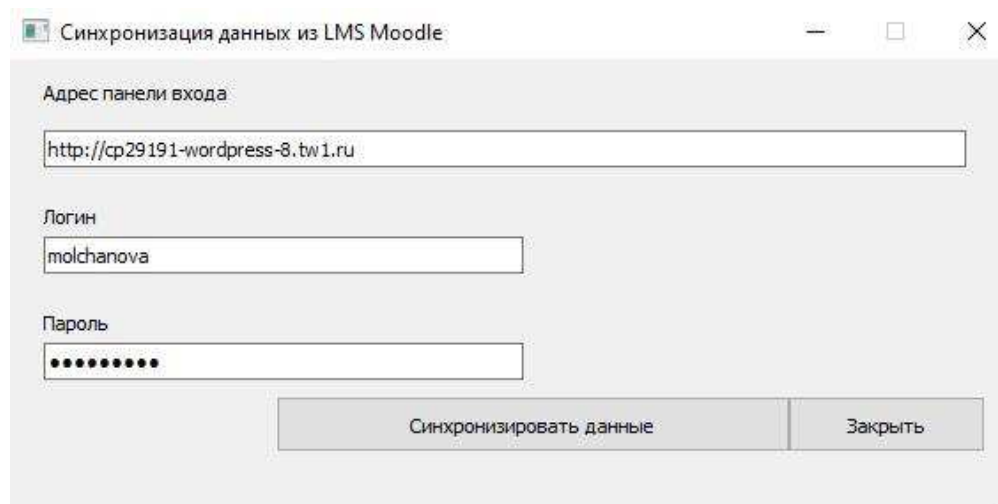


Рисунок 20 – Окно синхронизации с Moodle

Для того, чтобы посмотреть журнал оценок необходимо нажать на кнопку «Оценки» на главном окне, а далее кнопку «Открыть журнал» как показано на рисунке 21.

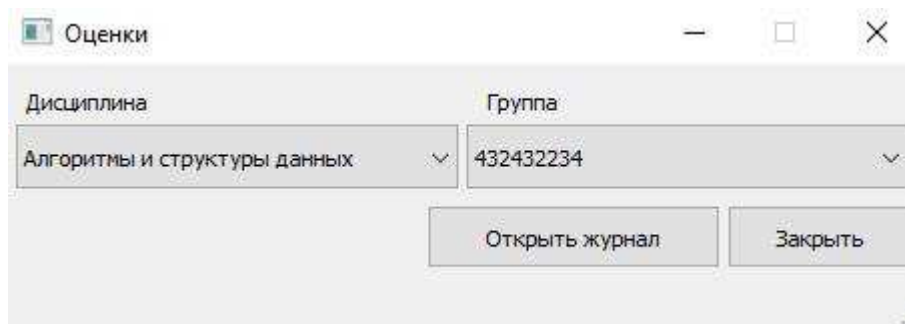


Рисунок 21 – Окно оценок

Для того, чтобы пользователь мог получить аналитику по успеваемости, ему нужно будет нажать на кнопку «Аналитика» на главном экране. Далее выбрать объект анализа и нажать кнопку далее, как показано на рисунке 22.

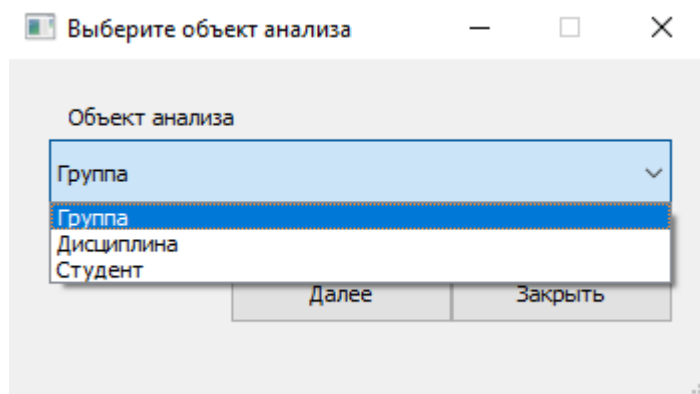


Рисунок 22 – Окно выбора объекта анализа

При выборе «Группа», пользователю будет необходимо настроить все параметры, по которым пользователь хочет получить аналитику, на рисунке 23, показано, как выглядит окно настройки успеваемости по группе.

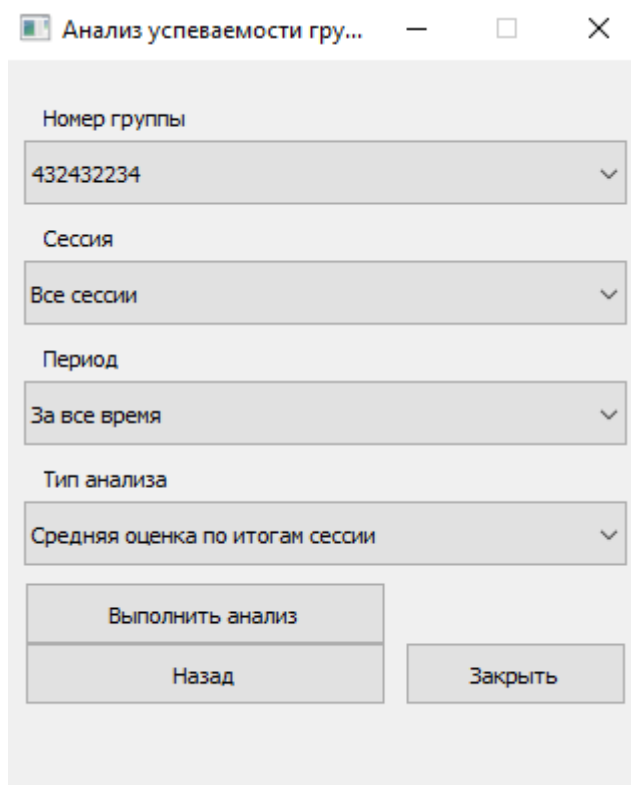
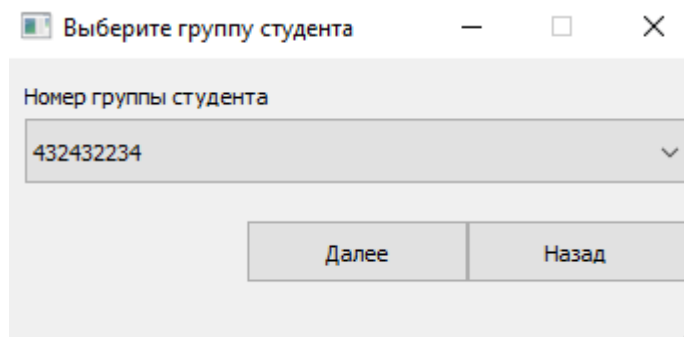


Рисунок 23 – Анализ успеваемости группы

При выборе «Студент», пользователь увидит окно настройки параметров для анализа по студентам (рисунок 24), пользователь должен будет выбрать группу и нажать кнопку «Далее».



Выберите группу студента

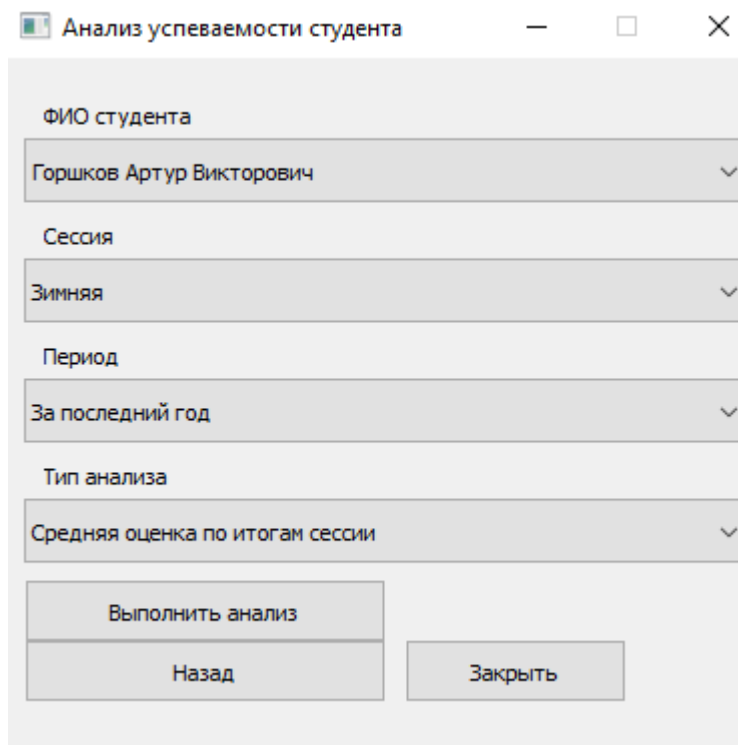
Номер группы студента

432432234

Далее Назад

Рисунок 24 – Анализ успеваемости студента, окно выбора группы

Далее (рисунок 25), пользователь должен выбрать необходимые параметры и нажать кнопку «Выполнить анализ».



Анализ успеваемости студента

ФИО студента

Горшков Артур Викторович

Сессия

Зимняя

Период

За последний год

Тип анализа

Средняя оценка по итогам сессии

Выполнить анализ

Назад Закрыть

Рисунок 25 – Анализ успеваемости студента

При выборе «Дисциплина» пользователю сначала необходимо будет выбрать дисциплину (рисунок 26)

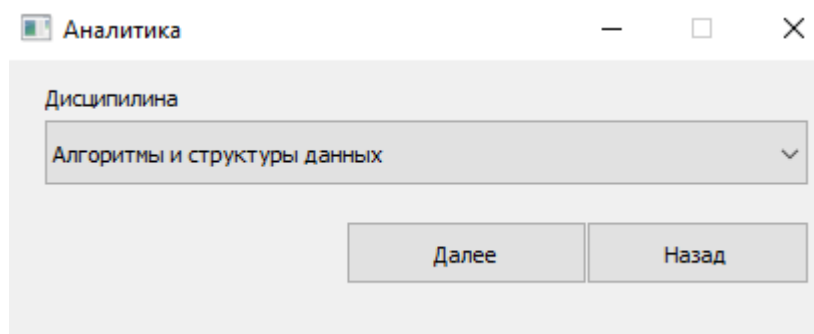


Рисунок 26 – Аналитика по дисциплине

Далее выбираем объект анализа (рисунок 27).

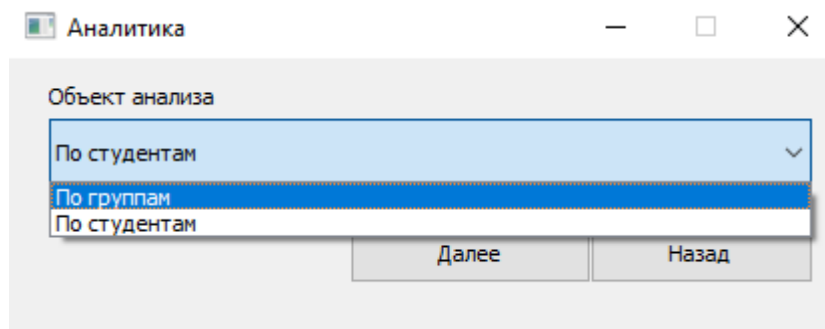


Рисунок 27 – Окно выбора объекта

Далее в зависимости от выбора, у пользователя будет два варианта продолжения. В случае выбора «По группам» пользователю сразу будет предложено сразу по какому параметру совершить анализ, а в случае выбора «по студента» сначала нужно будет выбрать группу, а далее уже выбрать по какому параметру совершить анализ.

Коротко о всех возможных вариантах параметров для анализа доступных пользователю.

По объектам:

- студент;
- дисциплина;
- группа.

По периоду:

- за последний год;
- за последние 2 года;
- за последние 3 года;
- за все время.

По сессии:

- зимняя;
- летняя.

По типу анализа:

- средняя оценка по итогам сессии;
- средняя оценка за работы в семестре;
- максимальная оценка по итогам сессии;
- максимальная оценка по итогам работы в семестре;
- минимальная оценка по итогам сессии;
- минимальная оценка по итогам работы в семестре;
- количество сданных работ в семестре.

Далее после выбора всех необходимых параметров, пользователь попадает в окно анализа (рисунок 28).

	Учебный год	Средняя оценка
1	2019-2020	61.6
2	2018-2019	63.17
3	2017-2018	52.25
4	2016-2017	52.21

Рисунок 28 – Окно анализа успеваемости студента

Здесь пользователь может либо отобразить диаграмму, либо сформировать отчет в зависимости от выбранной кнопки.

При нажатии на кнопку «Отобразить диаграмму» откроется окно с диаграммой (рисунок 29).

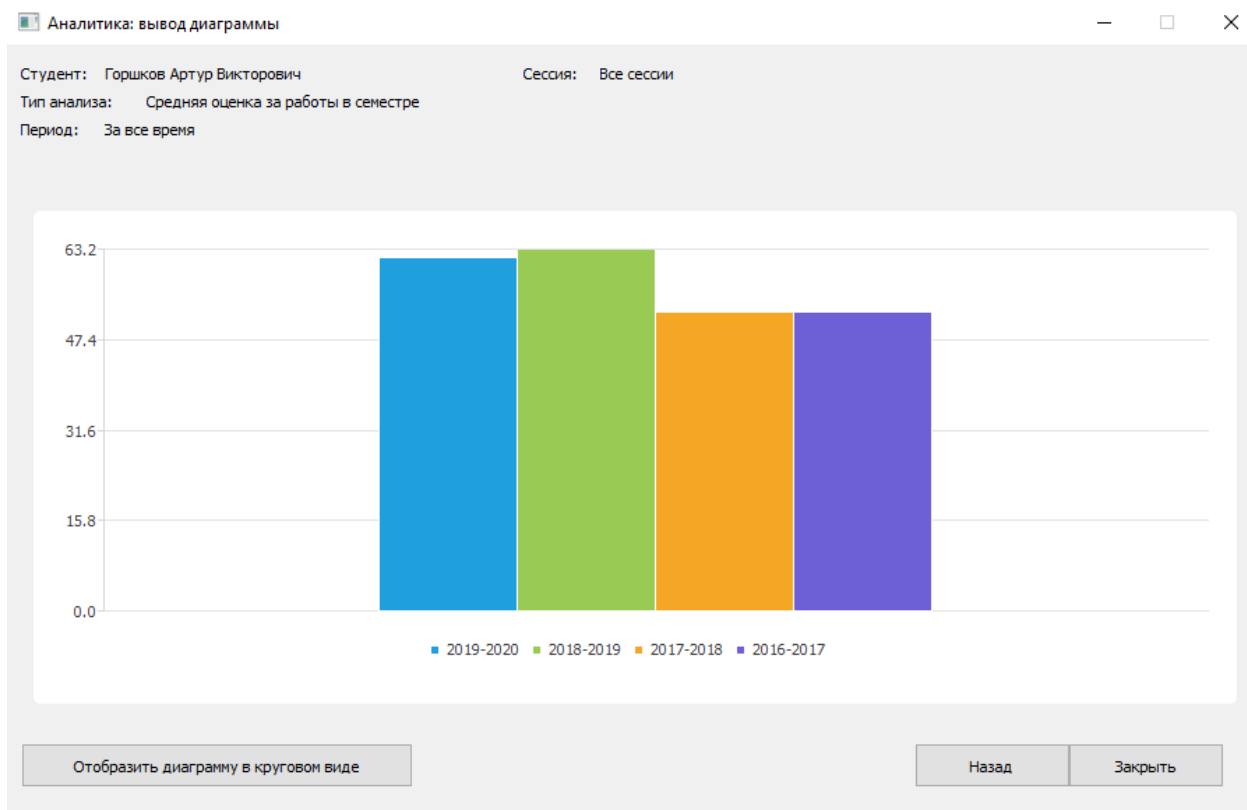


Рисунок 29 – Окно вывода диаграммы

При нажатии на кнопку «Сформировать отчет», сформируется отчет. Чтобы его сохранить, нужно открыть «Отчеты» с главного окна приложения, выбрать необходимый отчет и нажать кнопку «Сохранить» (рисунок 30).

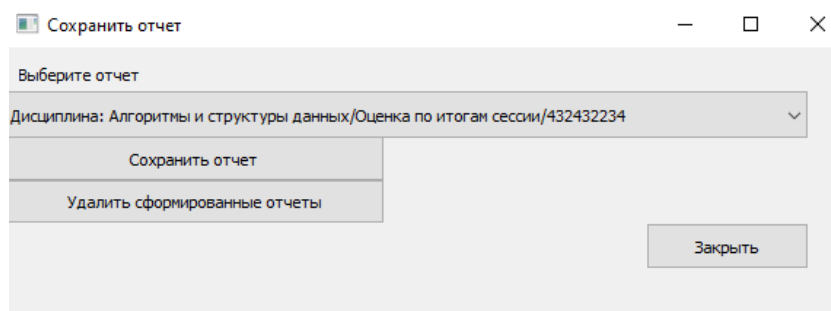


Рисунок 30 – Окно «Сохранить отчет»

Для выхода из приложения, пользователю необходимо нажать кнопку «Выход» на главном окне.

3.2 Инструкция разработчика

Для работы над приложением и его исходным кодом необходимо установить:

- PyCharm 2020.1 [21];
- Python 3.8 [22];
- Qt Creator 5.14.2 [23].

Необходимые библиотеки:

- Numpy 1.18.2;
- PyMySQL 0.9.3;
- PyQt5 5.14.2;
- PyQt5-stubs 5.14.2.0;
- PyQtChart 5.14.0;
- Pyqtgraph 0.10.0;
- SQLAlchemy 1.3.16.

Для установки данных библиотек, можно воспользоваться терминалом в PyCharm и с помощью файла requirements.txt установить необходимые библиотеки, либо программа сама предложить установить недостающие библиотеки. Необходимо так же в настройках выбрать необходимый интерпретатор, для этого нужно зайти в настройки, далее в «Project», нажать на Python interpreter и выбрать версию Python 3.8 (рисунок 31).

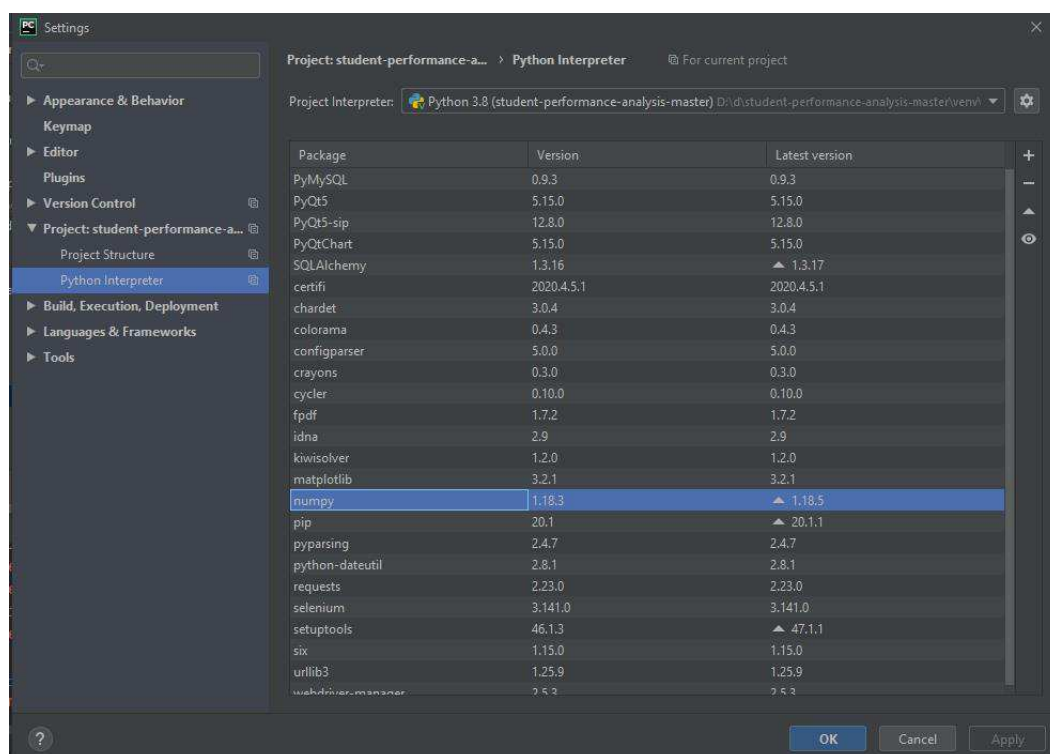


Рисунок 31 – Настройки проекта

В директории «db» располагается файл «models.py», в этом файле происходит работа с базой данных, каждый класс соответствует таблице в базе данных, а каждое поле класса соответствует полю в таблице базы данных. Для работы с таблицами базы данных, используются различные функций. Например, функция «Show_all» которая выводит список всех групп.

В директории «forms/py» находятся файлы, отвечающие за работу интерфейса, данные файлы были получены с помощью Qt и были сконвертированы в формат «py».

В файле «connecting.py» нужно внести данные для подключения к базе данных.

В файле «moodle.py» реализован импорт из LMS Moodle.

Qt Creator понадобится для внесения изменений в графический интерфейс приложения. Для этого нужно открыть в Qt Creator файл окна, который мы хотим изменить, из директории forms/ui. Далее, как файл будет отредактирован, необходимо сохранить и конвертировать в файл .py, для этого

в терминале необходимо выполнить команду: `pyuic5 -x UI/qt_ui/mainwindow.ui -o UI/py_ui/mainwindow.py`, где `mainwindow` – название вашего окна приложения.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы было спроектировано и реализовано приложение для мониторинга студентов.

В разработанном приложении были реализованы следующие функции:

- ввод/вывод данных через графический интерфейс;
- добавление/редактирование/удаление данных (группы, студенты);
- импорт данных в машиночитаемом формате (CSV);
- импорт данных из LMS Moodle;
- предоставление информации пользователю об отстающих студентах путем вывода не сданных работ;
- наглядное представление информации в различных разрезах: по студентам, по времени, по группе, по дисциплине, по сессии;
- базовые возможности анализа и визуализации данных, вычисление статистических показателей, построение диаграмм;
- формирование отчетов.

Приложение можно улучшить, путем модернизации уже реализованных функций:

- добавить представления информации в следующих разрезах: по виду элемента учебного плана, по виду занятия, по форме контроля;
- вывод данных в машиночитаемом формате.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Precision Campus: Customizable Reporting Software [Электронный ресурс]. – Режим доступа: <https://precisioncampus.com> (дата обращения: 28.02.2020).
2. Analysis Software Tableau [Электронный ресурс]. – Режим доступа: <https://www.tableau.com> (дата обращения: 28.02.2020).
3. The State of Data Education 2016 [Электронный ресурс]. – Режим доступа: <https://www.tableau.com/data-education-2016> (дата обращения: 10.03.2020).
4. SEAtS Software [Электронный ресурс]. – Режим доступа: <https://www.seatssoftware.com/student-attendance/> (дата обращения: 28.02.2020).
5. Creatrix Campus [Электронный ресурс]. – Режим доступа: <https://www.creatrixcampus.com/> (дата обращения: 28.02.2020).
6. Введение в JAVA. Язык программирования Java. [Электронный ресурс]. – Режим доступа: <https://ru.hexlet.io/blog/posts/yazyk-programmirovaniya-java-osobennosti-populyarnost-situatsiya-na-rynke-truda> (дата обращения: 05.03.2020).
7. Графический пользовательский интерфейс на Java [Электронный ресурс]. – Режим доступа: <http://journalpro.ru/articles/graficheskiy-polzovatelskiy-interfeys-na-java/> (дата обращения: 05.03.2020).
8. Стать Java разработчиком, а может быть выбрать Phyton? [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/477364/> (дата обращения: 04.03.2020)
9. Статьи о C++ [Электронный ресурс]. – Режим доступа: <https://urn.su/PC/code/cpp/> (дата обращения: 04.03.2020).
10. Qt for Python [Электронный ресурс]. – Режим доступа: https://wiki.qt.io/Qt_for_Python (дата обращения: 05.03.2020).

11. SQLite, MySQL и PostgreSQL: сравниваем популярные реляционные СУБД [Электронный ресурс]. – Режим доступа: <https://tproger.ru/translations/sqlite-mysql-postgresql-comparison/> (дата обращения: 04.03.2020).

12. ORM или как забыть о проектировании БД [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/237889/> (дата обращения: 05.03.2020).

13. ORM [Электронный ресурс]. – Режим доступа: [https://ru.bmstu.wiki/ORM_\(Object-Relational_Mapping\)](https://ru.bmstu.wiki/ORM_(Object-Relational_Mapping)) (дата обращения: 05.03.2020).

14. SQLALCHEMY краткое руководство [Электронный ресурс]. – Режим доступа: <https://coderlessons.com/tutorials/bazydannyykh/sqlalchemy/sqlalchemy-kratkoe-rukovodstvo> (дата обращения: 05.03.2020).

15. Сравниваем две популярные ORM на Python [Электронный ресурс]. – Режим доступа: https://otus.ru/nest/post/280/?admitad_uid=8ad274d1e418b8d315da7b7f5784e6d (дата обращения: 06.03.2020).

16. Peewee ORM, манипуляция базой данных [Электронный ресурс]. – Режим доступа: <https://python-scripts.com/peewee> (дата обращения: 06.03.2020).

17. Qt [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Qt> (дата обращения: 06.03.2020).

18. Model/view programming [Электронный ресурс]. – Режим доступа: <https://doc.qt.io/qt-5/model-view-programming.html> (дата обращения: 07.03.2020).

19. Знакомство с паттерном MVC (Model-view-controller) [Электронный ресурс]. – Режим доступа: <https://javarush.ru/groups/posts/2536-chastjh-7-znakomstvo-s-patternom-mvc-model-view-controller> (дата обращения: 15.04.2020).

20. База данных MySQL: что это такое? Преимущества перед другим СУБД. [Электронный ресурс]. – Режим доступа:

<https://webmasterie.ru/razrabotka/hosting/baza-dannih-mysql> (дата обращения: 23.04.2020).

21. PyCharm: the Python IDE for Professional Developers by JetBrains [Электронный ресурс]. – Режим доступа: <https://www.jetbrains.com/pycharm/> (дата обращения: 25.04.2020).

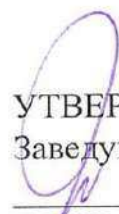
22. Welcome to Python.org [Электронный ресурс]. – Режим доступа: <https://www.python.org> (дата обращения 25.04.2020).

23. Get Qt [Электронный ресурс]. – Режим доступа: <https://www.qt.io/download> (дата обращения: 25.04.2020).

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

 О. В. Непомнящий

подпись

« ____ » _____ 2020 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 – «Информатика и вычислительная техника»

Приложение для мониторинга студентов


Руководитель


подпись, дата

канд. техн. наук,
доцент

А. И. Постников

Выпускник

 29.06.2020
подпись, дата

П. К. Толкачев

Консультант


подпись, дата

ст. преподаватель

К. В. Пушкарев

Нормоконтролер


подпись, дата

А. И. Постников

Красноярск 2020