

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»

Электронный журнал преподавателя

Руководитель	_____	канд. техн. наук, доцент	А. И. Постников
	подпись, дата		
Выпускник	_____		А. В. Дьяченко
	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 Дьяченко Артему Владимировичу.

Группа: КИ16-06Б. Направление (специальность): 09.03.01

«Информатика и вычислительная техника».

Тема выпускной квалификационной работы: «Электронный журнал преподавателя».

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 — электронный журнал преподавателя высшего учебного заведения.

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) учёт оценок, посещаемости, прохождения инструктажей по технике безопасности;
- б) просмотр истории изменения оценок;
- в) ввод/вывод данных через графический интерфейс, а также в машиночитаемом файловом формате (например, CSV, XML и т. п.);
- г) обмен информацией с LMS Moodle (загрузка и выгрузка оценок, списков работ и т. д.);
- д) синхронизация данных между устройствами;

е) автономная работа без доступа к сети.

Руководитель ВКР

подпись

А. И. Постников

Задание принял к исполнению

подпись

А. В. Дьяченко

30 декабря 2019 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Электронный журнал преподавателя» содержит 60 страниц текстового документа, 16 таблиц, 30 иллюстраций, 36 использованных источников.

ВЫСШЕЕ УЧЕБНОЕ ЗАВЕДЕНИЕ, ПРИЛОЖЕНИЕ, ЭЛЕКТРОННЫЙ ЖУРНАЛ, LMS MOODLE, БАЗА ДАННЫХ, PYTHON, QT, SQLITE, ORM.

Целью работы является разработка приложения — электронного журнала преподавателя высшего учебного заведения.

Для достижения цели в работе решаются следующие задачи:

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

В ходе работы проведен анализ различных электронных журналов преподавателя, рассмотрены существующие аналоги. На основе анализа были выделены основные требования к программному обеспечению.

Выполнен анализ различных инструментов для разработки, на основе которого выбраны инструменты для реализации приложения.

Спроектировано и реализовано настольное приложение – электронный журнал преподавателя высшего учебного заведения с графическим интерфейсом, поддержкой импорта из CSV и LMS Moodle. Составлены инструкции для пользователей и разработчиков.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Анализ задания на выпускную квалификационную работу	5
1.1 Обзор аналогичных программных продуктов	5
1.1.1 «Журнал преподавателя» (SJournal)	6
1.1.2 TeacherEase	7
1.1.3 Opengrade	8
1.1.4 Additio	9
1.1.5 Gnota	10
1.1.6 Teacher Gradebook	11
1.1.7 ThinkWave Administrator	13
1.1.8 Итоги обзора	14
1.2 Диаграмма прецедентов	18
1.3 Выбор инструментов	25
1.3.1 Выбор языка программирования	25
1.3.1.1 C++	25
1.3.1.2 Python	26
1.3.1.3 Java	27
1.3.1.4 Итог	27
1.3.2 Выбор ORM	28
1.3.2.1 SQLAlchemy	28
1.3.2.2 Django ORM	28
1.3.2.3 SQLAlchemy	29
1.3.2.4 Итог	29

2 Проектирование и реализация приложения.....	30
2.1 Архитектура приложения	30
2.2 Модульная структура	30
2.3 Структура базы данных.....	31
2.4 Диаграмма классов	39
2.5 Примеры интерфейса приложения	42
3 Инструкции.....	45
3.1 Инструкция пользователя	45
3.2 Инструкция разработчика	52
ЗАКЛЮЧЕНИЕ	56
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	57

ВВЕДЕНИЕ

Сегодня цифровые технологии интегрируются в нашу жизнь повсеместно. С их помощью автоматизируется большая часть задач, которые ранее выполнялись в ручном виде. Несмотря на это, далеко не все преподаватели высших учебных заведений используют информационные системы для автоматизации учета посещаемости и успеваемости студентов в повседневной жизни. Одной из причин является небольшой выбор готовых средств автоматизации учебного процесса с бесплатным доступом и гибкими системами настройки.

Целью работы является разработка приложения — электронного журнала преподавателя высшего учебного заведения.

Программа должна обладать следующими возможностями:

- поддержка операционных систем Windows и Linux;
- открытый исходный код;
- графический интерфейс пользователя;
- синхронизация данных между устройствами;
- поддержка ввода данных о дисциплине, группе, специальности, студенте и т.д. через графический интерфейс, а также в машиночитаемых форматах;
- учет оценок, посещаемости, прохождения инструктажей по технике безопасности;
- просмотр истории оценок;
- обмен информацией с LMS Moodle.

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

В соответствии с заданием необходимо разработать программный продукт: электронный журнал преподавателя высшего учебного заведения. Чтобы выполнить требование задания, был выполнен анализ аналогов, при этом отмечены их слабые и сильные стороны.

1.1 Обзор аналогичных программных продуктов

Единых рекомендаций и требований к электронным журналам в сфере высшего образования нет, поэтому вузы используют различные подходы и программные продукты. В ряде университетов Российской Федерации системы учета успеваемости студентов встроены в вузовскую структуру. Так, в МГТУ им. Н. Э. Баумана система «Электронный университет» обеспечивает контроль посещаемости студентами лекций и семинарских занятий [1], выполнение ими лабораторных работ и домашних заданий, курсового проекта и так далее. Доступ в такие системы носит закрытый характер и возможен только по логину и паролю преподавателя.

Существует также ряд программных продуктов, который может использовать преподаватель в своей работе:

- «Журнал преподавателя» (SJournal);
- TeacherEase;
- Opengrade;
- Addito;
- Gnota;
- Teacher Gradebook;
- ThinkWave.

Рассмотрим подробнее каждый программный продукт.

1.1.1 «Журнал преподавателя» (SJournal)

«Журнал преподавателя» (SJournal) – приложение, которое предназначено для преподавателей университетов. Предоставляет возможность вести журналы посещаемости и успеваемости учащихся [2].

Дополнительные возможности:

- настраиваемая сводная ведомость;
- студенты могут быть зачислены в несколько групп/подгрупп одновременно;
- настройка графического представления журнала (таблицы);
- импорт записей о студентах/группах из файлов;
- сохранение/восстановление базы данных (SD-card);
- графа «Среднее значение».

Положительными сторонами продукта является бесплатный доступ, простота установки из Play Market, мобильность, так как приложение разработано для ОС Android, интуитивно понятный интерфейс.

Недостатками данного ПО является отсутствие кроссплатформенности (ПО под ОС Android), отсутствие импорта данных о студентах и их оценках. Приложение является проприетарным программным обеспечением. Главное окно приложения Журнал преподавателя представлено на рисунке 1 [3].

The screenshot shows the SJournal application interface. At the top, there are navigation icons for 'ГРУППА', 'ДИСЦИПЛИНА', 'ВИД ЗАНЯТИЯ', 'НОВОЕ ЗАНЯТИЕ', and 'РЕЙТИНГ'. Below this is a header for the course: 'АТ-11-1 | ТОЭ 3ч - Теоретические основы электротехники | ЛБ'. The main part of the screen is a table with columns for student ID, FIO, and various assessment dates (ЛБ 1, ЛБ 2, Захист ЛБ 1, Захист ЛБ 2, ЛБ 3, ЛБ 4, ЛБ 5.1, ЛБ 5.2). The table contains 12 rows of student data and a summary row at the bottom. A dropdown menu is open on the right side, listing options like 'Семестр 1', 'Показать/скрыть панель', 'Дисциплины', 'Виды занятий', 'Группы', 'Студенты', 'Стиль таблицы', 'Экспорт', 'База данных', 'Настройки', 'Помощь', and 'О программе'.

№	Ф.И.О.	ЛБ 1 24.09.2013	ЛБ 2 01.10.2013	Захист ЛБ 1 08.10.2013	Захист ЛБ 2 15.10.2013	ЛБ 3 22.10.2013	ЛБ 4 29.10.2013	ЛБ 5.1 05.11.2013	ЛБ 5.2 12.11.2013
1	Белый Л.Г.			4	4				
2	Богданов С.С.			4	4	н	н	н	
3	Буц В.В.	н	н	4	4	н	н	н	н
4	Волошин А.Ю.		н	4	4		н		н
5	Иванов И.И.		н	4	4				н
6	Крамарев В.С.	н	н	4	4	н	н	н	н
7	Кулинич А.Ю.			4	4				
8	Петров Д.П.			4	4	6	н		н
9	Потапов Ю.В.			4	4	н	н	н	
10	Приморский В.А.								н
11	Романов С.М.	н	н	4	4	н	н	н	н
12	Степанов В.В.			4	4	н	н		н
Отсутствуют		3	5	0	0	6	8	5	8

Рисунок 1 – Главное окно приложения Журнал преподавателя

1.1.2 TeacherEase

TeacherEase – система оценок студентов [4]. Данная система создана в 2001 году, продукт активно поддерживается и в настоящее время. Система имеет только веб-интерфейс, в системе возможны три типа пользователей: преподаватель, студент и родитель.

К достоинствам данной системы стоит отнести встроенный модуль LMS, в котором преподаватель может настроить курс, а также длину курса от 1 до 4 семестров. Также стоит отметить встроенный модуль коммуникации студента и преподавателя, модуль анализа процесса обучения и оценок. Система предоставляет бесплатный доступ для личного использования преподавателям, для организаций система предоставляет платный доступ, цена доступа зависит от количества преподавателей, которые используют систему: чем больше преподавателей, тем дешевле стоимость одного аккаунта преподавателя. Система относится к проприетарному программному обеспечению.

К недостаткам системы относится отсутствие поддержки русского языка и невозможности импорта информации о студентах, экспорта оценок в .csv или .xlsx. Данный продукт в основном ориентирован на учебные заведения США и Канады, именно учебные заведения данных территорий фигурируют в настройках личного кабинета пользователя. Данная система не поддерживает сторонние LMS, в частности LMS Moodle. Экранная форма личного кабинета преподавателя системы оценок студентов TeacherEase представлена на рисунке 2 [5].

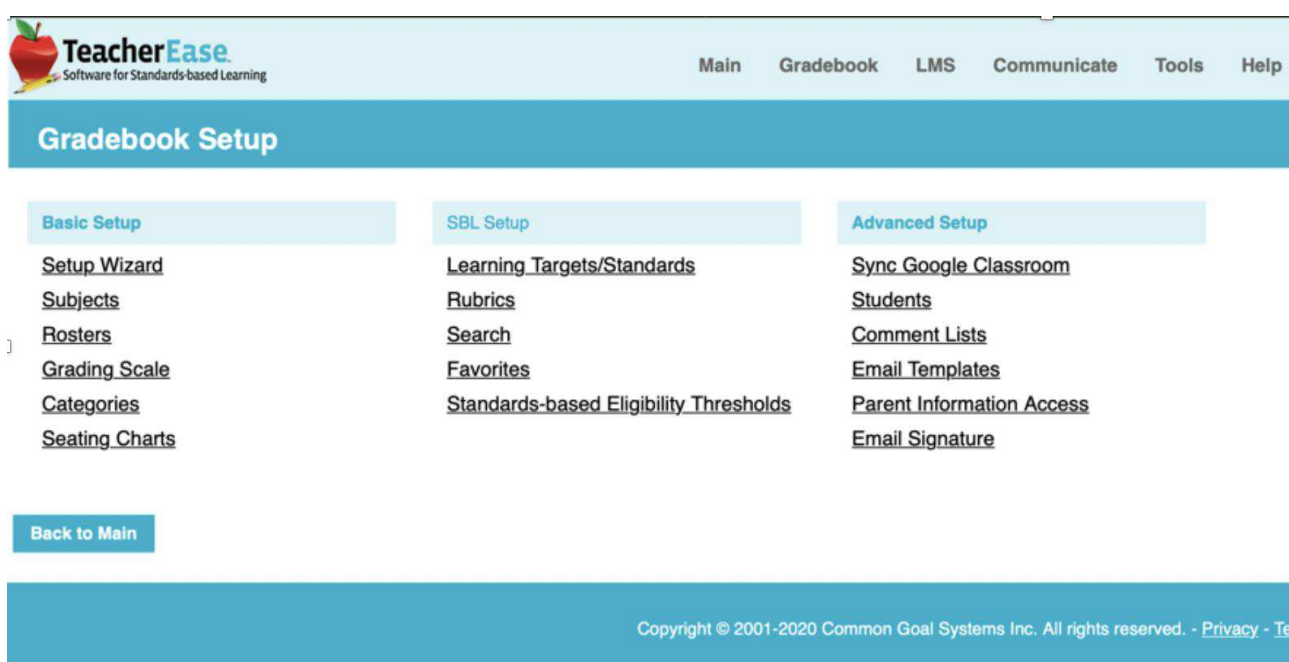


Рисунок 2 – Окно настроек личного кабинета преподавателя системы оценок студентов TeacherEase

1.1.3 Opengrade

Opengrade – программное обеспечение для учета оценок студентов, распространяющиеся под лицензией GPL v2. Opengrade позволяет создавать онлайн отчеты, вычислять оценки студентов по заданным формулам, защитить оценки студентов от взлома, учитывать посещаемость, экспортировать в электронную таблицу и другие форматы, выполнять автосохранение и

автоматическое резервное копирование [6]. Данный продукт имеет версии для Linux, MacOS, Windows и FreeBSD.

Интерфейс продукта не поддерживает русский язык. Экранная форма приложения представлена на рисунке 3 [6].

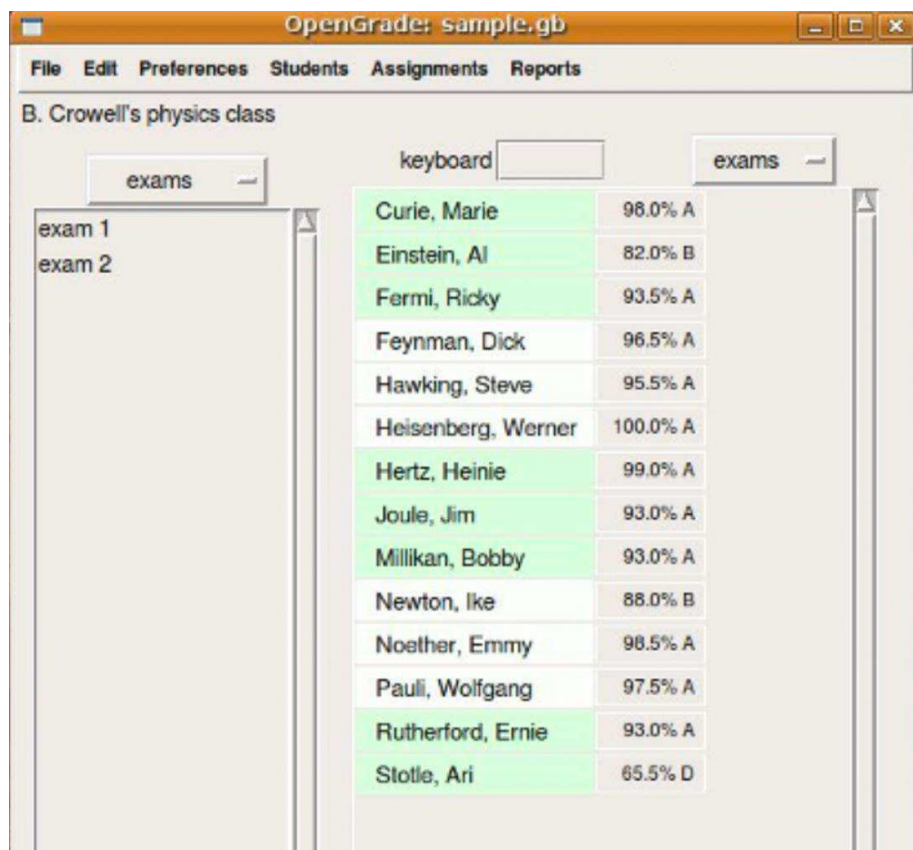


Рисунок 3 – Учет результатов экзамена в программе OpenGrade

1.1.4 Additio

Additio – приложение для IOS, Android и Web version [7]. Приложение оснащено следующими функциями: планировщик занятий на семестр, возможность экспертной оценки и самооценки обучающимся в выделенных разделах, настраиваемый шаблон оценки, индивидуальные отчеты, запись посещаемости, расписание, план рассадки и календарь, работает в автономном режиме на мобильных устройствах, интегрирован с Google Classroom. Google Classroom — это сервис Google Apps для образования, который позволяет

преподавателям быстро создавать и упорядочивать задания, выставлять оценки, оставлять комментарии и общаться с учащимися. В свою очередь, учащиеся могут хранить задания на «Google Диск», а также сдавать выполненные работы в «Классе» и напрямую общаться друг с другом и с преподавателями [8]. Приложение поддерживает функции общения со студентами и их родителями, экспорт данных в Excel и PDF, возможность связать с «Google Диск» и Microsoft OneDrive [9]. Приложение является проприетарным программным обеспечением.

Интерфейс приложения не поддерживает русский язык, отсутствует поддержка импорта оценок из LMS Moodle. Экранная форма web интерфейса представлена на рисунке 4 [10].

The screenshot shows the 'Digital gradebook' interface for 'Additio App'. It displays a table for '3A Language and Literature Classroom 3A' for 'Term 1'. The table includes columns for student names, various assessment categories, and final averages. The data is as follows:

3A Language and Literature Classroom 3A		Term 1								
		+ - 10%	Exercise 1	Exercise 2	Average exercise 30%	Exam 40%	Attendance 10%	Writing and reading 10%	Final Average	Fail-Pass
1.	Collins, Adam	6	7,4	8	7.7	8,4	9.5	8.2	8.04	Pass
2.	Stond, Simon	9	8	7	7.5	7	10	6.6	7.61	Pass
3.	Gordon, Martha	8	7	9 ↑	8	7,2	10	8.8	7.96	Pass
4.	Pierce, Justin	6	9	8.5	8.75	6,8	10	6.2	7.57	Pass
5.	Booth, Peter	7	7,4	7	7.2	6,7	10	6.8	7.22	Pass
6.	Klein, Roger	4	4	4.3	4.15	6	10	7.8	5.83	Pass
7.	Farr, Amy	8	8,5	9	8.75	6,6	10	7.8	7.85	Pass

Рисунок 4 – Цифровой журнал Additio

1.1.5 Gnota

Gnota – приложение для оценки успеваемости, разработанное под руководством Леандро Ламейро из Google. Данное приложение позволяет вести учет оценок, посещаемости, есть поддержка импорта студентов, возможность хранения личной информации о студентах, возможность добавления нескольких

предметов и настройка различных форм контроля по ним [11]. Приложение распространяется под лицензией GNU GPL v3.

Приложение имеет следующие недостатки: поддержка только ОС Linux, отсутствие поддержки русского языка, импорта данных из LMS Moodle и формирования отчетов. Экранная форма приложения представлена на рисунке 5 [12].

The screenshot shows a desktop application window titled 'Gnota'. The menu bar includes 'File', 'Edit', 'Manage', and 'Help'. The toolbar contains icons for 'Open', 'Save', 'Add Activity', 'Edit classes', and 'Edit Students'. A dropdown menu shows the current class: 'Introduction to Physics'. Below the toolbar is a table with the following data:

Name	Homework 1	Homework 2	Homework 3	Overall score	Missed classes	Approved
Bar, Mary				F	0	<input type="checkbox"/>
Baz, Richard				F	0	<input type="checkbox"/>
Foo, John	Pass	Pass	Pass	B-	0	<input checked="" type="checkbox"/>
Quux, Foo 0				F	0	<input type="checkbox"/>
Quux, Foo 1				F	0	<input type="checkbox"/>
Quux, Foo 2				F	0	<input type="checkbox"/>
Quux, Foo 3				F	0	<input type="checkbox"/>
Quux, Foo 4				F	0	<input type="checkbox"/>
Quux, Foo 5				F	0	<input type="checkbox"/>
Quux, Foo 6				F	0	<input type="checkbox"/>
Quux, Foo 7				F	0	<input type="checkbox"/>
Quux, Foo 8				F	0	<input type="checkbox"/>
Quux, Foo 9				F	0	<input type="checkbox"/>
Quux, Foo 10				F	0	<input type="checkbox"/>
Quux, Foo 11				F	0	<input type="checkbox"/>
Quux, Foo 12				F	0	<input type="checkbox"/>
Quux, Foo 13				F	0	<input type="checkbox"/>
Quux, Foo 14				F	0	<input type="checkbox"/>
Quux, Foo 15				F	0	<input type="checkbox"/>
Quux, Foo 16				F	0	<input type="checkbox"/>
Quux, Foo 17				F	0	<input type="checkbox"/>

Рисунок 5 – Приложение для оценки успеваемости Gnota

1.1.6 Teacher Gradebook

Teacher Gradebook – мобильное приложение для Android [13]. Особенности приложения: особый дизайн для планшетов, недельное расписание, экспорт в PDF и CSV, зашифрованная база данных для обеспечения безопасности данных,

управление студентами: добавить/удалить студентов, добавить комментариев, выбор случайного студента для опроса, поддержка различных цветовых схем для журнала оценок, позволяет группировать студентов или предметы, учет посещаемости: отсутствие, задержки, обоснованное отсутствие, оправданные задержки, резервное копирование, импорт студентов из CSV [14]. Поддерживает следующие программные продукты в сфере образования: ClassDojo [15], Kahoot! [16], Moodle [17], Edmodo [18], Socrative [19] и Google Classroom [8]. Приложение является проприетарным программным обеспечением.

Недостатками данного решения является отсутствие поддержки других операционных систем, в частности ОС для стационарных ПК. Отсутствует поддержка русского языка. Экранная форма приложения представлена на рисунке 6 [14].

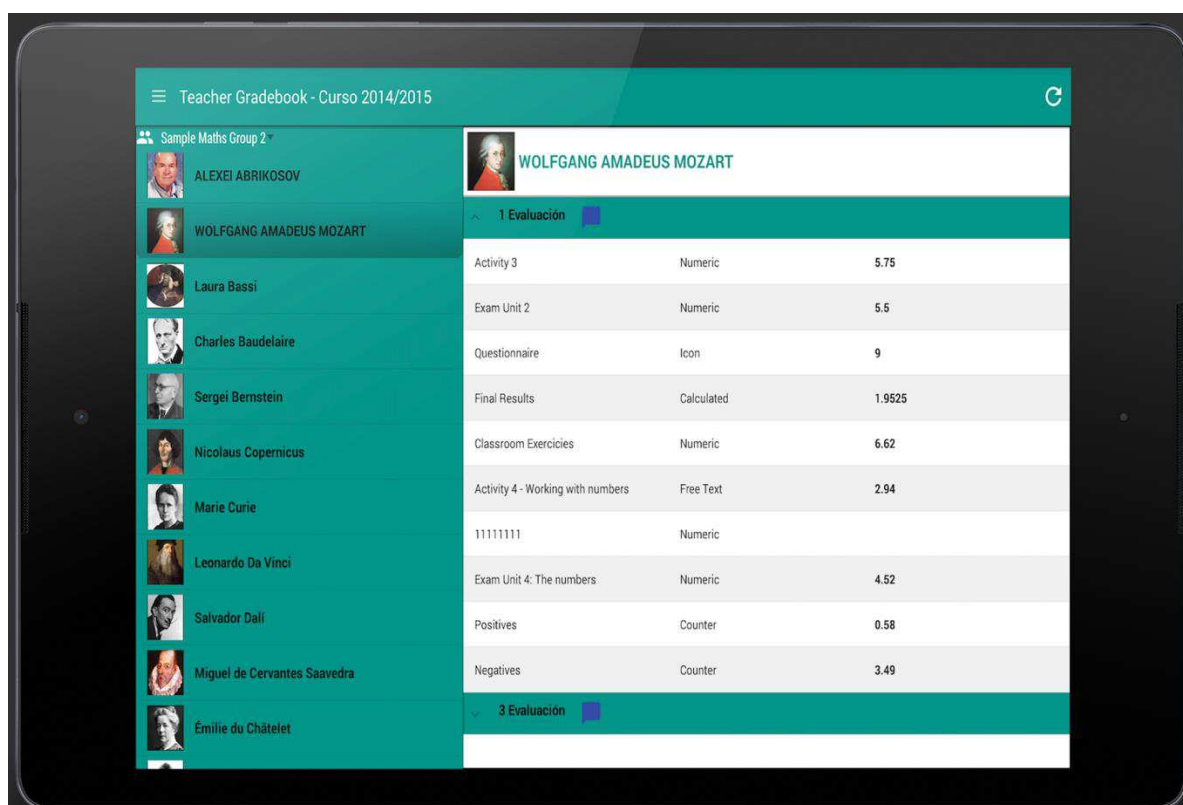


Рисунок 6 – Версия для планшета приложения Teacher Gradebook

1.1.7 ThinkWave Administrator

ThinkWave Administrator – это сетевая система учебного администрирования, которая автоматически собирает оценки, комментарии и посещаемость от учителей, что упрощает создание сводных таблиц успеваемости учащихся [20]. Данные из нескольких источников автоматически объединяются для создания отчетов, которые можно распечатать или отправить по электронной почте учащимся и родителям. Студент может иметь несколько итоговых оценок. Например, по математическому анализу студент может иметь общую оценку и дополнительные оценки для таких стандартов, как «Пространственное мышление» и «Решение проблем». Истории студентов содержат результаты обучения студентов на протяжении всего срока обучения. «Учет истории» в системе позволяет вузам записывать исторические данные прошлых лет для составления сводных отчетов по выпускам. Система поддерживает различные кабинеты пользователей, в том числе кабинет администратора и учителя, предоставляя возможность делегировать функции участникам образовательного процесса.

Система позволяет загружать файлы до 25 МБ каждый, размер базового онлайн-хранилища составляет 100 ГБ, который может быть увеличен при необходимости. Поддерживаются многие форматы файлов, включая .pdf, .doc, файлы изображений, файлы фильмов и другие. Работа с данной системой происходит через web-версию. Преподаватели могут отслеживать посещаемость, выставлять оценки, составлять отчеты, отправлять сообщения, прикреплять файлы, давать домашние задания, делать рассылки студентам.

Продукт является платным, цена зависит от количества учеников, использующих систему, и начинается от 299\$ в месяц. Отсутствует поддержка русского языка и офлайн доступа. Экранная форма приложения представлена на рисунке 7 [21].

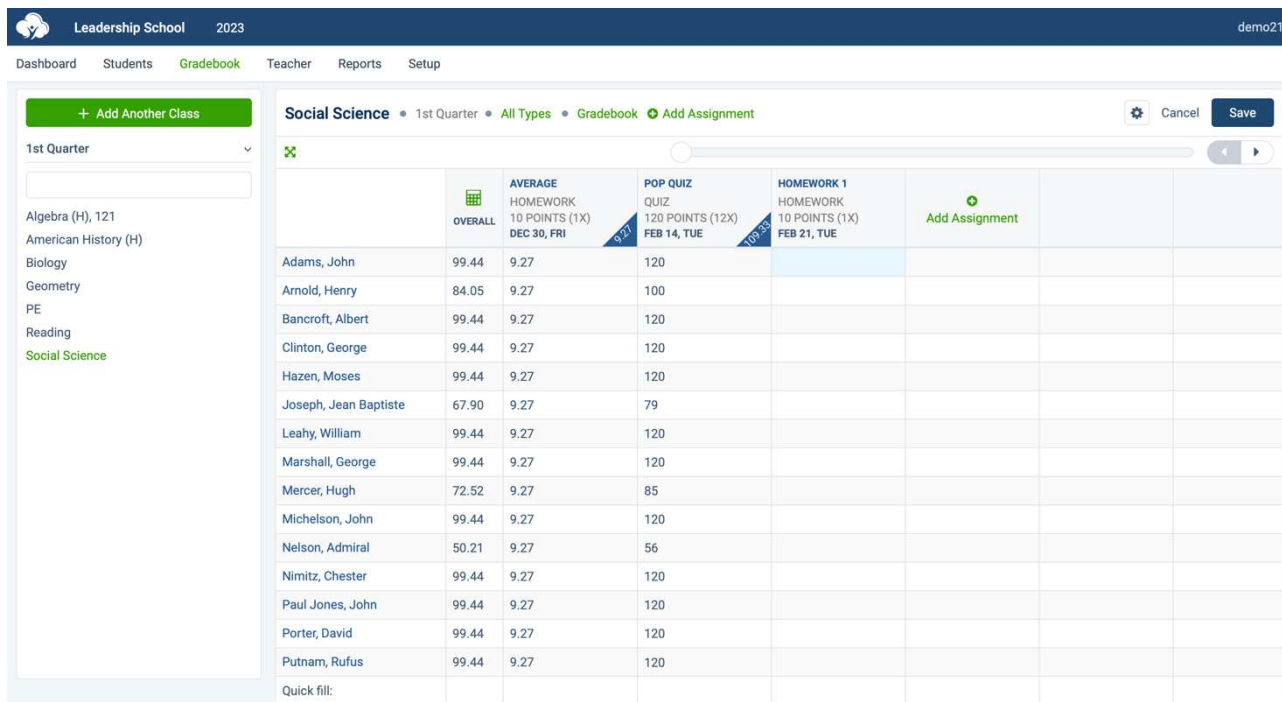


Рисунок 7 – Главная страница личного кабинета преподавателя ThinkWave в демонстрационном режиме

1.1.8 Итоги обзора

Из обзора программных продуктов, представленных в пунктах 1.1.1 – 1.1.7, можно сделать вывод, что все рассмотренные аналоги предназначены для учета успеваемости студента.

Каждое приложение позволяет добавлять студентов и выставлять ему оценки. Кроме того, некоторые приложения имеют особенные функции, не встречающиеся в других аналогах, так Teacher Gradebook поддерживает различные цветовые схемы для журнала оценок, Gnota и Teacher Gradebook позволяют импортировать данные из LMS Moodle. ThinkWave Administrator, Additio и TeacherEase являются системами, поддерживающими личные кабинеты преподавателя и студента. Системы личных кабинетов позволяют обеспечить взаимодействие студента и преподавателя в дистанционном режиме. Некоторые программные продукты (SJournal, TeacherEase, Opengrade, Teacher Gradebook)

поддерживают импорт и/или экспорт в машиночитаемом файловом формате, что можно было бы использовать в разрабатываемом проекте.

В результате анализа программных продуктов, был сделан вывод: в настоящее время не существует программного продукта, который обладал бы полным набором необходимых функций, в частности в большинстве продуктов отсутствует поддержка импорта данных из LMS Moodle, ввод и вывод через машиночитаемые файловые форматы CSV и XML. Большинство продуктов не являются кроссплатформенными и не поддерживают русский язык. Полученные данные из обзора были сведены в таблицу 1.

Таблица 1 – Основные сведения об аналогах

Свойство	SJournal	TeacherEase	Opengrade	Addito	Gnota	Teacher Gradebook	ThinkWave
Лицензия ПО	Про- приетарное	Проприетарное	GPL v2	Проприетарное	GNU GPL v3	Проприетарное	Проприетарное
Настраиваемые уведомления	Нет	Нет	Нет	Да	Нет	Да	Да
Обмен данными с LMS Moodle	Нет	Нет	Нет	Нет	Да	Да	Нет
Поддержка русского языка	Да	Нет	Нет	Нет	Нет	Нет	Нет
Возможность работы на стационарном ПК	Нет	Да	Да	Да	Да	Нет	Да
Графический интерфейс	Да	Да	Да	Да	Да	Да	Да

Окончание таблицы 1

Свойство	SJournal	TeacherEase	Opengrade	Addito	Gnota	Teacher Gradebook	ThinkWave
Ввод/вывод через CSV и XML	Да	Да	Да	Нет	Нет	Да	Нет
Вычисление оценок по заданным формулам	Нет	Да	Нет	Да	Нет	Да	Да
Создание отчетов	Нет	Нет	Нет	Да	Нет	Да	Да

1.2 Диаграмма прецедентов

На рисунке 8 изображена диаграмма прецедентов.

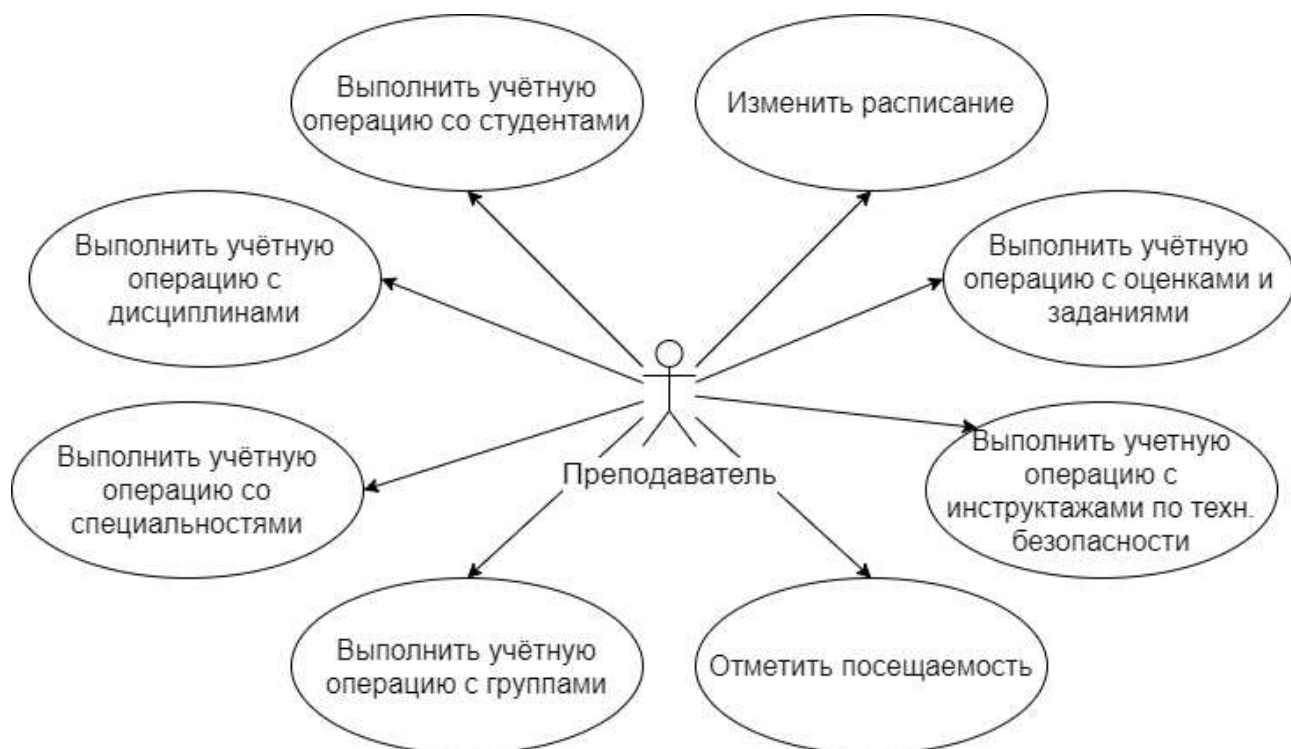


Рисунок 8 – Диаграмма прецедентов

Текстовое описание прецедентов приведено ниже.

Название прецедента: **Выполнить учётную операцию с дисциплинами.**

Цель сценария: отредактировать, добавить, удалить или посмотреть информацию о дисциплинах.

Предусловие: пользователь находится в главном окне приложения и нажимает кнопку «Дисциплины», открывается окно со списком дисциплин, после в открывшемся окне нажимает кнопку «Добавить», «Редактировать», вводит данные и нажимает на кнопку «Сохранить». Либо выбирает дисциплину и нажимает кнопку «Удалить».

Основной сценарий:

А. Данные проходят проверку на корректность и на наличие данных в базе данных.

В. Обновляются данные в базе данных.

Постусловие: пользователь попадает в окно «Дисциплины» с отображением обновленного списка дисциплин.

Условие ввода в действие альтернативных сценариев:

Условие 1. Проблема со связью с базой данных.

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

Условие 2. Данные введены некорректно.

А. Приложение выдает сообщение о некорректности введенных данных.

Условие 3. Дисциплина с таким названием уже существует (для добавления).

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

Название прецедента: **Выполнить учётную операцию с группами.**

Цель сценария: отредактировать, добавить, удалить или посмотреть информацию о группах.

Предусловие: пользователь находится в главном окне приложения, после чего нажимает на кнопку «Студенты», в открывшемся окне нажимает кнопку «Группы», появляется список групп, в окне «Группы» для добавления нажимает на кнопку «Импорт из .csv», в открывшемся окне выбирает .csv файл и нажимает на кнопку «Открыть». Либо пользователь нажимает кнопку «Импорт из Moodle», вводит данные для авторизации и нажимает кнопку «Импортировать». Также возможен ввод данных вручную, нажав кнопку «Добавить», далее требуется ввести все необходимые данные вручную. Для редактирования выбирает группу, нажимает кнопку «Редактировать», меняет необходимую информацию и нажимает кнопку «Сохранить». Для удаления выбирает группу и нажимает кнопку «Удалить».

Основной сценарий:

А. Данные проходят проверку на корректность и на наличие данных в базе данных.

В. Обновляются данные в базе данных.

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

Условие ввода в действие альтернативных сценариев:

Условие 1. Проблема со связью с базой данных.

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

Условие 2. Данные введены некорректно (для добавления или редактирования).

А. Приложение выдает сообщение о некорректности введенных данных.

Условие 3. Группа с таким номером уже существует (для добавления).

А. Приложение выдает сообщение о существовании группы с таким номером в базе данных.

Название прецедента: **Изменить расписание.**

Цель сценария: изменить расписание.

Предусловие: пользователь находится в главном окне, нажимает на кнопку «Расписание», в открывшемся окне отображается текущее расписание, для добавления занятия пользователь нажимает кнопку «Добавить единичное занятие» или «Добавить периодичное занятие». Для добавления единичного занятия в открывшемся окне пользователь указывает «Дисциплину», «Формат занятия», «Группу», «Дату», «Время начала занятия». Для добавления периодичного занятия необходимо дополнительно ввести «Дата окончания курса» и указать «Периодичность». Для удаления или редактирования занятия пользователь выбирает его с помощью курсора, кликнув по нему, нажимает кнопку «Удалить» или «Редактировать» соответственно.

Основной сценарий:

А. Данные проходят проверку на корректность.

В. Обновляются данные в базе данных.

Постусловие: пользователь попадает в окно «Расписание» с отображением обновленного расписания занятий.

Условие ввода в действие альтернативных сценариев:

Условие 1. Проблема с связью со базой данных.

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

Условие 2. Данные введены некорректно (для добавления или редактирования).

А. Приложение выдает сообщение о некорректности введённых данных.

Название прецедента: **Выполнить учётную операцию со специальностями.**

Цель сценария: отредактировать, добавить, удалить или посмотреть информацию о специальностях.

Предусловие: пользователь находится в главном окне приложения и нажимает кнопку «Студенты», в открывшемся окне нажимает кнопку «Специальности», открывается окно со списком специальностей, далее пользователь нажимает кнопку «Добавить» или «Редактировать» вводит данные и нажимает на кнопку «Добавить» или «Сохранить» соответственно. Для удаления пользователь выбирает специальность нажимает кнопку «Удалить».

Основной сценарий:

А. Данные проходят проверку на корректность и на наличие данных в базе данных.

В. Обновляются данные в базе данных.

Постусловие: пользователь попадает в окно «Специальности» с отображением обновленного списка специальностей.

Условие ввода в действие альтернативных сценариев:

Условие 1. Проблема со связью с базой данных.

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

Условие 2. Данные введены некорректно (для добавления и редактирования).

А. Приложение выдает сообщение о некорректности введённых данных.

Условие 3. Специальность с таким названием уже существует (для добавления).

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

Название прецедента: **Выполнить учётную операцию со студентами.**

Цель сценария: отредактировать, добавить, удалить или посмотреть информацию о студентах в группе.

Предусловие: пользователь находится в главном окне приложения, после чего нажимает на кнопку «Студенты», в открывшемся окне выбирает группу, нажимает кнопку «Посмотреть список группы», появляется список студентов, в этом окне нажимает на кнопку «Импорт из .csv», в открывшемся окне выбирает .csv файл и нажимает на кнопку «Открыть». Либо нажимает кнопку «Импорт из «Moodle», вводит данные для авторизации и нажимает кнопку «Импортировать». Тоже возможен ввод данных вручную, нажав кнопку «Добавить студента», после требуется ввести информацию о студенте. Для редактирования информации о студенте необходимо выбрать студента, кликнув на него курсором, нажать кнопку «Редактировать», внести изменения и нажать кнопку «Сохранить». Для удаления необходимо выбрать студента, кликнув на него курсором и нажать кнопку «Удалить».

Основной сценарий:

А. Данные проходят проверку на корректность и на наличие данных в базе данных.

В. Обновляются данные в базе данных.

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

Условие ввода в действие альтернативных сценариев:

Условие 1. Проблема со связью с базой данных.

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

Условие 2. Данные введены некорректно (для добавления и редактирования).

А. Приложение выдает сообщение о некорректности введенных данных.

Условие 3. Студент с такими данными (ФИО, номер зачетной книжки) уже существует (для добавления).

А. Приложение выдает сообщение о существовании студента с такими данными в базе данных.

Название прецедента: **Выполнить учётную операцию с оценками и заданиями.**

Цель сценария: отредактировать, добавить, удалить или посмотреть информацию об оценках студентов, заданий для студентов.

Предусловие: пользователь находится в главном окне, нажимает на кнопку «Успеваемость», в открывшемся окне выбирает дисциплину и номер группы, нажимает на кнопку «Оценки, появляется окно с текущими оценками. В открывшемся окне с оценками пользователь выбирает студента и работу, которую необходимо оценить и нажимает на кнопку «Добавить оценку», вводит данные, нажимает на кнопку «Добавить». Нажимает кнопку «История» для просмотра истории оценок. Нажимает на кнопку «Добавить работу», чтобы добавить новое задание, чтобы посмотреть подробности о задании (задание и заметка), необходимо, выбрав работу, нажать кнопку «Подробнее о работе». Для редактирования данных необходимо выбрать их с помощью курсора и кликнуть на них, далее нажать кнопку «Редактировать», внести изменения и нажать кнопку «Сохранить».

Основной сценарий:

А. Данные проходят проверку на корректность.

В. Обновляются данные в базе данных.

Постусловие: пользователь попадает в окно «Оценки» с отображением обновленного журнала оценок.

Условие ввода в действие альтернативных сценариев:

Условие 1. Проблема с связью со базой данных.

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

Условие 2. Данные введены некорректно.

А. Приложение выдает сообщение о некорректности введенных данных.

Название прецедента: **Выполнить учётную операцию с инструктажами по технике безопасности.**

Цель сценария: отредактировать, добавить, удалить или посмотреть информацию об инструктажах по технике безопасности.

Предусловие: пользователь находится в главном окне, нажимает на кнопку «Успеваемость», в открывшемся окне выбирает дисциплину и номер группы, нажимает на кнопку «Техника безопасности», появляется окно с журналом по технике безопасности. В открывшемся окне для отметки прохождения инструктажа пользователь нажимает кнопку «Добавить», выбирает тему инструктажа, студента, вводит дату и нажимает кнопку «Добавить». Для добавления новой темы инструктажа пользователь нажимает кнопку «Добавить тему», вводит данные и сохраняет новую тему инструктажа. Выбирает тему инструктажа, нажимает на кнопку «Удалить» для удаления темы инструктажа.

Основной сценарий:

А. Данные проходят проверку на корректность.

В. Обновляются данные в базе данных.

Постусловие: пользователь попадает в окно «Техника безопасности» с отображением обновленного журнала по технике безопасности.

Условие ввода в действие альтернативных сценариев:

Условие 1. Проблема с связью со базой данных.

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

Условие 2. Данные введены некорректно.

А. Приложение выдает сообщение о некорректности введенных данных.

Название прецедента: **Отметить посещаемость**

Цель сценария: отметить посещаемость занятий студентами.

Предусловие: пользователь находится в главном окне, нажимает на кнопку «Успеваемость», в открывшемся окне выбирает дисциплину и номер группы, нажимает на кнопку «Посещаемость», появляется окно с журналом посещаемости. В открывшемся окне для отметки посещения занятия

пользователь выбирает студента и занятие, нажимает кнопку «Посетил/Не посетил».

Основной сценарий:

А. Данные проходят проверку на корректность.

В. Обновляются данные в базе данных.

Постусловие: пользователь попадает в окно «Журнал посещаемости» с отображением обновленного журнала посещаемости.

Условие ввода в действие альтернативных сценариев:

Условие 1. Проблема с связью со базой данных.

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

1.3 Выбор инструментов

1.3.1 Выбор языка программирования

Согласно требованию задания, разрабатываемый продукт, должен иметь поддержку операционных систем Windows и Linux. С каждым языком/фреймворком связаны свои сложности и нюансы, достоинства и недостатки. Рассмотрим основные языки/фреймворки для написания кроссплатформенных приложений.

1.3.1.1 C++

C++ – компилируемый, статически типизированный язык программирования общего назначения. Плюсами данного языка являются высокая вычислительная производительность, поддержка различных стилей программирования, шаблоны (возможность построения обобщённых контейнеров и алгоритмов для разных типов данных), огромное количество учебной литературы. Из минусов можно выделить громоздкость синтаксиса и необходимость следить за памятью [22; 23].

C++ может быть использован для разработки кроссплатформенных приложений с использованием фреймворка Qt. Это кроссплатформенная среда разработки приложений для настольных, встраиваемых и мобильных устройств. Поддерживаемые платформы включают Linux, OS X, Windows, VxWorks, QNX, Android, iOS, BlackBerry, Sailfish OS и другие. Qt не является языком программирования сам по себе, это фреймворк, написанный на C++ [24]. Qt обладает всеми необходимыми возможностями для разработки программного обеспечения, включает в себя элементы графического интерфейса и классы для работы с сетью, базами данных и XML. Данный фреймворк активно развивается и поддерживается профессиональным сообществом.

1.3.1.2 Python

Python – это высокоуровневый язык программирования общего назначения, который можно применять ко многим различным классам задач [25].

Python может быть использован для разработки приложений, поддерживающих работу в операционных системах Windows/Linux. Так при разработке кроссплатформенных приложений может быть использован набор расширений к Qt посредством использования библиотек PyQt5 или PySide2. Данные библиотеки очень похожи, они отличаются размером сообщества, у PyQt5 профессиональное сообщество больше, однако для PySide2 имеется официальная документация на сайте Qt. PyQt5 практически полностью реализует возможности Qt: существующий набор виджетов графического интерфейса, стили виджетов, парсер XML, поддержку SVG.

По сравнению с C++, приложения на Python разрабатывать быстрее, синтаксис языка гораздо проще, код языка более читаемый [25].

1.3.1.3 Java

Java является одним из самых популярных языков программирования [26]. Он тоже может быть использован для написания кроссплатформенных приложений. С помощью виртуальной Java-машины, которая транслирует приложения в специальный байт-код, программы могут работать на любой компьютерной архитектуре, на которых Java-машина реализована. Для реализации графического интерфейса можно использовать библиотеку Swing [27]. Также для Java есть библиотека Qt Jambi, которая позволяет использовать Qt [28]. К недостаткам Java приложений стоит отнести низкую производительность, по сравнению с приложениями на C++. Синтаксис языка более сложный чем в Python [25].

1.3.1.4 Итог

Писать кроссплатформенные приложения можно на большинстве современных языков. Однако поддержка кроссплатформенности для большинства языков требует использования библиотек. Хорошим решением является C++, однако, данный язык скорее подходит для высоконагруженных систем, где особое внимание уделяется эффективному использованию ресурсов и размеру приложения [29]. Для написания небольших приложений скорее важна скорость разработки и высокая читабельность кода, которую может предоставить Python. К тому же, Python на сегодняшний день является одним из самых популярных языков программирования [26]. У него большое профессиональное сообщество, а Qt официально поддерживает надстройку Qt для Python. Поэтому для разработки приложения был выбран язык Python с библиотекой PyQt5.

1.3.2 Выбор ORM

Для того, чтобы упростить работу с базой данных, можно с помощью ORM (Object-Relational Mapping) объединить ее с концепцией объектно-ориентированных языков программирования [30]. ORM позволяет писать не «чистые» SQL запросы к БД, а посредством кода (например на Python). Также ORM поддерживает различные базы данных, поэтому возможна быстрая смена базы данных при использовании ORM. Так как для реализации приложения был выбран Python, то и ORM будем рассматривать, поддерживающие Python [31].

1.3.2.1 SQLAlchemy

SQLAlchemy - это набор инструментов SQL с открытым исходным кодом и ORM для языка программирования Python. В SQLAlchemy каждой таблице базы данных соответствует класс, каждому полю таблицы базы данных соответствует поле класса. Принцип Unit-of-work делает необходимым ограничить весь код манипулирования базой данных конкретным сеансом базы данных, который контролирует жизненные циклы каждого объекта в этом сеансе [31].

SQLAlchemy реализует паттерн Data Mapper. Работает back-end для баз данных: MySQL, PostgreSQL, SQLite, Oracle и других, между которыми можно переключаться изменением конфигурации.

1.3.2.2 Django ORM

Django ORM поддерживает большинство популярных баз данных в том числе MySQL, PostgreSQL, SQLite, Oracle. Ответом на запрос к базе данных у Django ORM является список объектов заданной модели, с которым удобно работать. Однако Django ORM тесно интегрирован с Django, а SQLAlchemy более сложная ORM, однако которая позволяет писать более производительные

запросы к БД. Самый главный недостаток Django ORM для разрабатываемого проекта, невозможность использовать ORM вне Django [31].

1.3.2.3 SQLAlchemy

В SQLAlchemy концепции базы данных отображаются в Python способом, очень похожим на Django ORM, где таблицы отображаются как классы, строки как экземпляры, а столбцы как атрибуты. Он также предоставляет язык запросов, основанный на объектах Python, который делает SQL более абстрактным, тем самым обеспечивая независимость базы данных для приложений. Из минусов можно выделить, что SQLAlchemy не поддерживает сеансы базы данных [31].

1.3.2.4 Итог

В настоящее время для Python существует большое количество ORM. У рассмотренных ORM достаточно большое сообщество, они обладают широким функционалом и хорошей документацией. Однако, несмотря на удобство использования Django ORM, использование вне Django не представляется возможным, так как проект будет разработана на Python с Qt. По сравнению с SQLAlchemy, Django ORM выделяется возможностью создания сессий. Данная концепция сводит к нулю случайные ошибки, связанные с синхронизацией при фиксации базы данных, поэтому из рассмотренных ORM была выбрана Django ORM.

2 Проектирование и реализация приложения

2.1 Архитектура приложения

Архитектура приложения реализована на основе паттерна проектирования Model-View (рисунок 9). Данный паттерн проектирования является модификацией паттерна Model-View-Controller и рекомендуется к использованию разработчиками Qt [32].

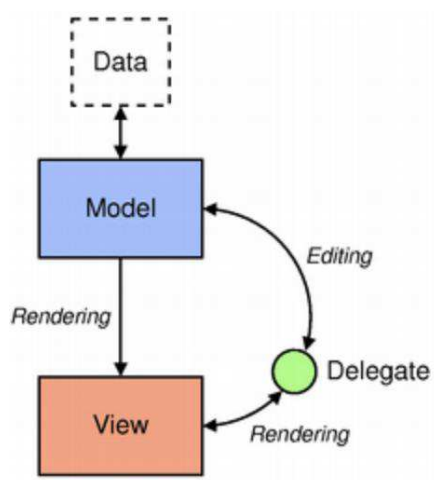


Рисунок 9 – Паттерн Model/View

Модель (Model) предоставляет интерфейс другим компонентам архитектуры к источнику данных. Представление (View) отвечает за отображение данных модели пользователю. Делегат (Delegate) используется во время редактирования (Editing), сообщает модели и представлению о состоянии редактора. [32].

2.2 Модульная структура

На рисунке 10 отображена структура приложения.

В приложении были выделены следующие модули:

- дисциплины;

- студенты;
- успеваемость;
- расписание.

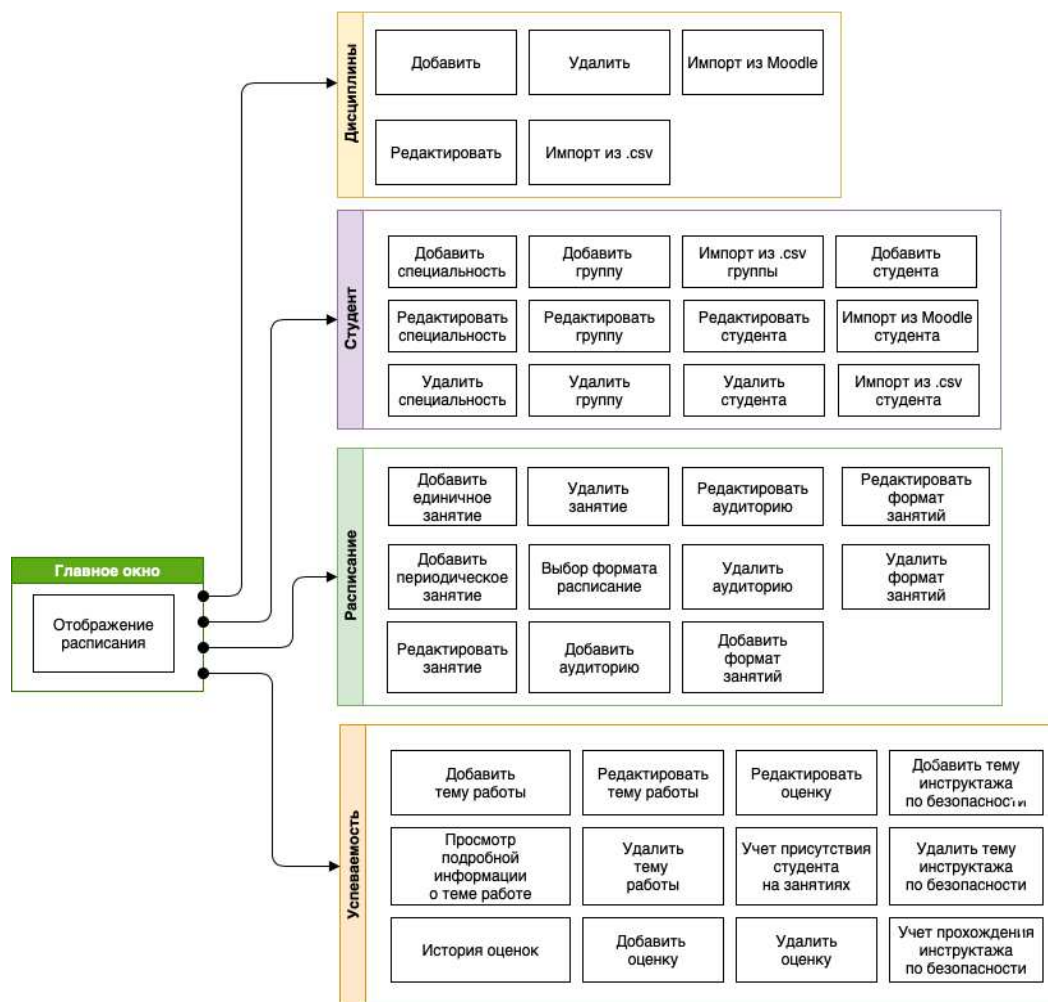


Рисунок 10 – Модульная структура приложения

2.3 Структура базы данных

В приложении используется встраиваемая СУБД – SQLite. БД SQLite является библиотекой, которая реализует ядро базы данных. БД SQLite - это встроенный движок базы данных SQL. В отличие от большинства других баз данных SQL, SQLite не имеет отдельного серверного процесса. SQLite читает и

пишет напрямую в обычные файлы на диске. Полная база данных SQL с несколькими таблицами, индексами, триггерами и представлениями содержится в одном файле на диске. Формат файла базы данных является кроссплатформенным [33]. Такой подход уменьшает накладные расходы, время отклика и упрощает программу.

После анализа предметной области были выделены следующие сущности:

- дисциплина – содержит информацию о дисциплине;
- специальность – содержит информацию о специальности;
- группа – содержит информацию об учебной группе. Группа создается по какой-либо специальности;

- студент – содержит информацию о студенте. Каждый студент обязательно должен состоять в группе. Для каждого студента помимо ФИО можно указать номер зачетки и написать заметку;

- задание – содержит информацию о задании для работы студентов. Каждое задание выдается для определенной группы и по определенной дисциплине. Также для каждой задачи можно выбрать шкалу оценивания и указать крайний срок сдачи;

- оценка – содержит информацию о полученной оценке студента. Оценка ставится определенному студенту по определенной работе. По каждой оценке можно написать отзыв и оставить заметку;

- история оценок – содержит информацию об изменении оценок, в том числе дату и тип изменения (добавление, редактирование, удаление оценки);

- шкала оценок – содержит информацию о максимальном и минимальном пороге оценки. Используется для выбора шкалы оценивания;

- формат занятий – содержит информацию о форматах проведения занятий. Сюда добавляются различные виды занятий, например, лекция, практика, лабораторная работа и прочее, которые в последующем используются при составлении расписания;

- день занятий – содержит информацию об учебных днях студентов. Используется для создания занятия;

- аудитория – содержит информацию об аудиториях. Отдельно создаются аудитории, которые содержат корпус и номер аудитории. В последующем используется при составлении расписания;

- занятие – содержит информацию о занятиях в расписании. Из таких занятий составляется расписание. Здесь хранится время, день, дисциплина, тип занятия, группа и аудитория;

- посещение – содержит информацию о посещении занятий студентом. Для каждого студента можно отметить посещение им какого-либо занятия;

- тема инструктажа техники безопасности – содержит информацию о темах инструктажей по технике безопасности студентов. Для каждой группы по каждой дисциплине можно назначать свои темы инструктажей;

- прохождение инструктажа по технике безопасности – содержит информацию о прохождении студентом инструктажа по технике безопасности. Здесь содержатся тема инструктажа, студент, который его прошел и дата прохождения.

Диаграмма сущностей и связей базы данных представлена на рисунке 11. На ней видно, как сущности связаны между собой.

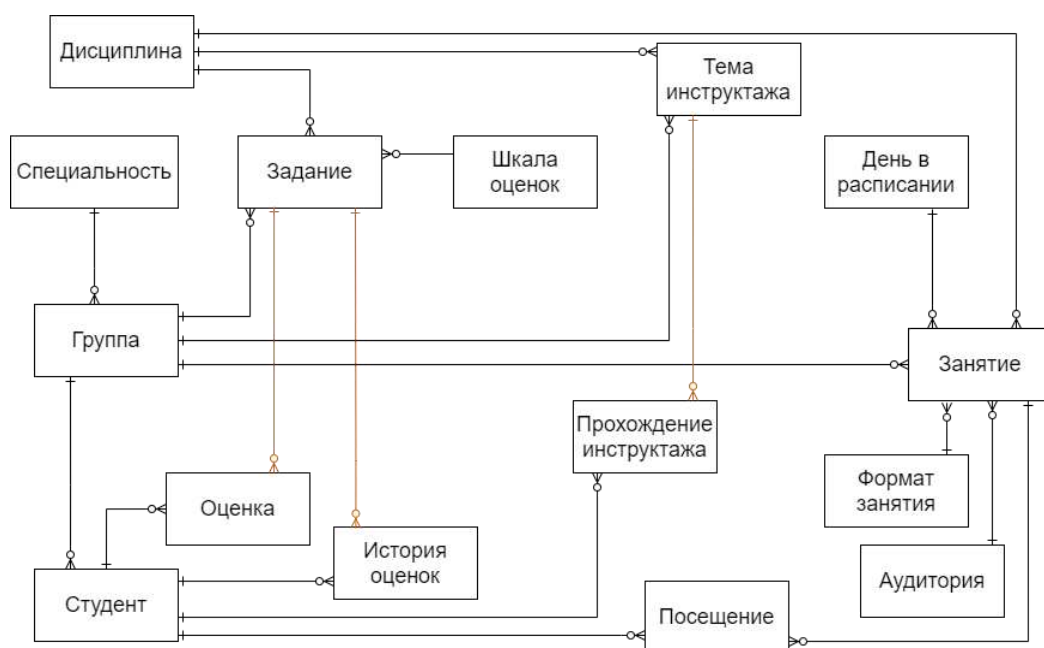


Рисунок 11 – Диаграмма сущностей и связей

На основе диаграммы сущностей и связей разработана логическая модель базы данных, которая изображена на рисунке 12. На ней отражено какую информацию должна хранить каждая сущность.

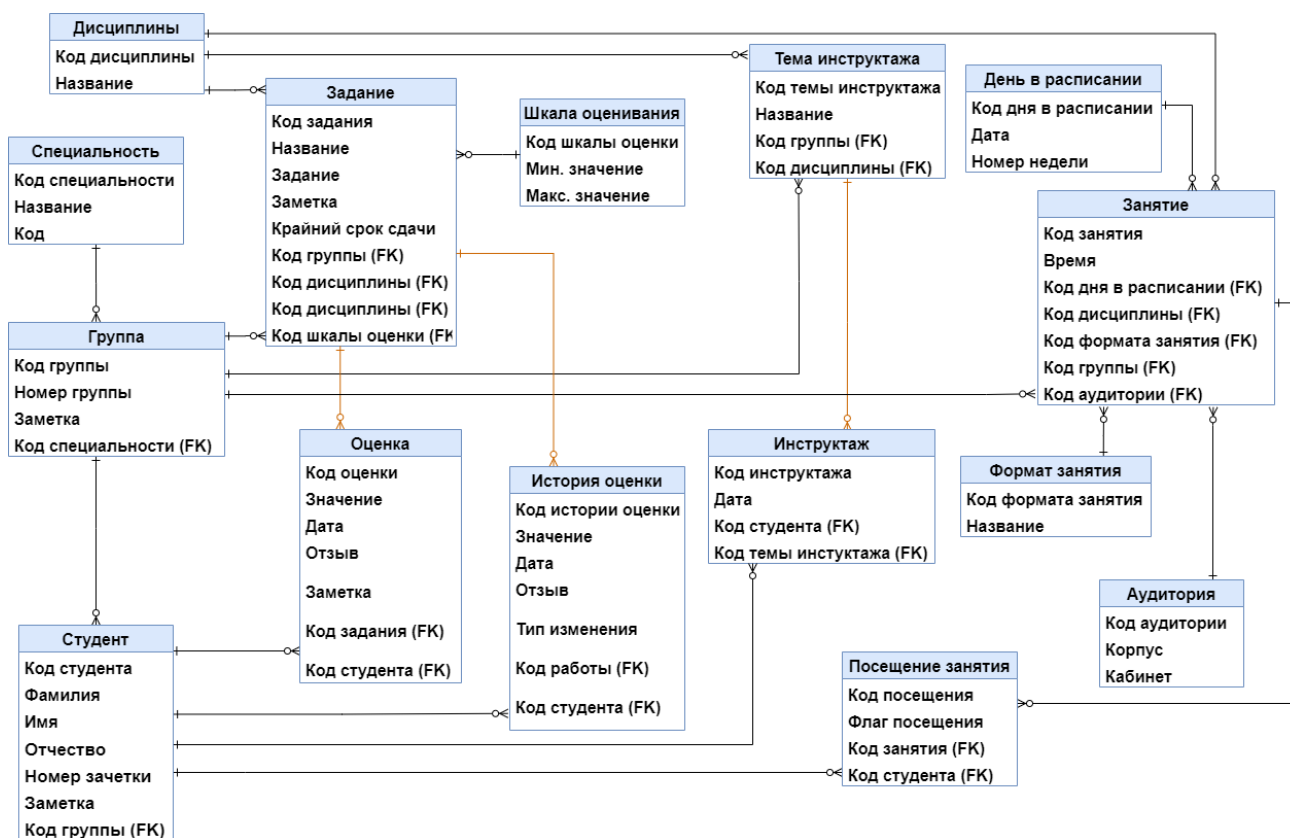


Рисунок 12 – Логическая модель базы данных

Далее разработана физическая модель базы данных, которая изображена на рисунке 13.

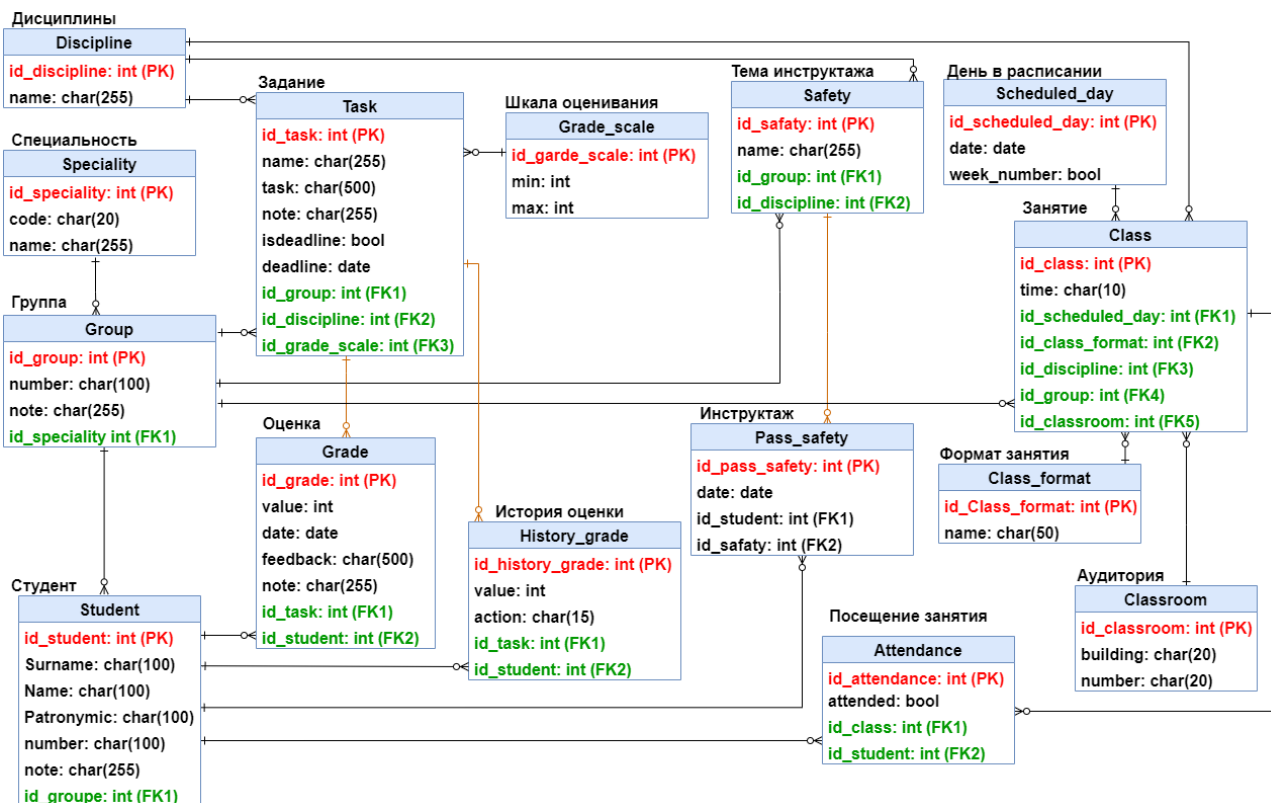


Рисунок 13 – Физическая модель базы данных

Ниже в таблицах 2-16 представлено подробное описание всех сущностей базы данных с названием полей и их назначением.

Таблица 2 – Описание Discipline (дисциплина)

Поле	Тип	Назначение
id_discipline	int	первичный ключ
name	char (255)	наименование дисциплины

Таблица 3 – Описание Speciality (специальность)

Поле	Тип	Назначение
id_speciality	int	первичный ключ
code	char (20)	код специальности
name	char (255)	наименование

Таблица 4 – Описание Group (группа)

Поле	Тип	Назначение
id_group	int	первичный ключ
number	char (100)	номер группы
note	char (255)	заметка
id_specialty	int	внешний ключ к таблице Specialty

Таблица 5 – Описание Student (студент)

Поле	Тип	Назначение
id_student	int	первичный ключ
Surname	char (100)	фамилия
Name	char (100)	имя
Patronymic	char (100)	отчество
number	char (100)	номер зачетной книжки
note	char (255)	заметка
id_group	int	внешний ключ к таблице Group

Таблица 6 – Описание GradeScale (шкала оценок)

Поле	Тип	Назначение
id_grade_scale	int	первичный ключ
min	int	минимальное значение оценки
max	int	максимальное значение оценки

Таблица 7 – Описание Task (задание)

Поле	Тип	Назначение
id_task	int	первичный ключ
name	char (255)	название работы
task	string (500)	задание
note	char (255)	заметка

Окончание таблицы 7

Поле	Тип	Назначение
isdeadline	bool	флаг установки крайнего срока
deadline	date	дата крайнего срока сдачи
id_group	int	внешний ключ к таблице Group
id_discipline	int	внешний ключ к таблице Discipline
id_grade_scale	int	внешний ключ к таблице GradeScale

Таблица 8 – Описание Grade (оценка)

Поле	Тип	Назначение
id_grade	int	первичный ключ
value	int	значение оценки
date	date	дата выставления
feedback	string (500)	отзыв
note	char (255)	заметка
id_work	int	внешний ключ к таблице Work
id_student	int	внешний ключ к таблице Student

Таблица 9 – Описание HistoryGrade (история оценок)

Поле	Тип	Назначение
id_history_grade	int	первичный ключ
value	int	значение оценки
date	date	дата изменения
action	char (15)	тип изменения
id_work	int	внешний ключ к таблице Work
id_student	int	внешний ключ к таблице Student

Таблица 10 – Описание Safety (тема инструктажа)

Поле	Тип	Назначение
id_safety	int	первичный ключ
name	char (255)	наименование темы
id_group	int	внешний ключ к таблице Group
id_discipline	int	внешний ключ к таблице Discipline

Таблица 11 – Описание PassSafety (инструктаж)

Поле	Тип	Назначение
id_pass_safety	int	первичный ключ
date	date	дата прохождения
id_student	int	внешний ключ к таблице Student
id_safety	int	внешний ключ к таблице Safety

Таблица 12 – Описание Classroom (аудитория)

Поле	Тип	Назначение
id_audience	int	первичный ключ
building	char (20)	корпус
number	char (20)	номер аудитории

Таблица 13 – Описание ClassFormat (формат занятий)

Поле	Тип	Назначение
id_class_format	int	первичный ключ
name	char (50)	наименование

Таблица 14 – Описание ScheduledDay (день в расписании)

Поле	Тип	Назначение
id_scheduled_day	int	первичный ключ
date	date	дата
week_number	bool	номер недели

Таблица 15 – Описание Class (занятие)

Поле	Тип	Назначение
id_class	int	первичный ключ
time	char (10)	время
id_scheduled_day	int	внешний ключ к таблице ScheduledDay
id_discipline	int	внешний ключ к таблице Discipline
id_class_format	int	внешний ключ к таблице ClassFormat
id_group	int	внешний ключ к таблице Group
id_audience	int	внешний ключ к таблице Audience

Таблица 16 – Описание Attendance (посещение)

Поле	Тип	Назначение
id_attendance	int	первичный ключ
attended	bool	флаг посещения
id_class	int	внешний ключ к таблице Class
id_student	int	внешний ключ к таблице Student

2.4 Диаграмма классов

Для каждого окна приложения есть свой класс, методы класса (функции) вызываются при нажатии соответствующей кнопки на этом окне. Также для каждой таблицы базы данных есть свой класс, где функции – это методы для работы с базой данных. На рисунке 14 изображена диаграмма классов для работы с дисциплинами, взаимодействие других сущностей происходит по такому же принципу.

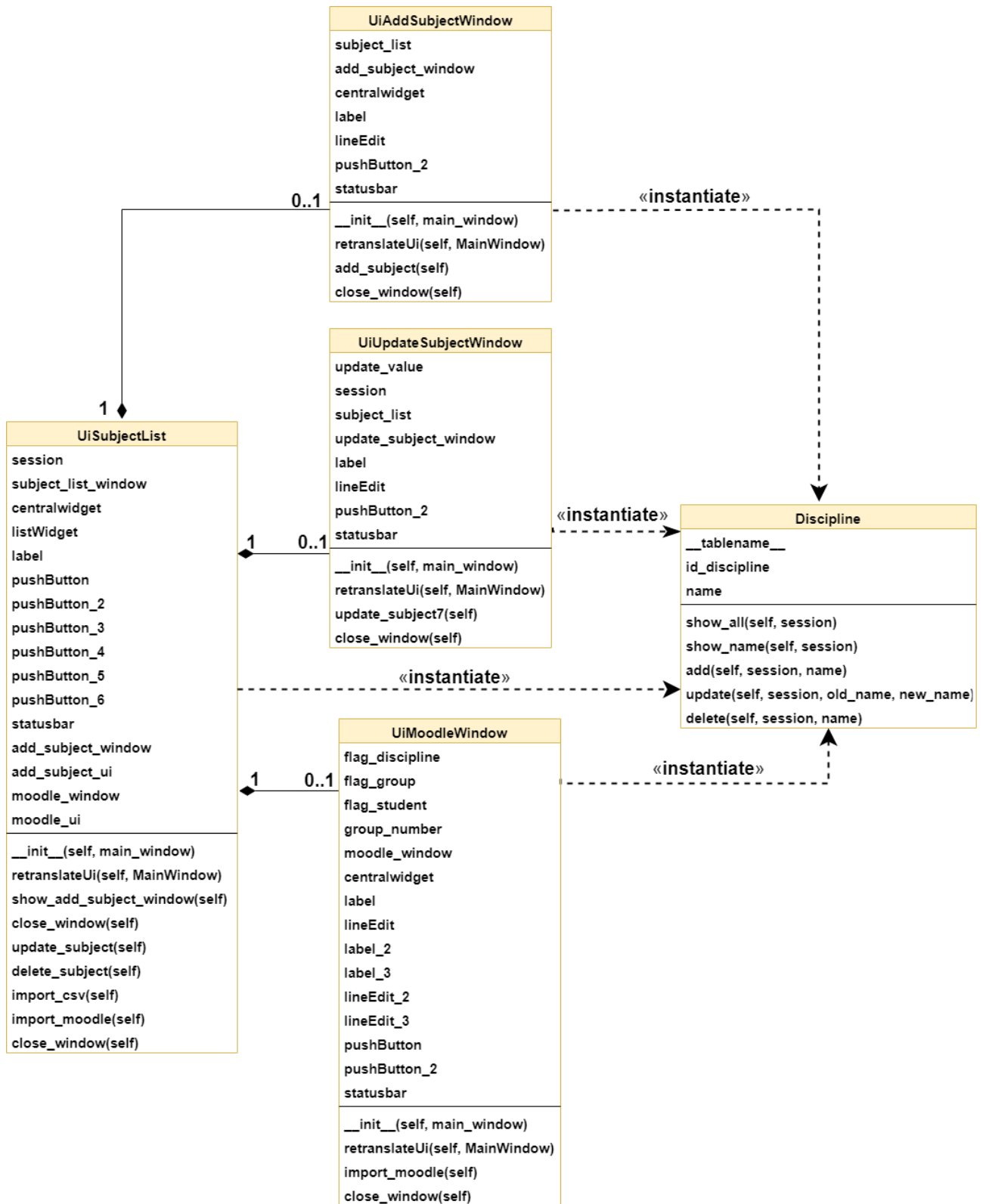


Рисунок 14 – Диаграмма классов для работы с дисциплинами

Класс UiSubjectList содержит графические элементы окна «Дисциплины» и функции для взаимодействия с интерфейсом. В функции `__init__(self,`

main_window) задается оформление окна, а также назначаются функции, которые будут вызываться по нажатию какой-либо кнопки, также в этой функции мы выводим список дисциплин, который получаем из базы данных с помощью класса Discipline. В функции retranslateUi(self, MainWindow) задаются названия кнопок и заголовков окна. Функция show_add_subject_window вызывается при нажатии на кнопку «Добавить» и открывает окно UiAddSubjectWindow для добавления новой дисциплины. Функция update_subject(self) вызывается при нажатии на кнопку «Редактировать» и открывает окно UiUpdateSubjectWindow для редактирования дисциплины. При нажатии на кнопку «Импорт из CSV» вызывается функция import_csv(self), в которой реализован импорт дисциплин из файлов формата CSV. Функция import_moodle(self) вызывается при нажатии на кнопку «Импортировать из Moodle», данная функция открывает окно для импорта из системы Moodle. Функция close_window(self) вызывается для закрытия окна.

Класс UiAddSubjectWindow содержит графические элементы окна «Добавить дисциплину» и функции для взаимодействия с интерфейсом. Функции __init__(self, main_window) и retranslateUi(self, MainWindow) аналогично классу UiSubjectList, за исключением вывода списка дисциплин. Функция add_subject(self) с помощью класса Discipline добавляет данные, которые мы ввели в этом окне в базу данных.

Класс UiUpdateSubjectWindow аналогично классу UiAddSubjectWindow, только здесь вместо функции добавления функция update_subject7(self), которая обновляет данные в базе данных.

Класс UiMoodleWindow содержит графические элементы окна «Импорт из Moodle». Функции __init__(self, main_window) и retranslateUi(self, MainWindow) аналогично классу UiSubjectList, за исключением вывода списка дисциплин. С помощью метода import_moodle(self) происходит импорт данных из системы Moodle. Для этого необходимо ввести данные личного кабинета в соответствующих полях данного окна и нажать на кнопку «Импортировать».

Атрибуты классов `UiSubjectList`, `UiAddSubjectWindow`, `UiUpdateSubjectWindow`, `UiMoodleWindow` содержат в себе элементы интерфейса: кнопки, поля для ввода текста, заголовки, а также ссылки на другие окна (если они их вызывают).

Класс `Discipline` содержит методы для работы с таблицей `Discipline` в базе данных. Имя таблицы задается в переменной `__tablename__`, код дисциплины в переменной `id_discipline`, название дисциплины в переменной `name`. Реализованы следующие функции: `show_all(self, session)` – получение списка всех дисциплин из базы данных, `show_name(self, session)` – получение списка названий дисциплин, `add(self, session, name)` – добавление новой дисциплины в базу данных, `update(self, session, old_name, new_name)` – изменение дисциплины (названия) в базе данных, `delete(self, session, name)` – удаление дисциплины из базы данных.

Функция `close_window(self)` одинаковая для всех классов, где она присутствует, с помощью нее можно закрыть текущее окно.

2.5 Примеры интерфейса приложения

На рисунках 15-19 изображен интерфейс приложения.

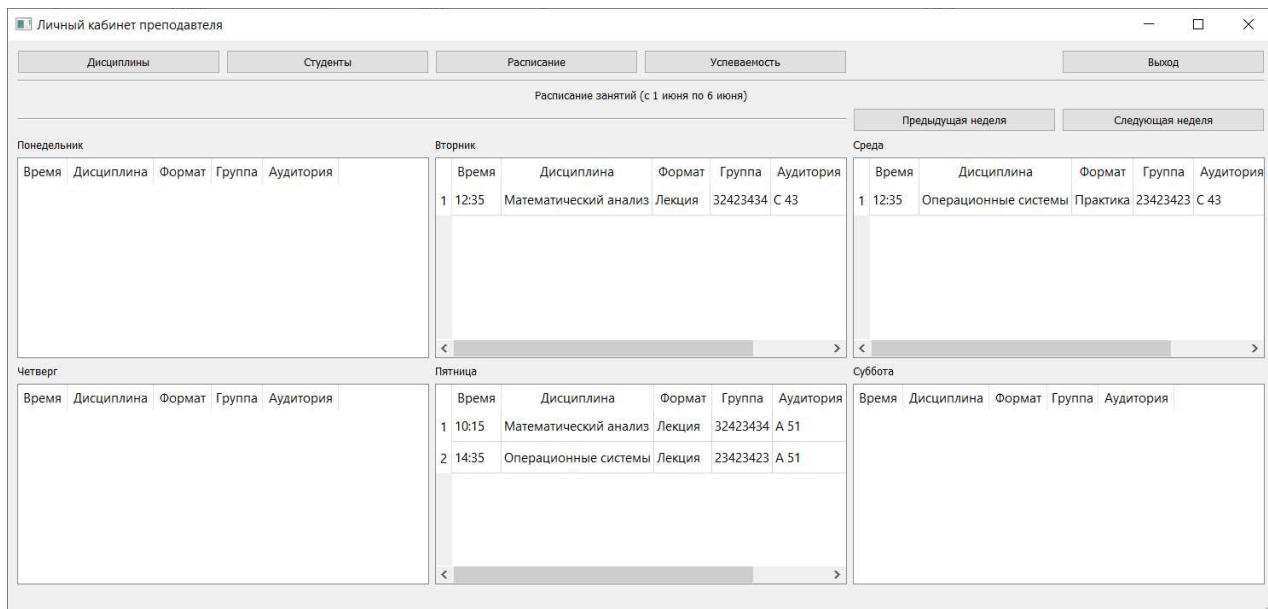


Рисунок 15 – Главное окно приложения

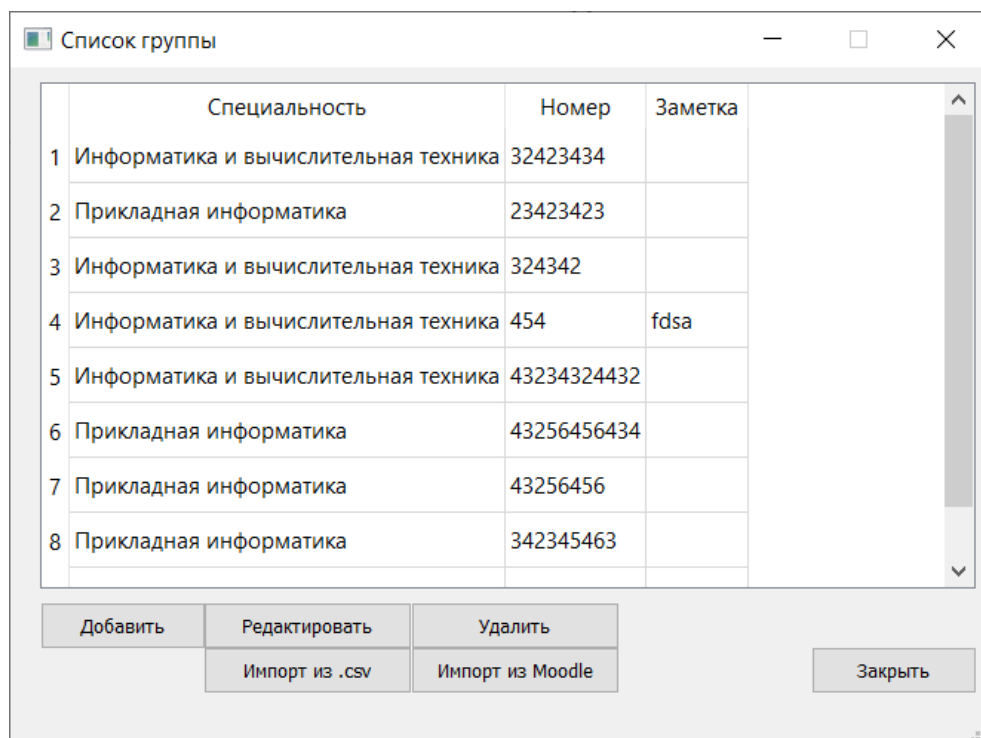


Рисунок 16 – Окно просмотра учебных групп преподавателя

Оценить

Студент: Франк Антон Игоревич

Дата: 05.06.2020

Работа: Лабораторная работа №1

Оценка (от 0 до 100):

Заметка:

Отзыв:

Прикрепить работу

Добавить

Рисунок 17 – Окно добавления оценки за работу студенту

История изменения оценок

	ФИО студента	Название работы	Тип изменения	Дата изменения	Оценка
1	Андреев Федор Онисимович	Лабораторная работа №1	Добавление	2020-08-05	50
2	Андреев Федор Онисимович	Лабораторная работа №1	Редактирование	2020-08-05	60
3	Борисов Антон Платонович	Лабораторная работа №3	Добавление	2020-08-05	5
4	Борисов Антон Платонович	Лабораторная работа №3	Удаление	2020-08-05	5
5	Франк Антон Игоревич	Лабораторная работа №3	Добавление	2020-05-08	9
6	Борисов Антон Платонович	Лабораторная работа №1	Добавление	2020-06-02	75

Заккрыть

Рисунок 18 – Окно просмотра истории изменения оценок

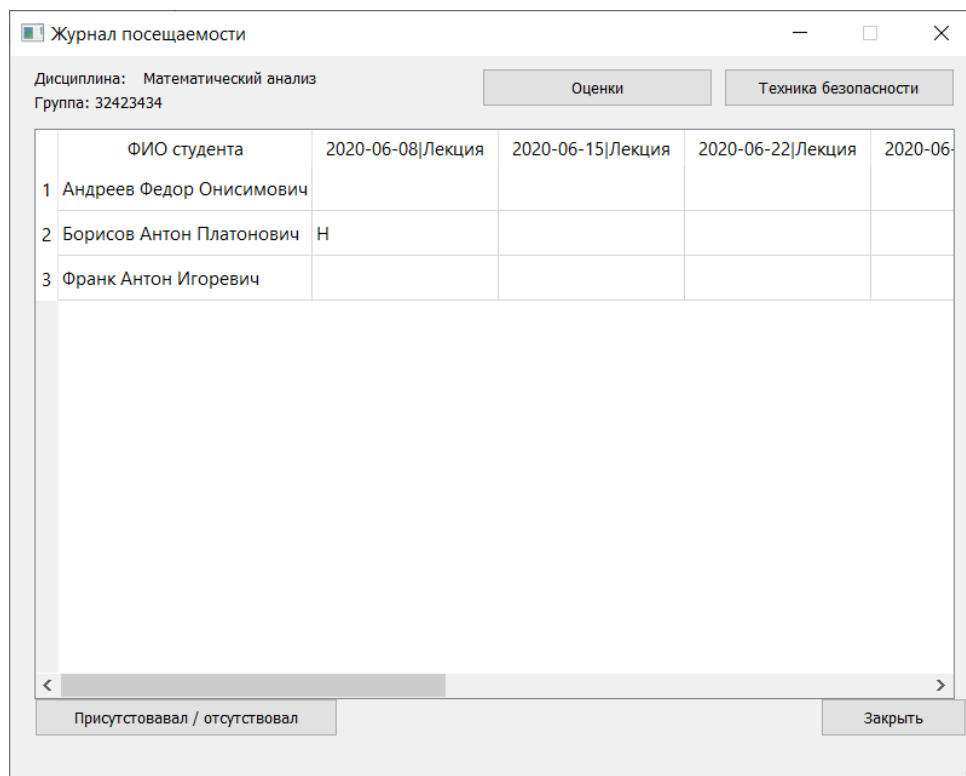


Рисунок 19 – Окно «Журнал посещаемости»

3 Инструкции

3.1 Инструкция пользователя

При открытии программы открывается главное окно программы, которое изображено ранее на рисунке 15. Для начала работы с приложением необходимо добавить дисциплины, группы, специальности и студентов.

Для добавления дисциплин необходимо нажать на кнопку «Дисциплины», затем на кнопку «Добавить», после ввести данные и сохранить их. Окно со списком дисциплин изображено на рисунке 20.

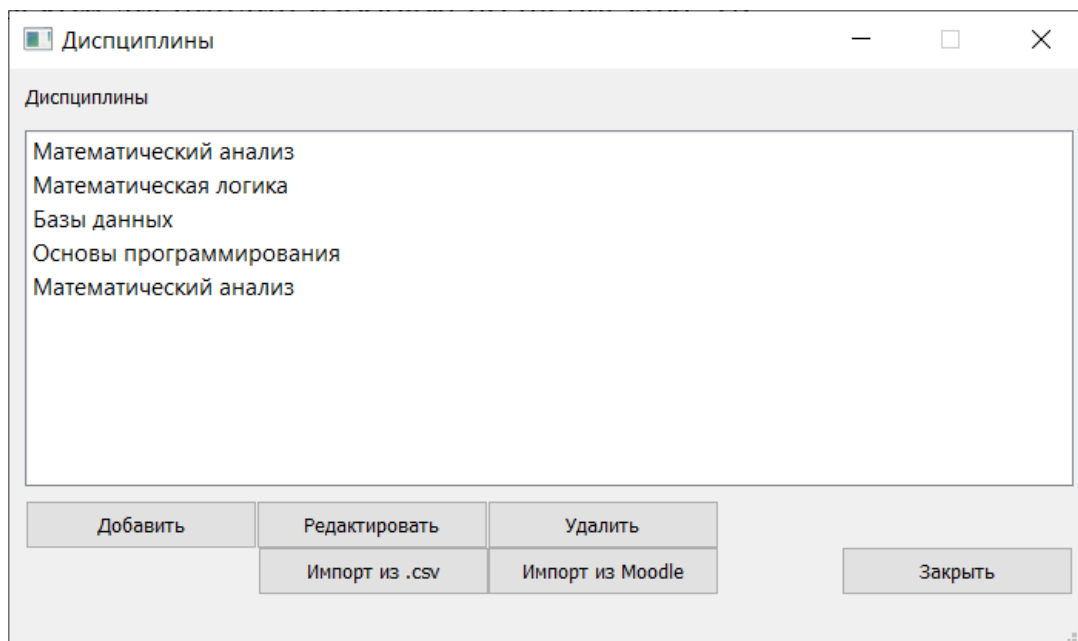


Рисунок 20 – Окно «Дисциплины»

Для того, чтобы добавить специальность, необходимо в главном окне программы нажать на кнопку «Студенты», далее кнопку «Специальности» и нажать на кнопку «Добавить». Окно со списком специальностей изображено на рисунке 21.

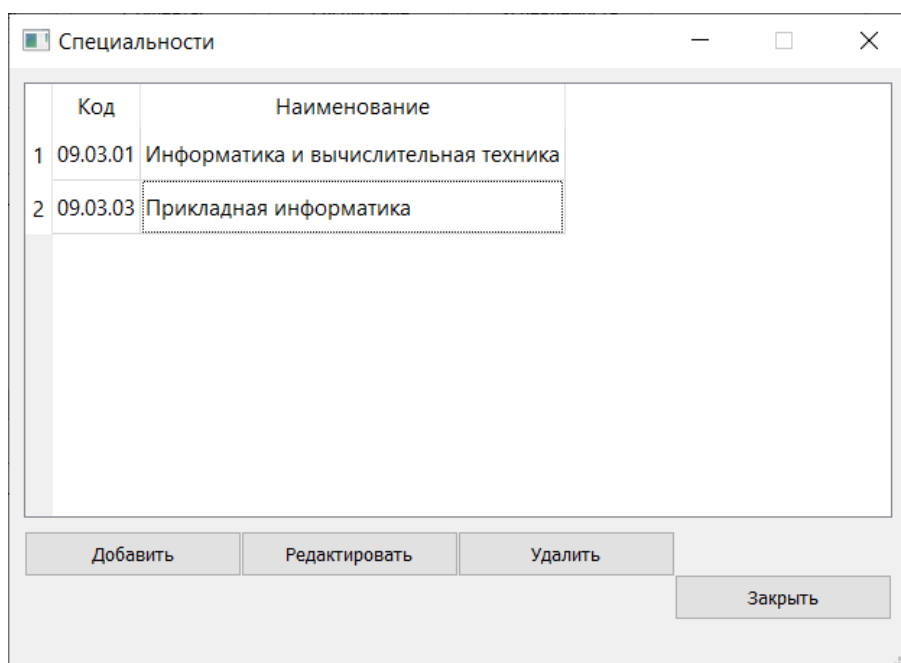


Рисунок 21 – Окно «Специальности»

Чтобы добавить группу, необходимо в главном окне программы нажать на кнопку «Студенты», далее «Группы» и нажать на кнопку «Добавить». Окно со списком групп изображено ранее на рисунке 16.

Чтобы добавить студентов, необходимо в главном окне программы нажать на кнопку «Студенты», выбрать номер учебной группы и нажать на кнопку «Просмотреть список группы», в открывшемся окне нажать на кнопку «Добавить». Данные также можно не только вводить в ручном режиме, а импортировать из CSV или LMS Moodle нажав на соответствующую кнопку в окне просмотра данных. Окно со списком студентов в группе изображено на рисунке 22.

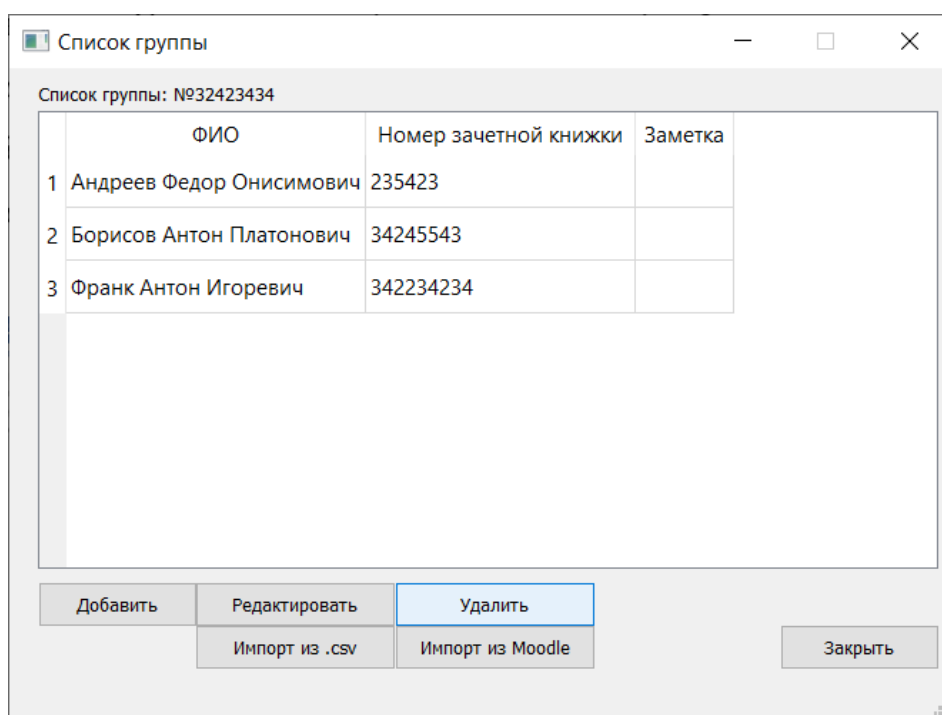


Рисунок 22 – Окно «Список группы»

Также в окнах просмотра данных: «Список дисциплин», «Список специальностей», «Список групп», «Список студентов» возможно отредактировать информацию или стереть, нажав соответственно на кнопку «Редактировать» или «Удалить».

Для работы с расписанием необходимо нажать на кнопку «Расписание» из главного окна приложения. Для добавления/редактирования/удаления информации об аудиториях, форматах занятий, формата планирования занятий необходимо выбрать раздел «Расписание» и нажать на кнопку «Параметры». Для того, чтобы добавить разовое занятие, необходимо нажать на кнопку «Добавить единичное занятие», для того чтобы добавить периодическое занятие необходимо нажать на кнопку «Добавить периодическое занятие». Для редактирования/удаления информации в расписании необходимо нажать на соответствующий пункт меню. Для навигации по расписанию (по неделям) необходимо использовать кнопки «Предыдущая неделя» и «Следующая неделя». Окно «Расписание» изображено на рисунке 23.

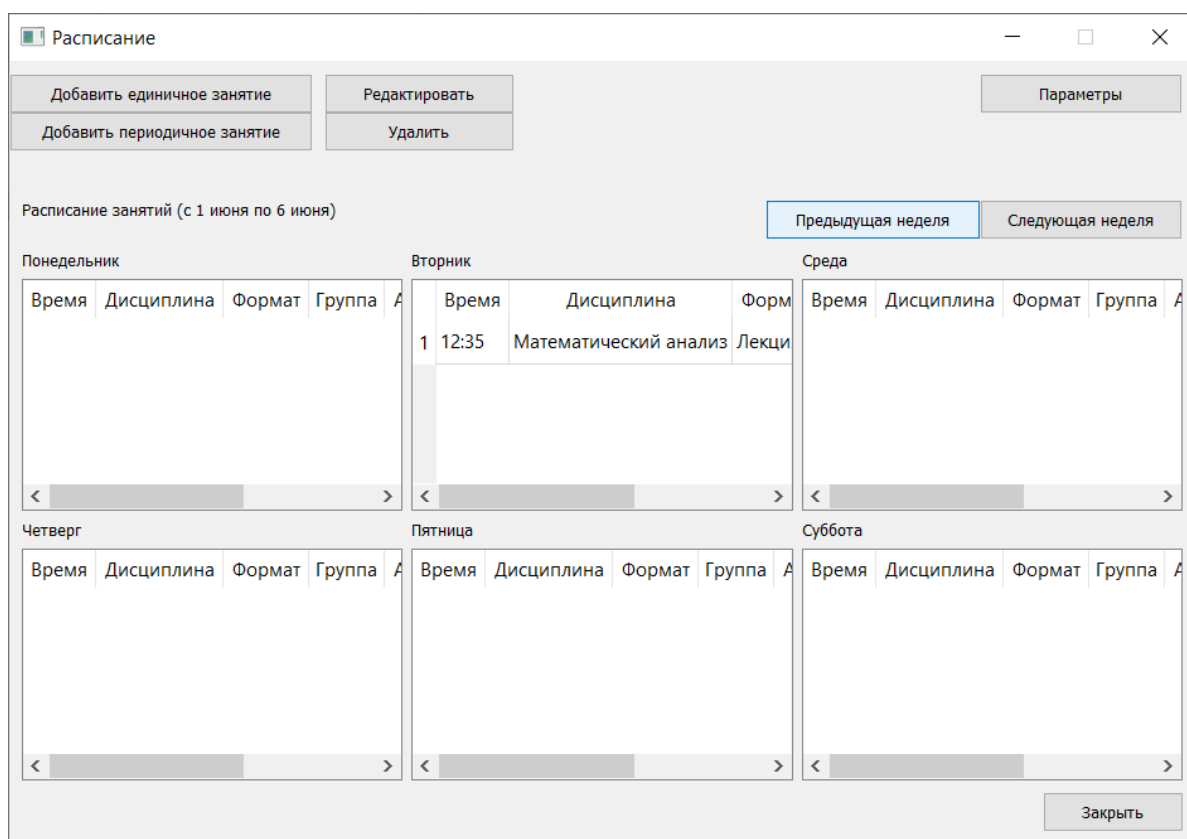


Рисунок 23 – Окно «Расписание»

Для работы с успеваемостью студента необходимо нажать на кнопку «Успеваемость» из главного окна приложения. Для навигации по журналам

оценок, посещаемости и техники безопасности необходимо в открывшемся окне выбрать дисциплину и группу и нажать соответствующую кнопку (рисунок 24).

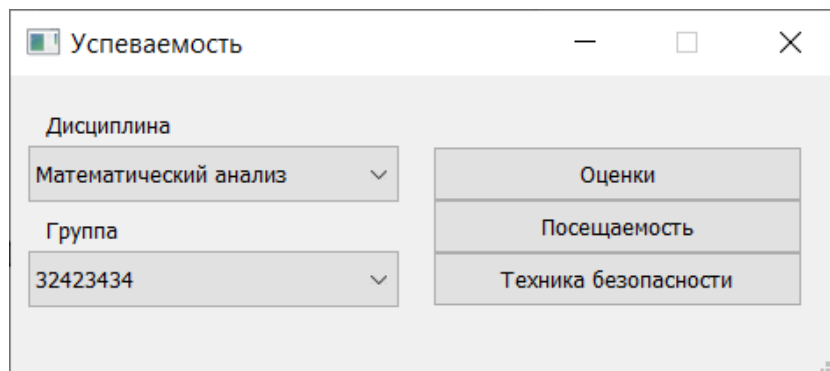


Рисунок 24 – Окно навигации по журналам оценок, посещаемости и техники безопасности

Для работы с журналом оценки необходимо добавить работу, нажав на кнопку «Добавить работу» (рисунок 25). Также возможно посмотреть подробную информацию о работе в том числе задание, заметку, установленный срок выполнения, нажав на кнопку «Подробнее о работе». Для оценивания студента необходимо выбрать работу и студента и нажать на кнопку «Добавить оценку» (рисунок 26). В данном окне возможно прикрепить работу, нажав на кнопку «Прикрепить работу». Для того чтобы редактировать/удалить, а также посмотреть подробную информацию об оценке, необходимо нажать на кнопку «Подробнее об оценке». Для того чтобы просмотреть историю изменения оценок необходимо нажать на кнопку «История оценок» (рисунок 18).

Добавить работу

Название работы

Шкала оценок

0-100

Срок выполнения

06.06.2020

Задание

Заметка

Добавить

Рисунок 25 – Окно «Добавить работу»

Оценить

Студент

Франк Антон Игоревич

Дата

06.06.2020

Работа

Лабораторная работа №1

Оценка (от 0 до 100)

Заметка

Отзыв

Прикрепить работу

Добавить

Рисунок 26 – Окно «Оценить»

Для работы с журналом посещаемости необходимо в окне навигации по журналам оценок, посещаемости и техники безопасности нажать на кнопку

«Посещаемость». В данном окне необходимо выбрать студента и занятие и нажать на кнопку «Присутствовал / отсутствовал» (рисунок 27).

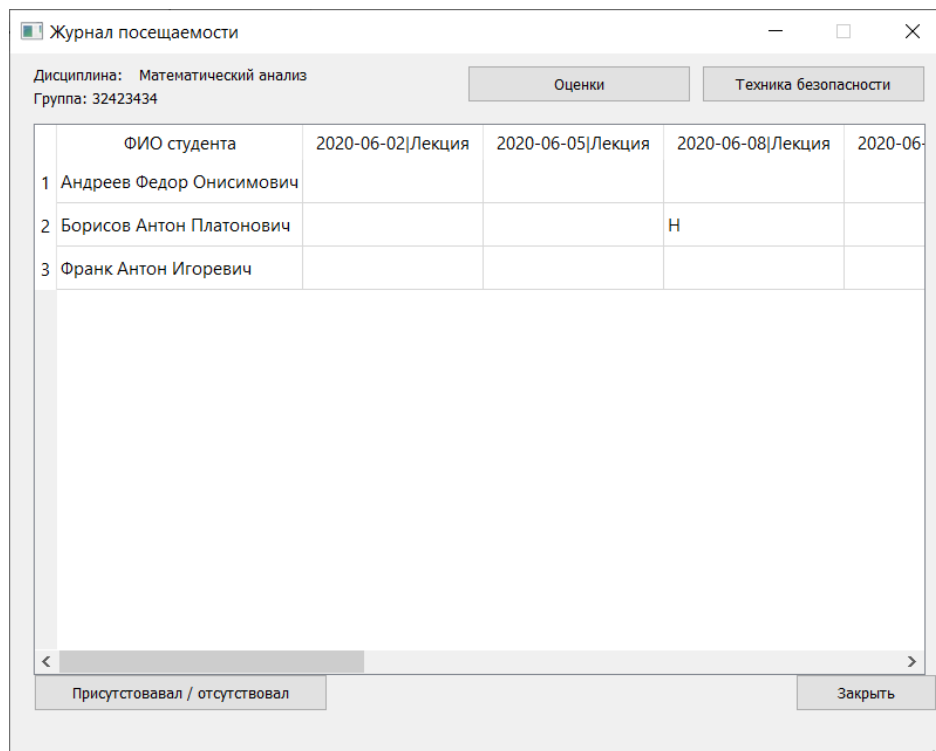


Рисунок 27 – Окно «Посещаемость»

Для работы с журналом прохождения техники безопасности необходимо в окне навигации по журналам оценок, посещаемости и техники безопасности нажать на кнопку «Техника безопасности». В открывавшемся окне для начала работы необходимо добавить тему инструктажа нажав на кнопку «Добавить тему инструктажа». Для указания, что студент прошел инструктаж необходимо выбрать тему инструктажа и студента и нажать на кнопку «Добавить». Для редактирования/удаления информации необходимо нажать на соответствующую кнопку (рисунок 28).

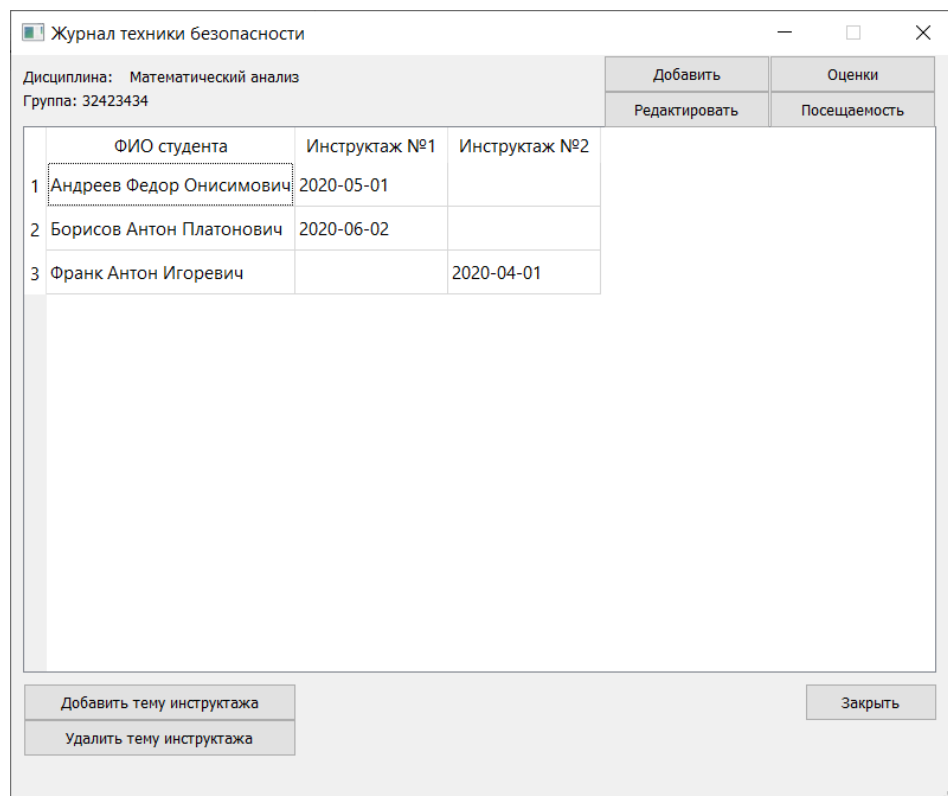


Рисунок 28 – Окно «Добавить оценку»

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

3.2 Инструкция разработчика

Для работы с исходным кодом необходимо программное обеспечение:

- PyCharm 2020.1 [34];
- Python 3.8 [35];
- Qt Creator 5.14.2 [36].

А также установленный набор библиотек:

- certifi 2020.4.5.1;
- chardet 3.0.4;
- configparser 5.0.0;
- crayons 0.3.0;
- idna 2.9;

- isort 4.3.21;
- PyQt5 5.14.2;
- PyQt5-sip 12.7.2;
- PyQt5-stubs 5.14.2.0;
- requests 2.23.0;
- selenium 3.141.0;
- shiboken2 5.14.1;
- SQLAlchemy 1.3.13;
- urllib3 1.25.9;
- webdriver-manager 2.5.3.

Для установки необходимо в PyCharm нажать на пункт меню File -> Settings. В открывшемся окне выбрать соответствующий пункт меню и добавить пакеты (рисунок 29). Также установить пакеты можно, установив их из файла requirements.txt через терминал.

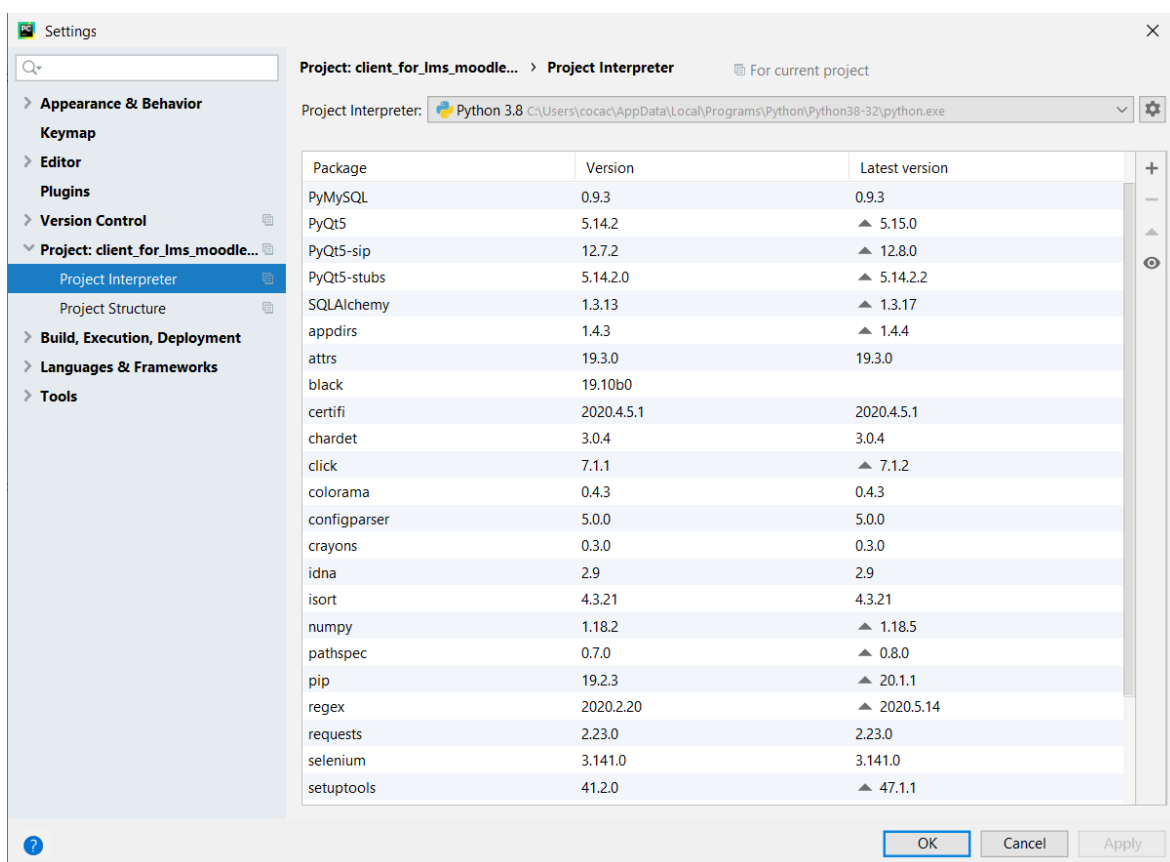


Рисунок 29 – Окно настроек проекта

На рисунке 30 изображен каталог файлов проекта.

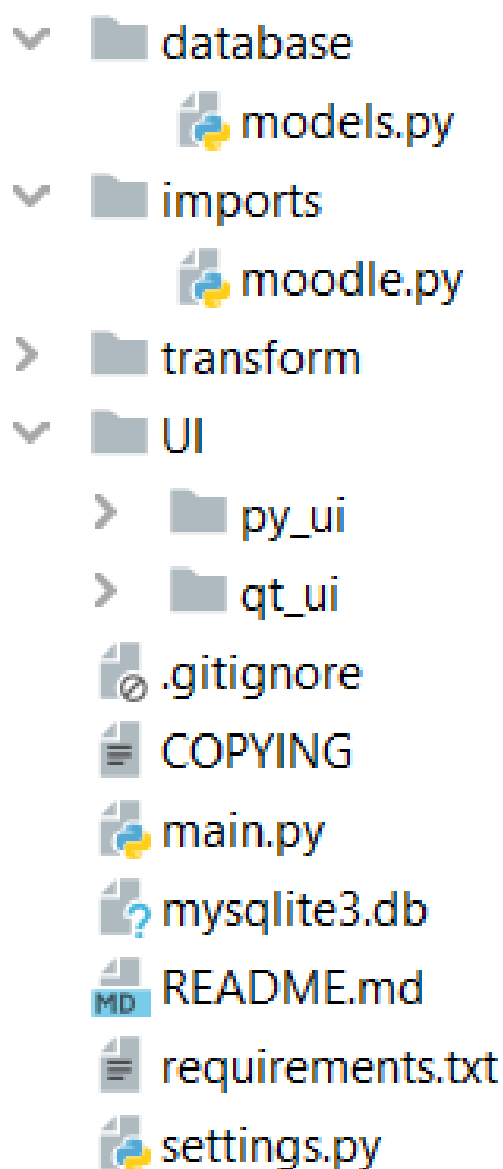


Рисунок 30 – Каталог файлов проекта

В директории database располагается файл models.py, в нем происходят все взаимодействия с базой данных. Каждый класс соответствует таблице в базе данных, а каждое поле класса соответствует полю в таблице базы данных. В каждом классе имеются функции для работы с базой данных, например, в классе Discipline, есть функция add, которая получает имя в качестве аргумента и добавляет дисциплину в базу данных.

В директории `imports` находится файл `moodle.py`, в котором находится реализация импорта данных из системы LMS Moodle.

В директории `UI/qt_ui` находятся файлы с интерфейсом составленные в Qt Creator. Для внесения изменений в графический интерфейс приложения необходимо открыть в Qt Creator одноименный файл окна из этой директории. После внесения изменений необходимо сохранить файл, а далее его 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.

Конвертированные файлы с интерфейсом находятся в директории `UI/py_ui`. Каждый файл — это окно приложения, также там описаны функции, которые обрабатывают нажатия кнопок в этом окне.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы было спроектировано и реализовано приложение — электронный журнал преподавателя высшего учебного заведения.

В разработанном приложении были реализованы следующие функции:

- добавление/редактирование/удаление дисциплин, специальностей, учебных групп, студентов, аудиторий, форматов занятий, единичных и периодических занятий, работ студентов, оценок за работы студентов;
- импорт данных в машиночитаемом формате (.csv);
- импорт данных из LMS Moodle;
- просмотр истории изменения оценок;
- ведение журнала посещаемости;
- ведение журнала прохождения инструктажей по технике безопасности.

Приложение можно улучшить путем добавления возможностей:

- оповещения о заданных событиях;
- экспорт данных в машиночитаемых форматах;
- создание отчетов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Научно-образовательный центр «Электронный университет» [Электронный ресурс]. – Режим доступа: <http://eun.bmstu.ru> (дата обращения: 27.02.2020).
2. Приложения в Google Play – Журнал преподавателя [Электронный ресурс]. – Режим доступа: <https://play.google.com/store/apps/details?id=com.drprog.sjournal&hl=ru> (дата обращения: 27.02.2020).
3. Журнал преподавателя на Android (SJournal) [Электронный ресурс]. – Режим доступа: <https://www.youtube.com/watch?v=-w-cBqfNT-I> (дата обращения: 11.03.2020).
4. Standards-based Learning and Student Information System - Common Goal Systems, Inc [Электронный ресурс]. – Режим доступа: <https://www.teacherease.com> (дата обращения: 27.02.2020).
5. Teacherease: Gredebook Setup [Электронный ресурс]. – Режим доступа: <https://www.teacherease.com/instructors/SetupGradebook.aspx?> (дата обращения: 11.03.2020).
6. OpenGrade [Электронный ресурс]. – Режим доступа: <http://www.lightandmatter.com/ogr/ogr.html> (дата обращения: 27.02.2020).
7. Additio App | Cuaderno de notas para el profesora [Электронный ресурс]. – Режим доступа: <https://www.additioapp.com/en/> (дата обращения: 27.02.2020).
8. Google Класс [Электронный ресурс]. – Режим доступа: <https://chrome.google.com/webstore/detail/google-classroom/mfhehppjhmmlfbborchdfldgimhfhk?> (дата обращения: 08.03.2020).
9. Приложения в Google Play – Teacher Gradebook – Additio [Электронный ресурс]. – Режим доступа: <https://play.google.com/store/apps/details?id=com.additioapp.additio&hl=ru> (дата обращения: 10.03.2020).
10. Features | Additio App [Электронный ресурс]. – Режим доступа: <https://www.additioapp.com/en/features/> (дата обращения: 11.03.2020).

11. GNota in Launchpad [Электронный ресурс]. – Режим доступа: <https://launchpad.net/gnota> (дата обращения: 27.02.2020).
12. Tuttle SVC: December 2007 [Электронный ресурс]. – Режим доступа: <http://www.tuttlesvc.org/2007/12/> (дата обращения: 11.03.2020).
13. Teacher Gradebook; Educational innovation project: digital gradebook. [Электронный ресурс]. – Режим доступа: <http://apps4edu.org/> (дата обращения: 08.03.2020).
14. Приложение в Google Play - Учительский журнал FREE [Электронный ресурс]. – Режим доступа: <https://play.google.com/store/apps/details?id=com.apolosoft.cuadernoprofesor&hl=ru> (дата обращения: 27.02.2020).
15. ClassDojo [Электронный ресурс]. – Режим доступа: <https://www.classdojo.com/ru-ru/> (дата обращения: 09.03.2020).
16. Kahoot! | Learning Games | Make Learning Awesome! [Электронный ресурс]. – Режим доступа: <https://kahoot.com> (дата обращения: 09.03.2020).
17. Moodle - Open-source learning platform | Moodle.org [Электронный ресурс]. – Режим доступа: <https://moodle.org/?lang=ru> (дата обращения: 09.03.2020).
18. Edmodo [Электронный ресурс]. – Режим доступа: <https://new.edmodo.com/?go2url=%2Fhome> (дата обращения: 09.03.2020).
19. Home – Socrative [Электронный ресурс]. – Режим доступа: <https://socrative.com/> (дата обращения: 09.03.2020).
20. ThinkWave | Cloud Based School Management Software | Free Gradebook [Электронный ресурс]. – Режим доступа: <https://www.thinkwave.com> (дата обращения: 27.02.2020).
21. ThinkWave | Gradebook and School Management System [Электронный ресурс]. – Режим доступа: <https://www.thinkwave.com/secure/gradebook/> (дата обращения: 11.03.2020).
22. C++ [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/C%2B%2B> (дата обращения: 17.06.2020).

23. Defective C++ [Электронный ресурс]. – Режим доступа: <http://yosefk.com/c++faq/defective.html#> (дата обращения: 17.06.2020).
24. Qt for Beginners [Электронный ресурс]. – Режим доступа: https://wiki.qt.io/About_Qt (дата обращения: 07.06.2020).
25. Программирование на Python [Электронный ресурс]. – Режим доступа: <https://roi4cio.com/de/categories/programmirovanie-na-python/> (дата обращения: 07.06.2020).
26. TIOBE Index for June 2020 [Электронный ресурс]. – Режим доступа: <https://www.tiobe.com/tiobe-index/> (дата обращения: 07.06.2020).
27. Введение в Java FX [Электронный ресурс]. – Режим доступа: <https://javarush.ru/groups/posts/2560-vvedenie-v-java-fx> (дата обращения: 20.03.2020).
28. Qt Jambi [Электронный ресурс]. – Режим доступа: https://wiki.qt.io/Qt_Jambi (дата обращения: 17.06.2020).
29. Что лучше: Java, C++ или Python? [Электронный ресурс]. – Режим доступа: <https://upread.ru/blog/articles-it/java-c-ili-python> (дата обращения: 17.06.2020).
30. ORM (Object-Relational Mapping) [Электронный ресурс]. – Режим доступа: [https://ru.bmstu.wiki/ORM_\(Object-Relational_Mapping\)](https://ru.bmstu.wiki/ORM_(Object-Relational_Mapping)) (дата обращения: 17.06.2020).
31. Python's SQLAlchemy vs Other ORMs [Электронный ресурс]. – Режим доступа: <https://www.pythoncentral.io/sqlalchemy-vs-orms/> (дата обращения: 07.06.2020).
32. Model/View Programming [Электронный ресурс]. – Режим доступа: <https://doc.qt.io/qt-5/model-view-programming.html> (дата обращения: 05.04.2020).
33. SQLite: БД в одной библиотеке [Электронный ресурс]. – Режим доступа: <http://hostinfo.ru/articles/web/rubric48/rubric55/rubric57/1319/> (дата обращения: 17.06.2020).

34. PyCharm: the Python IDE for Professional Developers by JetBrains [Электронный ресурс]. – Режим доступа: <https://www.jetbrains.com/pycharm/> (дата обращения: 15.04.2020).

35. Welcome to Python.org [Электронный ресурс]. – Режим доступа: <https://www.python.org> (дата обращения: 15.04.2020).

36. Get Qt [Электронный ресурс]. – Режим доступа: <https://www.qt.io/download> (дата обращения: 15.04.2020).

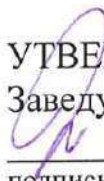
Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

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»

Электронный журнал преподавателя

Руководитель	 подпись, дата	канд. техн. наук, доцент	А. И. Постников
Выпускник	 29.06.2020 подпись, дата		А. В. Дьяченко
Консультант	 подпись, дата	ст. преподаватель	К. В. Пушкарев
Нормоконтролер	 подпись, дата		А. И. Постников

Красноярск 2020