**СОДЕРЖАНИЕ**

**ВВЕДЕНИЕ**3

1. **Описание предметной** **области**5
   1. Введение в предметную область5
   2. Анализ готовых решений7
2. **Проектирование системы**10
   1. Общее назначение системы10
   2. Требования к функциям, выполняемым системой11
   3. План тестирования16
3. **Реализация проекта системы**18
   1. Описание среды разработки18
      1. Выбор и описание программных инструментов18
      2. Обоснование выбора инструментария по разработке18
   2. Разработка программного модуля19
      1. Реализация пользовательского интерфейса программы19
      2. Описание кодом функциональных узлов модуля20
      3. Результат работы и тестирования21

**ЗАКЛЮЧЕНИЕ**22

**СПИСОК ИСПОЛЬЗОВАННЫХ** **ИСТОЧНИКОВ**23

**ПРИЛОЖЕНИЕ 1**24

**ПРИЛОЖЕНИЕ 2**27

**ПРИЛОЖЕНИЕ 3**36

**ВВЕДЕНИЕ**

В настоящее время информационные технологии играют ключевую роль во всех сферах жизни общества, включая образование. Одной из важнейших задач в сфере образования является учет успеваемости студентов, что позволяет эффективно контролировать процесс обучения и выявлять проблемы на ранних стадиях. В колледжах особенно важно иметь надежную систему учета успеваемости, которая бы помогала преподавателям и администрации вести точный контроль над успеваемостью студентов.

Целью курсового проекта является разработка программного модуля системы: «Учет успеваемости в колледже», который обеспечит конкурентоспособность колледжа на рынке образовательных услуг за счет повышения качества административных процессов, а также увеличит рентабельность за счет оптимизации ресурсов и времени, затрачиваемых на учет успеваемости.

Для достижения поставленной цели в рамках курсового проекта предстоит решить следующие задачи:

1. Проанализировать предметную область: изучить текущие системы учета успеваемости для выявления их функциональности, недостатков и потребностей конечных пользователей в условиях колледжа.
2. Проанализировать готовые решения: оценить существующие на рынке программные продукты, чтобы понять их преимущества и возможности для улучшения, что поможет создать более конкурентоспособное решение.
3. Подготовить техническое задание: определить и описать требования к разрабатываемому модулю, его функциональные и нефункциональные характеристики, а также условия его эксплуатации.
4. Подготовить план тестирования: разработать план тестирования, который включает различные методики и сценарии для гарантии надежности и корректной работы модуля.
5. Обосновать выбор инструментов и средств обработки: выбрать и обосновать применение конкретных технологий, языков программирования и инструментов разработки.
6. Описать реализацию технических задач: подробно описать процесс разработки, включая архитектуру системы, дизайн интерфейса, алгоритмы обработки.
7. Выполнить тестирование: провести тестирование разработанного модуля в различных условиях использования.

Объектом исследования является процесс автоматизации учета успеваемости студентов в образовательных учреждениях, в частности, в колледжах. Внедрение такого модуля позволит преподавателям и администрации колледжа сосредоточиться на более важных задачах, таких как повышение качества обучения и развитие студентов, а также обеспечит более точный и оперативный контроль успеваемости.

Предмет исследования — создание, хранение, анализ и обработка данных об успеваемости студентов в рамках разработки программного модуля, обеспечивающего автоматизацию этих процессов и интеграцию с другими системами колледжа.

Программный модуль позволит улучшить процессы учета успеваемости студентов, повысить эффективность работы колледжа, обеспечить его конкурентоспособность и увеличить рентабельность за счет оптимизации управления образовательными процессами. Актуальность данного проекта обусловлена необходимостью в современных и эффективных системах управления, которые адаптируются к динамичным требованиям образовательной среды.

Таким образом, разработка и внедрение программного модуля системы «Учет успеваемости в колледже» станет значимым шагом в повышении эффективности образовательного процесса и улучшении качества образования в колледже.

1. **Описание предметной области**
2. **Введение в предметную область**

Учет успеваемости в образовательных учреждениях имеет давнюю историю. Традиционно он осуществлялся в бумажном виде с использованием журналов, ведомостей и других документов. По мере развития информационных технологий, начиная с 1970-х годов, стали появляться первые автоматизированные системы учета успеваемости, которые постепенно вытесняли бумажный документооборот

Предметная область "Учет успеваемости в колледже" охватывает процессы сбора, хранения, анализа и управления данными о результатах учебной деятельности студентов. Включает в себя регистрацию оценок, посещаемости, выполнения заданий и экзаменационных результатов. Важной частью является автоматизация обработки этих данных, создание отчетов для преподавателей, студентов и администрации, а также обеспечение прозрачности и объективности оценки знаний.

Теоретическую основу предметной области составляют педагогика, психология, управление образованием и другие дисциплины, изучающие закономерности образовательного процесса. Современные теоретические подходы к системам учета успеваемости предполагают комплексное использование информационных технологий в управлении образовательным процессом. Это включает не только автоматизацию учета оценок, но и анализ данных об успеваемости для повышения эффективности обучения. Такие системы обеспечивают интеграцию данных, создавая единую информационную среду. Это требует анализа и проектирования не только программного обеспечения, но и организационных процессов учебного заведения, учитывая специфику образовательной деятельности и требования к учету успеваемости. В результате эффективная система учета успеваемости должна соответствовать педагогическим принципам и способствовать достижению образовательных целей.

Современные требования к информационным системам учета успеваемости в колледжах включают в себя: обеспечение полноты и достоверности данных, возможность оперативного доступа к информации, поддержку различных форм отчетности, интеграцию с другими информационными системами образовательной организации, соблюдение требований законодательства в области защиты персональных данных.

Законодательные аспекты.Разработка ИС регламентируется следующими нормативными документами:

1. Федеральный закон "О персональных данных" от 27.07.2006 N 152-ФЗ. [1]
2. Федеральный закон "Об информации, информационных технологиях и о защите информации" от 27.07.2006 N 149-ФЗ. [2]
3. Федеральный закон "О безопасности критической информационной инфраструктуры Российской Федерации" от 26.07.2017 N 187-ФЗ. [4]
4. ГОСТ 19.701–90 Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения. [5]
5. ГОСТ Р ИСО/МЭК 12207–2010 Системная и программная инженерия. Процессы жизненного цикла программных средств. [3]

Эти нормативные акты определяют требования к обработке персональных данных студентов, обеспечению информационной безопасности, а также к оформлению и ведению программной документации.

Методология разработки программного обеспечения (ПО). Учитывая специфику предметной области "Учет успеваемости в колледже", наиболее подходящей методологией разработки программного обеспечения является гибкая методология Agile.

Преимущества использования Agile-методологии в данном случае:

* Адаптивность к изменениям требований. Образовательные учреждения могут иметь специфические и динамично меняющиеся требования к системе учета успеваемости, которые лучше всего удовлетворяются с помощью итеративного и инкрементального подхода Agile.
* Тесное взаимодействие с заказчиком. Agile предполагает регулярные встречи и обратную связь от пользователей (преподавателей, администрации колледжа), что позволяет быстро адаптировать систему под их нужды.
* Быстрая разработка и внедрение. Agile-методология ориентирована на быструю разработку и поэтапное внедрение функциональности, что важно для оперативного решения задач учета успеваемости.
* Возможность итеративного тестирования. Agile позволяет проводить тестирование на каждом этапе разработки, что критично для систем, работающих с персональными данными студентов.
  1. **Анализ готовых решений**

Информационные системы учета успеваемости в колледжах применяются в образовательных учреждениях среднего профессионального образования. Это могут быть как государственные, так и частные колледжи, техникумы, училища и другие СПО организации.

Ниже приведены примеры ИС и их характеристика:

1. "1С: Колледж ПРОФ"

Комплексное решение на базе "1С: Предприятие" для автоматизации учебной и управленческой деятельности. Программное обеспечение включает модули для учета студентов, успеваемости, расписания, кадров и генерации отчетов. Система поддерживает автоматизацию различных процессов, интеграцию данных и управление учебным процессом.

Преимущества:

* Высокая масштабируемость благодаря модульной архитектуре, что позволяет легко добавлять новые функции при необходимости.
* Интеграция с другими системами "1С", что обеспечивает обмен данными между различными подсистемами, такими как бухгалтерия, кадровый учет и управление ресурсами.

Недостатки:

* Сложность внедрения из-за обширных настроек и требований к интеграции с существующими системами.
* Высокие затраты на лицензирование и обслуживание, что может быть проблемным для небольших учебных заведений, где возможности системы могут оказаться избыточными.

1. "Электронный колледж"

Веб-ориентированная система от компании "Кварта Технологии", обеспечивающая ведение электронных журналов, расписания, учет студентов и автоматическую генерацию отчетов. Система доступна через браузер и не требует установки на компьютеры пользователей.

Преимущества:

* Простота использования благодаря веб-интерфейсу, что снижает требования к технической поддержке и легко обучает пользователей.
* Быстрая настройка и минимальные требования к оборудованию, так как система работает в облаке.
* Возможность интеграции с системами видеонаблюдения и контроля доступа, что расширяет функциональность за счет дополнительных модулей.

Недостатки:

* Ограниченные возможности для глубокой кастомизации, что может быть проблемой для учебных заведений со сложными и специфическими требованиями.
* Зависимость от интернет-соединения, так как система работает исключительно через веб-интерфейс.

1. "ЭлЖур"

Закрытая информационная система для ведения электронных журналов и учета успеваемости. Программное обеспечение обеспечивает строгую регистрацию пользователей и образовательных учреждений, а также соблюдает требования безопасности.

Преимущества:

* Высокая масштабируемость, что позволяет адаптировать систему для больших учебных заведений с большим количеством пользователей.
* Интеграция с системами "1С", что упрощает управление данными и синхронизацию с другими программными решениями.

Недостатки:

* Сложность внедрения из-за необходимости настройки безопасности, интеграции с другими системами и обучения пользователей.
* Высокие затраты на внедрение и обслуживание, что может сделать систему недоступной для небольших учебных заведений с ограниченным бюджетом.

Таким образом, учет успеваемости в колледжах включает сбор, хранение и анализ данных о студентах. Переход на автоматизированные системы повысил их эффективность и прозрачность. Современные решения должны обеспечивать полноту данных, оперативный доступ, поддержку отчетности и интеграцию с другими системами, учитывая требования по защите данных. Методология Agile оптимальна для разработки ПО благодаря гибкости и взаимодействию с заказчиком. Выбор системы зависит от специфики образовательного учреждения.

Автоматизированные системы учета успеваемости позволяют преподавателям и администраторам получать актуальные данные в режиме реального времени, что способствует своевременному реагированию на проблемы и корректировке учебного процесса. Они также поддерживают различные формы отчетности и интеграцию с другими информационными системами.

Безопасность данных является критически важным аспектом, учитывая растущие угрозы кибербезопасности. Методология Agile, с ее акцентом на гибкость и взаимодействие с заказчиком, становится все более популярной в разработке ПО для образовательных учреждений. Выбор системы зависит от специфики образовательного учреждения, что требует тщательного анализа и адаптации к конкретным потребностям.

1. **Проектирование системы**
   1. **Общее назначение системы**

Система учета успеваемости в колледже предназначена для автоматизации процессов сбора, хранения и обработки данных об учебных достижениях студентов. Основная цель системы — улучшить управление учебным процессом, сократить трудозатраты со стороны преподавателей и администрации, а также обеспечить удобный доступ к информации для всех участников образовательного процесса.

Предполагаемый результат:

* Полная автоматизация учета успеваемости и посещаемости студентов.
* Сокращение времени на ручную обработку данных.
* Повышение точности и достоверности данных за счет исключения человеческого фактора.
* Интеграция с другими системами колледжа для создания единой информационной среды.

Эффект от внедрения:

* Ускорение процесса принятия решений за счет быстрого доступа к актуальной информации.
* Снижение нагрузки на преподавателей и администрацию, так как рутинные операции учета и анализа выполняются автоматически.
* Улучшение качества образовательного процесса за счет своевременного выявления и устранения проблем с успеваемостью студентов.
* Соответствие законодательным требованиям по защите персональных данных, что минимизирует риски нарушения конфиденциальности.
  1. **Требования к функциям, выполняемым системой**

Система учета успеваемости в колледже должна выполнять ряд ключевых функций, обеспечивающих автоматизацию процессов сбора, обработки и анализа данных о результатах обучения студентов. Основываясь на анализе предметной области, можно выделить следующие функциональные задачи программного модуля:

Основные функциональные задачи:

* 1. Ведение электронного журнала успеваемости:
* Ввод и хранение оценок, отметок посещаемости и комментариев преподавателей.
* Отслеживание динамики успеваемости студентов.
* Возможность корректировки данных при необходимости.
  1. Формирование отчетов:
* Поддержка различных форматов отчетов (Excel).
  1. Управление доступом пользователей:
* Разграничение прав доступа для разных категорий пользователей (преподаватели, студенты, администрация).
* Поддержка авторизации и аутентификации.
  1. Интеграция с внешними системами:
* Синхронизация данных с базами данных студентов и расписанием.

В системе учёта успеваемости колледжа задействовано несколько ключевых участников, каждый из которых выполняет определённые функции и взаимодействует с системой для общей эффективности. Диаграмма прецедентов (ПРИЛОЖЕНИЕ 1, рисунок 1) описывает основных актёров и их роли в системе.

Выбор типа пользователя:

* Актёры: студент, преподаватель, администратор.
* Описание: Пользователь при запуске приложения

1. Просмотр посещаемости
   * Актёр: Студент
   * Описание: Студент просматривает свою посещаемость по дисциплинам.
2. Просмотр оценок
   * Актёр: Студент
   * Описание: Студент просматривает свои оценки по дисциплинам.
3. Выставление оценки
   * Актёр: Преподаватель
   * Описание: Преподаватель выставляет или изменяет оценку студента по дисциплине.
4. Отметка посещаемости
   * Актёр: Преподаватель
   * Описание: Преподаватель отмечает посещаемость студентов на занятии.
5. Управление пользователями
   * Актёр: Администратор
   * Описание: Администратор добавляет, изменяет и удаляет пользователей (студентов, преподавателей, администраторов).
6. Управление группами
   * Актёр: Администратор
   * Описание: Администратор добавляет, изменяет и удаляет группы.

Основными объектами предметной области для системы учета успеваемости в колледже являются объекты, перечисленные в таблице 1.

Таблица 1 - Объекты предметной области

| № п/п | Объект | Действия |
| --- | --- | --- |
| 1 | Студент | Авторизация, просмотр успеваемости |
| 2 | Преподаватель | Внесение оценок, управление успеваемостью студентов, генерация отчетов |
| 3 | Администратор | Управление профилями пользователей, настройка прав доступа, обслуживание системы |
| 4 | Группа | Формирование списков, распределение студентов, учет численности |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| № п/п | Объект | Действия |
| 5 | Предмет | Создание учебных программ, расписание занятий, привязка к преподавателям |
| 6 | Оценка | Фиксация результатов, расчет среднего балла, формирование ведомостей |
| 7 | Посещаемость | Учет присутствия, формирование отчетов о пропусках, статистика посещений |

Для реализации функциональности, описанной в Таблице 1 "Объекты предметной области", система учёта успеваемости в колледже оперирует данными, хранящимися в базе данных.

Ниже переведены таблицы для описания каждой сущности, атрибутов, типа атрибутов и их краткого описания.

Таблица 2 - Атрибут сущности – «Пользователи»

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание |
| id | INT | Первичный ключ, автоматически увеличивающийся уникальный идентификатор пользователя. |
| username | VARCHAR(255) | Имя пользователя для входа. |
| password | VARCHAR(255) | Пароль пользователя |
| role | VARCHAR(255) | Роль пользователя |

Таблица 3 - Атрибут сущности – «Группа»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| group\_id | INT | Первичный ключ, автоматически увеличивающийся уникальный идентификатор группы. |
| group\_name | VARCHAR(255) | Название группы |
| speciality\_name | VARCHAR(255) | Название специальности |

Таблица 4 - Атрибут сущности – «Студент»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| user\_id | INT | Первичный ключ, внешний ключ, ссылающийся на `Users.id`. Уникальный идентификатор студента. |

Продолжение таблицы 4

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание |
| group\_id | INT | Внешний ключ, ссылающийся на `Group\_. group\_id`. Указывает группу, к которой принадлежит студент. |
| first\_name | VARCHAR(255) | Имя студента. |
| last\_name | VARCHAR(255) | Фамилия студента. |
| patronymic | VARCHAR(255) | Отчество студента. |

Таблица 5 - Атрибут сущности – «Преподаватель»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| user\_id | INT | Первичный ключ, внешний ключ, ссылающийся на `Users.id`. Уникальный идентификатор преподавателя. |
| first\_name | VARCHAR(255) | Имя преподавателя. |
| last\_name | VARCHAR(255) | Фамилия преподавателя. |
| patronymic | VARCHAR(255) | Отчество преподавателя. |

Таблица 6 - Атрибут сущности – «Предмет»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| discipline\_id | INT | Первичный ключ, автоматически увеличивающийся уникальный идентификатор дисциплины. |
| discipline\_name | VARCHAR(255) | Название дисциплины |
| teacher\_id | INT | Внешний ключ, ссылающийся на `Teacher.user\_id`. Указывает преподавателя, который ведет дисциплину. |
| group\_id | INT | Внешний ключ, ссылающийся на `Group\_. group\_id`. Указывает группу, которая изучает дисциплину. |

Таблица 7 - Атрибут сущности – «Оценка»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| grade\_id | INT | Первичный ключ, автоматически увеличивающийся уникальный идентификатор оценки. |

Продолжение таблицы 7

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание |
| student\_id | INT | Внешний ключ, ссылающийся на `Student.user\_id`. Указывает студента, получившего оценку. |
| discipline\_id | INT | Внешний ключ, ссылающийся на `Discipline.discipline\_id`. Указывает дисциплину, по которой выставлена оценка. |
| grade\_date | DATE | Дата выставления оценки. |
| grade\_value | INT | Значение оценки (например, 3, 4, 5). |
| grade\_type | VARCHAR(255) | Тип оценки (например, "экзамен", "зачет", "контрольная работа"). |

Таблица 8 - Атрибут сущности – «Посещаемость»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| attendance\_id | INT | Первичный ключ, автоматически увеличивающийся уникальный идентификатор группы. |
| student\_id | VARCHAR(255) | Внешний ключ, ссылающийся на Student.user\_id. Указывает студента, чья посещаемость записывается. |
| discipline\_id | VARCHAR(255) | Внешний ключ, ссылающийся на Discipline.discipline\_id. Указывает дисциплину, для которой записывается посещаемость. |
| attendance\_date | DATE | Дата записи посещаемости. |
| is\_present | BOOLEAN | Указывает, присутствовал ли студент или отсутствовал. |

Модель данных системы учёта успеваемости в колледже, включая все связи между таблицами, представлена в виде диаграммы IDEF1X (ПРИЛОЖЕНИЕ 1, рисунок 2). На ней показаны связи между таблицами базы данных и тип связей.

Представленная ER-диаграмма (ПРИЛОЖЕНИЕ 1, рисунок 3) изображает сущности, их атрибуты и связи, необходимые для моделирования системы «Учёт успеваемости в колледже». В данной системе выделены основные сущности, такие как Пользователи, Студенты, Преподаватели, Группы, Дисциплины, Оценки и Посещаемость. Каждая сущность обладает своими атрибутами, которые описывают важные характеристики данных объектов.

* 1. **План тестирования**

Данный план тестирования направлен на проверку корректности работы, производительности, удобства использования, безопасности и совместимости модуля "Учет успеваемости в колледже". Тестирование позволит выявить потенциальные проблемы и ошибки до развертывания системы, обеспечивая ее надежность и соответствие требованиям.

Задачи тестирования:

* + 1. Функциональное тестирование:

Цель: проверить соответствие функциональности модуля заявленным требованиям. Убедиться, что все функции работают корректно.

Тест-кейсы:

* Добавление, редактирование и удаление данных: проверка корректности работы с информацией о студентах, преподавателях, группах.
* Экспорт данных: проверка корректной работы экспорта файлов.

1. Тестирование производительности и стабильности:

Цель: оценить производительность модуля под нагрузкой.

Тест-кейсы:

* Стресс-тестирование: проверка поведения системы при экстремальных нагрузках, превышающих ожидаемые.
* Тестирование стабильности: длительное тестирование работы модуля для выявления потенциальных утечек памяти или других проблем, приводящих к нестабильной работе.

1. Тестирование удобства и юзабилити (графический интерфейс):

Цель: оценить удобство использования графического интерфейса.

Тест-кейсы:

Интуитивность интерфейса: проверка насколько легко пользователи могут ориентироваться в интерфейсе и выполнять необходимые действия.

* Визуальное оформление: оценка визуального оформления интерфейса, включая цветовую схему, шрифты и общую эстетику.
* Доступность: проверка соответствия интерфейса требованиям доступности для пользователей с ограниченными возможностями.

1. Обеспечение безопасности данных:

Цель: проверить безопасность данных, хранящихся в модуле, и предотвратить несанкционированный доступ.

Тест-кейсы:

* Аутентификация и авторизация: убедиться, что только авторизованные пользователи имеют доступ к определенным функциям и данным.
* Хранение паролей: проверка безопасности хранения паролей пользователей (хеширование) [15].

1. Совместимость с различными ПО и аппаратными средствами:

Цель: убедиться в корректной работе модуля на различных платформах и конфигурациях.

Тест-кейсы:

* Тестирование на различных операционных системах: проверка работы модуля на различных ОС (Windows, macOS). Убедиться, что установка, запуск и функциональность приложения работают корректно на каждой платформе.
* Тестирование на различных разрешениях экрана: убедиться, что все элементы интерфейса видны и доступны для взаимодействия на различных разрешениях экрана.

Таким образом, система учета успеваемости в колледже предназначена для автоматизации процессов сбора и обработки данных о студентах. Она обеспечивает удобный доступ к информации для всех участников образовательного процесса и сокращает трудозатраты.

1. **Реализация проекта системы**
   1. **Описание среды разработки**

Для реализации проекта системы учета успеваемости в колледже потребуется использование языка программирования и среды разработки, обеспечивающих эффективную разработку, отладку и управление кодом.

* + 1. **Выбор и описание программных инструментов**

В качестве языка программирования выбран Python – высокоуровневый язык программирования, известный своей простотой и читаемостью кода. Python широко используется в разработке веб-приложений, научных вычислений, автоматизации задач и других областях. Он обладает большой экосистемой библиотек, что упрощает разработку сложных приложений.

В качестве среды разработки выбрана PyCharm – мощная интегрированная среда разработки (IDE) от JetBrains, специально разработанная для Python. PyCharm предлагает широкий набор инструментов для упрощения разработки, отладки, тестирования и поддержки проектов на Python. Она включает в себя интеллектуальное авто дополнение кода, подсветку синтаксиса, навигацию по коду, рефакторинг, отладчик, инструменты для работы с системами контроля версий (например, Git), поддержку веб-разработки и многое другое.

* + 1. **Обоснование выбора инструментов**

### Библиотеки:

* **SQLite3**: Легковесная встраиваемая реляционная СУБД, идеальная для проектов с небольшой базой данных. Хранится в одном файле, поддерживает стандартный SQL.
* **PyQt6**: Набор Python-привязок для Qt, мощного инструментария для разработки GUI. Предоставляет обширный набор виджетов и функций для создания интерактивных приложений.

### Обоснование выбора:

* **Python [8], [10]**: Простой и мощный язык с большим количеством библиотек. Встроенная поддержка SQLite3.
* **PyCharm[11]**: Удобная среда разработки для Python, упрощающая написание и отладку кода.
* **SQLite3[7], [12], [14]**: Простая и эффективная база данных без необходимости настройки сервера, идеальная для небольших проектов.
* **PyQt6[9]**: Богатый набор инструментов для создания кроссплатформенных GUI, работающих на Windows, macOS и Linux без модификации кода.

Таким образом, использование Python, PyCharm, SQLite3 и PyQt6 обеспечивает эффективную и удобную среду для разработки приложения учета успеваемости в колледже.

* 1. **Разработка программного модуля**
     1. **Реализация пользовательского интерфейса программы**

При запуске приложения пользователь видит окно авторизации (ПРИЛОЖЕНИЕ 3, рисунок 4). После ввода логина и пароля определяется роль пользователя.

После авторизации пользователи могут иметь следующие роли:

1. **Администратор:**

* Управление пользователями (добавление, удаление, редактирование студентов и преподавателей).
* Управление группами, перемещение студентов между группами.

1. **Преподаватель:**
   * Просмотр и редактирование оценок студентов.
   * Экспорт журнала оценок в формат Excel.
2. **Студент:**
   * Просмотр своих оценок по различным дисциплинам.
   * Просмотр информации о своей группе и специальности.

Если студент успешно войдет в свой аккаунт, то он увидит свои фамилию, имя, отчество, группу и специальность, а также журнал с дисциплинами и оценками (ПРИЛОЖЕНИЕ 3, рисунок 5).

Если преподаватель успешно войдет в свой аккаунт, то изначально он увидит свои фамилию, имя, отчество, затем кнопку с цифрой группы которую ведет преподаватель, после нажатия на эту кнопку, ниже появятся кнопки с дисциплинами которые преподаватель ведет у этой группы, а при нажатии на кнопку дисциплины появится журнал по этой дисциплине (ПРИЛОЖЕНИЕ 3, рисунок 6–8).

Если администратор успешно войдет в свой аккаунт, то он увидит окно, в котором будет 9 вкладок: «Добавить студента», «Добавить преподавателя», «Удалить студента», «Удалить преподавателя», «Добавить группу», «Удалить группу», «Переместить студента», «Редактировать студента», «Редактировать преподавателя». На каждой вкладке реализован функционал, соответствующий ее названию (ПРИЛОЖЕНИЕ 3, рисунок 9–17).

* + 1. **Описание кодом функциональных узлов модуля**

При включении приложения отобразится окон авторизации, в котором нужно будет ввести логин и пароль пользователя, после чего программа сама определит какая роль у пользователя. Создание окна авторизации и функция авторизации (def login) находится в классе Main Window (ПРИЛОЖЕНИЕ 3, рисунок 18–21).

Если студент войдет в свой аккаунт, то для него откроется окно студента, в котором будет его дневник с оценками по дисциплинам. Отображение окна студента и создание дневника студента (def populate\_journal) прописано в коде в классе Student Window (ПРИЛОЖЕНИЕ 3, рисунок 22–24).

Если преподаватель войдет в свой аккаунт, то для него откроется окно преподавателя, в котором будут находиться дисциплины и журнал студентов по дисциплинам. Отображение окна преподавателя и создание журнала студентов (def show\_journal) прописано в коде в классе TeacherWindow (ПРИЛОЖЕНИЕ 3, рисунок 25–31).

Если администратор войдет в свой аккаунт, то он увидит вкладку добавления студента. Также в этом окне находятся еще 8 вкладок. Отображение окна администратора, отображение виджетов во вкладке создания студента (def create\_add\_student\_tab) и функция для добавления студента (def add\_student) находятся в классе AdminWindow (ПРИЛОЖЕНИЕ 3, рисунок 32–36).

Таким образом, разработанный программный модуль для системы учета успеваемости в колледже включает узлы для авторизации пользователей, отображения индивидуальных дневников студентов, журналов успеваемости для преподавателей и административных инструментов. Подробнее о каждой функции и детали приложения можно узнать на GitHub [6], [13].

* + 1. **Результат работы и тестирования**

Для начала проведем тестирование окна авторизации (ПРИЛОЖЕНИЕ 3, рисунок 37, 38)

Далее попробуем создать студента и проверить появится ли он в той группе, в которой нам нужно и будут ли у него те данные, которые введем (ПРИЛОЖЕНИЕ 3, рисунок 39, 40).

Теперь для проверки добавился ли студент в нужную группу и для проверки окна преподавателя авторизуемся за преподавателя, который ведет группу 316, в которую администратор добавил нового студента. Студент успешно был добавлен в группу 316, теперь он стоит в списке под 12 номером. (ПРИЛОЖЕНИЕ 3, рисунок 41). Далее попробуем выставить ему оценки (ПРИЛОЖЕНИЕ 3, рисунок 42).

Далее попробуем зайти на аккаунт студента, которого только что добавили и которому проставили оценки и проверить, появились ли оценки в его дневнике. Да, оценки появились в его дневнике и корректно отображаются (ПРИЛОЖЕНИЕ 3, рисунок 43).

Остальные позитивные и негативные тесты, в которых были проверены основные функции приложения, были проведены в (ПРИЛОЖЕНИЕ 2, таблица 9).

Таким образом, в результате тестирования было подтверждено корректное функционирование основных функций системы учета успеваемости в колледже.

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсового проекта была разработана система учета успеваемости в колледже, направленная на автоматизацию процессов сбора, хранения и обработки данных об учебных достижениях студентов.

Анализ предметной области позволил изучить существующие системы учета успеваемости, выявить их преимущества и недостатки, а также определить требования к разрабатываемому модулю. Проектирование системы включало разработку архитектуры программного модуля, проектирование базы данных, создание ER-диаграмм, а также описание требований к функциональности системы.

Реализация проекта осуществлялась с использованием языка программирования Python и среды разработки PyCharm. Для работы с базой данных была выбрана библиотека SQLite3, а для создания пользовательского интерфейса — PyQt6. В ходе разработки был создан пользовательский интерфейс, реализованы основные функциональные узлы, обеспечивающие авторизацию пользователей, отображение индивидуальных дневников студентов, журналов успеваемости для преподавателей и административные инструменты для управления данными.

Тестирование системы подтвердило корректное функционирование основных функций, включая авторизацию, добавление и редактирование данных о студентах и преподавателях, выставление оценок и просмотр успеваемости.

Таким образом, разработанная система учета успеваемости в колледже позволяет значительно упростить и ускорить процессы учета и анализа успеваемости студентов, повысить эффективность работы преподавателей и администрации, а также обеспечить прозрачность и объективность оценки знаний.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

Нормативно-правовые источники

1. Федеральный закон "О персональных данных" от 27.07.2006 N 152-ФЗ.
2. Федеральный закон "Об информации, информационных технологиях и о защите информации" от 27.07.2006 N 149-ФЗ.
3. ГОСТ Р ИСО/МЭК 12207–2010 Системная и программная инженерия. Процессы жизненного цикла программных средств.
4. Федеральный закон "О безопасности критической информационной инфраструктуры Российской Федерации" от 26.07.2017 N 187-ФЗ.
5. ГОСТ 19.701–90 Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.

Монографии

1. Кривицкий А. В. Информационная система «Учёт успеваемости в колледже», GitHub: https://github.com/keyvvest/KP-College-Diary.git

Онлайн-курсы

1. Интерактивный тренажер по SQL.
2. Поколение Python: курс для продвинутых.

Интернет-источники

* + 1. Riverbank Computing. PyQt6 Documentation. //https://www.riverbankcomputing.com/static/Docs/PyQt6//
    2. Python Software Foundation. Python Documentation. //https://www.python.org
    3. JetBrains. PyCharm Documentation. //https://www.jetbrains.com/pycharm/documentation/
    4. SQLZoo. SQL Documentation. //https://sqlzoo.net/wiki/
    5. GitHub. GitHub Documentation. //https://github.com/
    6. Sqlite3 Documentation. //https://docs.python.org/3/library/sqlite3.html
    7. Bcrypt. //https://zetcode.com/python/bcrypt/

**ПРИЛОЖЕНИЕ 1**

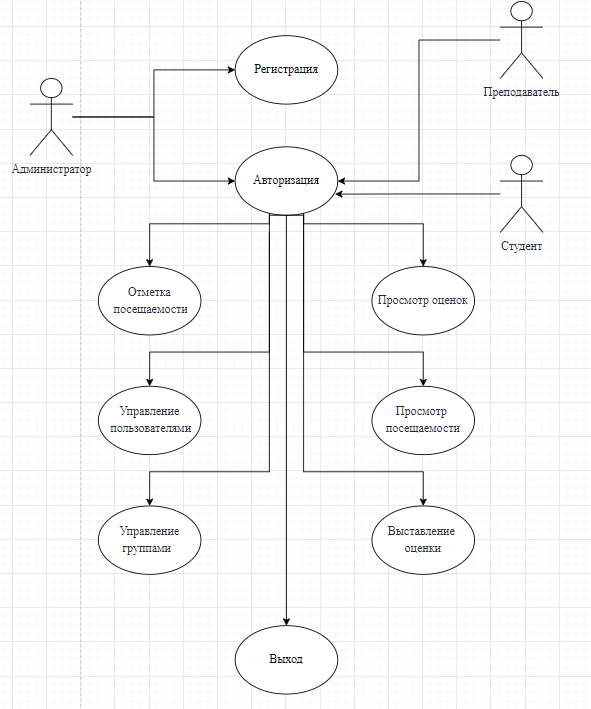


Рисунок 1 - Диаграмма прецедентов.

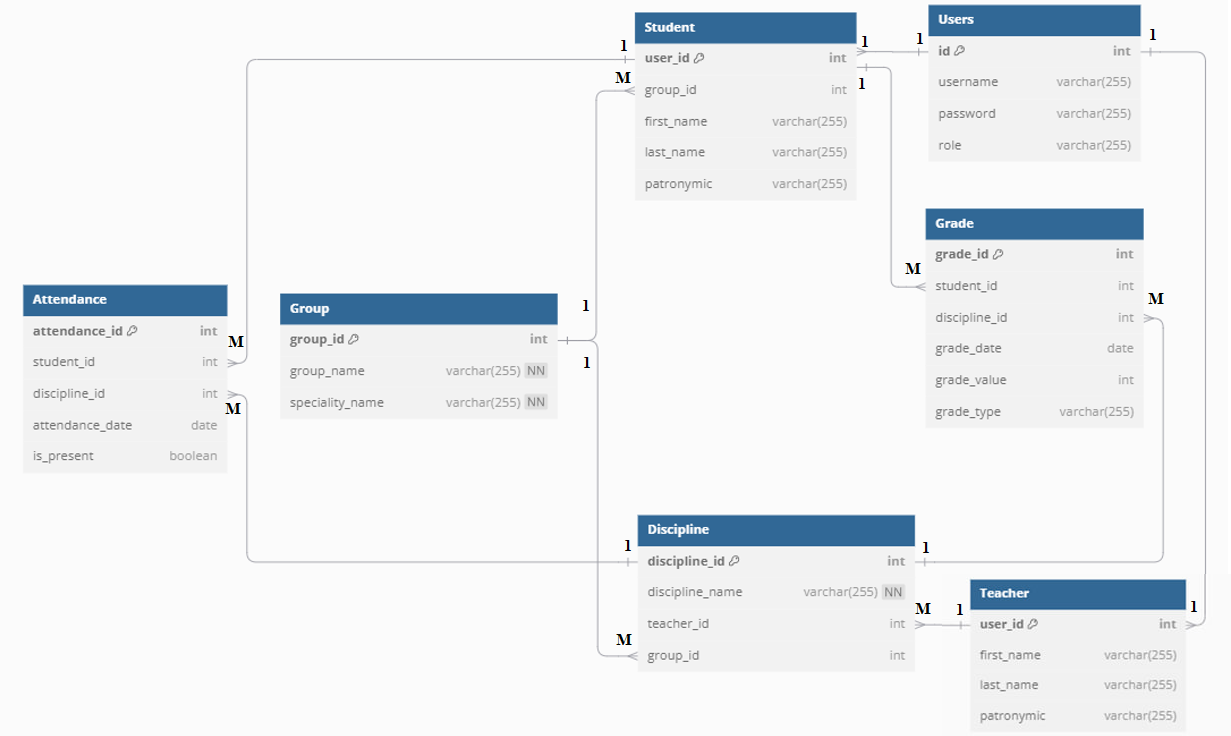
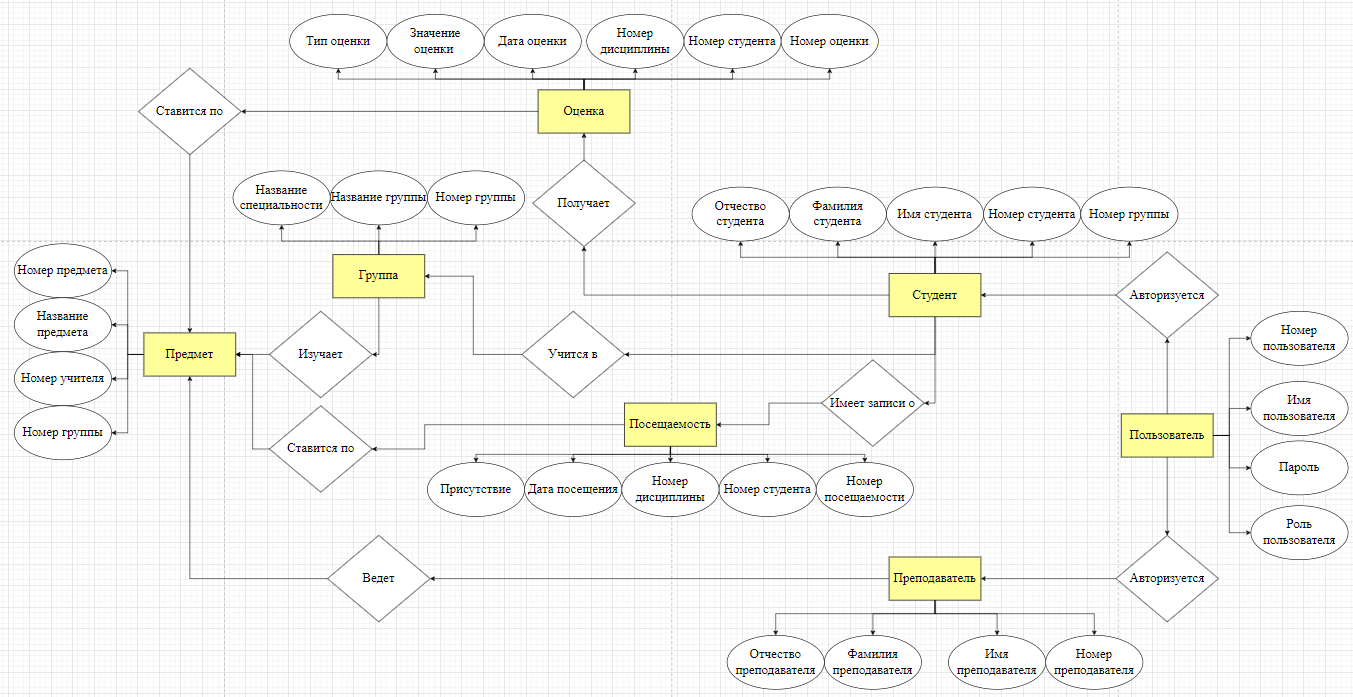


Рисунок 2 - Диаграмма IDEF1X.

Рисунок 3 - ER диаграмма.

**ПРИЛОЖЕНИЕ 2**

Таблица 9 - План тестирования

| № | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | Результат тестирования |
| --- | --- | --- | --- | --- | --- |
| 1 | Авторизация пользователя | Логин, пароль | Вводим корректные данные для входа пользователя  Логин: admin  Пароль: admin | Успешный вход | Успешный вход |
|  |  |  |  |  | |
|  |  |  | Вводим некорректные данные  Логин: ввв  Пароль: ввв | Ошибка: пользователь не найден | Ошибка: пользователь не найден |
|  |  |  |  |  | |
|  |  |  | Вводим неправильный пароль  Логин: admin  Пароль: d | Ошибка: неверный пароль | Ошибка: неверный пароль |

Продолжение таблицы 9

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | Результат тестирования |
|  |  |  |  |  | |
|  |  |  | Оставляем поле логина пустым | Ошибка: поле логин не может быть пустым | Ошибка: поле логин не может быть пустым |
|  |  |  |  |  | |
|  |  |  | Оставляем поле пароля пустым | Ошибка: поле пароль не может быть пустым | Ошибка: поле пароль не может быть пустым |
|  |  |  |  |  | |

Продолжение таблицы 9

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | | Результат тестирования |
| 2 | Добавить студента | Имя, Фамилия, Отчество, Номер группы, Логин, Пароль | Вводим корректные данные для добавления студента  Имя: Иван,  Фамилия: Иванов, Отчество: Иванович,  Номер группы: 316,  Логин: Ivanov,  Пароль: 12345 | Студент успешно добавлен | | Студент успешно добавлен |
|  |  |  |  |  | | |
|  |  |  | Оставляем все поля пустыми | Ошибка: Все поля должны быть заполнены | Ошибка: Все поля должны быть заполнены | |
|  |  |  |  |  | | |
|  |  |  | Вводим корректные данные для добавления студента  Имя: 1234,  Фамилия: 1234, Отчество: 12345 | Ошибка: поля Имя, Фамилия, Отчество должны содержать только текст | Ошибка: поля Имя, Фамилия, Отчество должны содержать только текст | |

Продолжение таблицы 9

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | Результат тестирования |
|  |  |  | Заполнить поле группа несуществующим номером группы  Номер группы: 2345 | Ошибка: Группа не найдена | Ошибка: Группа не найдена |
|  |  |  |  | Изображение выглядит как текст, снимок экрана, Шрифт, дизайн  Автоматически созданное описание | |
|  |  |  | Ввести логин, который уже занят | Ошибка: Пользователь с таким логином уже существует | Ошибка: Пользователь с таким логином уже существует |
|  |  |  |  |  | |
|  |  |  | Нажать кнопку сгенерировать пароль | Поле пароль будет заполнено | Поле пароль будет заполнено |
|  |  |  |  | **Изображение выглядит как текст, снимок экрана, Шрифт, линия  Автоматически созданное описание** | |
| 3 | Добавить преподавателя | Имя, Фамилия, Отчество, Логин, Пароль | Вводим корректные данные для добавления преподавателя  Имя: Петр,  Фамилия: Петров, Отчество: Петрович,  Логин: Petrov,  Пароль: 54321 | Преподаватель успешно добавлен | Преподаватель успешно добавлен |

Продолжение таблицы 9

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | Результат тестирования |
|  |  |  |  |  | |
|  |  |  | Заполнить поля: Имя, Фамилия, Отчество цифрами | Ошибка: поля Имя, Фамилия, Отчество должны содержать только текст | Ошибка: поля Имя, Фамилия, Отчество должны содержать только текст |
|  |  |  |  |  | |
|  |  |  | Ввести логин, который уже занят | Ошибка: Пользователь с таким логином уже существует | Ошибка: Пользователь с таким логином уже существует |
|  |  |  |  |  | |
| 6 | Добавить группу | Название группы, Специальность | Ввести корректные данные для добавления группы  Название группы: 315,  Специальность: Программирование | Группа добавлена успешно | Группа добавлена успешно |
|  |  |  |  |  | |

Продолжение таблицы 9

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | Результат тестирования |
|  |  |  | Оставить поля пустыми | Ошибка: название группы и специальность не могут быть пустыми | Ошибка: название группы и специальность не могут быть пустыми |
|  |  |  |  |  | |
| 7 | Удалить группу | Список группы | Выбрать группу, нажать на кнопку удалить группу и подтвердить свой выбор | Группа удалена успешно | Группа удалена успешно |
|  |  |  |  |  | |
| 8 | Переместить студента | Список студентов, список групп | Переместить студента в другую группу | Студент успешно перемещен | Студент успешно перемещен |
|  |  |  |  |  | |
|  |  |  | Переместить студента в ту группу, в которой он уже находится | Ошибка: Студент уже находится в этой группе | Ошибка: Студент уже находится в этой группе |

Продолжение таблицы 9

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | Результат тестирования |
|  |  |  |  |  | |
| 9 | Редактировать студента | Имя, Фамилия, Отчество, Логин, Пароль | Ввести корректные данные для редактирования | Данные студента обновлены успешно | Данные студента обновлены успешно |
|  |  |  |  |  | |
|  |  |  | Заполнить поля: Имя, Фамилия, Отчество цифрами  Имя: 123  Фамилия: 213  Отчество: 123 | Ошибка: поля Имя, Фамилия, Отчество должны содержать только текст | Ошибка: поля Имя, Фамилия, Отчество должны содержать только текст |
|  |  |  |  |  | |
|  |  |  | Ввести логин, который уже занят | Ошибка: Пользователь с таким логином уже существует | Ошибка: Пользователь с таким логином уже существует |
|  |  |  |  |  | |
| 10 | Редактировать преподавателя | Имя, Фамилия, Отчество, Логин, Пароль | Ввести корректные данные для редактирования | Данные преподавателя обновлены успешно | Данные преподавателя обновлены успешно |

Продолжение таблицы 9

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | Результат тестирования |
|  |  |  |  |  | |
|  |  |  | Заполнить поля: Имя, Фамилия, Отчество цифрами | Ошибка: поля Имя, Фамилия, Отчество должны содержать только текст | Ошибка: поля Имя, Фамилия, Отчество должны содержать только текст |
|  |  |  |  |  | |
|  |  |  | Ввести логин, который уже занят | Ошибка: Пользователь с таким логином уже существует | Ошибка: Пользователь с таким логином уже существует |
|  |  |  |  |  | |
| 11 | Заполнение журнала | Список студентов, оценки по дисциплинам | Поставить оценку за контрольную работу | Оценка появится в журнале | Оценка появится в журнале |

Продолжение таблицы 9

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | Результат тестирования |
|  |  |  |  |  | |
|  |  |  | Нажать на кнопку сохранить журнал | Журнал успешно сохранен | Журнал успешно сохранен |
|  |  |  |  |  | |
|  |  |  | Нажать на кнопку экспорт в Excel | Журнал по данной дисциплине будет экспортирован | Журнал по данной дисциплине будет экспортирован |
|  |  |  |  |  | |

**ПРИЛОЖЕНИЕ 3**

Скриншоты графического интерфейса и функций приложения.

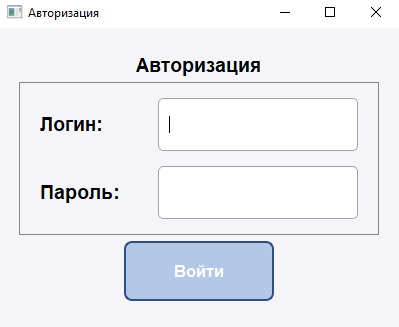


Рисунок 4 – Окно авторизации.

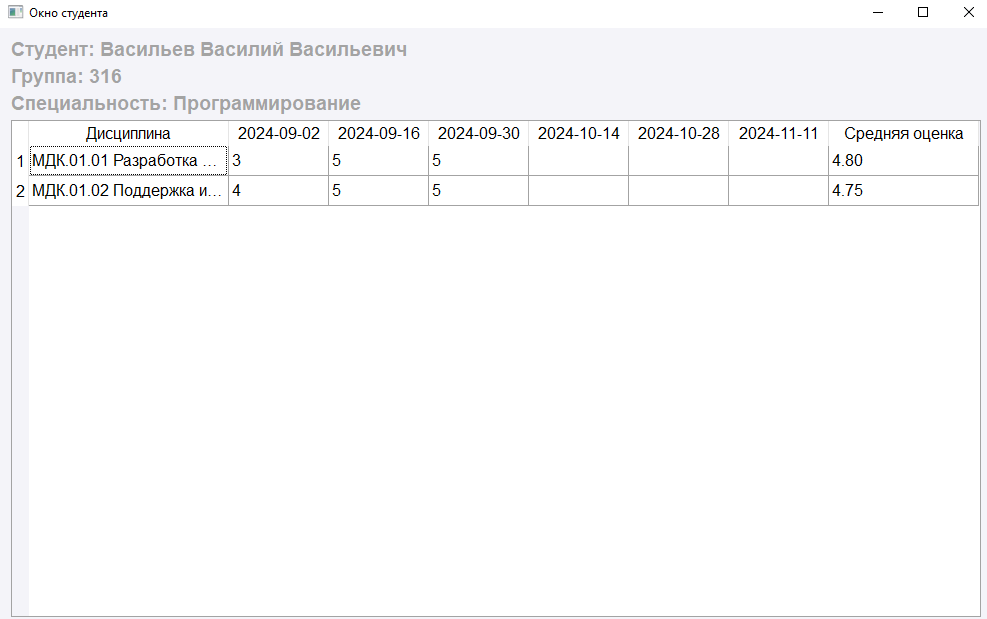
****

Рисунок 5 – Окно студента.

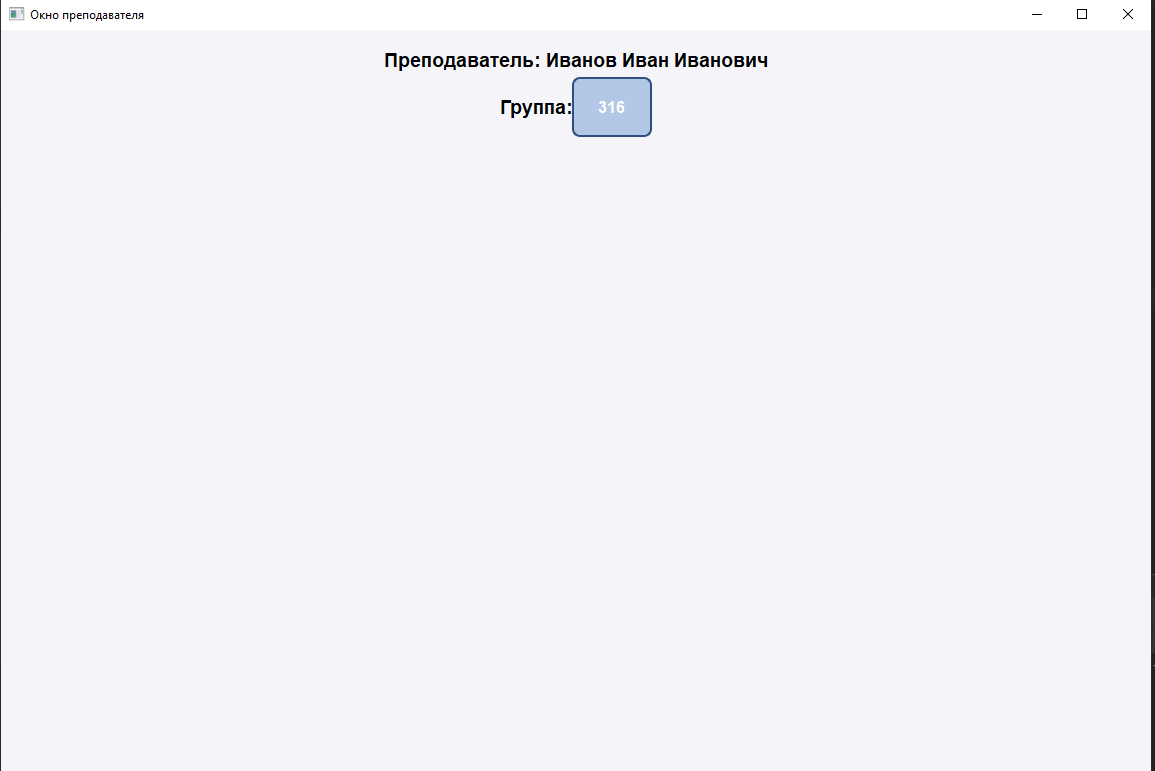
****

Рисунок 6 – Окно преподавателя при входе.

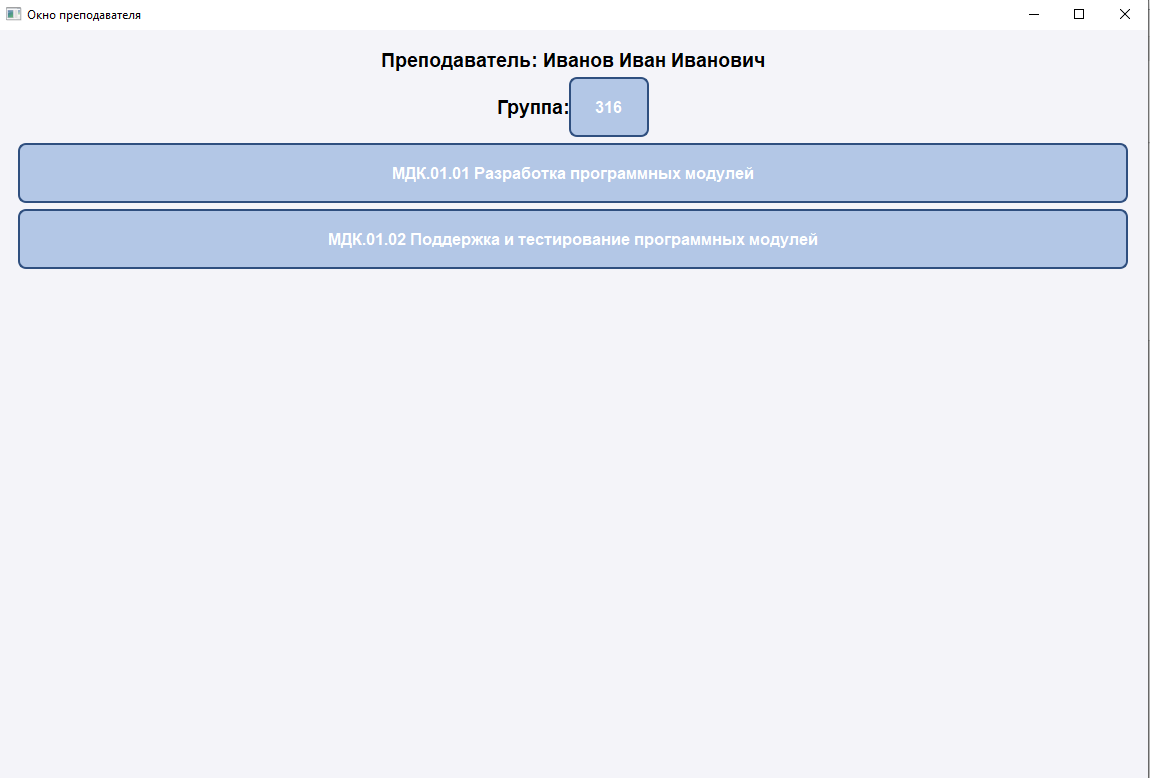


Рисунок 7 – Окно преподавателя после нажатия на кнопку с цифрой группы.

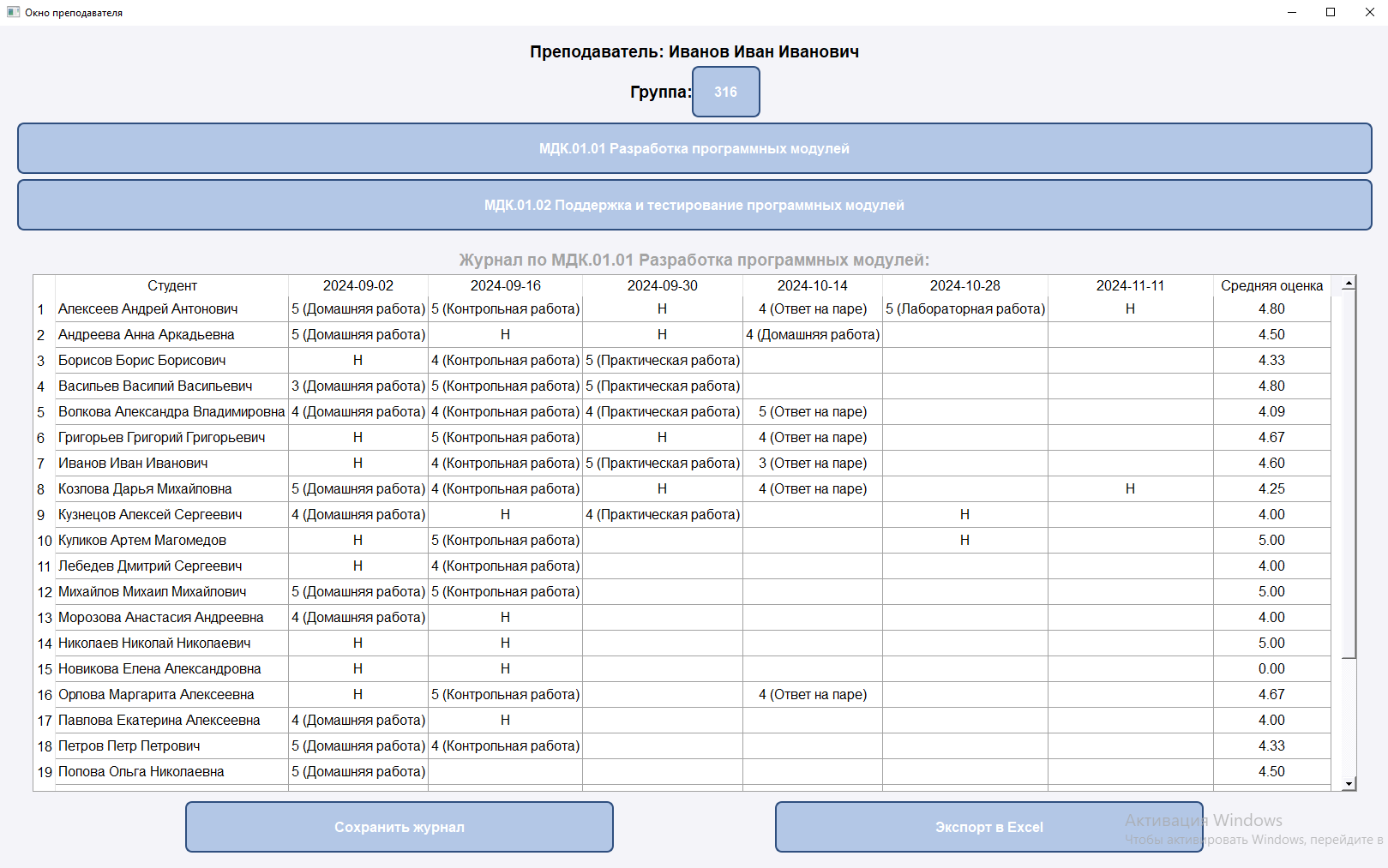


Рисунок 8 – Окно преподавателя после нажатия на кнопку с названием дисциплины.

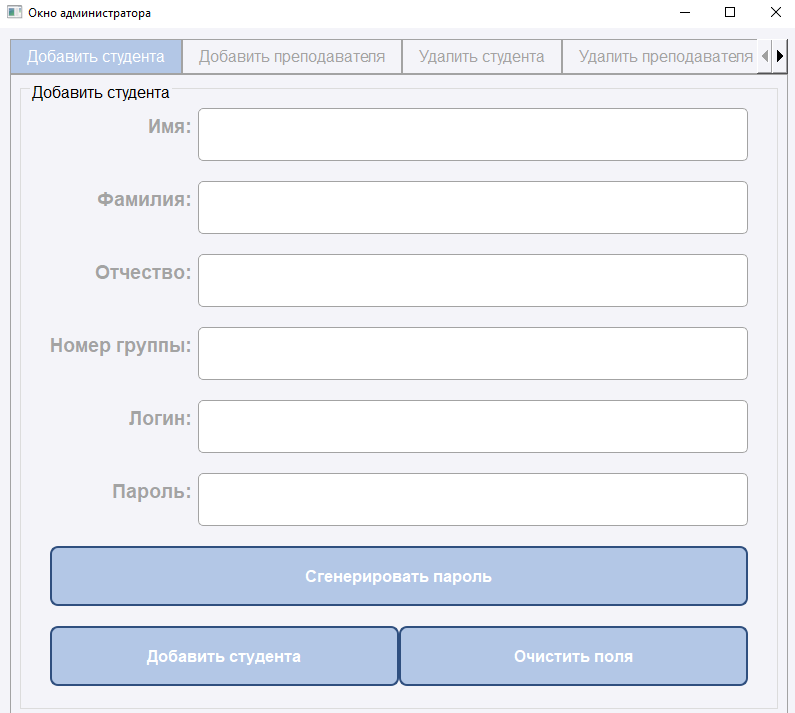


Рисунок 9 – Окно администратора, вкладка: «Добавить студента».

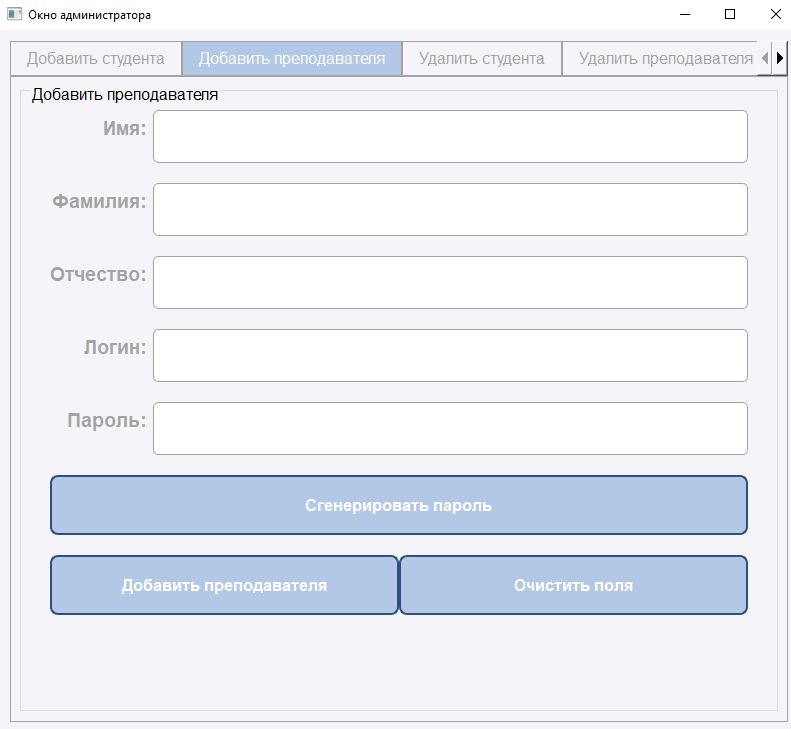


Рисунок 10 – Окно администратора, вкладка: «Добавить преподавателя».

Изображение выглядит как текст, снимок экрана, программное обеспечение, компьютер

Автоматически созданное описание

Рисунок 11 – Окно администратора, вкладка: «Удалить студента».

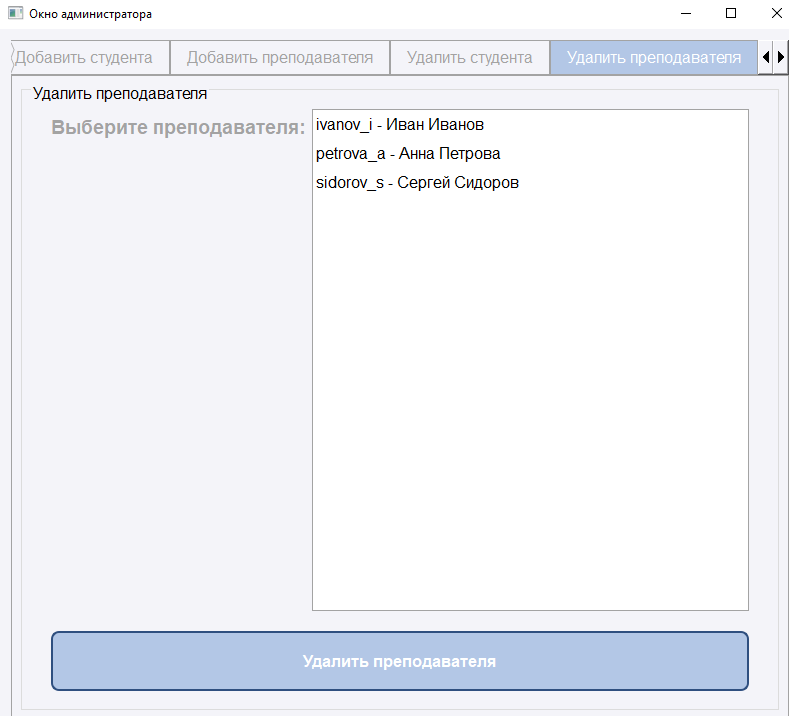


Рисунок 12 – Окно администратора, вкладка: «Удалить преподавателя».

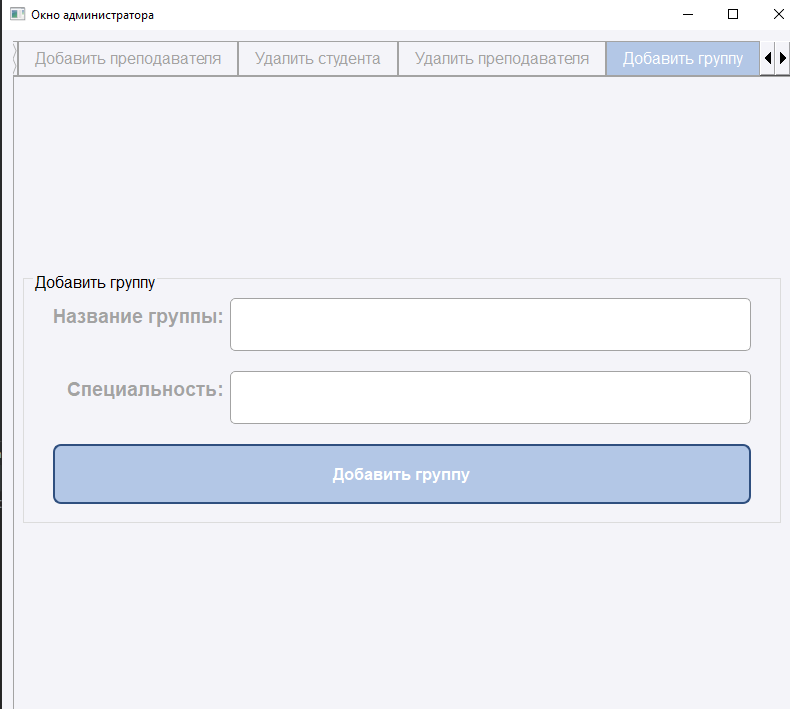


Рисунок 13 – Окно администратора, вкладка: «Добавить группу».

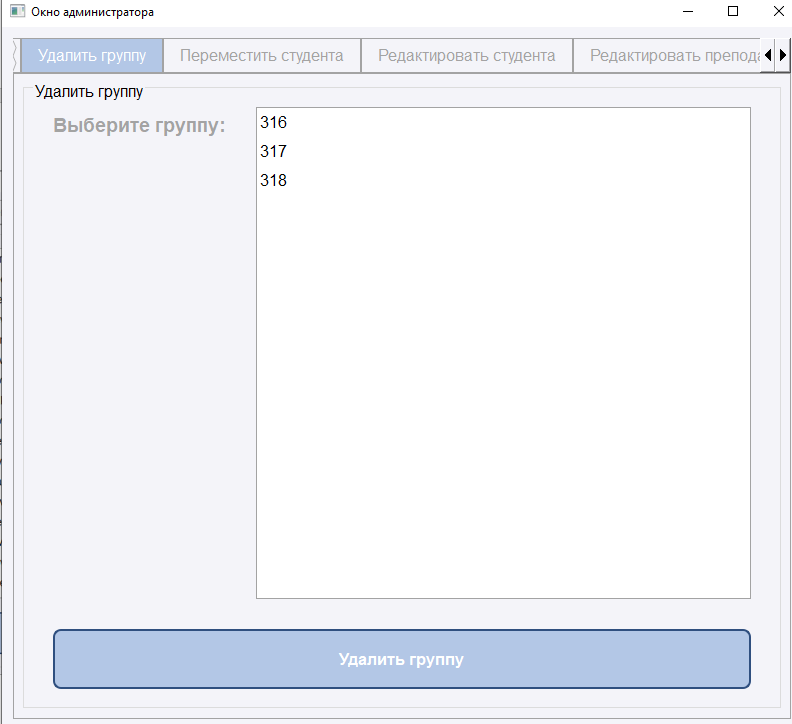


Рисунок 14 – Окно администратора, вкладка: «Удалить группу».

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Рисунок 15 – Окно администратора, вкладка: «Переместить студента».

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 16 – Окно администратора, вкладка: «Редактировать студента».

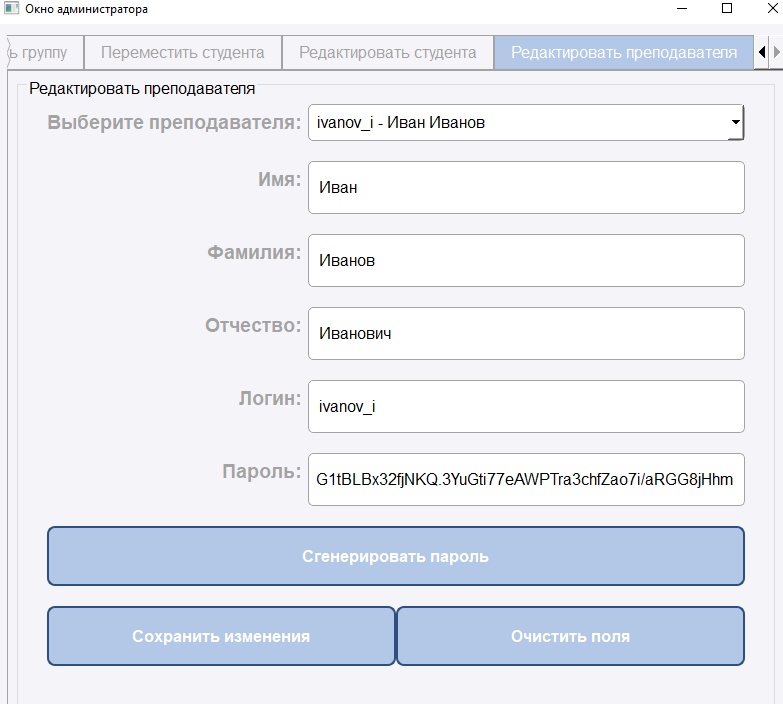


Рисунок 17 – Окно администратора, вкладка: «Редактировать преподавателя».



Рисунок 18 – Создание окна авторизации.

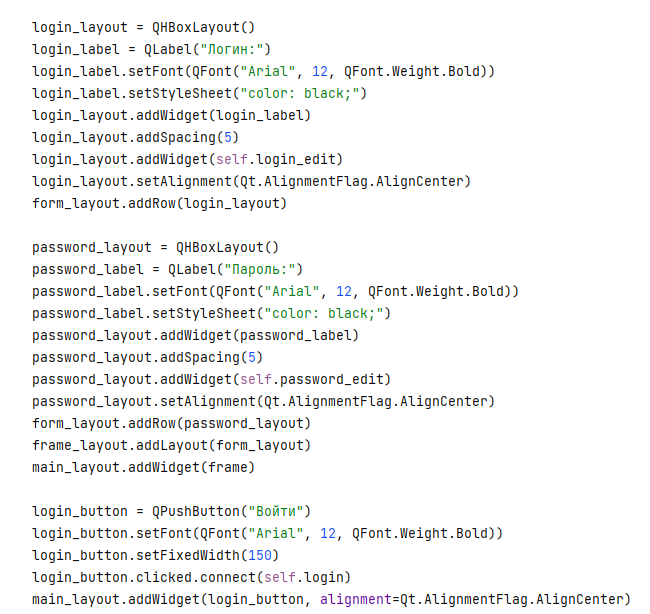


Рисунок 19 – Продолжение функции с созданием окна авторизации.



Рисунок 20 – Функция авторизации.



Рисунок 21 – Продолжение функции авторизации.



Рисунок 22 – Создание окна студента.

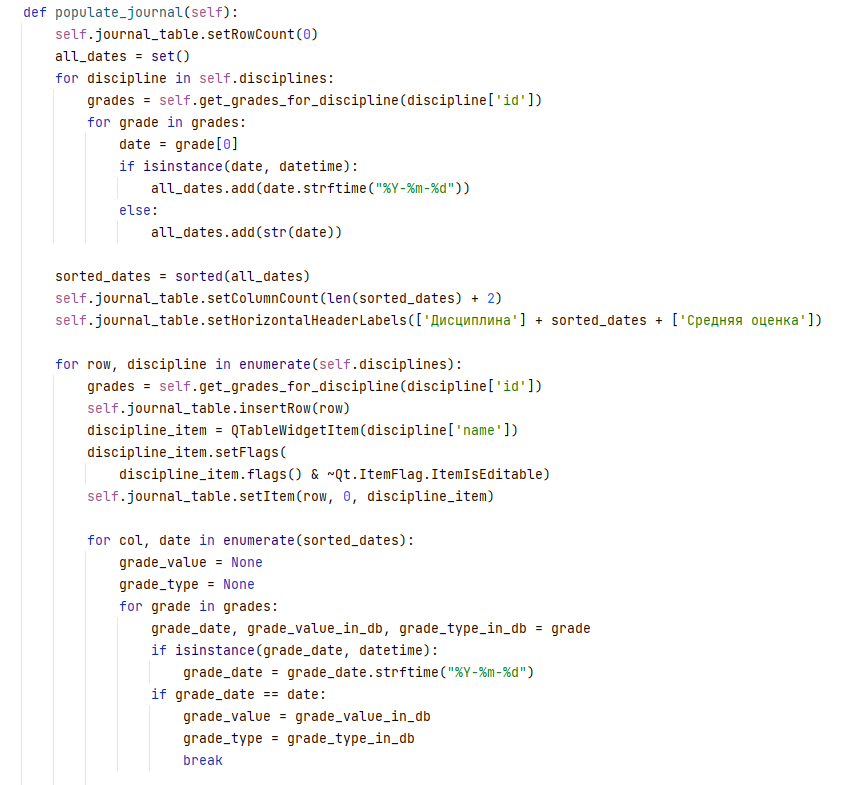


Рисунок 23 – Функция отображения дневника студента.



Рисунок 24 – Продолжения функции отображения дневника студента.

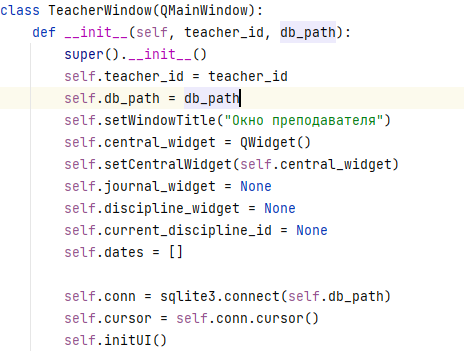


Рисунок 25 – Создание окна преподавателя.

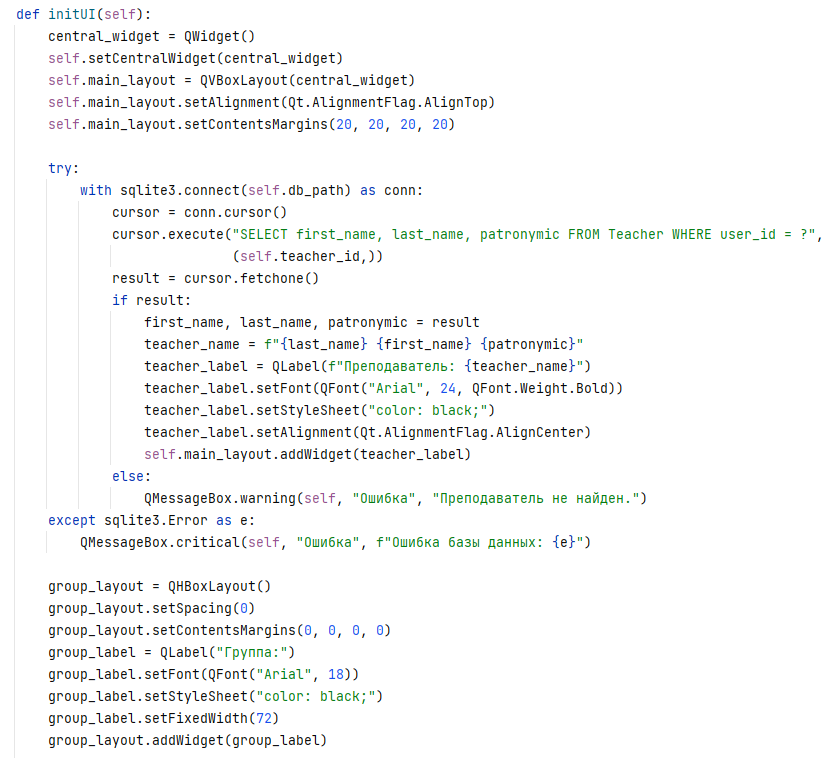


Рисунок 26 – Функция создания виджетов в окне преподавателя.

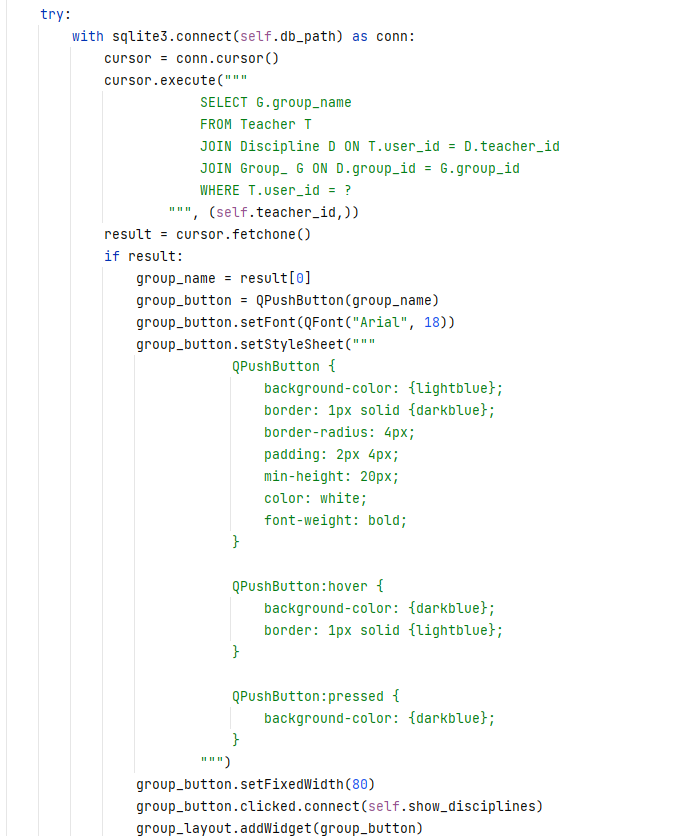


Рисунок 27 – Продолжение функции создания виджетов в окне преподавателя.

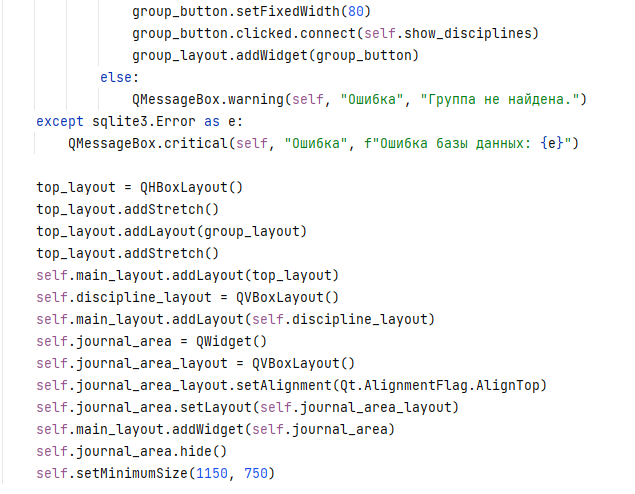


Рисунок 28 – Продолжение функции создания виджетов в окне преподавателя.

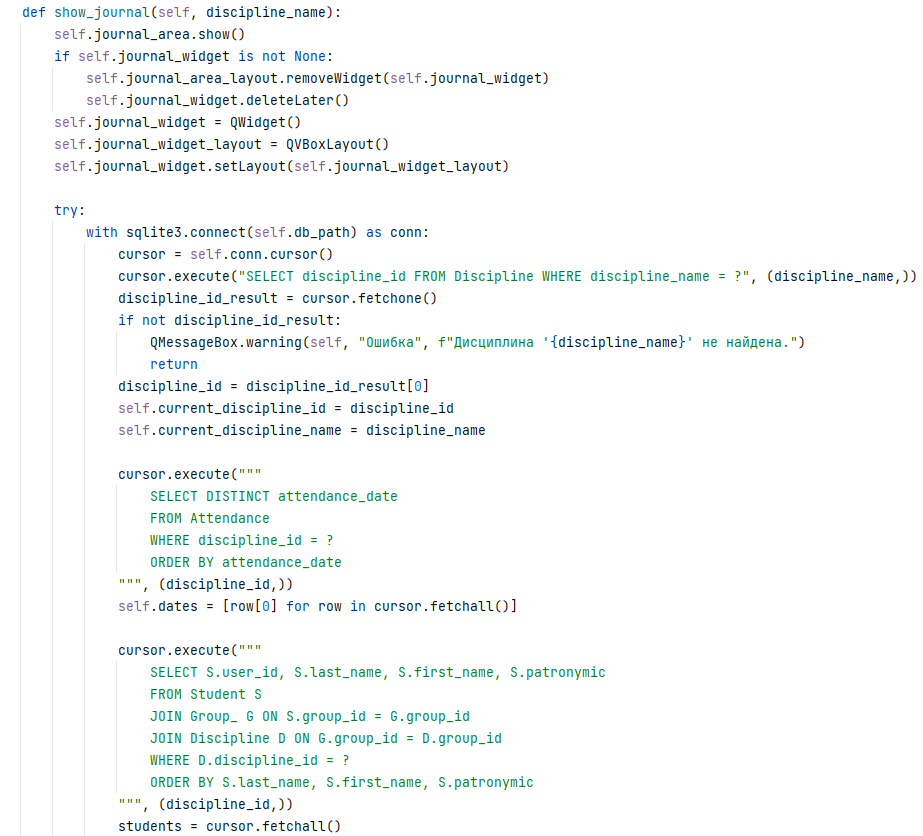


Рисунок 29 – Функция создания виджета журнала.

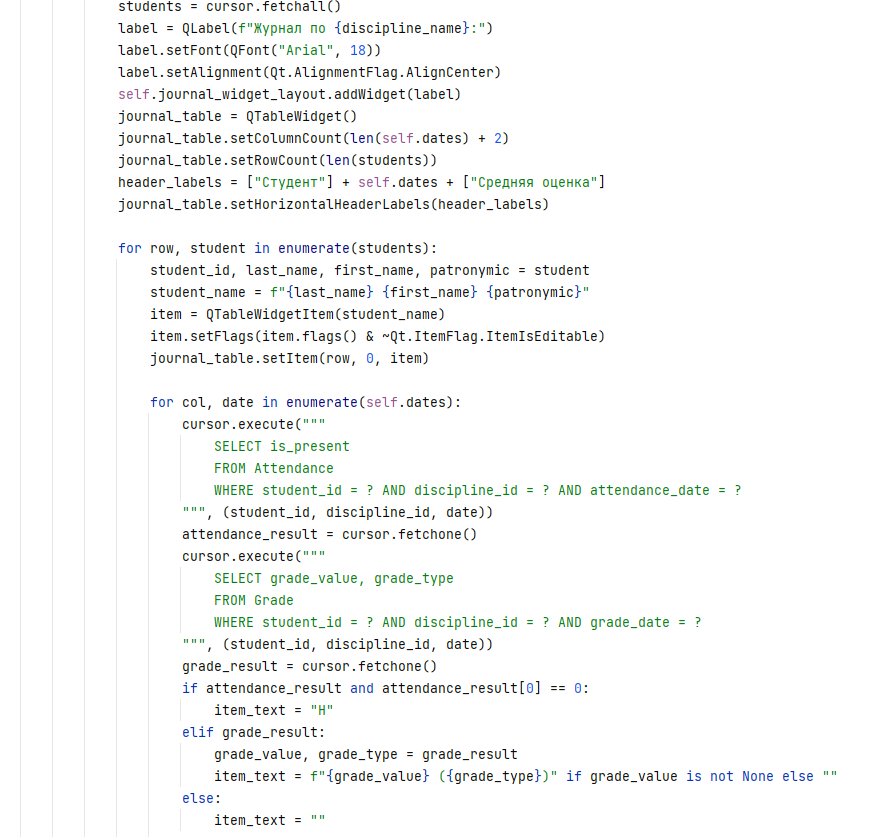


Рисунок 30 – Продолжение функции создания виджета журнала.



Рисунок 31 – Продолжение функции создания виджета журнала.



Рисунок 32 – Создание окна администратора.

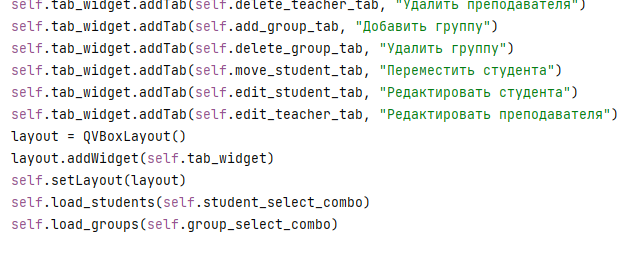


Рисунок 33 – Продолжение создания окна администратора.

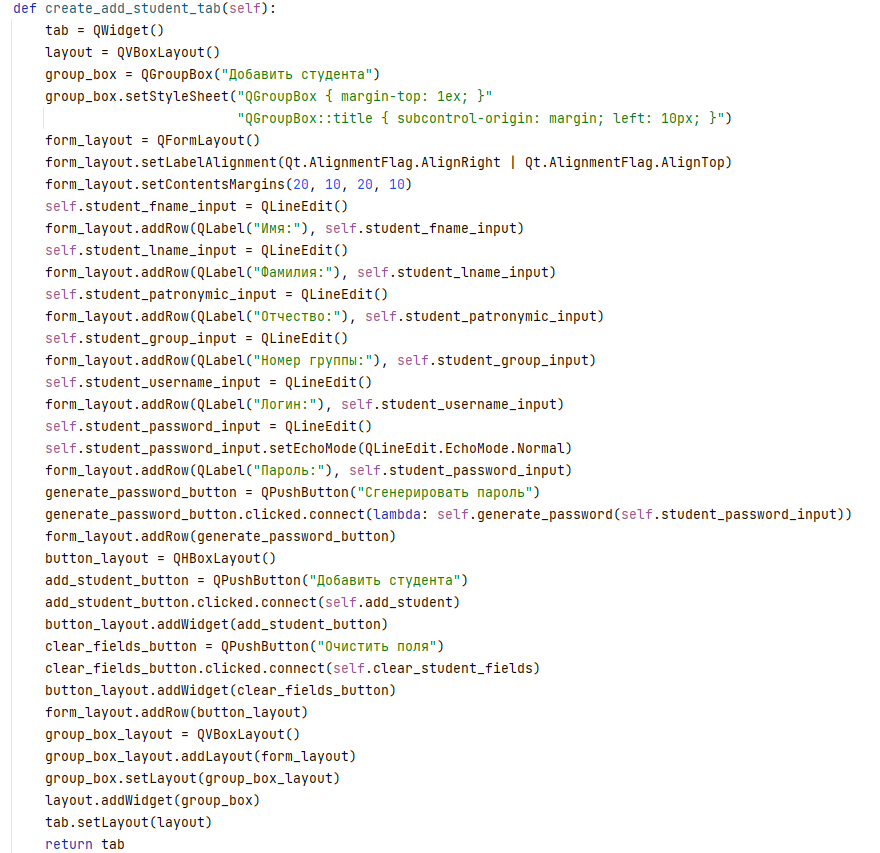


Рисунок 34 – Функция для отображения виджетов во вкладке добавления студента.

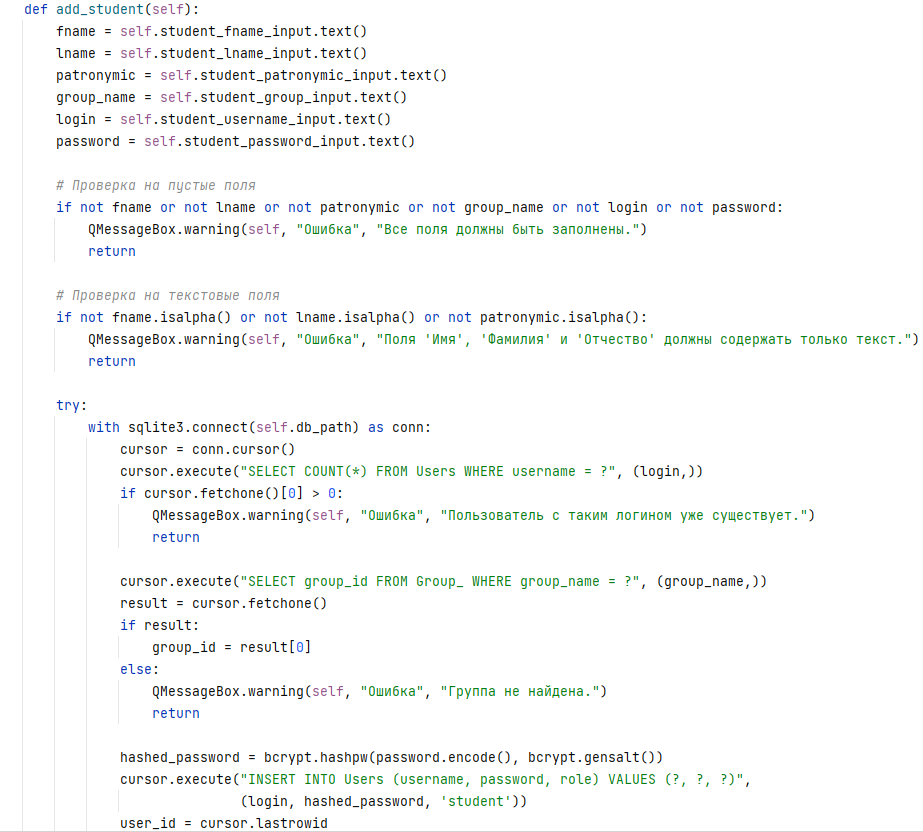


Рисунок 35 – Функция для добавления студента.

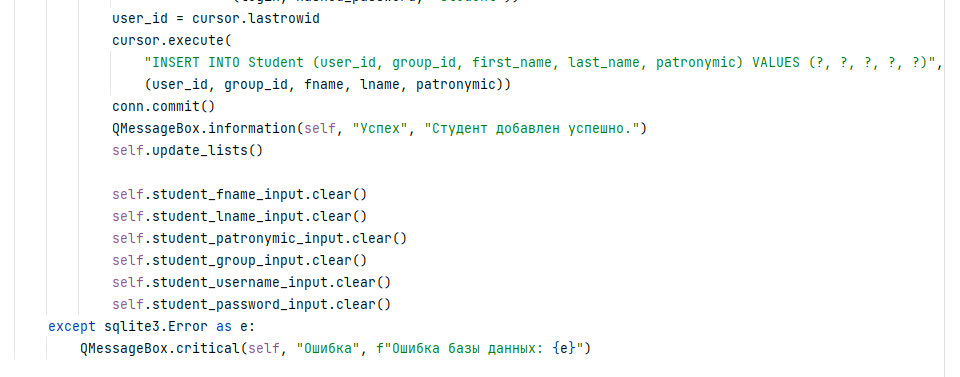


Рисунок 36 – Продолжение функции для добавления студента.

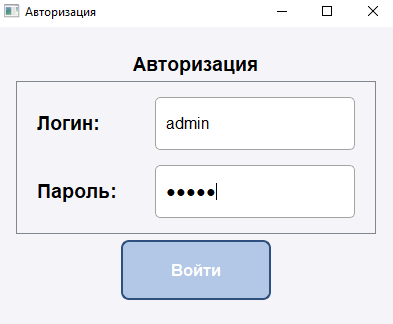


Рисунок 37 – Тестирование окна авторизации.

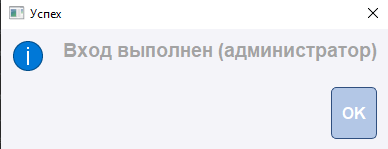


Рисунок 38 – Результат после ввода корректных данных для входа в окно администратора.

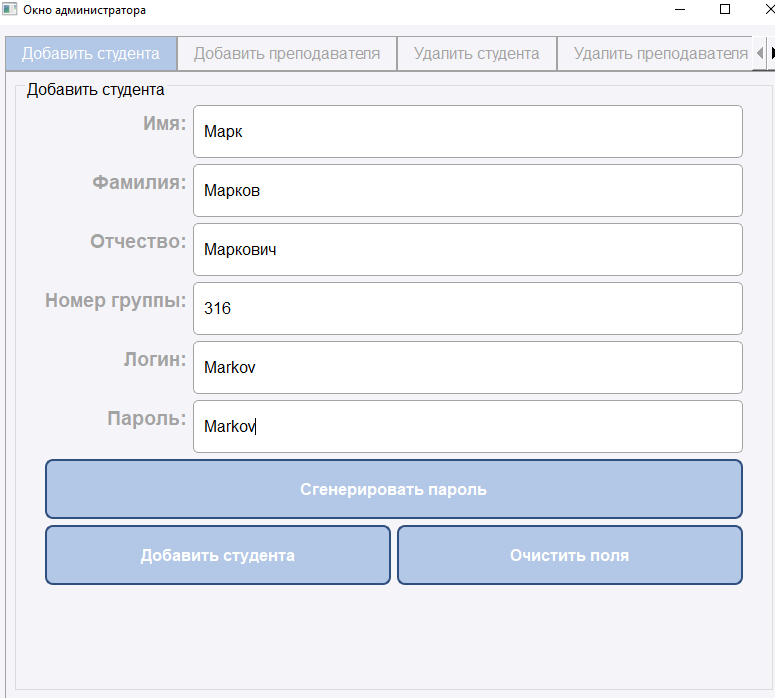


Рисунок 39 – Тестирование создания студента.

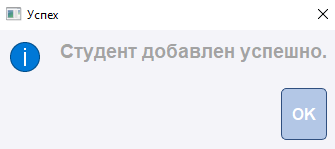


Рисунок 40 – Результат после создания студента.

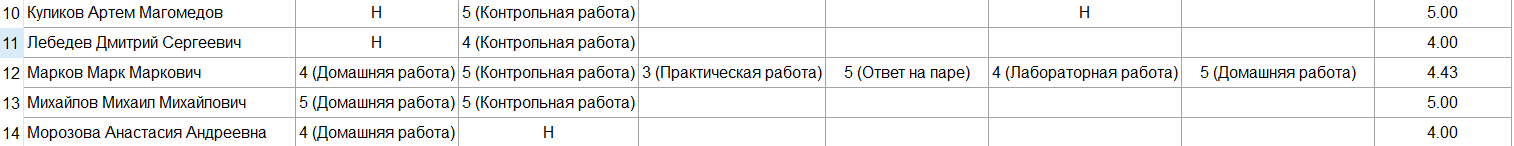
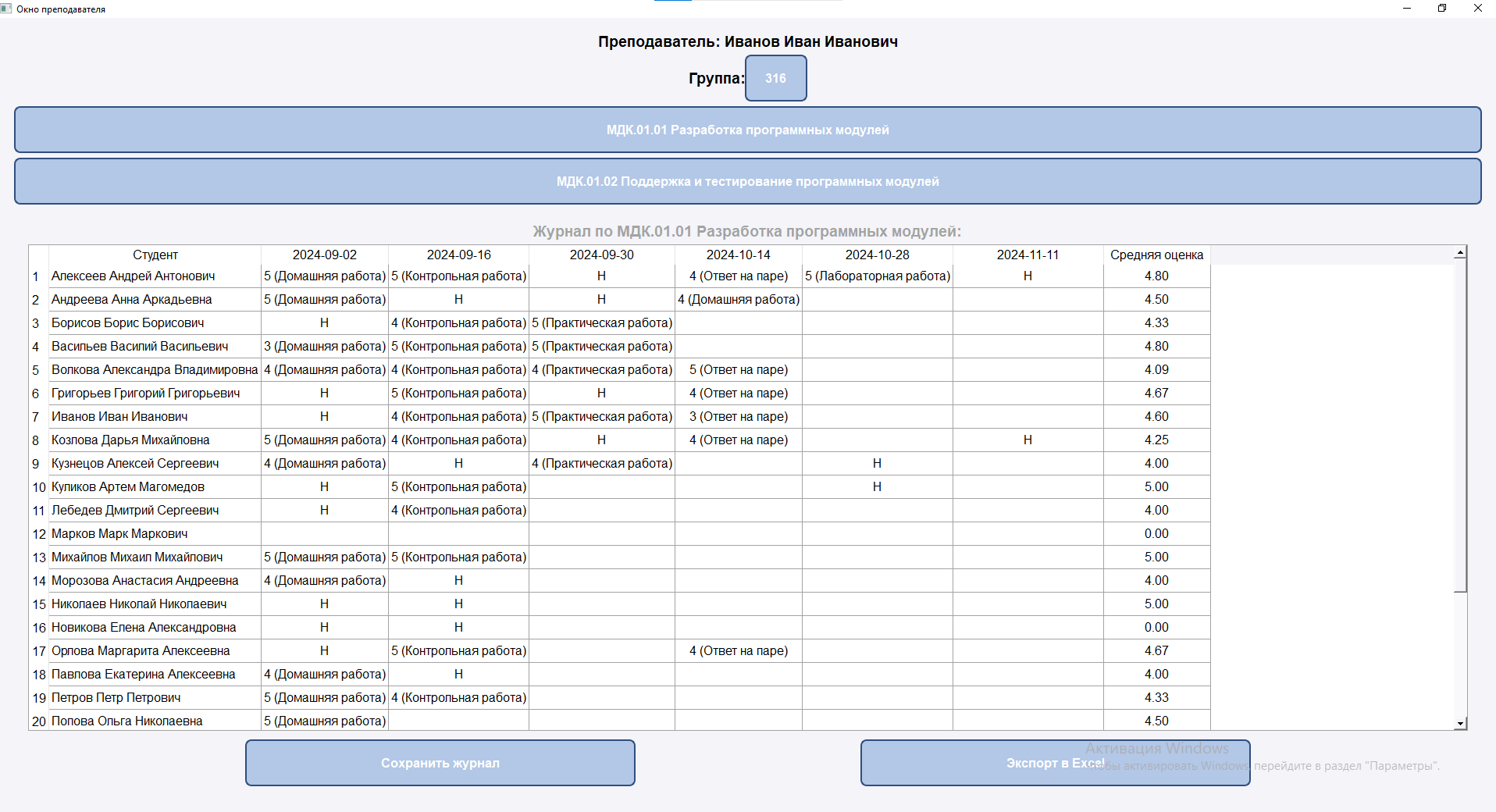
****Рисунок 41 – Результат после создания студента, студент отобразился в журнале под номером 12.

Рисунок 42 – Преподаватель успешно проставил оценки только что добавленному студенту.

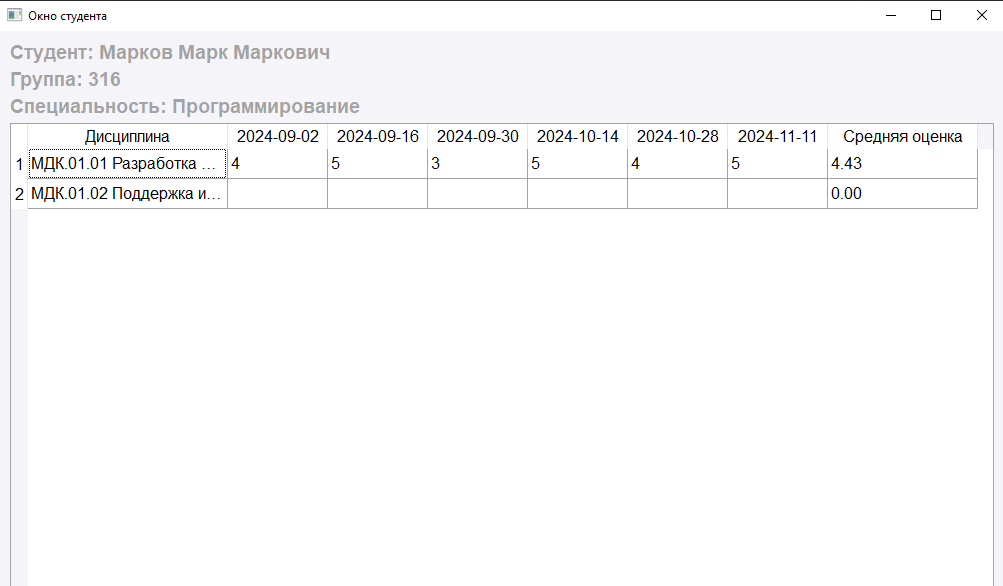


Рисунок 43 – Оценки отображаются у студента.