



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Экономики и управления на предприятии природопользования»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(бакалаврская работа)
по направлению подготовки 09.03.03 Прикладная информатика
(квалификация – бакалавр)

На тему «Разработка информационной системы «Электронная зачётка»

Исполнитель Шайков Михаил Васильевич

Руководитель к.г.н., доцент по кафедре Информатики и прикладной математики
Полупанов Владимир Николаевич

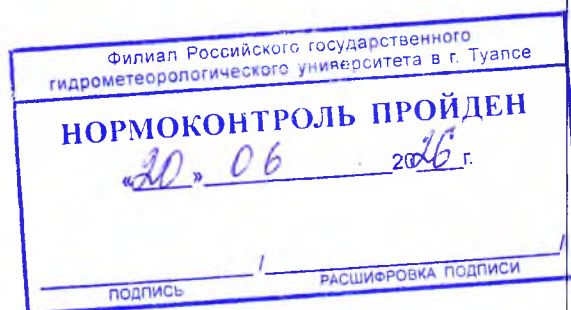
«К защите допускаю»

Руководитель кафедры _____

кандидат экономических наук

Майборода Евгений Викторович

«22» 06 2026 г.



Туапсе
2026

ОГЛАВЛЕНИЕ

Введение.....	4
1 Аналитическая часть.....	7
1.1 Анализ предметной области	7
1.2 Анализ существующего процесса учёта успеваемости студентов	8
1.3 Анализ проблем и обоснование необходимости автоматизации.....	10
1.4 Анализ существующих программных решений	11
1.5 Постановка задачи разработки информационной системы «Электронная зачётка»	15
2 Проектирование и разработка информационной системы	17
2.1 Обоснование выбора средств разработки.....	17
2.2 Проектирование базы данных.....	20
2.2.1 Структура базы данных.....	20
2.2.2 Логическая модель данных	22
2.3 Архитектура программного обеспечения.....	25
2.3.1 Общая структура проекта.....	25
2.3.2 Описание программных модулей	27
2.4 Реализация функциональных возможностей системы.....	29
2.4.1 Авторизация пользователя.....	29
2.4.2 Управление данными.....	32
2.4.3 Электронная зачётная книжка	35
2.4.4 Статистика и дополнительные возможности системы	36
2.5 Руководство пользователя.....	39
2.5.1 Описание интерфейса	39

2.5.2 Порядок работы с системой	42
3 Обоснование экономической эффективности результатов разработки	44
3.1 Выбор методики оценки эффективности	44
3.2 Расчёт показателей экономической эффективности	45
3.3 Анализ результатов внедрения информационной системы	48
Заключение	50
Список литературы	52

Введение

Характеризуясь высокой интенсивностью на современном этапе, внедрение цифровых решений в деятельность организаций различных сфер обеспечивает ускорение процессов обработки информации, снижение количества неточностей при выполнении операций и формирование удобного доступа к необходимым данным [7]. В образовательной среде, где постоянно осуществляется формирование, корректировка и анализ крупных информационных потоков, относящихся к учебной деятельности, роль автоматизации проявляется наиболее заметно.

Фиксация академической успеваемости учащихся представляет один из ключевых компонентов образовательного процесса. За период обучения накапливается существенный объем сведений: личные данные студентов, перечень осваиваемых курсов, отметки по экзаменам и зачетам, средний балл, информация о задолженностях и ходе выполнения учебного плана. К подобным сведениям предъявляются требования сохранения, систематического обновления, упорядочивания и предоставления пользователям в понятном виде.

Для работы с указанной информацией нередко применяются бумажные носители, табличные редакторы или разрозненные файлы, размещенные на персональных компьютерах сотрудников. Данным подходом обеспечивается осуществление основных учетных действий, однако с ростом числа учащихся и объема данных недостатки его становятся все более очевидными.

Затраты времени на поиск необходимых записей возрастают, усложняется проверка достоверности информации, увеличивается вероятность появления ошибок при ручном вводе и обработке.

В связи с этим возрастает востребованность профильных информационных систем, созданных для автоматизации учебных процессов. Результатами внедрения таких систем выступают организация единого хранилища данных, упрощение выполнения рутинных задач и оперативное получение требуемой информации.

Автоматизация снижает нагрузку на сотрудников, уменьшает количество рутинных действий и способствует повышению эффективности работы с учебными данными.

В рамках выпускной квалификационной работы рассматривается разработка информационной системы «Электронная зачётка», предназначенной для автоматизации учёта успеваемости студентов. Разрабатываемое приложение представляет собой настольную информационную систему для операционной системы Windows, созданную с использованием языка программирования Python и библиотеки PySide6. Хранение данных организовано на основе встроенной реляционной базы данных SQLite.

Функциональные возможности системы охватывают основные операции, необходимые для ведения сведений об успеваемости обучающихся. В приложении реализованы механизмы авторизации пользователей, инструменты управления информацией о студентах, дисциплинах и оценках, электронная зачётная книжка, средства расчёта среднего балла и определения академических задолженностей. Дополнительно предусмотрены формирование статистической информации, импорт и экспорт данных в формате Excel, экспорт зачётной книжки в PDF-документ, а также создание резервных копий базы данных.

Объектом исследования является процесс учёта успеваемости студентов в образовательной организации.

Предметом исследования выступают методы, модели и программные средства, используемые для автоматизации хранения, обработки и представления информации об учебных результатах обучающихся.

Цель выпускной квалификационной работы заключается в разработке информационной системы «Электронная зачётка», обеспечивающей автоматизацию учёта успеваемости студентов и повышение эффективности работы с учебными данными.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ предметной области и существующего процесса учёта успеваемости студентов;
- исследовать существующие программные решения и определить требования к разрабатываемой информационной системе;
- спроектировать структуру базы данных, архитектуру программного обеспечения и пользовательский интерфейс приложения;
- реализовать программные модули для работы со студентами, дисциплинами, оценками и электронной зачётной книжкой;
- выполнить оценку результатов разработки и определить экономическую эффективность внедрения информационной системы.

Практическая значимость работы определяется созданием функционирующего программного продукта, который может быть использован для автоматизации процессов учёта успеваемости студентов. Система создана прежде всего для небольших образовательных организаций и филиалов вузов, где с зачётными данными занимается ограниченное число сотрудников. Локальное настольное приложение помогает упростить организацию хранения данных, а также повседневные операции, без необходимости в отдельной серверной инфраструктуре.

1 Аналитическая часть

1.1 Анализ предметной области

Цифровизация образовательной сферы постепенно меняет подходы к организации учебного процесса. Если ещё сравнительно недавно большая часть документации велась на бумажных носителях, то сегодня всё чаще используются специализированные информационные системы, позволяющие хранить, обрабатывать и анализировать данные в электронном виде. Такой подход упрощает работу сотрудников образовательных организаций и обеспечивает более оперативный доступ к необходимой информации. Использование информационных систем позволяет организовать хранение, обработку и представление данных в форме, удобной для пользователей и управленческих процессов [10].

Одним из важных направлений деятельности любого учебного заведения является учёт успеваемости студентов. Этот процесс сопровождает обучающегося на протяжении всего периода освоения образовательной программы и включает накопление, хранение и обработку сведений о результатах обучения.

Учёт успеваемости используется для фиксации результатов обучения и контроля выполнения учебного плана. Эти сведения нужны студентам, преподавателям и сотрудникам учебных подразделений.

С каждым годом обучения увеличивается количество информации о студентах, их оценки, зачёты и т.п. При большом количестве обучающихся обработка такой информации без применения специализированных программных средств становится трудоёмкой задачей, требующей значительных временных затрат.

Традиционно сведения об успеваемости могут храниться в бумажных зачётных книжках, экзаменационных ведомостях, журналах учёта или электронных таблицах. Несмотря на простоту такого подхода, он имеет ряд недостатков.

В рассматриваемой предметной области важную роль играет зачётная книжка. В ней находится информация обо всех пройденных дисциплинах и полученных результатах аттестации.

Учебные заведения стремятся сократить объём бумажного документооборота и применять информационные системы, которые позволяют централизованно хранить данные, быстро находить нужные сведения и автоматизировать повторяющиеся операции. Использование таких решений повышает точность обработки информации и уменьшает время, необходимое для выполнения повседневных задач.

Предметная область данной работы включает процессы, связанные с учётом успеваемости студентов в образовательной организации. Анализ показывает, что качественная работа с учебными данными невозможна без применения современных информационных технологий, обеспечивающих надёжное хранение, быструю обработку и удобное представление информации пользователям. Поэтому создание специализированной информационной системы «Электронная зачётка» является актуальной задачей, направленной на улучшение организации учёта результатов обучения.

1.2 Анализ существующего процесса учёта успеваемости студентов

Учёт успеваемости студентов можно рассматривать как постоянный процесс, включающий сбор, хранение, обработку и последующий анализ сведений о результатах обучения. От того, насколько грамотно организована эта работа, зависит достоверность информации об учебных достижениях обучающихся, а также скорость и качество подготовки учебной и отчётной документации.

Данный процесс опирается на несколько групп данных. В первую очередь к ним относятся сведения о студентах: персональная информация, учебная группа и курс обучения.

Работа с учётом начинается с внесения информации о составе обучающихся. Для каждого студента формируется отдельная запись, включающая данные, необходимые для дальнейшей работы. Одновременно создаётся список дисциплин. По каждой дисциплине указываются основные данные - название, курс, семестр и тип зачёта.

На основе результатов обучения формируются сведения об успеваемости студентов за выбранный период, рассчитываются необходимые показатели, подготавливаются документы и отчёты. Чем больше становится объём данных, тем выше требования к скорости их обработки и удобству доступа к ним.

Одним из центральных элементов этого процесса является ведение зачётной книжки. В ней отражаются дисциплины, изученные студентом, результаты экзаменов и зачётов, а также иные сведения, связанные с выполнением учебного плана. Именно зачётная книжка позволяет получить общее представление об образовательных результатах конкретного обучающегося.

Важную роль играет и анализ успеваемости. Сотрудникам образовательной организации необходимо не просто хранить данные, но и быстро оценивать результаты обучения. Для этого определяется средний балл, выявляются неудовлетворительные оценки и устанавливается наличие академических задолженностей. Полученная информация используется при организации учебного процесса и контроле качества подготовки студентов.

В процессе учёта отдельные элементы системы постоянно взаимодействуют между собой. Данные о студентах связаны с дисциплинами, дисциплины — с результатами аттестации, а сведения об обучении применяются при формировании зачётных книжек и отчётных документов. Такая взаимосвязь требует поддержания целостности данных и исключения противоречий между различными информационными объектами.

На практике учёт успеваемости часто ведётся сразу в нескольких источниках.

1.3 Анализ проблем и обоснование необходимости автоматизации

Одна из главных проблем - раздельность данных. Информация о студентах, дисциплинах и оценках находится в разных документах, а иногда и на разных устройствах. Поиск нужной информации занимает больше времени, а риск ошибок между источниками увеличивается. Приходится вручную проверять данные на правильность и актуальность.

Проблемы возникают и при внесении новой информации. Большое количество информации делает изменения более трудозатратными по времени, а также повышают риск ошибки, что может повлечь неразбериху при сверке данных.

Существенным недостатком традиционной организации учёта остаётся необходимость вручную выполнять различные расчёты. Чтобы определить средний балл, проанализировать результаты обучения или выявить академические задолженности, пользователю приходится самостоятельно обрабатывать данные либо обращаться к дополнительным программным средствам.

Проблемы возникают и на этапе подготовки итоговых документов.

Для зачётной книжки необходимо собирать данные из разных источников, проверять и систематизировать. Если делать это вручную – это займёт много времени, особенно при большом количестве студентов, а любые изменения требуют вложения сил и времени.

Бумажные документы, а также разные файлы для хранения информации также не лишены недостатков – бумажные документы могут быть повреждены, либо же утеряны. Отдельные файлы не гарантируют достаточный уровень защиты, сохранение данных, а также возможность резервного копирования информации.

Чаще всего они проявляются одновременно и оказывают комплексное влияние на организацию учёта успеваемости. В результате сотрудники тратят больше времени на выполнение рутинных операций, снижается оперативность

работы с данными и возрастает вероятность возникновения ошибок при обработке информации.

Устранить перечисленные недостатки можно за счёт внедрения специализированной информационной системы. Проектирование такой системы включает формализацию предметной области, определение состава обрабатываемых данных и подбор средств автоматизации прикладных процессов [1]. Единая база данных позволяет централизованно хранить сведения о студентах, учебных дисциплинах и результатах обучения. Автоматизация основных операций уменьшает объём ручной работы, ускоряет поиск нужной информации и повышает точность обработки записей.

Все эти недостатки подтверждают, что для ведения данных о зачётах требуется отдельное приложение. ИС «Электронная зачётка» поможет хранить информацию о студентах, предметах и оценках в одном месте, сократить объём ручной работы и повысить удобство обработки учебной информации.

1.4 Анализ существующих программных решений

Перед разработкой собственной системы необходимо рассмотреть решения, которые уже применяются в образовательной сфере. Это позволяет понять, какие задачи они закрывают, в чём их преимущества и почему не каждое из таких решений подходит для ведения электронной зачётной книжки.

Сравнение аналогов помогает уточнить требования к разрабатываемому приложению и определить, какие функции должны быть реализованы в первую очередь [3].

В качестве одного из примеров можно рассмотреть Moodle. Эта платформа хорошо известна в образовательных организациях и чаще всего используется для дистанционного или смешанного обучения.

В ней можно размещать учебные материалы, создавать задания и тесты, контролировать выполнение работ и организовывать взаимодействие между преподавателями и студентами.

Популярность Moodle во многом объясняется её гибкостью и широкими возможностями настройки. Пользователи могут адаптировать систему под особенности конкретной образовательной организации, подключать дополнительные модули и расширять базовый функционал. Платформа поддерживает большое количество образовательных сценариев и активно развивается благодаря открытому программному коду.

Однако при решении задач, связанных непосредственно с ведением зачётных данных, возможности Moodle оказываются не всегда удобными. Основной акцент в системе сделан на организацию учебного процесса и электронного обучения.

Функции централизованного ведения электронной зачётной книжки, оперативного контроля академических задолженностей и быстрого формирования сведений об успеваемости не относятся к её ключевым направлениям использования.

Другой распространённой категорией программных продуктов являются решения семейства 1С:Университет и 1С:Колледж. Эти системы предназначены для комплексной автоматизации деятельности образовательных учреждений и охватывают широкий спектр задач: управление контингентом обучающихся, формирование расписания занятий, учёт результатов аттестации, подготовку отчётности и сопровождение учебного процесса.

К преимуществам решений на платформе 1С относятся высокая функциональная насыщенность, возможность интеграции с другими информационными системами и соответствие требованиям образовательных организаций. Пользователи получают единое программное пространство для решения большинства административных и учебных задач.

Вместе с тем подобные системы ориентированы на комплексную автоматизацию деятельности образовательного учреждения. Их внедрение требует значительных организационных и финансовых ресурсов. Необходимы приобретение лицензий, настройка программного обеспечения, обучение персонала и дальнейшее сопровождение системы. Для решения сравнительно

узкой задачи учёта успеваемости студентов такой подход может оказаться избыточным.

Помимо широко распространённых программных комплексов многие образовательные организации используют собственные локальные решения. В ряде случаев для хранения информации применяются электронные таблицы, специализированные базы данных или небольшие приложения, разработанные под внутренние потребности конкретного учреждения. Такие системы обычно отличаются простотой внедрения и не требуют значительных затрат на эксплуатацию.

В результате система становится более сложной в освоении и эксплуатации, чем это требуется для выполнения повседневных операций по работе с зачётными данными.

С другой стороны, простые локальные решения не всегда обеспечивают необходимый уровень автоматизации и удобства работы. Часто им не хватает средств централизованного хранения информации, автоматического расчёта показателей успеваемости, формирования электронной зачётной книжки и подготовки документов в распространённых форматах.

Для обобщения результатов анализа существующих программных решений целесообразно представить их основные характеристики в сравнительном виде.

Сравнение выполнено по общему назначению системы, характеру использования и применимости к задаче ведения зачётных данных.

Таблица 1 - Характеристика существующих программных решений

Программное решение	Общее назначение	Основные возможности	Особенности применения
Moodle	Электронное обучение	Учебные материалы, тестирование, задания, взаимодействие участников	Ориентирована на дистанционное обучение

Продолжение таблицы 1

1С:Университет	Автоматизация деятельности вуза	Контингент, учебный процесс, аттестация, отчётность	Используется для широкого круга задач
1С:Колледж	Автоматизация СПО	Обучающиеся, расписание, аттестация, отчётность	Предназначена для комплексной автоматизации
«Электронная зачётка»	Учёт успеваемости студентов	Студенты, дисциплины, оценки, зачётная книжка, экспорт данных	Ориентирована на ведение зачётных данных

Как видно из таблицы 1, рассмотренные программные решения имеют различное назначение.

Moodle ориентирована преимущественно на организацию электронного обучения, решения семейства 1С применяются для комплексной автоматизации образовательных организаций, а информационная система «Электронная зачётка» предназначена для решения более узкой задачи — ведения и обработки зачётных данных.

Таким образом, результаты проведённого анализа подтверждают целесообразность разработки специализированного программного продукта, ориентированного на учёт успеваемости студентов.

Такая система должна сочетать простоту эксплуатации, минимальные требования к программно-техническому обеспечению и достаточный набор функций для хранения, обработки и анализа учебных данных.

1.5 Постановка задачи разработки информационной системы «Электронная зачётка»

Проведённый анализ предметной области показал, что процессы учёта успеваемости студентов связаны с обработкой большого объёма информации. В образовательной организации необходимо хранить сведения об обучающихся, учебных дисциплинах, результатах промежуточной аттестации, среднем балле и академических задолженностях. При использовании бумажной документации, электронных таблиц или разрозненных файлов выполнение этих операций требует дополнительных временных затрат и сопровождается риском возникновения ошибок.

Для устранения выявленных недостатков поставлена задача разработки информационной системы «Электронная зачётка», предназначенной для автоматизации хранения, обработки и представления сведений об успеваемости студентов. При постановке задачи разработки информационной системы необходимо определить её назначение, состав обрабатываемых данных, основные функции и требования к программной реализации [2].

Разрабатываемая система должна функционировать в виде настольного приложения для операционной системы Windows. Такой подход обеспечивает простоту эксплуатации и позволяет использовать программный продукт без развёртывания сложной серверной инфраструктуры.

Хранение информации должно осуществляться в локальной базе данных, обеспечивающей быстрый доступ к сведениям и их централизованную обработку.

Основным пользователем системы является сотрудник образовательной организации, работающий с данными об успеваемости студентов.

В процессе эксплуатации пользователь должен иметь возможность выполнять полный цикл операций по ведению учебной информации: добавлять, редактировать и удалять сведения о студентах, дисциплинах и оценках, а также использовать поиск и фильтрацию данных.

Одной из ключевых функций системы является электронная зачётная книжка. На основании сведений, содержащихся в базе данных, приложение должно автоматически формировать информацию об успеваемости выбранного студента, отображать перечень дисциплин, результаты аттестации, средний балл и сведения о наличии академических задолженностей.

Для повышения удобства работы необходимо реализовать импорт и экспорт данных в формате Microsoft Excel, а также экспорт электронной зачётной книжки в PDF-документ. Дополнительно предусматриваются отображение статистической информации, создание резервных копий базы данных, поддержка различных тем оформления интерфейса и другие сервисные функции.

Таким образом, задача разработки информационной системы «Электронная зачётка» заключается в создании программного продукта, обеспечивающего автоматизацию основных процессов учёта успеваемости студентов. Реализация системы должна способствовать сокращению трудозатрат пользователей, повышению точности обработки данных и улучшению качества работы с учебной информацией в образовательной организации.

2 Проектирование и разработка информационной системы

2.1 Обоснование выбора средств разработки

Выбор технологий и программных средств является одним из наиболее ответственных этапов создания любой информационной системы. От используемых инструментов зависят не только функциональные возможности будущего программного продукта, но и удобство его сопровождения, производительность, надёжность и перспективы дальнейшего развития. При разработке информационной системы «Электронная зачётка» выбор технологий осуществлялся с учётом особенностей поставленной задачи, требований к простоте эксплуатации, доступности программного обеспечения и возможности реализации всего необходимого функционала. Выбор средств разработки должен соответствовать назначению системы, составу решаемых задач и условиям её дальнейшей эксплуатации [8].

В качестве основного языка программирования был выбран Python версии 3.12. Python широко применяется при разработке прикладного программного обеспечения благодаря читаемому синтаксису, развитой экосистеме библиотек и удобству сопровождения программного кода [19]. На сегодняшний день Python является одним из самых востребованных языков разработки благодаря сочетанию простого синтаксиса, широких функциональных возможностей и большого количества готовых библиотек. Его активно используют при создании настольных приложений, веб-сервисов, информационных систем и программ автоматизации. Для данного проекта Python оказался особенно подходящим благодаря высокой скорости разработки и удобству сопровождения исходного кода. Использование Python позволяет разрабатывать прикладные программы с понятной структурой и эффективно применять сторонние библиотеки для решения практических задач [21]. Это позволило сосредоточиться непосредственно на реализации бизнес-логики приложения, не затрачивая значительных ресурсов на решение низкоуровневых задач. В официальной документации Python подробно представлены

стандартные возможности языка и встроенные библиотеки, которые применяются при создании прикладного программного обеспечения [23].

Для разработки современного пользовательского интерфейса потребовалось выбрать устойчивый и функциональный графический фреймворк. В качестве такого инструмента была использована библиотека PySide6 — официальный набор привязок Python к платформе Qt. Документация Qt for Python описывает возможности PySide6 при создании графических интерфейсов и взаимодействии с компонентами Qt в Python-приложениях [24]. Данная библиотека предоставляет разработчику широкий набор средств для создания настольных программ и позволяет использовать таблицы, кнопки, выпадающие списки, диалоговые окна и другие элементы управления, необходимые для удобной работы пользователя.

Выбор PySide6 был обусловлен рядом факторов. Во-первых, библиотека обеспечивает стабильную работу интерфейса и обладает достаточной производительностью для настольного приложения. Во-вторых, Qt относится к числу наиболее распространённых фреймворков для разработки desktop-программ, поэтому по нему доступна качественная документация, примеры реализации и обучающие материалы. Наконец, использование PySide6 позволило реализовать современный интерфейс с поддержкой различных тем оформления и удобной навигацией между разделами системы.

Для хранения данных была выбрана система управления базами данных SQLite. В рамках рассматриваемой задачи использование полноценного серверного решения оказалось нецелесообразным, поскольку информационная система ориентирована на автономную работу на персональном компьютере пользователя. SQLite представляет собой встроенную реляционную базу данных, которая не требует отдельной установки и настройки серверных компонентов. Согласно официальной документации, SQLite хранит базу данных в обычном файле и применяется как встраиваемая СУБД для локальных приложений [27]. Вся информация хранится в одном файле, что существенно упрощает развёртывание и перенос приложения между устройствами.

Дополнительным преимуществом SQLite является поддержка языка SQL и механизмов обеспечения целостности данных. Описание синтаксиса SQL, поддерживаемого SQLite, представлено в официальном справочнике языка SQLite [28]. Несмотря на простоту использования, данная СУБД обладает достаточной производительностью для хранения сведений о студентах, дисциплинах и результатах обучения. Для задач, решаемых информационной системой «Электронная зачётка», её возможностей более чем достаточно.

В процессе эксплуатации приложения пользователям необходимо работать с электронными таблицами Microsoft Excel. Для реализации функций импорта и экспорта данных была выбрана библиотека OpenPyXL. Данная библиотека предназначена для чтения и записи файлов Excel формата XLSX средствами Python [29]. Она позволяет создавать, изменять и считывать файлы формата XLSX непосредственно средствами Python без необходимости установки дополнительных программных компонентов. Использование OpenPyXL обеспечивает удобный обмен информацией между системой и внешними документами, а также значительно упрощает первоначальное заполнение базы данных.

Для формирования электронных документов в формате PDF использовалась библиотека ReportLab. Инструменты ReportLab позволяют программно создавать PDF-документы, включая текстовые блоки, таблицы и другие элементы оформления [30]. Выбор данного инструмента связан с необходимостью автоматического создания электронной зачётной книжки студента. Библиотека предоставляет широкие возможности по программному формированию документов различной структуры, включая текстовые блоки, таблицы и элементы оформления. Благодаря этому удалось реализовать функцию экспорта зачётной книжки в удобный для хранения, передачи и печати формат.

Завершающим этапом разработки стала подготовка приложения к распространению и эксплуатации на компьютерах пользователей. Для подготовки приложения к дальнейшему использованию была применена

утилита PyInstaller. С её помощью проект, разработанный на Python, был преобразован в отдельный исполняемый файл формата EXE. Благодаря этому пользователю не нужно самостоятельно устанавливать интерпретатор Python, подключать библиотеки или настраивать дополнительные зависимости: приложение можно запускать сразу после установки.

Также рассматривалась разработка приложения средствами VBA и Microsoft Access. Этот подход вполне может использоваться для небольших ИС, но имеет зависимость от установленного офисного ПО и создаёт ограничения дальнейшего изменения архитектуры.

Использование Python, PySide6 и SQLite позволяет создать отдельное приложение, не требующее дополнительных программ на рабочем компьютере. Работа с Excel-файлами выполняется благодаря средствам библиотеки OpenPyXL, а само приложение является EXE-файлом, что упрощает распространение ПО.

Выбранный технологический стек соответствует назначению проекта и требованиям, предъявляемым к разрабатываемой системе. Python обеспечил удобство разработки и последующего сопровождения программного кода, PySide6 позволил создать современный графический интерфейс, а SQLite — организовать надёжное локальное хранение данных.

2.2 Проектирование базы данных

2.2.1 Структура базы данных

От качества проектирования БД напрямую зависят скорость обработки данных, удобство выполнения запросов, надёжность хранения информации и возможность дальнейшего расширения функциональности системы.

При разработке структуры базы данных был проведён анализ предметной области, в результате которого были определены основные информационные объекты, участвующие в процессе учёта успеваемости студентов. Определение информационных объектов предметной области является одним из основных

этапов проектирования информационной системы [9]. Исследование показало, что для решения поставленных задач необходимо организовать хранение сведений об обучающихся, учебных дисциплинах и результатах промежуточной аттестации. На основании этого были выделены три основные сущности: «Студенты», «Дисциплины» и «Оценки».

В качестве системы управления базами данных используется SQLite. Выбор СУБД при разработке информационной системы должен учитывать объём данных, характер выполняемых операций, требования к развёртыванию и условия эксплуатации программного продукта [18]. Данное решение позволяет хранить всю информацию в одном файле базы данных, не требует установки дополнительного серверного программного обеспечения и обеспечивает достаточную производительность для выполнения операций, предусмотренных функциональными возможностями приложения. Такой подход упрощает процесс развёртывания системы и делает её полностью автономной.

Центральное место в структуре базы данных занимает таблица студентов. Она предназначена для хранения основной информации об обучающихся и содержит сведения, необходимые для их идентификации и дальнейшей работы с результатами обучения. Для каждого студента в базе данных сохраняется уникальный идентификатор, фамилия, имя, отчество, учебная группа и курс обучения. Наличие уникального идентификатора позволяет однозначно связывать студента с другими объектами системы и обеспечивает корректную работу механизмов хранения данных.

Следующим элементом структуры является таблица дисциплин. Её назначение заключается в хранении сведений об учебных предметах, изучаемых студентами в рамках образовательной программы. Для каждой учебной дисциплины в базе сохраняются основные характеристики: название, курс, семестр изучения и форма контроля. Размещение дисциплин в отдельной таблице позволяет не повторять одни и те же сведения в разных записях и

использовать их централизованно во всех функциональных разделах приложения.

С точки зрения учёта успеваемости ключевое значение имеет таблица оценок. В ней фиксируются результаты промежуточной аттестации студентов. Каждая запись включает сведения о конкретном обучающемся, дисциплине, выставленной оценке и дате её получения.

При проектировании базы данных важно было не только распределить сведения по таблицам, но и сохранить связь между ними. Для этого в каждой таблице используется поле `id`, которое позволяет однозначно определить конкретную запись, что снижает риск допущения ошибок.

В базе данные разделены на три самостоятельные таблицы: студенты, дисциплины и оценки. Благодаря этому сведения о дисциплине или студенте не повторяются в каждой записи об оценке. Например, дисциплина добавляется в справочник один раз, а затем используется при выставлении оценок разным студентам.

На её основе реализованы управление студентами, дисциплинами и оценками, формирование электронной зачётной книжки, расчёт среднего балла, определение академических задолженностей, получение статистических сведений, а также импорт и экспорт данных.

Таким образом, спроектированная структура базы данных полностью соответствует особенностям рассматриваемой предметной области и обеспечивает надёжное хранение информации об успеваемости студентов. Выбранный подход позволяет эффективно организовать работу с данными, поддерживать их целостность и создавать основу для дальнейшего развития информационной системы «Электронная зачётка».

2.2.2 Логическая модель данных

После определения структуры базы данных следующим этапом проектирования стала разработка логической модели данных. Её назначение

заключается в формальном описании основных сущностей предметной области, их атрибутов и взаимосвязей. Грамотно построенная логическая модель позволяет обеспечить целостность информации, исключить избыточное хранение данных и создать основу для эффективной работы информационной системы.

При разработке информационной системы «Электронная зачётка» логическая модель строилась на основании результатов анализа предметной области, выполненного в первой главе работы. Проведённое исследование показало, что ключевыми объектами процесса учёта успеваемости являются студенты, учебные дисциплины и результаты промежуточной аттестации. Данные объекты были выделены в качестве основных сущностей будущей БД.

В логической модели выделены три основные сущности: «Студент», «Дисциплина» и «Оценка». Сущность «Студент» хранит сведения об обучающемся: ФИО, учебную группу и курс. Сущность «Дисциплина» содержит название предмета, курс, семестр и форму контроля. Сущность «Оценка» связывает студента с конкретной дисциплиной и фиксирует результат аттестации вместе с датой его получения.

Данное разделение вполне удобно для работы приложения: сведения о студенте или дисциплине не приходится повторять в каждой записи об оценке. При формировании зачётной книжки система обращается к связанным таблицам и получает уже готовый набор без лишних данных по выбранному студенту.

Благодаря этому обеспечивается хранение полной информации о результатах промежуточной аттестации и возможность дальнейшего анализа успеваемости.

Между сущностями модели существуют определённые связи. В реляционной модели данных такие связи позволяют описывать отношения между объектами предметной области и обеспечивать согласованность хранимой информации [13]. Связь между студентами и оценками относится к типу «один ко многим». Это показывает, что один обучающийся может иметь

несколько оценок по разным учебным дисциплинам, при этом каждая отдельная оценка относится только к одному конкретному студенту. Такая структура соответствует логике образовательного процесса, поскольку в течение обучения студент проходит множество предметов и по каждому из них получает определённые результаты аттестации.

По тому же принципу построена связь между дисциплинами и оценками. Одна дисциплина может использоваться во множестве записей об успеваемости разных студентов, но каждая конкретная оценка относится только к одной учебной дисциплине. Подобная организация данных делает модель более логичной и позволяет использовать сведения о дисциплинах без их повторного внесения в разные записи.

Если рассматривать взаимосвязь между студентами и дисциплинами напрямую, то она имеет тип «многие ко многим».

В реляционных базах данных такие связи обычно реализуются через промежуточные таблицы, позволяющие хранить дополнительные сведения об отношении между объектами [16]. Один студент изучает множество дисциплин, а каждая дисциплина, в свою очередь, изучается большим количеством студентов. Для реализации такой связи используется промежуточная сущность «Оценка», которая связывает обучающихся с изучаемыми предметами и одновременно хранит результаты их обучения.

Использование подобной модели обеспечивает гибкость работы с данными и позволяет выполнять широкий спектр операций. На её основе реализуются механизмы поиска информации, формирования электронной зачётной книжки, расчёта среднего балла, определения академических задолженностей, фильтрации данных и получения статистических сведений.

Разработанная логическая модель также соответствует принципам реляционного проектирования баз данных. Каждая сущность выполняет строго определённую функцию, а связи между объектами отражают реальные отношения, существующие в предметной области. Благодаря этому

обеспечивается структурированность информации, упрощается сопровождение базы данных и создаются условия для дальнейшего расширения системы.

Данная логическая модель соответствует функционалу созданного прототипа, но при переходе к полноценному внедрению её следует расширить, выделить учебную группу в отдельную сущность, а в таблице студентов хранить ссылку на нужную запись группы.

Для определения списка дисциплин будет нужна таблица связи между группами и дисциплинами. Такое нововведение поможет учитывать учебный план, не отталкиваясь на наличие оценок, выполнять разные запросы, например, для поиска студентов без оценок на текущий момент.

Таким образом, логическая модель данных отражает основные процессы учёта успеваемости студентов и описывает взаимосвязи между ключевыми объектами предметной области. Логическое проектирование базы данных позволило определить состав данных и связи между ними до этапа физической реализации [12].

2.3 Архитектура программного обеспечения

2.3.1 Общая структура проекта

Разработка информационной системы «Электронная зачётка» осуществлялась с использованием модульного подхода. Такой подход соответствует общим принципам проектирования информационных систем, при которых функциональность программного продукта разделяется на взаимосвязанные компоненты [4]. Его применение позволило разделить функциональность приложения на отдельные компоненты, каждый из которых отвечает за выполнение определённого набора задач. Такое решение упрощает сопровождение программного продукта, облегчает внесение изменений в код и создаёт условия для дальнейшего развития системы.

Информационная система реализована в виде настольного приложения для операционной системы Windows. Архитектура проекта построена на

взаимодействии пользовательского интерфейса, программной логики и базы данных. Совместная работа этих компонентов обеспечивает выполнение всех операций, связанных с хранением и обработкой сведений об успеваемости студентов.

Данная версия использует локальную однопользовательскую архитектуру, Приложение со всеми файлами находятся на одном рабочем компьютере, операции с информацией ведутся одним человеком. Создание независимых копий приложения на других компьютерах не предусмотрено, из-за неминуемой несогласованности экземпляров БД.

В будущем возможен переход к архитектуре клиент-сервер с общей базой информации. В таком случае несколько пользователей смогут работать с общим набором данных, а доступ к функционалу будет определяться ролью пользователя в системе.

Запуск приложения осуществляется через главный программный модуль. Он отвечает за инициализацию системы, загрузку необходимых настроек и открытие окна авторизации. После успешного входа пользователь получает доступ к основным функциям приложения и может приступить к работе с данными.

Для организации взаимодействия с базой данных используется отдельный программный модуль.

В приложении логика работы распределена между отдельными модулями, связанными с основными объектами системы. Для студентов, дисциплин и оценок предусмотрены самостоятельные компоненты, через которые выполняются операции добавления, изменения, удаления и получения данных.

Такой подход позволяет не смешивать разные группы функций и упрощает дальнейшую доработку отдельных частей программы.

Пользовательский интерфейс также разделён по функциональным областям. Для входа в систему, работы со студентами, дисциплинами, оценками, электронной зачётной книжкой и статистикой предусмотрены

отдельные окна. Благодаря этому пользователь работает с каждой задачей в отдельном разделе, а интерфейс не перегружается лишними элементами.

2.3.2 Описание программных модулей

Программная часть информационной системы «Электронная зачётка» включает несколько основных модулей, каждый из которых отвечает за определённый участок работы приложения. Такое разделение удобно для проекта на Python, поскольку позволяет поддерживать понятную структуру кода и при необходимости изменять отдельные компоненты без полной переработки всей системы [20].

При разработке программных модулей необходимо обеспечить чёткое распределение функций системы и согласованную работу всех её компонентов [5]. Такой подход делает структуру проекта более понятной, упрощает сопровождение программного продукта и позволяет в дальнейшем расширять его возможности без значительной переработки уже написанного кода.

Основным элементом проекта выступает главный программный модуль, через который выполняется запуск приложения. Он отвечает за инициализацию ключевых компонентов системы, загрузку интерфейса и открытие окна авторизации. После успешной проверки учётных данных пользователь автоматически переходит к главному окну программы.

Существенное значение для работы приложения имеет модуль взаимодействия с базой данных. Он выполняет создание необходимых таблиц при первом запуске системы, устанавливает соединение с базой SQLite и обеспечивает выполнение операций сохранения, изменения, удаления и получения сведений. Выделение данного функционала в отдельный модуль позволяет централизовать работу с данными и избежать повторения однотипного программного кода.

Для реализации основных возможностей системы используются модули, связанные с ключевыми сущностями предметной области. Отдельные

компоненты отвечают за управление данными о студентах, дисциплинах и результатах обучения. Они обеспечивают добавление, редактирование, удаление и поиск записей, а также их отображение в графическом интерфейсе приложения.

Авторизация реализована в отдельном окне входа. Пользователь вводит логин и пароль, после чего система проверяет данные и при успешном результате открывает главное окно приложения. Для удобства предусмотрены отображение и скрытие пароля, вход по клавише Enter, а также возможность сохранить данные для последующего запуска программы.

Раздел работы со студентами предназначен для ведения списка обучающихся. В нём пользователь может просматривать записи, добавлять новых студентов, редактировать ранее внесённые сведения и удалять ненужные данные. Дополнительно реализованы поиск по ФИО, фильтрация по курсу, а также импорт и экспорт списка студентов через Excel-файлы.

Дисциплины вынесены в отдельный раздел приложения. В нём находятся основные данные - названия учебных дисциплин, курс, семестр и форма контроля. Можно легко добавлять новые дисциплины, удалять неактуальные записи, а также выполнять импорт и экспорт данных в формате Excel.

В окне оценок пользователь выбирает студента и дисциплину из списков, указывает оценку и дату. В этом же разделе доступны редактирование, удаление записей, фильтрация по студенту, курсу и семестру, а также импорт и экспорт Excel-файлов.

В разделе электронной зачётной книжки пользователь выбирает студента из списка, после чего приложение выводит связанные с ним дисциплины и оценки. В этом же окне рассчитывается средний балл и отображается статус по каждой дисциплине. Если необходимо сохранить результат отдельно, данные можно выгрузить в PDF-документ.

Окно статистики предназначено для быстрого просмотра общей информации по базе данных. В нём отображается количество студентов, дисциплин и оценок, рассчитывается общий средний балл, а также

показываются сведения о задолженностях. Такой раздел нужен не для ввода данных, а для общей оценки текущего состояния системы.

Благодаря этому пользователь может быстро оценить текущее состояние базы данных и общую картину результатов обучения.

Сервисные функции приложения реализованы в дополнительных модулях. К ним относятся инструменты резервного копирования базы данных, журнал изменений программы, механизмы переключения между светлой и тёмной темами оформления, а также компоненты, необходимые для подготовки проекта к распространению в виде исполняемого EXE-файла.

Модульная организация позволяет в будущем дополнить систему управлением пользователями, а также проверкой прав доступа для разных пользователей без серьёзных изменений в технической составляющей приложения.

Таким образом, программная структура информационной системы построена на взаимодействии специализированных модулей, каждый из которых выполняет свой набор задач. Такое решение повышает устойчивость работы приложения, упрощает сопровождение исходного кода и создаёт основу для дальнейшего развития информационной системы «Электронная зачётка».

2.4 Реализация функциональных возможностей системы

2.4.1 Авторизация пользователя

Взаимодействие пользователя с информационной системой «Электронная зачётка» начинается с авторизации. Этот механизм используется для предоставления доступа к функциям приложения и позволяет организовать работу с системой сразу после её запуска в понятной для пользователя форме.

После запуска программы открывается окно авторизации, в котором размещены поля для ввода логина и пароля. Внешний вид окна авторизации пользователя показан на рисунке 1.

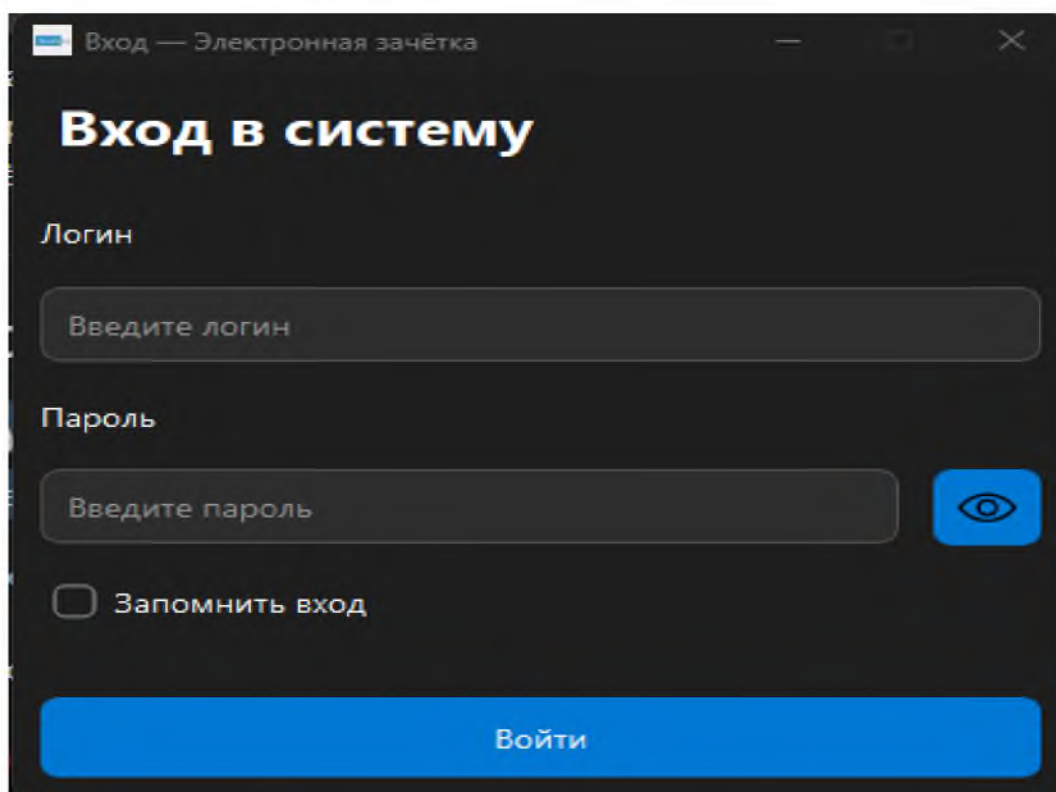


Рисунок 1- Окно авторизации пользователя

Средства Qt for Python позволяют разрабатывать интерактивные окна приложения и обрабатывать действия пользователя через элементы графического интерфейса [26]. Интерфейс окна авторизации был спроектирован с ориентацией на простоту, наглядность и удобство повседневного использования.

Основные элементы управления расположены в логической последовательности, что позволяет пользователю быстро выполнить вход в систему без лишних действий.

Процесс авторизации заключается во вводе учётных данных и их последующей проверке.

После успешного подтверждения введённой информации пользователь получает доступ к главному окну приложения, через которое осуществляется работа со студентами, дисциплинами, оценками и другими компонентами системы.

Для повышения удобства эксплуатации реализована возможность входа по нажатию клавиши Enter. Благодаря этому пользователь может выполнить

авторизацию без дополнительного использования кнопки входа, что ускоряет процесс начала работы с приложением.

Дополнительным элементом интерфейса является функция отображения и скрытия пароля. При необходимости пользователь может временно сделать вводимые символы видимыми, чтобы проверить правильность введенных данных. Такая возможность помогает избежать ошибок при вводе пароля и делает процесс авторизации более комфортным.

В системе также предусмотрен механизм запоминания учетных данных. При активации соответствующей функции логин и пароль сохраняются для последующего использования. Во время следующего запуска приложения сохранённые значения автоматически подставляются в поля ввода, что позволяет сократить время, необходимое для входа в систему.

При разработке окна авторизации особое внимание уделялось минимизации количества действий, выполняемых пользователем. Все основные операции доступны непосредственно из стартового окна, а процесс входа организован таким образом, чтобы не вызывать затруднений даже при первом использовании программы.

Реализованный механизм авторизации обеспечивает удобный доступ к информационной системе и создаёт единое пространство для дальнейшей работы с данными об успеваемости студентов. Использование данного компонента позволяет организовать процесс взаимодействия пользователя с приложением в простой и понятной форме.

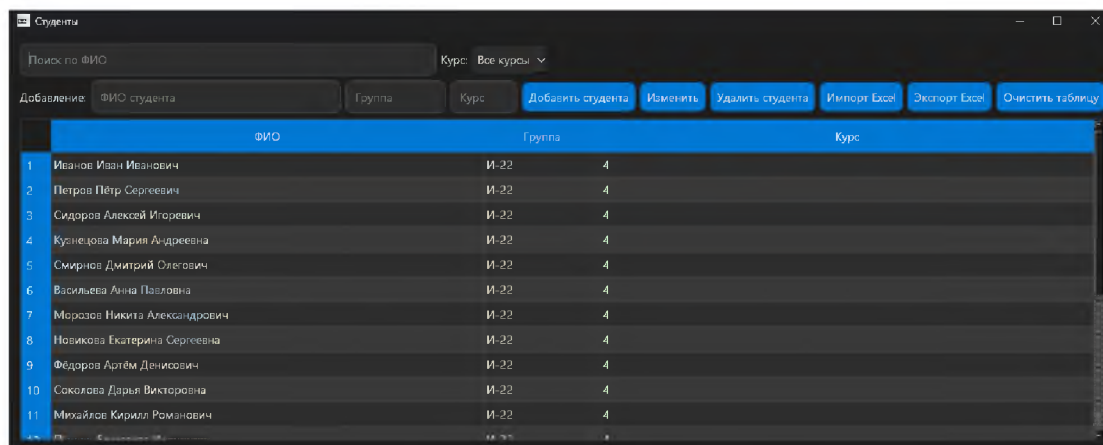
В текущей версии авторизация нужна для ограничения доступа к приложению, хотя разделение по ролям ещё не предусмотрено. В текущий момент пользователь получает все функции, предусмотренные системой.

При последующей разработке планируется добавить другие роли. Администратор будет также иметь доступ ко всем функциям приложения. Преподаватель получит доступ к дисциплине, которую ведёт сам. Студент получит доступ к просмотру своих оценок и не более.

2.4.2 Управление данными

Одним из основных направлений работы информационной системы «Электронная зачётка» является управление данными, связанными с учётом успеваемости студентов. Для решения этой задачи в приложении реализованы отдельные модули работы со студентами, дисциплинами и оценками. Все сведения хранятся в единой базе данных и используются при формировании электронной зачётной книжки, расчёте статистических показателей и выполнении других операций обработки информации. Использование единого хранилища данных обеспечивает согласованность информации и упрощает выполнение операций поиска, изменения и анализа записей [17].

Работа со студентами осуществляется через специализированный модуль, предназначенный для хранения и редактирования сведений об обучающихся. После открытия соответствующего раздела пользователю отображается таблица, содержащая информацию о зарегистрированных студентах. Для каждой записи выводятся фамилия, имя, отчество, учебная группа и курс обучения. Окно управления сведениями о студентах представлено на рисунке 2.



	ФИО	Группа	Курс
1	Иванов Иван Иванович	И-22	4
2	Петров Пётр Сергеевич	И-22	4
3	Сидоров Алексей Игоревич	И-22	4
4	Кузнецова Мария Андреевна	И-22	4
5	Смирнов Дмитрий Олегович	И-22	4
6	Васильева Анна Павловна	И-22	4
7	Морозов Никита Александрович	И-22	4
8	Новикова Екатерина Сергеевна	И-22	4
9	Фёдоров Артём Денисович	И-22	4
10	Соколова Дарья Викторовна	И-22	4
11	Михайлов Кирилл Романович	И-22	4

Рисунок 2 - Окно управления студентами

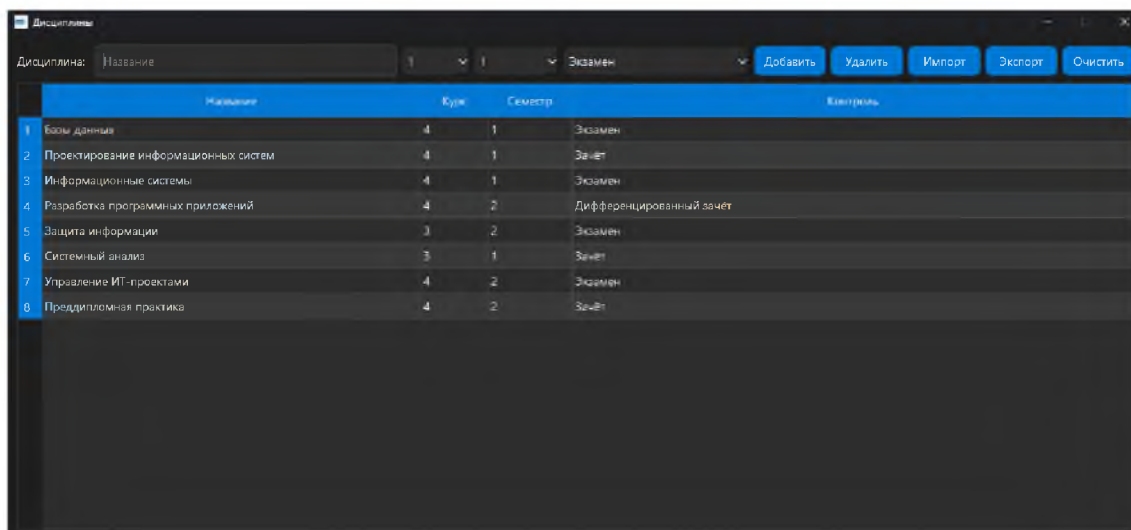
Для внесения новых сведений предусмотрена функция добавления студентов. Пользователь заполняет необходимые поля формы, после чего информация сохраняется в базе данных и становится доступной для использования в других разделах системы. При необходимости ранее

введённые данные могут быть изменены с помощью функции редактирования.

Для удаления неактуальных или ошибочно созданных записей предусмотрен отдельный механизм удаления. Дополнительно реализованы инструменты поиска по фамилии, имени и отчеству, а также фильтрация по курсу обучения. Эти возможности позволяют быстро находить нужные сведения даже при большом количестве записей.

С целью ускорения работы с информацией в модуле предусмотрены функции импорта и экспорта данных в формате Microsoft Excel. Автоматизация подобных рутинных операций средствами Python позволяет сократить объём ручного ввода и повысить скорость обработки данных [22]. Импорт позволяет автоматически загружать подготовленные списки студентов, а экспорт используется для сохранения информации во внешний файл и её дальнейшей обработки в сторонних приложениях.

Не менее важным компонентом системы является модуль управления дисциплинами. Он предназначен для ведения справочника учебных предметов, используемых при учёте успеваемости студентов. Окно управления учебными дисциплинами представлено на рисунке 3.



	Название	Кредит	Семестр	Контроль
1	Базы данных	4	1	Экзамен
2	Проектирование информационных систем	4	1	Зачет
3	Информационные системы	4	1	Экзамен
4	Разработка программных приложений	4	2	Дифференцированный зачет
5	Защита информации	3	2	Экзамен
6	Системный анализ	3	1	Зачет
7	Управление ИТ-проектами	4	2	Экзамен
8	Преддипломная практика	4	2	Зачет

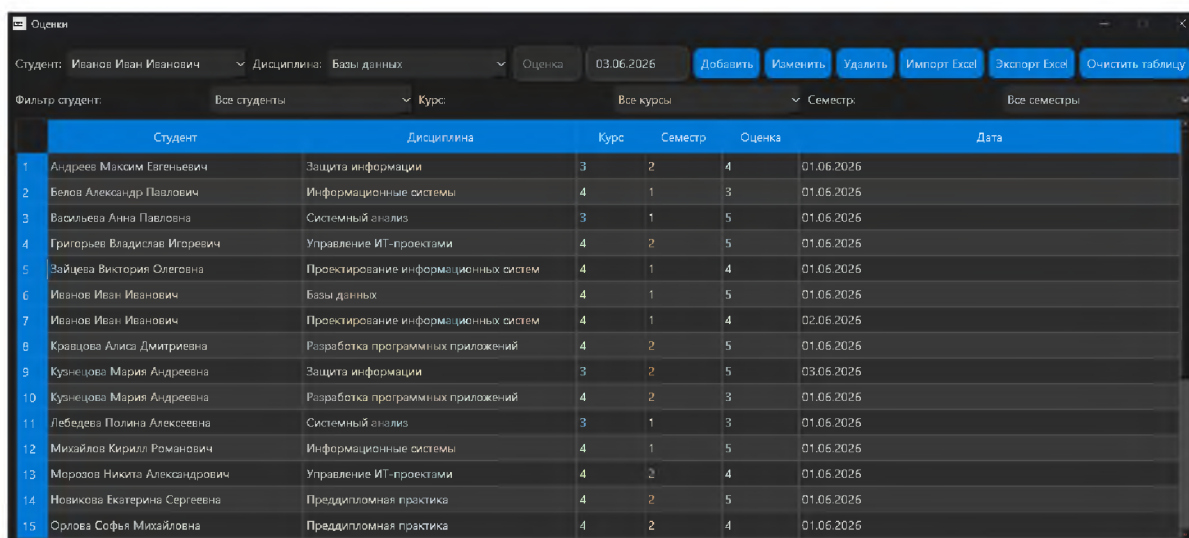
Рисунок 3- Окно управления дисциплинами

Централизованное хранение таких данных позволяет избежать дублирования информации и обеспечивает единообразие сведений во всех разделах приложения.

После открытия раздела пользователю отображается таблица со списком дисциплин. Для каждой записи сохраняются наименование дисциплины, курс обучения, семестр и форма контроля. Добавление новых дисциплин выполняется через специальную форму ввода, а изменение и удаление информации осуществляется с помощью соответствующих функций редактирования.

Для повышения удобства работы курс обучения, семестр и форма контроля выбираются из заранее подготовленных списков. Такое решение уменьшает риск ошибок при заполнении полей и помогает поддерживать единый формат данных в базе. По аналогии с модулем управления студентами, здесь также предусмотрены функции импорта и экспорта сведений в формате Excel.

Отдельный функциональный блок-системы связан с управлением оценками. Данный модуль предназначен для хранения результатов промежуточной аттестации студентов. Эти сведения используются при расчёте среднего балла, выявлении академических задолженностей и формировании электронной зачётной книжки. Окно управления оценками показано на рисунке 4.



	Студент	Дисциплина	Курс	Семестр	Оценка	Дата
1	Андреев Максим Евгеньевич	Защита информации	3	2	4	01.06.2026
2	Белов Александр Павлович	Информационные системы	4	1	3	01.05.2026
3	Васильева Анна Павловна	Системный анализ	3	1	5	01.05.2026
4	Григорьев Владислав Игоревич	Управление ИТ-проектами	4	2	5	01.05.2026
5	Зайцева Виктория Олеговна	Проектирование информационных систем	4	1	4	01.05.2026
6	Иванов Иван Иванович	Базы данных	4	1	5	01.05.2026
7	Иванов Иван Иванович	Проектирование информационных систем	4	1	4	02.05.2026
8	Кравцова Алиса Дмитриевна	Разработка программных приложений	4	2	5	01.05.2026
9	Кузнецова Мария Андреевна	Защита информации	3	2	5	03.05.2026
10	Кузнецова Мария Андреевна	Разработка программных приложений	4	2	3	01.05.2026
11	Лебедева Полина Алексеевна	Системный анализ	3	1	3	01.05.2026
12	Михайлов Кирилл Романович	Информационные системы	4	1	5	01.05.2026
13	Морозов Никита Александрович	Управление ИТ-проектами	4	2	4	01.05.2026
14	Новикова Екатерина Сергеевна	Преддипломная практика	4	2	5	01.05.2026
15	Орлова Софья Михайловна	Преддипломная практика	4	2	4	01.05.2026

Рисунок 4- Окно управления оценками

В этом разделе пользователь работает с таблицей, в которой

отображаются все внесённые в систему оценки. В таблице оценок каждая запись показывает, какому студенту выставлен результат, по какой дисциплине, за какой курс и семестр. Это позволяет быстро просматривать результаты обучения, а также их корректность.

Новая оценка добавляется после выбора студента и дисциплины из выпадающих списков. Затем пользователь указывает оценку и дату. При необходимости можно изменить или удалить запись.

Также можно находить нужную информацию через фильтр, что очень помогает при поиске в большом массиве данных.

Помимо этого, пользователь может импортировать и экспортировать данные об оценках с использованием файлов Excel [29].

Таким образом, реализованные инструменты управления данными позволяют выполнять полный цикл операций со сведениями о студентах, дисциплинах и результатах обучения. Единая база данных в сочетании со специализированными средствами обработки информации повышает удобство работы пользователей и обеспечивает основу для функционирования других компонентов информационной системы «Электронная зачётка».

2.4.3 Электронная зачётная книжка

Одним из основных элементов информационной системы «Электронная зачётка» выступает модуль электронной зачётной книжки. Он предназначен для отображения полной и актуальной информации об успеваемости выбранного студента в одном интерфейсе. Благодаря этому пользователь может быстро получить сведения о результатах обучения, не переходя между разными разделами приложения и не выполняя дополнительный поиск.

Работа с данным модулем начинается с выбора студента в выпадающем списке. После этого система автоматически обращается к базе данных и загружает сведения, связанные с результатами его промежуточной аттестации. Пользователю не нужно отдельно формировать запрос или вручную обновлять

информацию — необходимые данные выводятся системой автоматически.

Основная область окна представлена таблицей, в которой размещаются сведения об изученных дисциплинах и полученных результатах. По каждой записи отображаются название дисциплины, курс, семестр, форма контроля, выставленная оценка и дата её получения. Такой формат позволяет удобно просматривать информацию и быстро оценивать учебные результаты конкретного студента. Интерфейс электронной зачётной книжки представлен на рисунке 5.

	Дисциплина	Курс	Семестр	Форма контроля	Оценка	Дата	Статус
1	Проектирование информационных систем	4	1	Зачёт	4	01.06.2026	Сдано
2	Информационные системы	4	1	Экзамен	5	02.06.2026	Сдано

Рисунок 5- Электронная зачётная книжка студента

Одной из значимых возможностей модуля является автоматическое вычисление среднего балла. После загрузки сведений о выбранном студенте система самостоятельно выполняет расчёты и отображает итоговый показатель успеваемости. Если пользователь добавляет новую оценку, изменяет уже существующую запись или удаляет её, средний балл пересчитывается без дополнительных действий, что позволяет поддерживать актуальность отображаемых данных.

Для контроля результатов обучения в системе предусмотрен механизм выявления академических задолженностей. На основе внесённых оценок приложение анализирует успеваемость студента и определяет соответствующий статус. Это позволяет пользователю быстро обнаружить дисциплины, по которым имеются проблемы, и своевременно контролировать выполнение учебного плана.

Применение электронной зачётной книжки заметно упрощает работу с учебными сведениями. Вся необходимая информация размещается в одном разделе приложения и становится доступной сразу после выбора студента. Благодаря этому сокращается время поиска данных и повышается удобство их последующего анализа.

Дополнительно модуль поддерживает экспорт зачётной книжки в PDF-формат [30]. При запуске данной функции система автоматически создаёт документ, в который включаются сведения о студенте, список дисциплин, результаты аттестации, средний балл и информация о наличии академических задолженностей. Сформированный файл можно использовать для хранения, передачи другим пользователям или печати.

PDF-документ формируется на основании сведений, уже содержащихся в базе данных. Такой подход исключает необходимость ручного заполнения отчётной формы и уменьшает вероятность ошибок при подготовке документации. Все данные переносятся автоматически, поэтому итоговый файл содержит полную и актуальную информацию.

Таким образом, модуль электронной зачётной книжки обеспечивает централизованное отображение сведений об успеваемости студентов, автоматизирует расчёт основных учебных показателей и предоставляет удобный инструмент для подготовки отчётных документов.

2.4.4 Статистика и дополнительные возможности системы

Помимо основных операций с данными, в приложении реализованы статистика и резервное копирование базы данных. Использование статистических данных позволяет быстро оценить ситуацию без необходимости просмотра большого количества отдельных записей.

В окне статистики отображаются сведения о количестве зарегистрированных студентов, числе дисциплин и общем количестве оценок, хранящихся в базе данных. Одновременно выполняется расчёт среднего балла

по всем результатам обучения, зарегистрированным в системе. Полученные показатели обновляются автоматически и отражают текущее состояние базы данных. Окно статистики информационной системы представлено на рисунке 6.

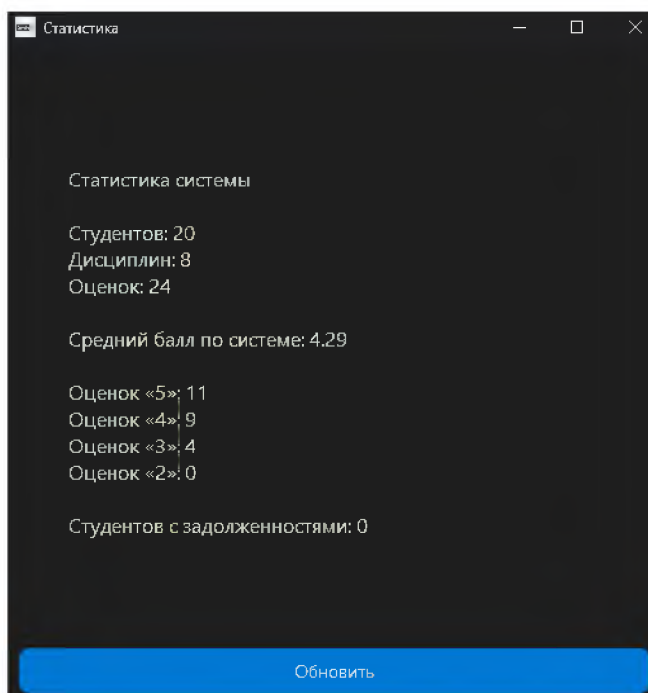


Рисунок 6 - Окно статистики

Для более детального анализа предусмотрен подсчёт количества оценок по каждому уровню успеваемости. Пользователь может получить информацию о числе оценок «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно». Также отображается количество студентов, имеющих академические задолженности. Подобные сведения позволяют быстро оценить общую успеваемость и выявить проблемные участки образовательного процесса.

С целью обеспечения актуальности информации реализована возможность обновления статистических данных в процессе работы приложения. После внесения изменений в базу данных пользователь может получить обновлённые показатели без перезапуска программы.

Одной из значимых дополнительных возможностей является создание резервной копии базы данных. В ходе эксплуатации системы постепенно накапливается большой объём сведений, и их потеря может серьёзно

затруднить дальнейшую работу с учебной информацией. В разработанном приложении пользователь может воспользоваться специальной функцией интерфейса и сохранить копию файла базы данных SQLite. Это повышает защищённость данных и снижает вероятность их утраты при программных сбоях, ошибочных действиях пользователя или неисправности оборудования.

Для более комфортной работы в приложении предусмотрено несколько вариантов оформления интерфейса.

Таким образом, реализованные дополнительные возможности расширяют функциональность информационной системы «Электронная зачётка», обеспечивают удобство работы с данными и способствуют повышению надёжности хранения информации. Их использование делает программный продукт более практичным и удобным для повседневного применения в образовательной организации.

2.5 Руководство пользователя

2.5.1 Описание интерфейса

Пользовательский интерфейс информационной системы «Электронная зачётка» разработан с использованием библиотеки PySide6 и представляет собой совокупность взаимосвязанных окон, обеспечивающих доступ ко всем функциям приложения. Модуль QtWidgets предоставляет основные элементы графического интерфейса, включая окна, кнопки, таблицы, поля ввода и выпадающие списки [25]. При проектировании интерфейса основное внимание уделялось удобству работы пользователя, простоте навигации и логичному расположению элементов управления.

После запуска приложения пользователю отображается окно авторизации. В нём расположены поля для ввода логина и пароля, кнопка входа в систему, элемент управления отображением пароля и функция сохранения учётных данных. Интерфейс окна организован таким образом, чтобы процесс

входа занимал минимальное количество времени и не вызывал затруднений даже при первом использовании программы.

После успешной авторизации открывается главное окно приложения, которое является центральным элементом интерфейса. Через него осуществляется доступ ко всем основным разделам информационной системы. В главном окне размещены кнопки перехода к управлению студентами, дисциплинами и оценками, работе с электронной зачётной книжкой, просмотру статистики и другим функциям системы. Также здесь отображается краткая информация о текущем состоянии базы данных и доступны настройки оформления интерфейса. Главное окно информационной системы представлено на рисунке 7.

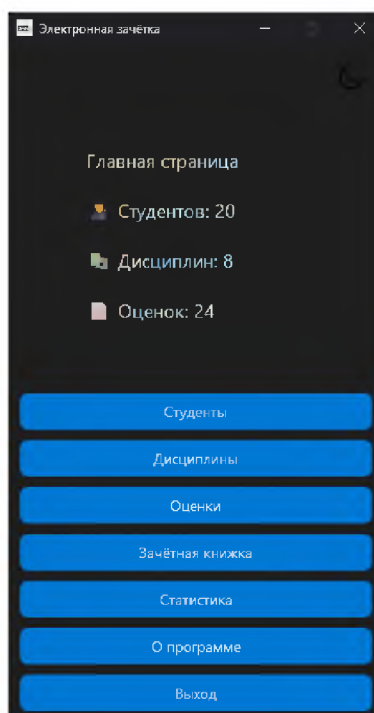


Рисунок 7- Главное окно информационной системы

Для работы со сведениями об обучающихся предусмотрено отдельное окно управления студентами. Основную часть интерфейса занимает таблица со списком студентов. Дополнительно реализованы поля ввода данных, средства поиска по фамилии, имени и отчеству, фильтр по курсу обучения, а также кнопки добавления, редактирования, удаления, импорта и экспорта информации.

Окно управления дисциплинами предназначено для ведения справочника учебных предметов. В его состав входят таблица с перечнем дисциплин, поля ввода информации и выпадающие списки для выбора курса, семестра и формы контроля. Пользователь может выполнять добавление, изменение и удаление записей, а также импортировать и экспортировать данные в формате Excel.

Для работы с результатами промежуточной аттестации используется окно управления оценками. В данном разделе предусмотрены элементы выбора студента и дисциплины, поля ввода оценки и даты её выставления, а также инструменты фильтрации информации. Интерфейс позволяет быстро выполнять все основные операции по работе с оценками и обеспечивает удобный просмотр накопленных данных.

Окно электронной зачётной книжки предназначено для отображения информации об успеваемости выбранного студента. В центральной части окна располагается таблица с перечнем дисциплин и результатами аттестации. Дополнительно отображаются сведения о среднем балле и наличии академических задолженностей. Из данного раздела пользователь может выполнить экспорт зачётной книжки в формат PDF.

Для работы с аналитическими сведениями в системе предусмотрено отдельное окно статистики. В нём выводятся основные показатели приложения: количество студентов, дисциплин и оценок, общий средний балл по базе данных, а также информация об академических задолженностях. Такой раздел позволяет пользователю быстро оценить текущее состояние системы и получить обобщённое представление о накопленных учебных данных.

Дополнительной возможностью интерфейса является поддержка светлой и тёмной тем оформления. Пользователь может выбрать вариант отображения, который будет наиболее удобен с учётом условий работы и личных предпочтений. Смена темы выполняется непосредственно во время использования приложения и не требует дополнительных настроек.

Таким образом, интерфейс информационной системы «Электронная зачётка» предоставляет удобный доступ ко всем основным функциям

приложения, имеет логичную структуру и позволяет эффективно выполнять операции по учёту успеваемости студентов без специальной подготовки пользователя.

2.5.2 Порядок работы с системой

Работа с информационной системой «Электронная зачётка» начинается с открытия приложения. После запуска программы пользователь попадает в окно авторизации, где требуется указать логин и пароль. Если введённые данные проходят проверку, система открывает главное окно, через которое становится доступен весь функционал приложения.

Перед началом полноценной работы рекомендуется заполнить основные справочники. Сначала пользователь переходит в раздел управления студентами и вносит сведения об обучающихся. Данные можно добавить вручную через поля ввода либо загрузить из файла Excel. После сохранения записи помещаются в базу данных и могут использоваться в других разделах системы.

Следующим шагом заполняется справочник дисциплин. В данном разделе пользователь добавляет учебные предметы, указывая их название, курс обучения, семестр и форму контроля. Как и при работе со списком студентов, сведения можно ввести вручную или импортировать из заранее подготовленной электронной таблицы.

После заполнения справочников пользователь может перейти к внесению результатов промежуточной аттестации. Для этого предназначен раздел управления оценками. В нём выбираются студент и дисциплина из доступных списков, указывается полученная оценка и дата её выставления, после чего запись сохраняется. Введённые сведения автоматически включаются в общую базу данных и становятся доступными для остальных компонентов приложения.

Для просмотра учебных результатов используется модуль электронной зачётной книжки. После выбора конкретного студента система выводит список

изученных дисциплин, полученные оценки, средний балл и информацию о наличии академических задолженностей. Все показатели рассчитываются автоматически на основе данных, которые уже содержатся в базе.

При необходимости пользователь может сформировать электронную зачётную книжку в PDF-формате. Для этого используется функция экспорта: после её запуска система самостоятельно создаёт документ со сведениями об успеваемости выбранного студента. Полученный файл можно сохранить, передать другим пользователям или распечатать.

Для просмотра обобщённых сведений о состоянии базы данных предусмотрен раздел статистики. В нём отображаются основные показатели работы системы: количество студентов, дисциплин и оценок, средний балл, а также сведения об академических задолженностях.

Это позволяет оперативно оценивать результаты работы и контролировать актуальность информации.

В процессе эксплуатации приложения пользователь может использовать дополнительные возможности системы. К ним относятся импорт и экспорт данных в формате Excel, создание резервных копий базы данных, просмотр журнала изменений и переключение между светлой и тёмной темами оформления интерфейса.

Работа с информационной системой построена по принципу последовательного выполнения основных операций учёта успеваемости. Описание порядка работы пользователя является важной частью проектной документации, поскольку позволяет связать реализованные функции системы с практическими сценариями её эксплуатации [6]. Интерфейс приложения интуитивно понятен, а выполнение основных действий не требует специальных технических знаний, что обеспечивает удобство использования системы в повседневной деятельности образовательной организации.

3 Обоснование экономической эффективности результатов разработки

3.1 Выбор методики оценки эффективности

Одним из завершающих этапов разработки информационной системы является оценка её эффективности. Проведение такой оценки позволяет определить практическую значимость созданной информационной системы, а также определить, как её использование помогает в работе над процессами обработки информации.

ИС «Электронная зачётка» создана для упрощения учёта успеваемости студентов, хранения учебных данных и формирования необходимых документов. Внедрение системы направлено на сокращение времени выполнения повседневных операций, уменьшение объёма ручной работы и повышение удобства взаимодействия пользователей с информацией.

Для оценки эффективности был выбран простой показатель — время выполнения типовой операции до и после автоматизации. Такой вариант подходит для данной системы, потому что основной эффект от её использования связан с сокращением ручной работы при поиске, расчёте и подготовке сведений об успеваемости.

Суть выбранной методики заключается в сопоставлении времени, затрачиваемого на выполнение одинаковых задач при традиционном способе работы и при использовании разработанной информационной системы. В качестве объектов сравнения рассматриваются операции, непосредственно связанные с учётом успеваемости студентов: поиск необходимых сведений, внесение результатов аттестации, расчёт показателей успеваемости, выявление академических задолженностей и формирование итоговых документов.

Во время оценки результатов разработки нужно учитывать как сокращение времени выполнения отдельных операций, так и затраты, связанные с созданием и началом использования продукта. К этому также относятся трудозатраты на разработку и тестирование, а также исправление проблем, установку и заполнение данных в приложение.

В текущей версии приложение рассматривается как локальное, созданное для автоматизации отдельных частей учёта успеваемости. На начальном этапе использования не предполагает полного отказа от бумажных вариантов ведения учёта успеваемости. Из-за этого, во время внедрения, электронный и бумажный способ учёта могут использоваться параллельно, что создаст дополнительные временные затраты на сверку данных.

Учитывая вышесказанное, оценка проводится по двум направлениям. Первое будет связано с трудозатратами до полного введения в среду учебного отдела, а второе будет связано уже с уменьшением трудозатрат при всех операциях при ведении данных учётов студентов.

При использовании традиционных методов значительная часть перечисленных действий выполняется вручную. Пользователю необходимо самостоятельно искать информацию в различных источниках, выполнять расчёты и подготавливать документы. Это приводит к увеличению трудозатрат и повышает вероятность возникновения ошибок при обработке данных.

Использование информационной системы позволяет автоматизировать большую часть указанных операций. Сведения хранятся в единой базе данных, расчёт показателей выполняется автоматически, а формирование необходимых документов осуществляется средствами программного обеспечения.

Оценивать качество будем показателем экономии времени. Такой подход позволяет объективно оценить результаты внедрения информационной системы «Электронная зачётка» и подтвердить целесообразность её использования в образовательной организации.

3.2 Расчёт показателей экономической эффективности

Для оценки эффективности разработанной информационной системы «Электронная зачётка» было проведено сравнение временных затрат на выполнение основных операций учёта успеваемости студентов до и после автоматизации рассматриваемого процесса.

До внедрения информационной системы работа с учебными данными выполнялась с использованием бумажных документов, электронных таблиц и отдельных файлов. Поиск информации о студенте, анализ результатов обучения, расчёт среднего балла и подготовка итоговых сведений требовали выполнения ряда последовательных действий вручную. При увеличении объёма данных такие операции занимали значительное время и повышали вероятность возникновения ошибок.

После внедрения информационной системы большая часть указанных операций выполняется автоматически. Пользователь получает доступ к централизованной базе данных, а обработка информации осуществляется средствами программного обеспечения. Это позволяет существенно сократить время выполнения повседневных задач и повысить удобство работы с данными.

Для количественной оценки эффективности рассмотрим операцию формирования сведений об успеваемости одного студента. Данная операция включает поиск необходимой информации, просмотр результатов обучения по дисциплинам, расчёт среднего балла и подготовку итоговых данных для дальнейшего использования.

При традиционном способе работы выполнение перечисленных действий занимает в среднем около 10 минут. Значительная часть времени расходуется на поиск информации в различных источниках и выполнение расчётов вручную.

При использовании информационной системы необходимые сведения формируются автоматически после выбора студента в модуле электронной зачётной книжки. Среднее время получения аналогичного результата составляет около 1 минуты.

Для наглядного представления изменения временных затрат рассмотрим сравнение выполнения одной из типовых операций до и после использования информационной системы. В качестве примера выбрана операция

формирования сведений об успеваемости одного студента.

Таблица 2 - Сравнение временных затрат до и после автоматизации

Операция	До автоматизации	После автоматизации	Изменение
Формирование сведений об успеваемости одного студента	10 минут	1 минута	Сокращение на 9 минут

Как видно из таблицы 2, использование информационной системы позволяет значительно сократить время выполнения рассматриваемой операции. Для количественной оценки полученного результата рассчитаем экономию времени.

Экономия времени рассчитывается по формуле:

$$\text{Э} = ((T1 - T2) / T1) \times 100 \%, \quad (1)$$

где Э — эффективность автоматизации, %;

T1 — время выполнения операции до автоматизации;

T2 — время выполнения операции после автоматизации.

Подставляя имеющиеся значения, получаем:

$$\text{Э} = ((10 - 1) / 10) \times 100 = 90 \, \%.$$

Данные вычисления показывают, что после заполнения БД и освоения приложения пользователем время формирования сведений об успеваемости одного студента может сократиться с 10 до 1 минуты. Экономия времени составляет 90% если сравнивать с бумажным способом ведения данных.

Данная эффективность учитывает именно освоенное использование ПО, в расчёт не входят затраты на установку, настройку и освоение приложения.

В начале внедрения общая эффективность может оказаться ниже, ибо сотрудникам понадобится время на освоение продукта и заполнение

необходимых данных. По мере заполнения таблиц и набора опыта в использовании приложения повысится и эффективность в работе с информацией, тем самым уменьшится затрачиваемое на это время.

Полученный результат обусловлен тем, что система автоматически выполняет поиск информации, расчёт среднего балла, определение академических задолженностей и формирование итоговых сведений об успеваемости. Пользователю больше не требуется выполнять эти действия вручную, что значительно снижает трудоёмкость работы.

Дополнительный эффект достигается за счёт использования единой базы данных. Централизованное хранение данных уменьшает время, необходимое для поиска нужной информации, упрощает её обновление и обеспечивает оперативный доступ к сведениям даже при значительном объёме накопленных записей.

3.3 Анализ результатов внедрения информационной системы

Электронная зачётная книжка стала одним из основных практических результатов разработки. После выбора студента можно узнать его дисциплины, оценки, средний балл и определить наличие задолженностей. Не требуется вручную собирать эти сведения из разных таблиц или документов, поэтому подготовка данных об успеваемости выполняется быстрее.

Проведённая оценка показывает, что разработанная система помогает уменьшить время выполнения операций, связанных с учётом студентов, дисциплин, оценок и зачётов. Больше всего заметен результат после формирования электронной зачётной книжки, ведь все необходимые данные отображаются автоматически, при выборе студента.

Текущую версию приложения стоит учитывать как локальный прототип и вспомогательный инструмент. При введении система не исключает продолжения ввести параллельно также и бумажный вариант документации, параллельно с электронным.

Можно получить более полный эффект, но только после дальнейшей доработки приложения. В перспективе улучшение приложения в сторону разделения ролей на администратора, преподавателя и студента, а также переход к общей БД и организации одновременной работы нескольких сотрудников. Эти правки позволят уменьшить объём повторяющихся операций, а также расширит область практического применения системы.

Таким образом, такие проектные решения можно считать обоснованными для текущей версии прототипа. Экономия времени, равная 90% относится к отдельной автоматизированной операции, в то время как общая эффективность уже зависит от условий эксплуатации и дальнейшего развития ИС.

Заключение

Во время выпускной квалификационной работы была разработана ИС «Электронная зачётка», которая помогает в автоматизации учёта данных в образовательной организации.

Был проведён анализ предметной области, связанной с хранением и обработкой учебных данных. Анализ позволил выделить недостатки при использовании бумажных носителей информации, а также хранения данных в разрозненных файлах. При увеличении данных рос и объём времени, необходимый на обработку данных, а также увеличивался шанс на ошибки при внесении изменений в существующие данные.

Для выбора оптимального способа устранения выявленных недостатков проведён анализ программных решений, используемых в образовательной сфере. Результаты показали - многие современные системы обладают развитым функционалом, но чаще всего ориентированы на повышенную автоматизацию работы образовательной организации. Из-за этого они включают большое количество возможностей, которые не всегда требуются при решении задач, связанных именно с учётом успеваемости студентов. Это подтвердило необходимость разработки специализированного программного продукта, направленного на выполнение конкретной прикладной задачи.

На основе результатов анализа была спроектирована структура информационной системы, разработана база данных и определены средства программной реализации проекта. В качестве основных технологий были выбраны язык программирования Python, библиотека PySide6 для создания графического интерфейса пользователя и система управления базами данных SQLite. Для работы с файлами Excel использовалась библиотека OpenPyXL, а формирование документов в формате PDF было реализовано с помощью библиотеки ReportLab.

Результатом выполненной работы стала информационная система «Электронная зачётка», представляющая собой настольное приложение для

операционной системы Windows. В разработанном программном продукте реализованы механизмы авторизации пользователей, управление сведениями о студентах, дисциплинах и оценках, ведение электронной зачётной книжки, расчёт среднего балла успеваемости и определение академических задолженностей. Дополнительно предусмотрены функции импорта и экспорта данных в формате Excel, формирования PDF-документов, отображения статистической информации, резервного копирования базы данных и настройки оформления пользовательского интерфейса.

Проведённая оценка эффективности показала, что использование разработанной системы позволяет существенно сократить временные затраты на выполнение основных операций по обработке учебной информации. Автоматизация процессов хранения, поиска и анализа данных способствует снижению трудоёмкости работы пользователей и повышает удобство взаимодействия с информацией.

Поставленная цель выпускной квалификационной работы была достигнута, а все сформулированные задачи успешно решены. Созданная информационная система является работоспособным программным продуктом и может использоваться для автоматизации учёта успеваемости студентов в образовательных организациях.

Дальнейшее развитие проекта может быть связано с расширением функциональных возможностей системы. Перспективными направлениями являются реализация многопользовательского режима работы, разграничение прав доступа пользователей, интеграция с другими образовательными информационными системами, а также развитие инструментов аналитической обработки данных и формирования отчётности.

Список литературы

1. Чистов, Д. В. Проектирование информационных систем: учеб. и практикум для вузов / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук. — Москва: Юрайт, 2024. — 293 с.
2. Чистов, Д. В. Проектирование информационных систем: учеб. для вузов / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук. — Москва: Юрайт, 2026. — 273 с.
3. Григорьев, М. В. Проектирование информационных систем: учеб. для вузов / М. В. Григорьев, И. И. Григорьева. — Москва : Юрайт, 2025. — 278с.
4. Григорьев, М. В. Проектирование информационных систем: учеб. пособие для вузов / М. В. Григорьев, И. И. Григорьева. — Москва: Юрайт, 2024. — 278 с.
5. Грекул, И. В. Проектирование информационных систем: учеб. и практикум / И. В. Грекул, Н. Л. Коровкина, Г. А. Левочкина. — Москва: Юрайт, 2024. — 423 с.
6. Грекул, И. В. Проектирование информационных систем. Практикум: учеб. пособие / И. В. Грекул, Н. Л. Коровкина, Ю. В. Куприянов. — Москва: Национальный открытый университет «ИНТУИТ», 2012. — Текст: электронный. — Режим доступа: <https://publications.hse.ru/books/53441017> (дата обращения: 05.04.2026).
7. Советов, Б. Я. Информационные технологии: учеб. для вузов / Б. Я. Советов, В. В. Цехановский. — Москва: Юрайт, 2023. — 327 с.
8. Гвоздева, В. А. Проектирование информационных систем: учеб. пособие / В. А. Гвоздева. — Москва: Форум, 2022. — 320 с.
9. Зараменских, Е. П. Основы проектирования информационных систем : учебное пособие / Е. П. Зараменских. — Москва: Юрайт, 2023. — 187с.
10. Зараменских, Е. П. Информационные системы в бизнесе: учеб. и практикум / Е. П. Зараменских. — 2-е изд., перераб. и доп. — Москва: Юрайт,

2026. — 470 с. — Текст: электронный. — Режим доступа: <https://urait.ru/bcode/587744> (дата обращения: 12.04.2026).

11. Кузнецов, С. Д. Базы данных: учеб. для вузов / С. Д. Кузнецов. — Москва : Академия, 2022. — 496 с.

12. Илюшечкин, В. М. Основы использования и проектирования баз данных: учеб. для вузов / В. М. Илюшечкин. — Москва: Юрайт, 2025. — 213 с. — Текст: электронный. — Режим доступа: <https://urait.ru/bcode/559613> (дата обращения: 19.04.2026).

13. Нестеров, С. А. Базы данных: учеб. и практикум / С. А. Нестеров. — 2-е изд. — Москва : Юрайт, 2026. — 258 с. — Текст: электронный. — Режим доступа: <https://urait.ru/bcode/587742> (дата обращения: 26.04.2026).

14. Советов, Б. Я. Базы данных: учеб. для вузов / Б. Я. Советов. — Москва : Юрайт, 2023.

15. Гринченко, Н. Н. Базы данных. Проектирование моделей данных: учеб. пособие / Н. Н. Гринченко. — Москва, 2024.

16. Введение в реляционные базы данных : учебное пособие. — Москва: ИНФРА-М, 2023.

17. Кузнецов, С. Д. Базы данных. Вводный курс / С. Д. Кузнецов. — Текст: электронный. — Режим доступа: https://citforum.ru/database/advanced_intro/ (дата обращения: 03.05.2026).

18. Маркин, А. В. Программирование на SQL: учеб. и практикум для вузов / А. В. Маркин. — 3-е изд., перераб. и доп. — Москва: Юрайт, 2024. — 805 с. — Текст : электронный. — Режим доступа: <https://urait.ru/bcode/534873> (дата обращения: 10.05.2026).

19. Федоров, Д. Ю. Программирование на Python : учеб. пособие / Д. Ю. Федоров. — Москва: Юрайт, 2024. — 187 с.

20. Чернышев, С. А. Основы программирования на Python: учеб. пособие / С. А. Чернышев. — Москва : Юрайт, 2024.

21. Лутц, М. Изучаем Python / М. Лутц. — Санкт-Петербург: Питер, 2024.

22. Свейгарт, Э. Автоматизация рутинных задач с помощью Python / Э. Свейгарт. — Санкт-Петербург : Питер, 2021.
23. Python Software Foundation. Python Documentation [Электронный ресурс]. — Режим доступа: <https://docs.python.org/3/> (дата обращения: 17.05.2026).
24. Qt Company. Qt for Python Documentation [Электронный ресурс]. — Режим доступа: <https://doc.qt.io/qtforpython-6/> (дата обращения: 24.05.2026).
25. Qt Company. PySide6.QtWidgets Documentation [Электронный ресурс]. — Режим доступа: <https://doc.qt.io/qtforpython-6/PySide6/QtWidgets/> (дата обращения: 24.05.2026).
26. Qt Company. Qt for Python Official Documentation [Электронный ресурс]. — Режим доступа: <https://www.qt.io/development/qt-framework/python-bindings> (дата обращения: 24.05.2026).
27. SQLite Documentation [Электронный ресурс]. — Режим доступа: <https://sqlite.org/docs.html> (дата обращения: 31.05.2026).
28. SQLite Language Reference [Электронный ресурс]. — Режим доступа: <https://sqlite.org/lang.html> (дата обращения: 31.05.2026).
29. OpenPyXL Documentation [Электронный ресурс]. — Режим доступа: <https://openpyxl.readthedocs.io/> (дата обращения: 31.05.2026).
30. ReportLab User Guide [Электронный ресурс]. — Режим доступа: <https://docs.reportlab.com/> (дата обращения: 31.05.2026).