

**Разработка клиентского приложения учета средств безопасности
управления информатизации
для ФГБОУ ВО «ПГУ имени Шолом-Алейхема»**

Анишкова Анастасия Сергеевна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Целью данной работы является разработка информационной системы учета средств безопасности для Управления информатизации, которая позволит автоматизировать процессы документооборота, обеспечить прозрачность и полноту данных, а также повысить эффективность работы отдела информационной безопасности. Для реализации выбран язык программирования C# в сочетании со средой разработки Microsoft Visual Studio. Этот выбор обусловлен высокой эффективностью, надежностью и широкими функциональными возможностями данной технологии при создании оконных (десктопных) приложений. C# – это современный объектно-ориентированный язык программирования, разработанный компанией Microsoft и предназначенный для работы в рамках платформы .NET Framework и более новой кроссплатформенной версии .NET Core / .NET 5+. Он предоставляет богатый набор инструментов для создания различных типов приложений, включая Windows Forms, WPF (Windows Presentation Foundation), веб-приложения и сервисы. Результат может быть применён в обучающих целях.

Ключевые слова: клиентское приложение, учет средств безопасности, Windows Forms, C#, Visual Studio.

**Development of a client application for accounting of information security
management tools for the Sholom-Aleichem Priamurskiy State University**

Anishkova Anastasia Sergeevna

Sholom-Aleichem Priamurskiy State University

Student

Abstract

The purpose of this work is to develop an information security accounting system for Information Management, which will automate document management processes, ensure transparency and completeness of data, and improve the efficiency of the information security department. The C# programming language in combination with the Microsoft Visual Studio development environment is selected for implementation. This choice is due to the high efficiency, reliability and wide functionality of this technology when creating windowed (desktop) applications. C#

is a modern object-oriented programming language developed by Microsoft and designed to work within the framework of the platform.NET Framework and a newer cross-platform version .NET Core / .NET 5+. It provides a rich set of tools for creating various types of applications, including Windows Forms, WPF (Windows Presentation Foundation), web applications and services. The result can be when

Keywords: client application, security accounting, Windows Forms, C#, Visual Studio.

1 Введение

1.1 Актуальность

В условиях постоянного развития информационных технологий и увеличения числа киберугроз обеспечение информационной безопасности становится одной из приоритетных задач образовательных учреждений. Федеральное государственное бюджетное образовательное учреждение высшего образования «ПГУ имени Шолом-Алейхема» не является исключением — вузы обладают большим объемом конфиденциальной информации, включая персональные данные сотрудников и студентов, научные разработки, а также административно-управленческую информацию. Эффективное управление средствами обеспечения информационной безопасности требует системного подхода, включающего учет, контроль и мониторинг используемых программных и аппаратных решений. На данный момент отсутствует единая клиентская система, позволяющая централизованно учитывать и управлять такими средствами, что может приводить к дублированию функционала, неэффективному использованию ресурсов и повышению рисков нарушения безопасности. Разработка клиентского приложения учета средств обеспечения информационной безопасности позволит автоматизировать процессы управления, повысить прозрачность и оперативность принятия решений, снизить административные затраты и обеспечить соответствие требованиям законодательства в сфере защиты информации.

1.2 Обзор исследований

Р. И. Баженов, А. А. Глухова продемонстрировали разработку информационной системы учета заказов в компьютерной мастерской [1]. Разработка автоматизированной системы учета стажеров описали А.В.Борисенко, А. Н. Петрова [2]. А. В. Борисенко, А. Н. Петрова разработали систему учета развлекательного центра «Киров парк» [3]. Роль языка программирования с# в создании программного обеспечения описал Б.Б.Исамухамедов [4]. Е. А. Пасюкова провела Обзор технических средств разработки графического интерфейса на языке программирования С [5].

1.3 Цель исследования

Целью данной работы является разработка информационной системы учета средств безопасности для Управления информатизации, которая

позволит автоматизировать процессы документооборота, обеспечить прозрачность и полноту данных, а также повысить эффективность работы отдела информационной безопасности.

2 Материалы и методы

Для реализации проекта выбран язык программирования C# в комбинации со средой разработки Microsoft Visual Studio. Такой выбор обусловлен высокой производительностью, надежностью и широким функциональным потенциалом данной технологии при разработке десктопных (оконных) приложений.

3 Результаты

Информационная система представляет собой программный комплекс, предназначенный для автоматизации процессов учета и управления средствами криптографической защиты информации (СКЗИ), информационными системами персональных данных (ИСПДн) и согласиями на обработку персональных данных (ПДн) в рамках отдела информационной безопасности.

Система реализована в виде локального оконного приложения, разработанного с использованием языка C# и предназначенного для работы на компьютере начальника отдела информационной безопасности. В качестве основного механизма хранения данных используется локальная база данных, построенная на СУБД SQLite, что обеспечивает простоту развертывания, высокую производительность и отсутствие необходимости в дополнительной инфраструктуре.

Функционал системы реализован двумя основными модулями:

1. Модуль базы данных отвечает за хранение, обработку и обеспечение целостности всей информации, связанной с учетом СКЗИ, ИСПДн и согласий на обработку ПДн.

2. Клиентское приложение предоставляет удобный графический интерфейс, позволяющий пользователям эффективно взаимодействовать с данными. Интерфейс реализован в виде форм Windows Forms и ориентирован на упрощение и оптимизацию рабочих процессов сотрудников отдела ИБ.

При запуске приложения пользователю предлагается пройти процедуру аутентификации (см.рис.2.3). Возможность входа предоставляется только заранее зарегистрированным пользователям. Регистрация новых пользователей возможна только администратором системы – начальником отдела информационной безопасности, что обеспечивает контроль доступа и повышает уровень информационной безопасности.

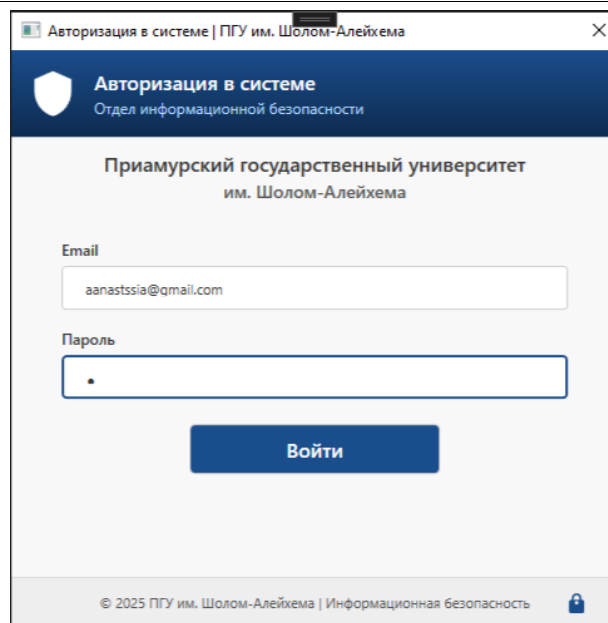


Рисунок 3. Авторизация пользователя

При успешном входе в систему пользователь попадает в главное меню (см.рис.4), которое представляет собой центральную точку навигации и включает в себя кнопки доступа к основным разделам приложения: «СКЗИ», «ИСПДн», «Справочники», «ПДн», «Заметки» и «Журнал изменений». При нажатии на соответствующую кнопку открывается форма, предоставляющая функционал для работы с выбранным разделом.

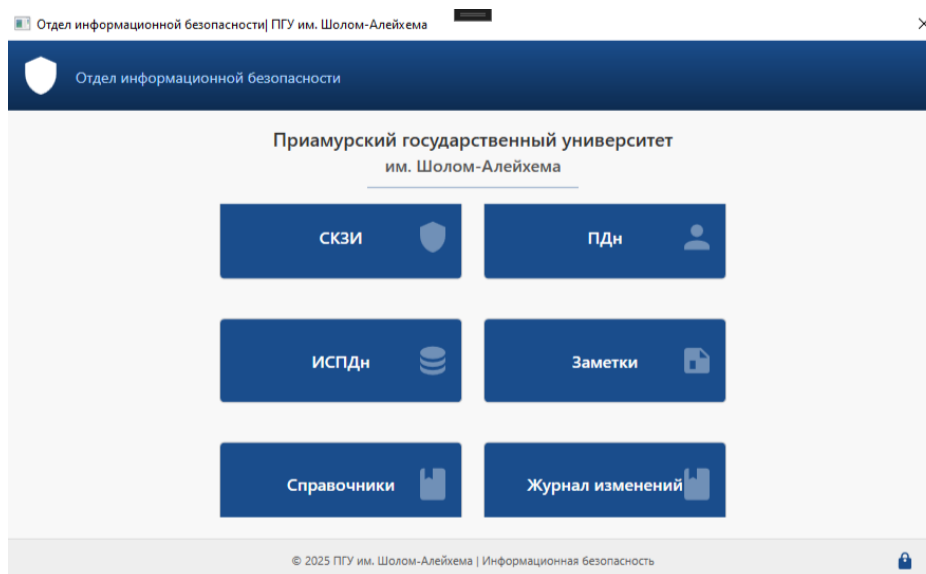


Рисунок 4. Главное меню

Перейдя по кнопке «СКЗИ» откроется форма «Учет средств криптографической защиты информации». Форма работы со средствами криптографической защиты информации (СКЗИ) реализована в виде оконного приложения с использованием технологии WPF и языка программирования C#. Она предназначена для обеспечения удобного и эффективного взаимодействия пользователя с данными, связанными с учетом и

отслеживанием СКЗИ (см. рис. 5). Интерфейс формы содержит таблицу для отображения списка записей, поля ввода и редактирования данных, а также элементы управления для фильтрации, навигации и формирования отчетов.

Учет средств криптографической защиты информации (СКЗИ)

Фiltrация

Наименование: Серийный номер: Номер экземпляра:

От кого: Кому рассылка: Дата ввода (от-до): -

Основные данные

Наименование СКЗИ: Серийный номер:

Номер экземпляра:

Получение | Рассылка | Ввод/вывод | Возврат | Уничтожение | Примечание

Дата ввода: Дата вывода:

Список СКЗИ

Наименование	Серийный №	Экземпляр №	От кого	Кому	Дата письма	Номер письма	Дата рассылки	Номер рассылки	Дата подтв.	Номер подтв.
Шифрование	SN123456	ЭКЗ-001	АО «ШифрСервис»	Иванов	26.05.2025	4	30.04.2025	4	08.05.2025	3
Аутентификация	SN123457	ЭКЗ-002	ЗАО «ЗащитаИнфо»	Иванов	26.05.2025	5	30.04.2025	4	08.05.2025	3
Хранение ключей	SN123459	ЭКЗ-004	АО «ШифрСервис»	Смирнов	01.01.0001	6	15.04.2025	6	10.05.2025	5
Аутентификация	6	44	ООО «СигналСекью»	Леонтьев	01.01.0001	78	01.01.0001		01.01.0001	

Добавить запись | Удалить запись | Сохранить изменения | Отчет | Начало | Конец | Применить фильт | Сбросить фильт

Рисунок 5. Форма «СКЗИ»

При запуске формы автоматически загружаются данные из локальной базы данных SQLite, включая список всех СКЗИ и необходимые справочники, такие как наименования типов СКЗИ, организации-поставщики и сотрудники, которым отправляется рассылка. Все данные представлены в удобном для восприятия виде: информация о каждом СКЗИ отображается в строке таблицы, а при выборе конкретной записи соответствующие данные загружаются в поля ввода для просмотра или редактирования.

Форма предоставляет широкий функционал: добавление новых записей, обновление существующих, удаление и применение фильтров по различным параметрам, таким как название СКЗИ, серийный номер, организация-отправитель, ФИО получателя и диапазон дат ввода в эксплуатацию. Все операции сопровождаются проверкой корректности вводимых данных, включая обязательные поля и формат дат. Это позволяет свести к минимуму ошибки при работе с системой.

Для повышения информационной безопасности и прозрачности все действия пользователей – такие как создание, редактирование и удаление записей – регистрируются в JSON-файле. Журнал содержит информацию о пользователе, типе операции, идентификаторе объекта, а также старые и новые значения ключевых полей. Это позволяет отслеживать изменения и контролировать доступ к данным.

Форма также поддерживает удобную навигацию по списку записей с помощью специальных кнопок перехода к первой или последней записи, что особенно актуально при работе с большим объемом информации. Кроме того,

реализована возможность сортировки данных по любому столбцу таблицы, что упрощает поиск и анализ информации.

Еще одной важной функцией является создание отчета в формате PDF по всем текущим записям, который может быть сохранен на устройстве пользователя или распечатан. Для этого нужно нажать на кнопку «Отчет», тогда откроется форма для формирования отчета (см. рис.6). Отчет формируется с использованием заранее заданного шаблона, что гарантирует структурированное и наглядное представление данных.

Рисунок 6. Форма для формирования отчета по журналу СКЗИ

Форма FormOtchet – это окно WPF-приложения, предназначенное для формирования отчета по данным из таблицы Skzi базы данных. Форма позволяет пользователю выбрать диапазон дат, указать ФИО и должность ответственного лица, а также путь сохранения PDF-файла. При нажатии на кнопку «Создать» происходит фильтрация данных и генерация отчета с помощью библиотеки QuestPDF. Отчет состоит из трех страниц: титульного листа, информационной страницы с описанием нормативной базы и таблицы с данными о СКЗИ. Таблица включает подробные сведения об экземплярах средств криптографической защиты информации, включая даты получения, рассылки, вывода из эксплуатации и уничтожения. Форма также предоставляет возможность выбора пути сохранения файла через стандартный диалог (см. рис.7, 8).

Следующая кнопка «ИСПДн», при нажатии откроется форма «Учет информационных систем персональных данных (ИСПДн)» (см. рис.8). Форма работы с информационными системами персональных данных (ИСПДн) реализована в виде оконного приложения на платформе WPF с использованием языка C#. Её основное назначение – обеспечение удобного и эффективного взаимодействия пользователя с данными об ИСПДн, включая операции добавления, редактирования, удаления записей, а также

отслеживания сроков действия систем. Интерфейс формы содержит таблицу для отображения списка систем, поля для ввода и редактирования информации, а также элементы управления для выполнения действий над записями.

Номер	Наименование	Дата старта	Дата окончания	Статус
0001	Маска	01.01.2024	07.06.2025	Просрочена
0002	ЛВС645	05.09.2023	10.04.2026	Активна
00002	ФИС ГНА РДО	08.09.2022	13.06.2025	Заканчивается
0003	ООКД	01.08.2022	31.07.2025	Активна

Рисунок 8. Форма «Учет СКЗИ»

При запуске формы происходит загрузка всех доступных записей из локальной базы данных SQLite. Каждая запись представляет собой ИСПДн с такими параметрами, как номер, наименование, дата начала и дата окончания действия системы. Данные отображаются в таблице, а при выборе конкретной строки информация о ней загружается в соответствующие поля ввода, что позволяет просматривать или изменять данные.

Форма реализует функцию автоматического уведомления о сроках действия ИСПДн (см. рис.9). Система проверяет актуальность каждой записи и выводит предупреждения, если срок действия системы истек или истекает в ближайшие дни. Уведомления могут отображаться как в интерфейсе формы, так и в виде всплывающих сообщений, обеспечивая своевременное информирование пользователя о необходимости продления или закрытия системы.

Форма предоставляет пользователю возможность выполнять полный цикл работы с записями: добавлять новые ИСПДн, редактировать существующие и удалять ненужные. Все операции сопровождаются проверкой корректности вводимых данных, особенно обязательных полей – номера и наименования системы, а также дат начала и окончания. Это обеспечивает целостность и достоверность хранимой информации.

Кроме того, форма поддерживает удобную навигацию по списку записей, а также выделение записей с истекшим или приближающимся сроком действия с помощью цветовых индикаторов, что повышает наглядность и

упрощает работу с данными. Также реализована возможность двойного щелчка по строке для быстрого перехода к редактированию выбранной записи.

Вся работа пользователя с формой регистрируется, что позволяет отслеживать изменения и обеспечивает дополнительный уровень контроля за внесенными данными. Регистрация выполняется с указанием имени пользователя, типа операции, идентификатора объекта и измененных значений.

Следующая кнопка «ПДн» при нажатии на нее открывается форма «Учет персональных данных». Форма работы с персональными данными (ПДн) (см. рис.9) реализована в виде оконного приложения на платформе WPF с использованием языка программирования C#. Её основное назначение – обеспечение удобного и структурированного взаимодействия пользователя с информацией о физических лицах, данные которых обрабатываются в рамках информационной системы.

Учет персональных данных (ПДн)

Фильтрация

Фамилия: **Сбросить фильтр**

Персональные данные

Фамилия: ☐ Согласие на обработку ПДн

Имя: ☐ Согласие на распространение ПДн

Отчество: Факультет (для студента):

Дата рождения: Выбор даты 15 Подразделение (для сотрудника):

Список персональных данных

Фамилия	Имя	Отчество	Дата рождения	Согласие на обработку	Согласие на распространение	Факультет
Анишкова	Анастасия	Сергеевна	16.01.2001	True	False	Математики, инфс
Пайдершев	Алексей	Константинович	01.10.2004	True	False	Филологии, истор
Екимова	Яна	Сергеевна	09.03.2000	True	False	Математики, инфс
Матвеева	Дарья	Сергеевна	16.03.2000	True	False	Математики, инфс

Добавить запись **Удалить запись** **Сохранить изменения** **Импорт** **Экспорт**

Рисунок 9. Форма учета персональных данных

Форма предоставляет функционал для добавления, редактирования, удаления записей, а также импорта и экспорта данных в формате Excel.

Интерфейс формы включает в себя таблицу для отображения списка персональных данных, поля для ввода информации о каждом лице и элементы управления для выполнения операций над записями. При запуске форма автоматически загружает данные из локальной базы данных SQLite, включая список сотрудников или лиц, чьи персональные данные обрабатываются, а также справочники факультетов и связанных с ними подразделений. Все

записи отображаются в виде таблицы, что обеспечивает наглядность и упрощает работу с информацией.

Каждая запись содержит такие поля, как фамилия, имя, отчество, дата рождения, наличие согласия на обработку персональных данных, согласие на их распространение, принадлежность к факультету и подразделению. При выборе строки в таблице данные о лице загружаются в соответствующие поля ввода, где их можно просматривать или редактировать. Также реализована возможность двойного щелчка по строке для быстрого перехода к редактированию выбранной записи.

Форма поддерживает функцию фильтрации данных по фамилии, что позволяет пользователю быстро находить нужную информацию. Реализованы кнопки для очистки фильтров и полей ввода, что повышает удобство использования интерфейса. Кроме того, предусмотрена проверка корректности вводимых данных, особенно обязательных полей – фамилии, имени и даты рождения. Это помогает избежать ошибок и гарантирует целостность хранимой информации.

Все действия пользователя – создание, редактирование и удаление записей – подробно регистрируются в JSON-файле, что обеспечивает прозрачность изменений и дополнительный уровень контроля за внесенными данными. Регистрация включает в себя имя пользователя, тип операции, идентификатор объекта, а также старые и новые значения полей, что позволяет эффективно отслеживать историю изменений.

Форма также поддерживает экспорт текущих данных в формате Excel, что позволяет сохранять выгрузку для последующего анализа или передачи другим сотрудникам (см. рис. 10). Кроме того, реализована возможность импорта данных из файла Excel, что значительно упрощает массовое добавление новых записей и интеграцию с внешними системами (см. рис. 11).

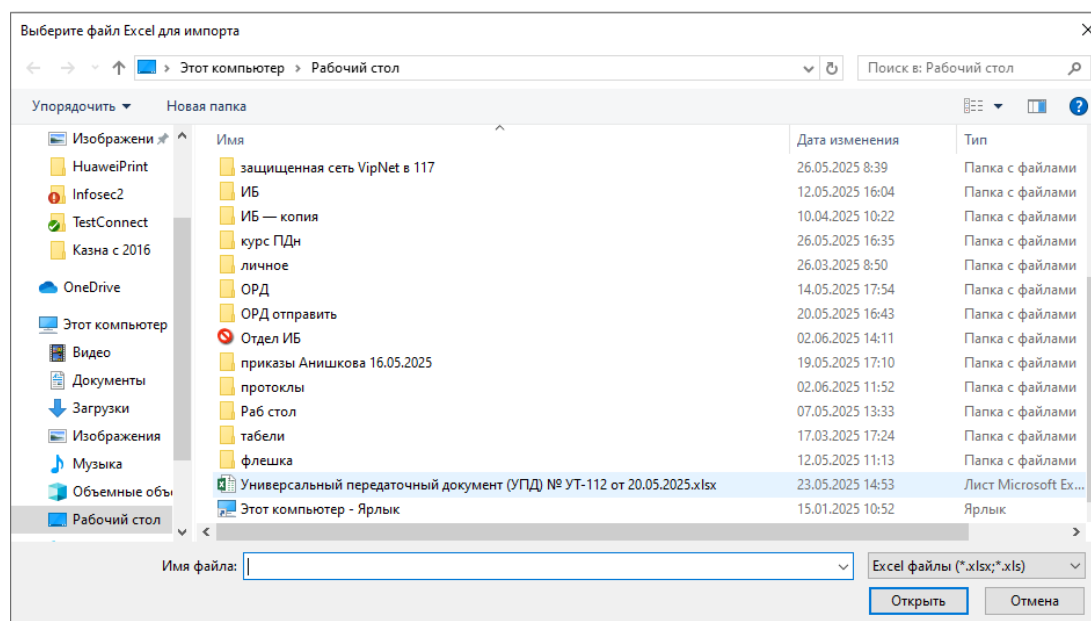


Рисунок 10. Импорт файла

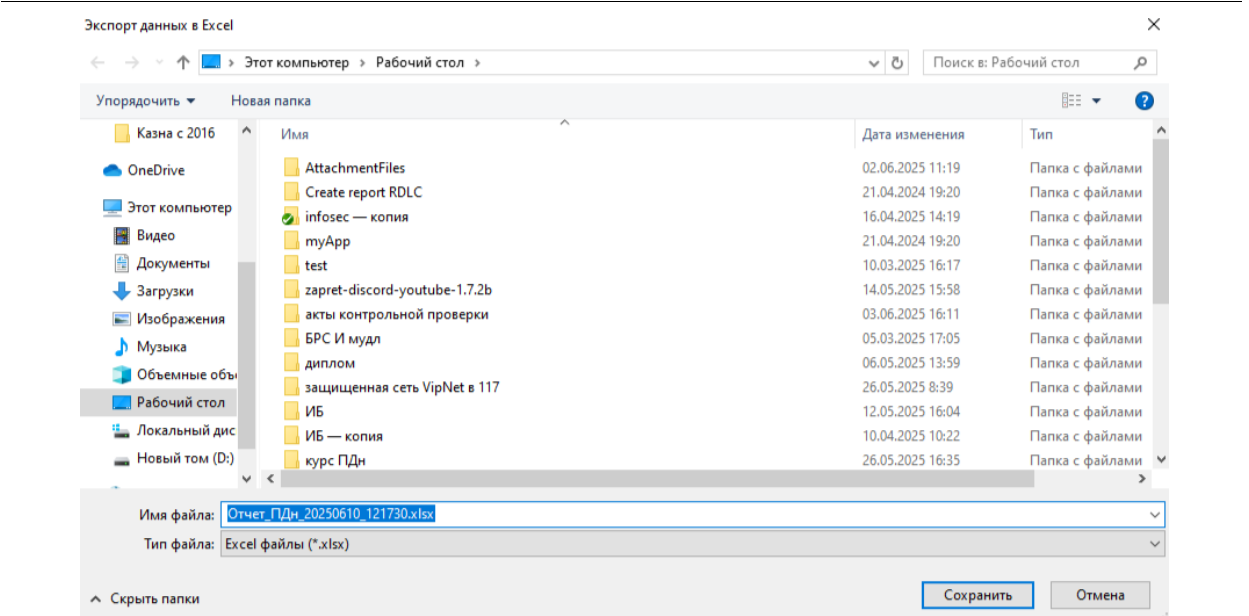


Рисунок 11. Экспорт файла

Следующая кнопка «Заметки», при нажатии на нее открывается форма «Управление заметками» (см. рис.12). Форма «Управление заметками» представляет собой оконный интерфейс, реализованный в среде WPF с использованием языка программирования C#. Она предназначена для удобного хранения, просмотра, добавления, редактирования и удаления заметок, связанных с деятельностью отдела информационной безопасности. Форма интегрирована в общую информационную систему и взаимодействует с локальной базой данных SQLite через Entity Framework Core.

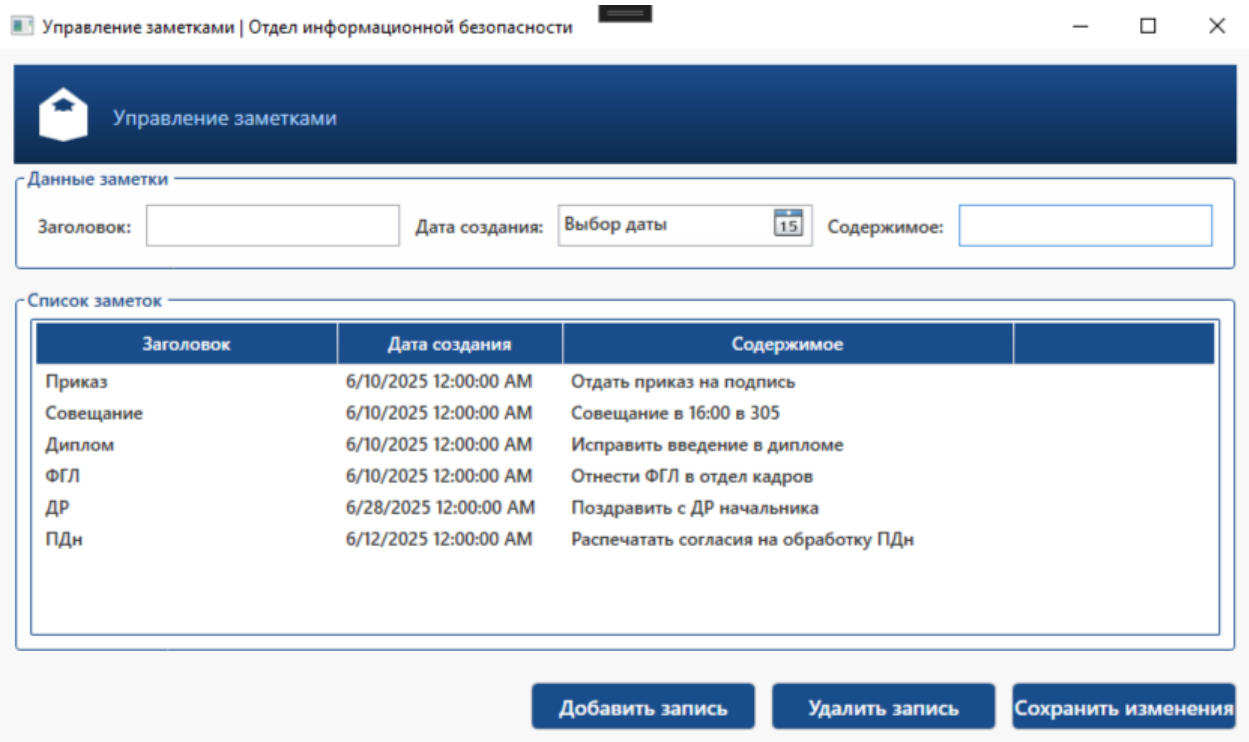


Рисунок 1 Форма управления заметками

Интерфейс формы состоит из основных элементов: таблицы,

отображающей список всех заметок, полей ввода для заполнения или изменения данных, а также кнопок управления, обеспечивающих выполнение операций над записями. При запуске формы происходит автоматическая загрузка всех доступных заметок из базы данных, которые отображаются в виде списка, где каждая запись содержит информацию о названии, дате создания и содержании заметки.

Пользователь может выбрать конкретную заметку из списка, после чего её данные будут загружены в поля ввода для просмотра или редактирования. Также предусмотрена возможность двойного щелчка по строке списка для быстрого перехода к редактированию выбранной записи. Все действия пользователя – такие как создание, изменение или удаление заметок – подробно регистрируются в JSON-файле, что обеспечивает прозрачность изменений и дополнительный уровень контроля за внесёнными данными.

Форма поддерживает стандартные функции работы с данными:

1. Добавление новой заметки с обязательным указанием названия, даты и текста;
2. Редактирование существующих записей с сохранением изменений в базе данных и соответствующей записью в журнале изменений;
3. Удаление записей после подтверждения пользователем с последующим логированием удалённых данных;
4. Обновление списка заметок после каждой операции для отображения актуальных данных.

Все операции сопровождаются проверкой корректности вводимых данных, особенно обязательных полей – названия и содержания заметки. Это позволяет свести к минимуму ошибки при работе с системой и гарантирует целостность хранимой информации.

Далее кнопка «Журнал изменений» она открывает форму «Журнал изменений». Форма «Журнал изменений» (см. рис.13) представляет собой модуль информационной системы, предназначенный для просмотра и анализа журнала изменений, ведущегося в рамках работы с различными разделами приложения. Реализованная в виде оконного интерфейса на платформе WPF с использованием языка C#, форма предоставляет пользователю возможность отслеживать действия, выполненные в системе, такие как добавление, редактирование или удаление записей. Все изменения регистрируются в JSON-файле с указанием типа операции, имени пользователя, времени выполнения действия, а также старых и новых значений изменённых данных.

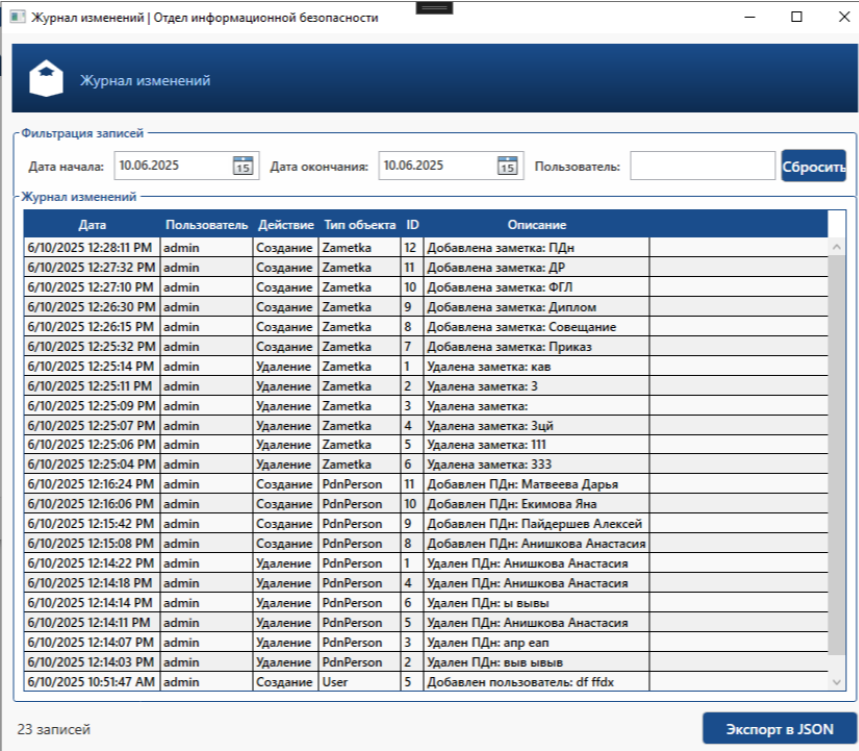


Рисунок 13. Форма «Журнал изменений»

Интерфейс формы включает элементы фильтрации записей по дате, имени пользователя и типу действия, что позволяет удобно анализировать журнал и находить нужные события. Пользователь может задать временной диапазон, выбрать конкретного пользователя или тип операции, или оставить фильтры общими для получения полной картины изменений. Результаты отображаются в таблице, где каждая запись содержит информацию о дате и времени события, пользователе, типе действия и кратком описании произошедшего.

Кроме того, форма поддерживает функцию экспорта содержимого журнала в формате JSON (см. рис.14, 15), что обеспечивает возможность последующего хранения, передачи или анализа данных вне приложения. Также реализована возможность сброса фильтров до начальных значений, что упрощает повторный просмотр всего доступного списка изменений.

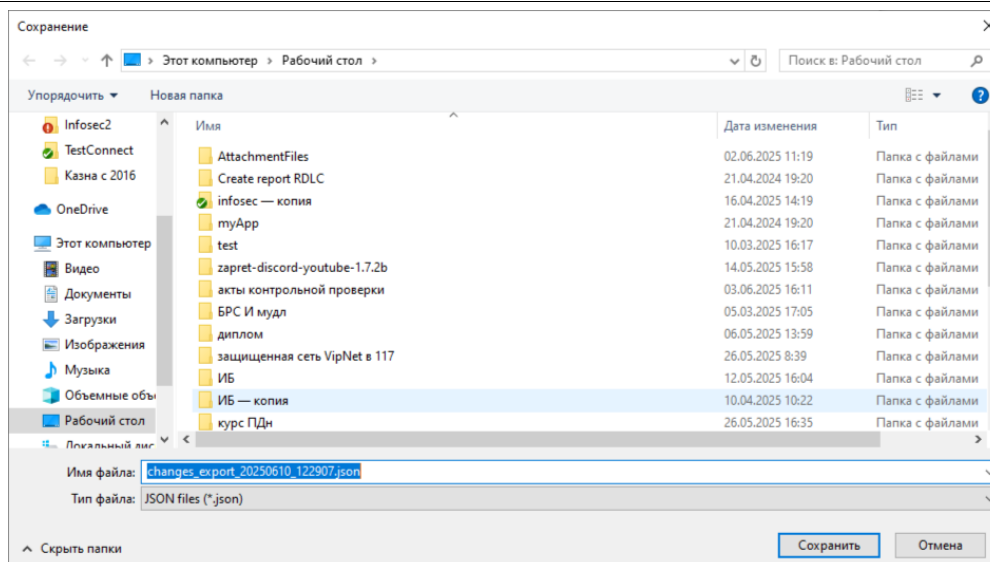


Рисунок 14. Сохранение файла

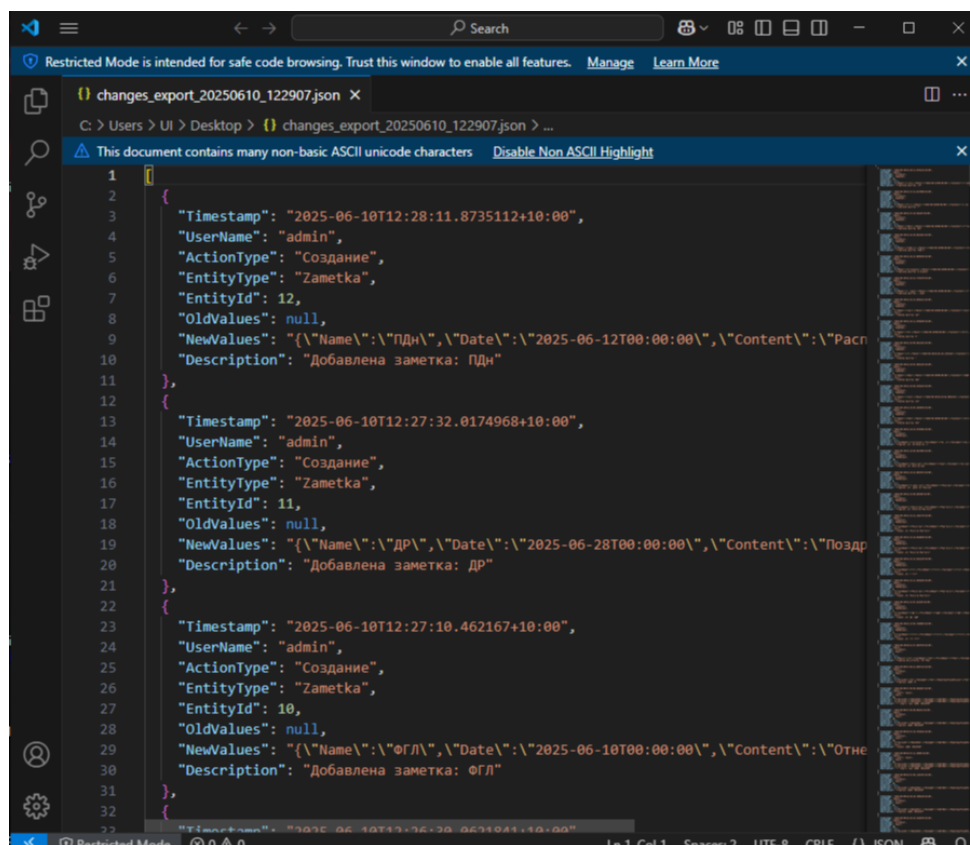


Рисунок 15. Файл

Форма «Журнал изменений» играет важную роль в обеспечении прозрачности и безопасности использования системы, предоставляя администраторам и другим ответственным лицам инструменты для контроля за вносимыми изменениями, аудита действий пользователей и выявления возможных ошибок или несанкционированных операций. Благодаря продуманному дизайну и функциональным возможностям эта форма значительно повышает уровень доверия к данным и способствует эффективному управлению информационной системой в целом.

Кнопка «Справочники» открывает форму «Справочники системы информационной безопасности» (см. рис.16). Форма «Справочники» представляет собой центральный интерфейс информационной системы, реализованный в виде оконного приложения на платформе WPF с использованием языка программирования C#. Данная форма выполняет роль справочного меню, обеспечивающего навигацию по различным модулям и функциональным разделам системы. Интерфейс формы содержит набор кнопок, каждая из которых предназначена для открытия соответствующего окна приложения – будь то формы для работы со справочниками.

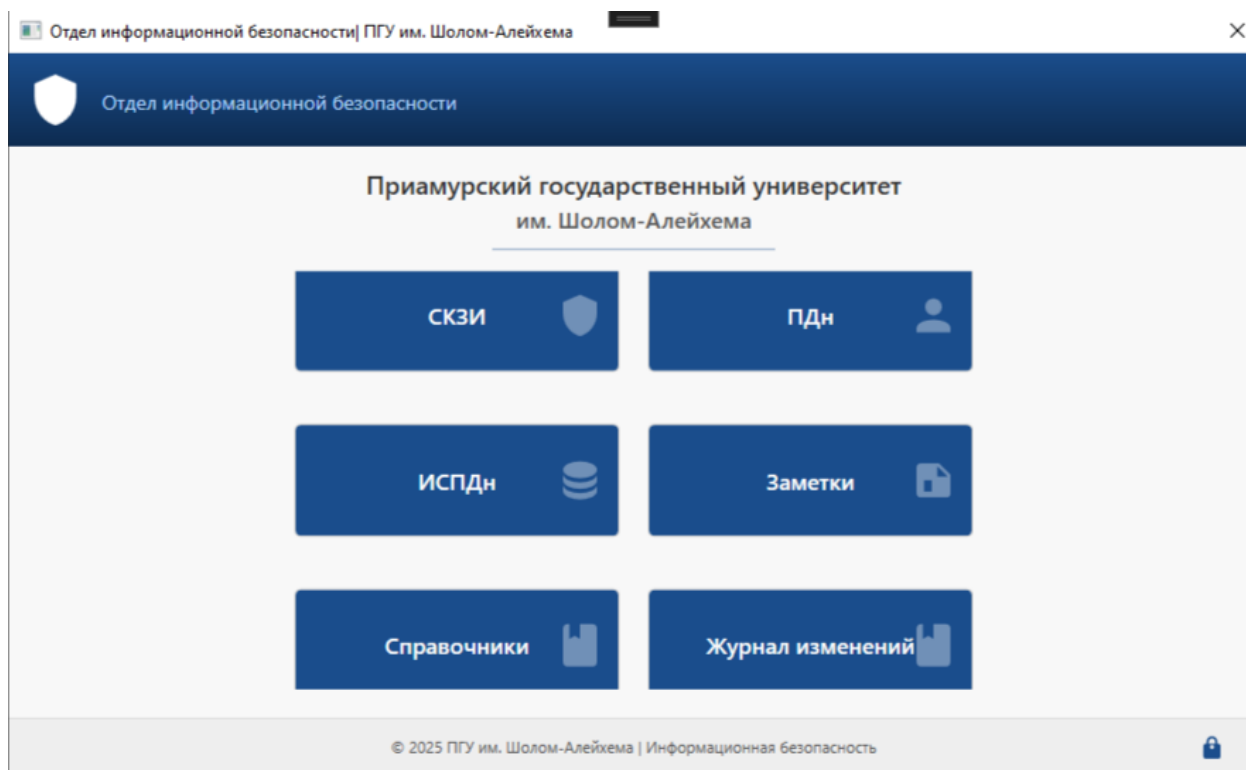


Рисунок 16. Справочная форма

Форма «Факультет» (см. рис.17) предназначена для управления списком факультетов в приложении, обеспечивая возможность просмотра, добавления, редактирования и удаления записей. При загрузке формы данные о факультетах считываются из базы данных и отображаются в списке `FakultetList`, а поле `InputName` используется для ввода или изменения названия выбранного факультета. Добавление новой записи осуществляется с помощью кнопки «Добавить», редактирование – через кнопку «Обновить / Редактировать», а удаление выполняется после подтверждения с использованием соответствующей кнопки. Все изменения логируются с указанием типа операции, имени пользователя, идентификатора и новых или старых значений изменённой записи. Форма также предусматривает обработку ошибок, включая проверку наличия выделенного элемента перед выполнением операций и корректную обработку исключений при взаимодействии с базой данных. Двойное нажатие на элемент списка позволяет быстро загрузить его данные для редактирования. Таким образом,

форма предоставляет удобный интерфейс для администрирования справочника факультетов с возможностью отслеживания всех изменений.

Управление факультетами | Отдел информационной безопасности

Управление факультетами

Данные факультета

Наименование:

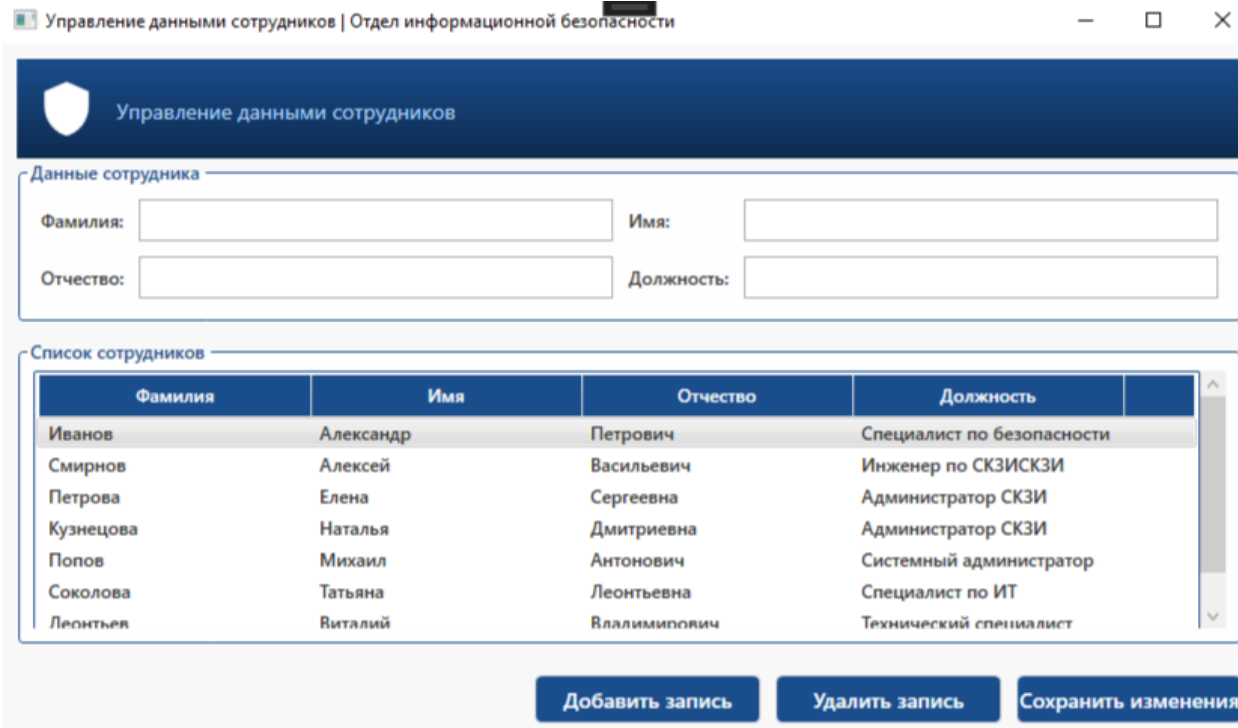
Список факультетов

Наименование
Математики, информационных технологий и техники
Филологии, истории и журналистики
Педагогики и психологии
Экономики, экологии и права
Информационных и промышленных технологий программ среднего профессионального образования
Промышленно-экономический факультет программ среднего профессионального образования
Лицей

Добавить запись Удалить запись Сохранить изменения

Рисунок 17. Форма «Факультеты»

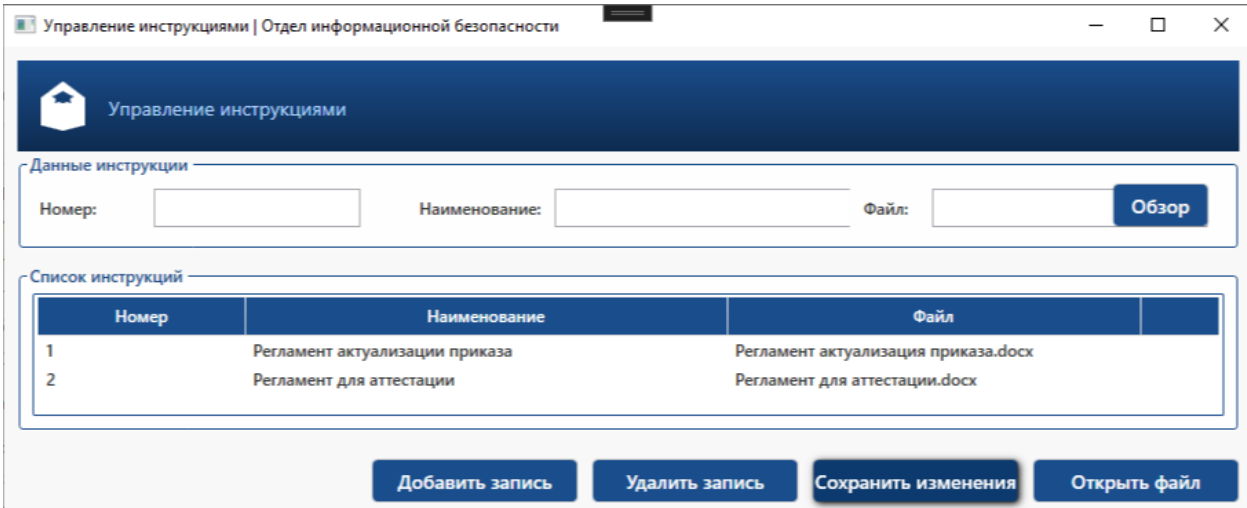
Форма «Управление данными сотрудников» (см. рис.18) предназначена для работы с информацией о сотрудниках, включая добавление, редактирование, удаление и просмотр записей, используемых, например, для рассылок или учёта персонала. При запуске формы загружается список сотрудников из базы данных, который отображается в таблице `fioRassilkiList`, а поля ввода предназначены для отображения и изменения данных выбранной записи: фамилии, имени, отчества и должности. Новая запись добавляется с помощью кнопки «Добавить», при этом данные сохраняются в базе данных и операция логируется с указанием типа действия, имени пользователя и изменённых данных. Удаление выполняется после подтверждения пользователем и также сопровождается записью в журнал изменений. Редактирование осуществляется через обновление полей и нажатие кнопки «Обновить/Редактировать», что приводит к сохранению изменений в базе данных и логированию старых и новых значений. Двойное нажатие на элемент списка автоматически загружает его данные в форму для редактирования. Все операции реализованы с использованием Entity Framework и включают обработку возможных ошибок, обеспечивая стабильную и безопасную работу с данными. Форма предоставляет удобный и понятный интерфейс для управления списком сотрудников с возможностью отслеживания всех выполненных изменений.



Фамилия	Имя	Отчество	Должность
Иванов	Александр	Петрович	Специалист по безопасности
Смирнов	Алексей	Васильевич	Инженер по СКЗИСКЗИ
Петрова	Елена	Сергеевна	Администратор СКЗИ
Кузнецова	Наталья	Дмитриевна	Администратор СКЗИ
Попов	Михаил	Антонович	Системный администратор
Соколова	Татьяна	Леонтьевна	Специалист по ИТ
Леонтьев	Виталий	Владимирович	Технический специалист

Рисунок 18. Форма управления сотрудниками

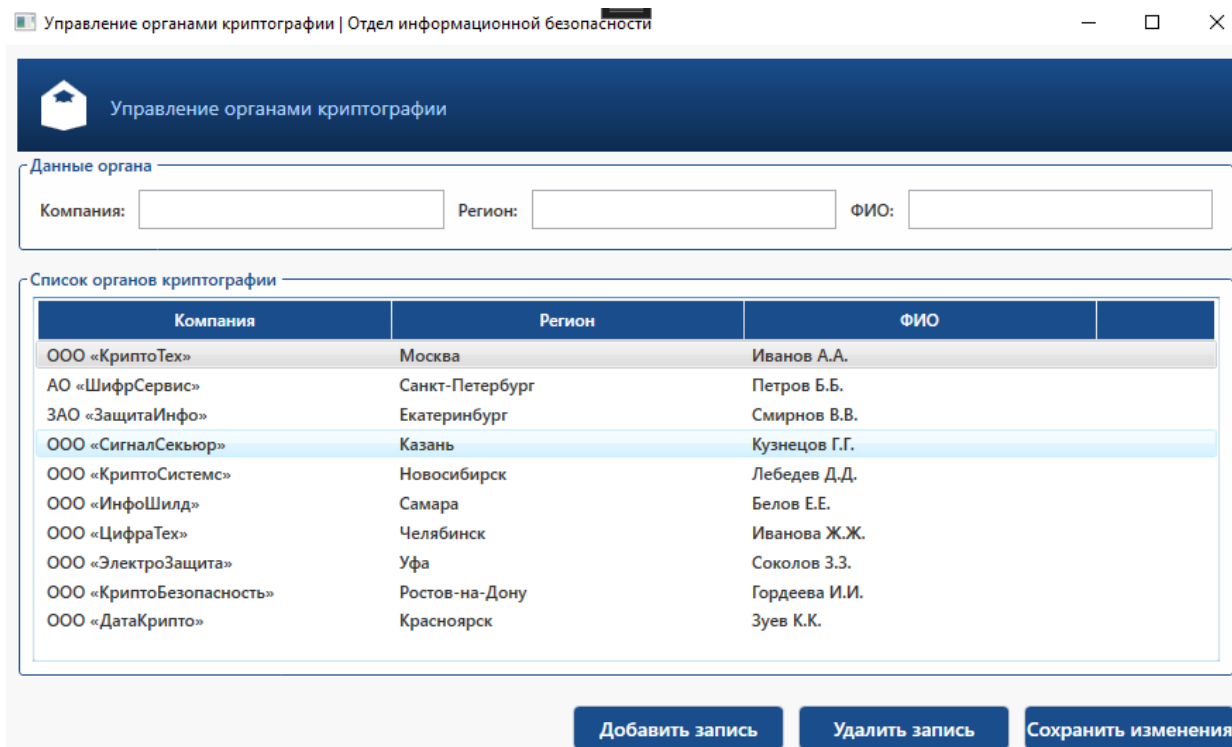
Форма «Инструкции» (см. рис.19) представляет собой интерфейс для управления списком инструкций или руководств в приложении, позволяя пользователю добавлять, редактировать, удалять и просматривать записи, сопровождаемые файлами. При открытии формы загружается список уже существующих записей из базы данных, каждая из которых содержит номер, наименование и, при необходимости, прикрепленный файл. Пользователь может выбрать запись для редактирования или удаления, а также добавить новую инструкцию, указав обязательные поля – номер и наименование, и при необходимости привязав к ней файл. Файлы сохраняются локально в специальной папке внутри каталога `AppData`, что обеспечивает их безопасное хранение и последующее открытие по мере необходимости. Кнопка «Обзор» позволяет выбрать нужный файл с компьютера, который затем копируется в рабочую директорию приложения. Редактирование записи предусматривает обновление всех её полей, включая замену файла с автоматическим удалением старого, если он был изменён. Удаление выполняется после подтверждения и сопровождается удалением как записи в базе данных, так и связанного файла с диска. Открытие файла реализовано через двойной клик по элементу списка или нажатие кнопки «Открыть», что запускает его в соответствующем приложении по умолчанию. Все операции – создание, редактирование и удаление – логируются с помощью класса `JsonChangeLogger`, где фиксируются тип действия, имя пользователя, идентификатор записи, старые и новые значения, а также описание произошедшего изменения. Форма обеспечивает понятный и удобный интерфейс для работы с документами, включая обработку ошибок, проверку вводимых данных и корректную работу с файловой системой и базой данных.



Номер	Наименование	Файл
1	Регламент актуализации приказа	Регламент актуализация приказа.docx
2	Регламент для аттестации	Регламент для аттестации.docx

Рисунок 19. Форма управление инструкциями

Форма «Управление органами криптографии» (см. рис.20) предназначена для управления списком организаций, связанных с криптографическими средствами, предоставляя пользователю возможность просматривать, добавлять, редактировать и удалять записи. При запуске формы из базы данных загружается список всех зарегистрированных организаций, который отображается в элементе управления `organCryptoList`, а поля ввода – `InputCompany`, `InputRegion` и `InputFIO` – используются для отображения и изменения информации о выбранной организации. Новая запись создаётся с помощью кнопки «Добавить», при этом данные сохраняются в базе данных, а операция логируется с указанием типа действия, имени пользователя, идентификатора записи и новых значений. Удаление выполняется после подтверждения, обеспечивая удаление записи из базы данных и занесение соответствующей записи в журнал изменений. Редактирование доступно через кнопку «Обновить/Редактировать» и позволяет изменить наименование организации, регион её расположения и контактное лицо, при этом старые значения также сохраняются в журнале для отслеживания изменений. Выбор элемента списка автоматически заполняет поля ввода данными выбранной организации, а двойной щелчок по записи открывает возможность быстрого редактирования или вызывает окно просмотра информации. Форма реализована с использованием Entity Framework, что обеспечивает стабильную работу с данными, а также содержит обработку ошибок и проверку корректности ввода информации. Всё это делает форму удобным и надёжным инструментом для администрирования справочника организаций, связанных с криптографическими системами, с возможностью полного отслеживания всех выполненных действий.



Компания	Регион	ФИО
ООО «КриптоТех»	Москва	Иванов А.А.
АО «ШифрСервис»	Санкт-Петербург	Петров Б.Б.
ЗАО «ЗащитаИнфо»	Екатеринбург	Смирнов В.В.
ООО «СигналСекьюр»	Казань	Кузнецов Г.Г.
ООО «КриптоСистемс»	Новосибирск	Лебедев Д.Д.
ООО «ИнфоШилд»	Самара	Белов Е.Е.
ООО «ЦифраТех»	Челябинск	Иванова Ж.Ж.
ООО «ЭлектроЗащита»	Уфа	Соколов З.З.
ООО «КриптоБезопасность»	Ростов-на-Дону	Гордеева И.И.
ООО «ДатаКрипто»	Красноярск	Зуев К.К.

Рисунок 20. Форма управление органами криптографии

Форма «Подразделение» (см. рис.21) предназначена для управления списком подразделений в приложении, предоставляя пользователю возможность просматривать, добавлять, редактировать и удалять записи. При запуске формы из базы данных загружается список всех существующих подразделений, который отображается в элементе 'PodrazdelenieList', а поле 'InputName' используется для отображения или изменения названия выбранного подразделения. Новая запись создаётся с помощью кнопки «Добавить», после чего данные сохраняются в базе данных, а операция логируется с указанием типа действия, имени пользователя, идентификатора записи и нового значения. Удаление выполняется после подтверждения, обеспечивая удаление как самой записи, так и соответствующей информации в журнале изменений. Редактирование доступно через кнопку «Обновить/Редактировать» и позволяет изменить наименование подразделения, при этом старое значение сохраняется в логе для отслеживания изменений. Выбор элемента списка автоматически заполняет поля ввода данными выбранной записи, а двойной щелчок по элементу также позволяет быстро перейти к редактированию. Форма реализована с использованием Entity Framework, что обеспечивает стабильную работу с данными, а также содержит обработку ошибок и проверку наличия выделенной строки перед выполнением операций. Всё это делает форму удобным и надёжным инструментом для администрирования справочника подразделений с возможностью полного отслеживания всех выполненных действий.

Управление подразделениями | Отдел информационной безопасности

Управление подразделениями

Данные подразделения

Наименование:

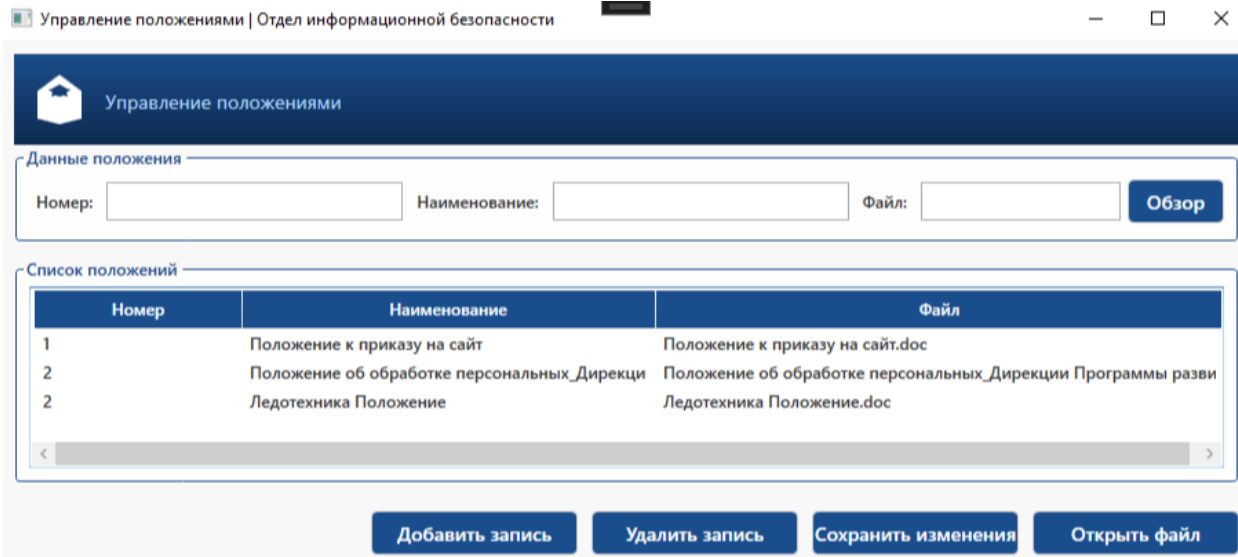
Список подразделений

Наименование
Управление бухгалтерского учета и финансового контроля
Учебно-методическое управление
Управление научно-исследовательской и инновационной деятельностью
Управление экономики и имущественных отношений
Отдел информационной безопасности
Отдел учебных вычислительных лабораторий
Отдел информационных технологий
Отдел сервиса компьютерных средств и связи

Добавить запись Удалить запись Сохранить изменения

Рисунок 21. Форма управления подразделениями

Форма «Положение» (см. рис.22) представляет собой интерфейс для работы с документами, относящимися к различным положениям организации, позволяя пользователю добавлять, редактировать, удалять и просматривать записи с возможностью прикрепления файлов. При запуске формы из базы данных загружается список всех сохранённых положений, отображаемых в элементе `PolozhenieList`, а поля `InputNomer`, `InputName` и `InputFile` используются для отображения или изменения информации о выбранной записи, включая путь к связанному файлу. Новая запись создаётся с помощью кнопки «Добавить», после чего данные сохраняются в базе данных, а при необходимости – файл копируется в специальную папку внутри каталога `AppData` для последующего доступа к нему. Удаление выполняется после подтверждения пользователем, при этом вместе с записью из базы данных удаляется и связанный с ней файл с диска, если он существует. Редактирование осуществляется через соответствующую кнопку и позволяет обновить номер, наименование и заменить прикреплённый файл, при этом старый файл автоматически заменяется на новый, а все изменения логируются с указанием типа операции, имени пользователя, идентификатора записи и старых или новых значений. Открытие файла реализовано как через двойной клик по элементу списка, так и через отдельную кнопку, что обеспечивает удобство и быстроту доступа к документам. Форма реализована с использованием Entity Framework, содержит обработку ошибок, проверяет корректность ввода данных и обеспечивает стабильное взаимодействие с файловой системой и базой данных, предоставляя тем самым надёжный и интуитивно понятный интерфейс для управления нормативными документами с возможностью полного отслеживания всех выполненных изменений.



Управление положениями | Отдел информационной безопасности

Управление положениями

Данные положения

Номер: Наименование: Файл: **Обзор**

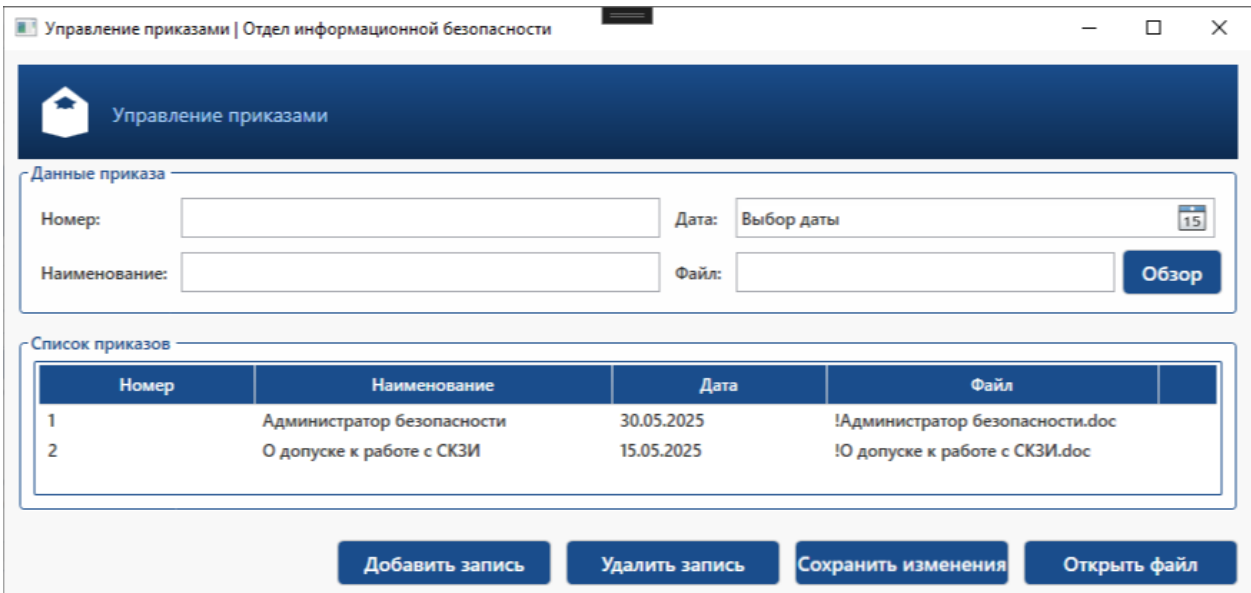
Список положений

Номер	Наименование	Файл
1	Положение к приказу на сайт	Положение к приказу на сайт.doc
2	Положение об обработке персональных Дирекци	Положение об обработке персональных Дирекции Программы разви
2	Ледотехника Положение	Ледотехника Положение.doc

Добавить запись **Удалить запись** **Сохранить изменения** **Открыть файл**

Рисунок 2 Форма управление положениями

Форма «Приказ» (см. рис.23) предназначена для работы с документами, представляющими собой приказы организации, предоставляя пользователю возможность добавлять, редактировать, удалять и просматривать записи с привязкой к ним файлов. При запуске формы из базы данных загружается список всех существующих приказов, отображаемый в элементе управления 'PrikazList', а поля ввода – 'InputNomer', 'InputName', 'InputDate' и 'InputFile' – используются для отображения и изменения информации о выбранном приказе. Новая запись создаётся с помощью кнопки «Добавить», после чего данные сохраняются в базе данных, а связанный файл копируется в специальную папку внутри каталога 'AppData' для последующего использования. Удаление выполняется после подтверждения, при этом вместе с записью из базы данных удаляется и соответствующий файл с диска, если он был прикреплен. Редактирование доступно через кнопку «Обновить/Редактировать» и позволяет изменить номер, наименование, дату и заменить прикрепленный файл, при этом все изменения логируются с указанием типа действия, имени пользователя, идентификатора записи и старых или новых значений. Открытие файла реализовано как двойным щелчком по элементу списка, так и через отдельную кнопку, что обеспечивает удобство и быстроту доступа к документам. Форма реализована с использованием Entity Framework и содержит обработку ошибок, проверку корректности ввода данных и стабильное взаимодействие с файловой системой, предоставляя таким образом надёжный и интуитивно понятный интерфейс для управления приказами с возможностью полного отслеживания всех выполненных изменений.



Номер	Наименование	Дата	Файл
1	Администратор безопасности	30.05.2025	!Администратор безопасности.doc
2	О допуске к работе с СКЗИ	15.05.2025	!О допуске к работе с СКЗИ.doc

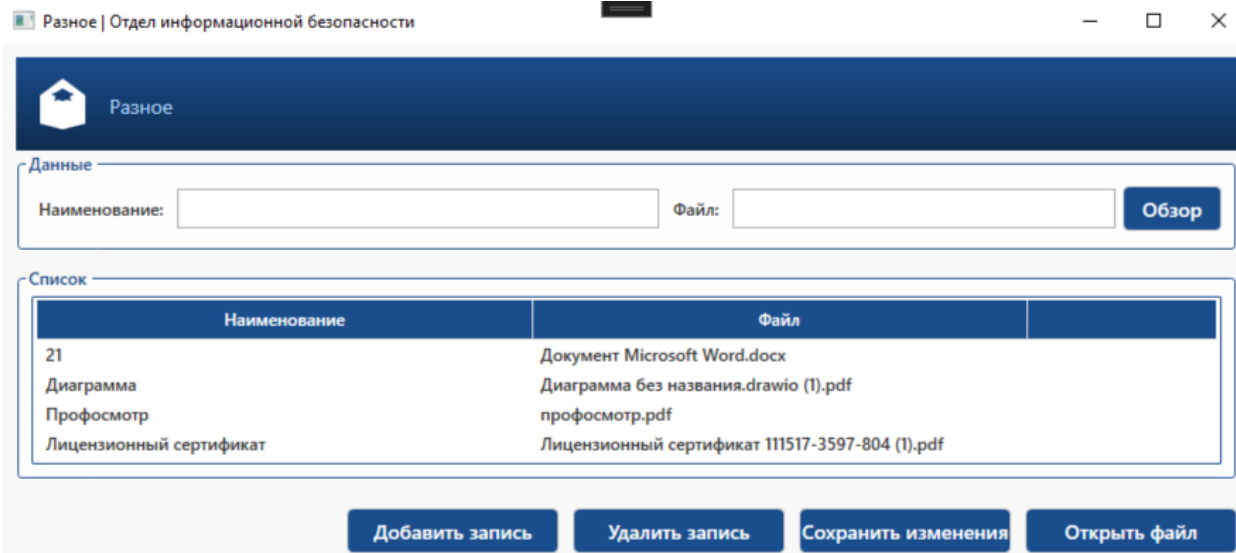
Рисунок 23. Форма управление приказами

Форма «Программное обеспечение» (см. рис.24) предназначена для работы с информацией о программном обеспечении, связанном с образовательными программами или внутренними нормативными документами организации, предоставляя пользователю возможность добавлять, редактировать, удалять и просматривать записи с привязкой к ним файлов. При запуске формы из базы данных загружается список всех существующих записей, который отображается в элементе управления `ProgObList`, а поля ввода – `InputNomer`, `InputName` и `InputFile` – используются для отображения и изменения информации о выбранной записи. Новая запись создаётся с помощью кнопки «Добавить», после чего данные сохраняются в базе данных, а при необходимости – файл копируется в специальную папку внутри каталога `AppData` для последующего доступа к нему. Удаление выполняется после подтверждения пользователем, при этом вместе с записью из базы данных удаляется и связанный с ней файл с диска, если он существует. Редактирование осуществляется через кнопку «Обновить/Редактировать» и позволяет изменить номер, наименование и заменить прикрепленный файл, при этом все изменения логируются с указанием типа действия, имени пользователя, идентификатора записи и старых или новых значений. Открытие файла реализовано как двойным щелчком по элементу списка, так и через отдельную кнопку, что обеспечивает удобство и быстроту доступа к документам. Форма реализована с использованием Entity Framework и содержит обработку ошибок, проверку корректности ввода данных и стабильное взаимодействие с файловой системой, предоставляя таким образом надёжный и интуитивно понятный интерфейс для управления записями о программном обеспечении с возможностью полного отслеживания всех выполненных изменений.

Номер	Наименование	Файл
ПО-001/2023	Программа ИСПДн-001	docsprogob_001.pdf
ПО-002/2023	Программа ИСПДн-002	docsprogob_002.pdf
ПО-003/2023	Программа ИСПДн-003	docsprogob_003.pdf

Рисунок 24. Форма «Программное обеспечение»

Форма «Разное» (см. рис.25) предназначена для работы с произвольными документами или файлами, не входящими в строго определённые категории других форм, предоставляя пользователю возможность добавлять, редактировать, удалять и просматривать записи с прикрепленными файлами. При запуске формы из базы данных загружается список всех существующих записей, который отображается в элементе управления `RaznoeList`, а поле `InputName` используется для отображения или изменения наименования выбранной записи, а также указания пути к связанному файлу. Новая запись создаётся с помощью кнопки «Добавить», после чего данные сохраняются в базе данных, а при необходимости – файл копируется в специальную папку внутри каталога `AppData` для последующего доступа к нему. Удаление выполняется после подтверждения пользователем, при этом вместе с записью из базы данных удаляется и связанный с ней файл с диска, если он существует. Редактирование осуществляется через кнопку «Обновить/Редактировать» и позволяет изменить наименование и заменить прикрепленный файл, при этом все изменения логируются с указанием типа действия, имени пользователя, идентификатора записи и старых или новых значений. Открытие файла реализовано как двойным щелчком по элементу списка, так и через отдельную кнопку, что обеспечивает удобство и быстроту доступа к документам. Форма реализована с использованием Entity Framework и содержит обработку ошибок, проверку корректности ввода данных и стабильное взаимодействие с файловой системой, предоставляя таким образом надёжный и интуитивно понятный интерфейс для управления различными документами с возможностью полного отслеживания всех выполненных изменений.



Разное | Отдел информационной безопасности

Разное

Данные

Наименование: Файл: **Обзор**

Список

Наименование	Файл
21	Документ Microsoft Word.docx
Диаграмма	Диаграмма без названия.drawio (1).pdf
Профосмотр	профосмотр.pdf
Лицензионный сертификат	Лицензионный сертификат 111517-3597-804 (1).pdf

Добавить запись **Удалить запись** **Сохранить изменения** **Открыть файл**

Рисунок 25. Форма «Разное»

Форма «Руководство» (см. рис.26) предназначена для работы с информацией о руководствах или инструкциях, предоставляя пользователю возможность добавлять, редактировать, удалять и просматривать записи, каждая из которых содержит номер, наименование и путь к связанному файлу. При запуске формы из базы данных загружается список всех существующих записей, который отображается в элементе управления 'RukovodstvoList', а поля ввода – 'InputNomer', 'InputName' и 'InputFile' – используются для отображения и изменения информации о выбранной записи. Новая запись создаётся с помощью кнопки «Добавить», после чего данные сохраняются в базе данных при условии, что все обязательные поля заполнены, а указанный файл существует. Удаление выполняется после подтверждения, обеспечивая удаление записи из базы данных и автоматическое обновление списка. Редактирование доступно через соответствующую кнопку и позволяет изменить номер, наименование и путь к файлу, при этом все изменения логируются с указанием типа действия, имени пользователя, идентификатора записи и старых или новых значений. Форма также предоставляет возможность открытия файла по двойному щелчку на элементе списка или нажатием кнопки «Открыть», что запускает его в соответствующем приложении по умолчанию. Форма реализована с использованием Entity Framework, содержит обработку ошибок, проверку корректности ввода данных и обеспечивает стабильную работу с файловой системой и базой данных, предоставляя тем самым надёжный и интуитивно понятный интерфейс для управления записями о руководствах с возможностью полного отслеживания всех выполненных изменений.

Номер	Наименование	Файл
3	Новый текстовый документ	C:\Users\UI\Desktop\Новый текстовый до...
5555	Согласие2	C:\Users\UI\Desktop\согласие 2.pdf

Рисунок 26. Форма «Руководства»

Форма «Наименование СКЗИ» (см. рис.27) предназначена для управления справочником средств криптографической защиты информации (СКЗИ), предоставляя пользователю возможность просматривать, добавлять, редактировать и удалять записи, каждая из которых содержит информацию о типе, производителе и версии программного обеспечения. При запуске формы происходит загрузка всех доступных записей из базы данных, которые отображаются в списке `SkziNameList`, а поля ввода – `InputType`, `InputFirma` и `InputVersion` – служат для отображения и изменения параметров выбранной записи. Новая запись создаётся с помощью кнопки «Добавить», после чего данные сохраняются в базе данных, а операция логируется с указанием типа действия, имени пользователя, идентификатора записи и новых значений. Удаление выполняется после подтверждения, при этом удаляемая запись также фиксируется в журнале изменений с сохранением своих предыдущих значений. Редактирование осуществляется через кнопку «Обновить», позволяя изменять все основные параметры СКЗИ, при этом старые значения также сохраняются в логе для отслеживания изменений. Выбор элемента списка автоматически заполняет поля ввода данными выбранной записи, что обеспечивает удобство работы с информацией. Форма реализована с использованием Entity Framework, содержит обработку ошибок и проверку наличия выделенной строки перед выполнением операций, предоставляя тем самым стабильный и интуитивно понятный интерфейс для администрирования справочника СКЗИ с возможностью полного отслеживания всех выполненных изменений.

Наименования СКЗИ | Отдел информационной безопасности

Наименования СКЗИ

Данные СКЗИ

Тип: Фирма: Версия:

Список наименований СКЗИ

Тип	Фирма	Версия
Шифрование	КриптоПро	версия 4.0
Аутентификация	ViPNet	v5.1
Защита данных	Информационное поле	версия 2.3
Подпись цифровая	Криптосигнал	v3.0
Обнаружение	Безопасное обнаружение	версия 1.8
Хранение ключей	Хранилище ключей	версия 2.1
Передача данных	Ссылка на DataSecureLink	версия 1.9
Сетевой мониторинг	Сетевой страж	версия 2.0
Управление доступом	Контроль доступа	версия 3.2
Архивирование	Резервная копия хранилища	версия 1.7

Добавить запись Удалить запись Сохранить изменения

Рисунок 27. Форма «Наименование СКЗИ»

4 Вывод

Разработанное клиентское приложение является современным программным решением, которое полностью соответствует поставленным целям и требованиям. Оно может быть использовано не только в образовательных учреждениях, таких как ПГУ имени Шолом-Алейхема, но и в других организациях, где требуется централизованный учет и контроль средств информационной безопасности. Применение языка C# и среды Microsoft Visual Studio обеспечило надежность, производительность и простоту сопровождения приложения. Результаты работы могут быть использованы в учебном процессе для изучения методов разработки информационных систем, а также служить основой для последующего развития и внедрения в реальные бизнес-процессы.

Библиографический список

1. Баженов Р. И., Глухова А. А. Разработка информационной системы учета заказов в компьютерной мастерской //Современные научные исследования и инновации. 2014. №. 6. С. 30-30.
2. Беккер В. А., Полежаева О. А. Разработка автоматизированной системы учета стажеров // Современные информационные технологии. 2016. №. 23. С. 41-44.
3. Борисенко А. В., Петрова А. Н. Разработка системы учета развлекательного центра «Киров парк» //Молодежь и наука: актуальные проблемы фундаментальных и прикладных исследований. 2019. С. 230-233.
4. Исамухамедов Б. Б. Роль языка программирования C# в создании

программного обеспечения //Modern Science and Research. 2025. Т. 4. №. 2. С. 72-74.

5. Пасюкова Е. А. Обзор технических средств разработки графического интерфейса на языке программирования С //Постулат. 2021. №. 2.