

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика
(код и наименование направления подготовки, специальности)

Разработка программного обеспечения
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка системы для управления событиями и мероприятиями»

Обучающийся

А.В. Солончак

(Инициалы Фамилия)

(личная подпись)

Руководитель

Канд. техн. наук Д.Г. Токарев

ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия

Тольятти 2025

Аннотация

Тема: «Разработка системы для управления событиями и мероприятиями».

В ходе выполнения работы была спроектирована архитектура приложения, включая модули управления мероприятиями, участниками, уведомлениями и отчетностью. Реализована система на языке C# с использованием .NET Framework и SQL Server, обеспечивающая интеграцию с корпоративной инфраструктурой ЦИТ. Проведено тестирование функциональности, безопасности и производительности системы.

Во введении обоснована актуальность темы, определены цель, задачи и методы исследования.

Первый раздел посвящен анализу предметной области, включая изучение процессов управления мероприятиями в ЦИТ, исследование существующих решений и формулировку требований к разрабатываемой системе.

Во втором разделе представлено проектирование архитектуры системы, разработка ключевых модулей и интеграция с корпоративной инфраструктурой (Active Directory, почтовые сервисы, календари).

Третий раздел содержит описание методологии тестирования, включая результаты тестирования, а также анализ результатов проверки работоспособности и безопасности системы.

В заключении подведены итоги работы, сформулированы выводы и обозначены перспективы дальнейшего развития системы.

Бакалаврская работа состоит из введения, трех разделов, заключения, списка используемых источников и приложений. Общий объем работы составляет 56 страниц, содержит 1 таблицу, 31 рисунок и 31 источников литературы.

Содержание

Введение	4
1. Анализ предметной области и объекта исследования	7
1.1. Анализ деятельности ЦИТ и потребностей в автоматизации.....	7
1.2. Анализ существующих решений	10
1.3. Формулировка требований к системе	11
1.4. Выбор технологий	14
2. Разработка системы	18
2.1. Проектирование архитектуры	18
2.2. UML моделирование системы.....	22
2.3. Реализация функционала	25
2.4. Интеграция с инфраструктурой ЦИТ	29
3. Тестирование системы.....	37
3.1. Методология тестирования.....	37
3.2. Результаты тестирования	38
Заключение	50
Список используемых источников.....	54

Введение

Современный мир характеризуется беспрецедентным по своей скорости и масштабам развитием информационных технологий, которые проникают во все сферы человеческой деятельности, трансформируя традиционные подходы к организации бизнес-процессов, государственному управлению и социальному взаимодействию. В условиях глобальной цифровизации экономики и общества особую актуальность приобретают эффективные ИТ-решения, способные не только автоматизировать рутинные операции, но и создавать принципиально новые возможности для повышения производительности труда, оптимизации ресурсов и обеспечения устойчивых конкурентных преимуществ.

Центр информационных технологий (ЦИТ) занимает стратегически важное положение в этом процессе, выступая в качестве ключевого интегратора технологических инноваций и бизнес-потребностей. Как специализированная организация, ЦИТ выполняет комплексную функцию по разработке, внедрению и сопровождению информационных систем, помогая предприятиям различных отраслей адаптироваться к вызовам цифровой трансформации. Деятельность центра охватывает полный жизненный цикл ИТ-решений - от анализа потребностей заказчиков и проектирования архитектуры до технической поддержки и обучения пользователей.

Миссия ЦИТ заключается в создании инновационных технологических решений, направленных на:

- автоматизацию бизнес-процессов и повышение операционной эффективности;
- обеспечение высокого качества ИТ-услуг и сервисов;
- создание надежных систем информационной безопасности;
- внедрение цифровых технологий в управленческие процессы.

В условиях высокой динамики технологических изменений и усиливающейся конкуренции на рынке ИТ-услуг, ЦИТ последовательно реализует стратегию технологического лидерства, основанную на:

- применении современных методологий разработки (Agile, DevOps);
- использовании передовых платформ и инструментов (ITSM-системы, облачные технологии);
- внедрении аналитических систем для поддержки принятия решений;
- постоянном развитии компетенций сотрудников.

Актуальность настоящего исследования обусловлена несколькими ключевыми факторами:

- возрастающей потребностью предприятий в комплексных решениях для управления событиями и мероприятиями;
- необходимостью интеграции разрозненных систем планирования и учета;
- важностью автоматизации процессов взаимодействия с участниками мероприятий;
- потребностью в аналитических инструментах для оценки эффективности мероприятий.

Особую значимость работа приобретает в контексте цифровой трансформации экономики, когда традиционные подходы к организации мероприятий требуют принципиального пересмотра с учетом:

- распространения гибридных (очно-заочных) форматов;
- необходимости обеспечения кибербезопасности;
- требований к масштабируемости решений;
- интеграции с существующими корпоративными системами.

Практическая значимость исследования заключается в разработке универсальной платформы для управления мероприятиями, которая:

- позволяет централизовать все процессы планирования и проведения событий;

- обеспечивает прозрачность и контролируемость всех этапов;
- автоматизирует ключевые операции (регистрация, уведомление, отчетность);
- предоставляет аналитические инструменты для оценки эффективности.

Теоретическая база исследования основана на современных концепциях:

- системного анализа и проектирования ИТ-решений;
- управления жизненным циклом программного обеспечения;
- проектирования баз данных и интерфейсов [5];
- обеспечения информационной безопасности.

Методологическую основу работы составляют:

- принципы объектно-ориентированного анализа и проектирования;
- методы тестирования и обеспечения качества ПО;
- подходы к оценке экономической эффективности ИТ-решений.

Таким образом, настоящее исследование не только решает конкретную прикладную задачу по созданию системы управления мероприятиями, но и вносит вклад в развитие методологии цифровой трансформации организаций, предлагая комплексный подход к автоматизации данного класса бизнес-процессов. Полученные результаты могут быть использованы как для дальнейшего развития системы в ЦИТ, так и для тиражирования опыта в других организациях, сталкивающихся с аналогичными вызовами цифровой эпохи.

Перспективы развития работы связаны с:

- расширением функциональности системы (интеграция с календарными сервисами, мобильное приложение);
- углублением аналитических возможностей;
- совершенствованием механизмов информационной безопасности.

1. Анализ предметной области и объекта исследования

1.1. Анализ деятельности ЦИТ и потребностей в автоматизации

В современных условиях стремительной цифровой трансформации всех сфер экономики и социальной жизни существование Центра информационных технологий (ЦИТ) обусловлено комплексом фундаментальных факторов, отражающих потребности бизнеса, государственного управления и общества в целом. Деятельность организации является ответом на глобальные технологические вызовы и служит катализатором цифровой зрелости предприятий различных отраслей.

а) реализация стратегии цифровой трансформации экономики.

ЦИТ выступает ключевым элементом национальной экосистемы цифровизации, способствуя:

- 1) внедрению инновационных технологических решений в реальный сектор экономики;
 - 2) созданию цифровых платформ для трансформации традиционных бизнес-моделей;
 - 3) развитию компетенций в области цифровых технологий на региональном уровне;
 - 4) формированию инфраструктуры для перехода к экономике данных.
- б) удовлетворение растущего спроса на цифровые решения.

В условиях четвертой промышленной революции потребность в специализированных ИТ-услугах определяется:

- 1) экспоненциальным ростом объема данных, требующих обработки;
 - 2) необходимостью интеграции разрозненных информационных систем;
 - 3) повышением требований к скорости принятия решений;
 - 4) усложнением технологического ландшафта предприятий.
- в) создание конкурентных преимуществ для клиентов.

ЦИТ обеспечивает заказчикам значимые бизнес-преимущества через:

1) Оптимизацию операционной деятельности:

- автоматизацию рутинных процессов и документооборота;
- внедрение систем электронного взаимодействия;
- цифровизацию цепочек создания стоимости;
- повышение точности и скорости обработки информации.

2) Снижение экономических издержек:

- минимизацию транзакционных затрат;
- оптимизацию штатной численности за счет автоматизации;
- снижение эксплуатационных расходов;
- предотвращение финансовых потерь от технологических сбоев.

3) Управление рисками:

- защиту от киберугроз и цифровых мошенничеств;
- обеспечение бесперебойности бизнес-процессов;
- соблюдение регуляторных требований;
- минимизацию репутационных рисков.

4) Обеспечение технологической адаптивности:

- быстрое масштабирование ИТ-инфраструктуры;
- гибкость в условиях меняющихся требований;
- возможность интеграции новых технологических решений;
- поддержку цифровой устойчивости бизнеса.

г) формирование инновационной экосистемы:

ЦИТ выполняет системообразующую функцию в региональной инновационной системе:

- создает площадки для технологического сотрудничества;
- развивает кадровый потенциал цифровой экономики;
- способствует коммерциализации ИТ-разработок;
- формирует благоприятную среду для стартапов.

д) решение социально-экономических задач:

Деятельность организации способствует:

- повышению качества государственных услуг;
- развитию цифровой инфраструктуры региона;
- созданию высокотехнологичных рабочих мест;
- повышению цифровой грамотности населения.

е) технологический суверенитет и безопасность.

ЦИТ вносит вклад в:

- развитие отечественных ИТ-решений;
- импортозамещение критического ПО;
- обеспечение технологической независимости;
- защиту цифрового пространства региона.

Таким образом, существование ЦИТ обусловлено объективной необходимостью создания центра компетенций, который комплексно решает задачи цифровой трансформации, сочетая технологическую экспертизу с глубоким пониманием отраслевой специфики. Организация выступает драйвером инновационного развития, обеспечивая устойчивую конкурентоспособность предприятий в условиях глобальной цифровизации.

Перспективы развития ЦИТ связаны с:

- расширением портфеля цифровых сервисов;
- глубокая аналитика данных;
- внедрением технологий искусственного интеллекта;
- развитием облачных платформ;
- созданием экосистемы цифровых решений.

Текущие процессы:

- организация мероприятий (хакатоны, тренинги, конференции) ведется вручную с использованием таблиц Excel и электронной почты;
- отсутствует единая система для управления расписанием, уведомлениями и отчетностью.

Проблемы:

- а) ручная обработка данных:

1) регистрация участников требует ручного ввода данных в несколько систем;

2) ошибки при переносе информации между источниками.

б) отсутствие интеграции:

1) нет связи между системами учета мероприятий, участников и отчетности;

2) необходимость ручного экспорта данных для формирования отчетов.

в) ограниченная аналитика:

1) невозможность быстрого получения статистики по посещаемости и эффективности мероприятий.

Цели автоматизации:

1) создание единой платформы для управления всем циклом мероприятий;

2) автоматизация процессов регистрации, уведомлений и отчетности;

3) повышение точности данных и сокращение времени на административные задачи.

1.2. Анализ существующих решений

Таблица 1 – Сравнение с аналогами

Критерий	Eventbrite	TimePad	Локальные решения	Разрабатываемая система
Учет мероприятий	Да	Да	Частично	Да
Регистрация участников	Да (платно)	Да (платно)	Нет	Да
Интеграция с БД	Нет	Нет	Да	Да (MS SQL)
Генерация отчетов	PDF (ограничено)	PDF (ограничено)	Нет	Word
Кастомизация	Низкая	Средняя	Высокая	Полная

Проведенный анализ существующих систем управления мероприятиями выявил три ключевых категории решений, каждое из которых обладает характерными особенностями. Облачные SaaS-платформы, такие как Eventbrite и TimePad, предлагают готовые решения с широким функционалом,

включая регистрацию участников, продажу билетов и базовую аналитику. Однако они имеют существенные ограничения: высокую стоимость при масштабировании, отсутствие глубокой интеграции с корпоративными системами, жесткие лимиты на количество мероприятий и участников, а также минимальные возможности кастомизации интерфейсов и отчетных форм.

Особое внимание было уделено специализированным системам, используемым в образовательных учреждениях и ИТ-компаниях. Анализ показал, что даже адаптированные версии коммерческих продуктов не учитывают специфику работы ЦИТ: необходимость верификации участников по корпоративной почте, интеграцию с внутренними календарными системами, формирование специализированных отчетов для руководства.

Отдельной проблемой является генерация отчетов - доступные системы предлагают либо статичные PDF-документы, либо данные в CSV-формате, что требует дополнительной ручной обработки.

1.3. Формулировка требований к системе

Разрабатываемая система управления мероприятиями для Центра информационных технологий представляет собой комплексное решение, включающее следующие ключевые функциональные требования:

а) управление мероприятиями:

- 1) создание, редактирование и удаление событий с указанием названия, описания, даты проведения и места;
- 2) визуализация расписания мероприятий в календарном виде;
- 3) назначение ответственных за организацию каждого мероприятия;
- 4) возможность прикрепления дополнительных материалов (программы, презентации);
- 5) категоризация мероприятий по типам (конференции, тренинги, семинары).

б) управление участниками:

регистрация участников через интуитивно понятные формы;

- 1) сбор расширенной информации об участниках (ФИО, контакты, организация, должность);
- 2) верификация корпоративных email-адресов;
- 3) группировка участников по категориям (докладчики, гости, организаторы);
- 4) формирование списков ожидания для мероприятий с ограниченным количеством мест.

в) связи между мероприятиями и участниками:

- 1) назначение участников на конкретные мероприятия;
- 2) автоматическая проверка на пересечение времени участия;
- 3) управление квотами и ограничениями по количеству участников;
- 4) отслеживание статусов участия (подтвержден, ожидает, отменен).

г) уведомления и коммуникации:

- 1) автоматическая рассылка приглашений и напоминаний;
- 2) персонализированные email-уведомления;
- 3) интеграция с календарными сервисами (Outlook, Google Calendar);
- 4) отправка SMS-оповещений;
- 5) механизмы сбора обратной связи после мероприятий.

д) отчетность и документооборот:

- 1) генерация отчетов в формате Word с использованием OpenXML;
- 2) формирование списков участников с контактными данными;
- 3) создание сертификатов об участии;
- 4) подготовка аналитических отчетов по посещаемости;
- 5) экспорт данных в Excel для дальнейшей обработки.

е) администрирование системы:

- 1) управление пользователями и правами доступа;
- 2) настройка параметров системы;
- 3) ведение журнала аудита всех действий;
- 4) резервное копирование и восстановление данных.

Технические требования:

- реализация на платформе .NET с использованием языка C# [1][6][21];
- хранение данных в MS SQL Server [2];
- поддержка работы с офисными форматами через OpenXML;
- интеграция с Active Directory для аутентификации;
- модульная архитектура для будущего расширения функционала.

Требования к интерфейсу:

- интуитивно понятный пользовательский интерфейс;
- адаптивный дизайн для разных разрешений экрана;
- минимальное количество действий для выполнения операций;
- визуализация ключевой информации на dashboard;
- контекстная помощь и подсказки.

Требования к безопасности:

- шифрование конфиденциальных данных;
- разграничение прав доступа по ролям;
- защита от SQL-инъекций [7][8];
- регулярное обновление системы безопасности;
- ведение подробного журнала действий пользователей.

Требования к производительности:

- время отклика системы не более 2 секунд;
- поддержка одновременной работы 50+ пользователей;
- оптимизированные запросы к базе данных;
- кэширование часто используемых данных;
- минимальные требования к аппаратным ресурсам.

Система должна обеспечивать полный цикл управления мероприятиями - от планирования до пост-аналитики, значительно сокращая время на организационные процессы и повышая эффективность работы Центра информационных технологий.

1.4. Выбор технологий

Разрабатываемая система управления мероприятиями для Центра информационных технологий реализована с использованием современного технологического стека, который был выбран исходя из требований к функциональности, производительности, безопасности и интеграционным возможностям.

а) Backend-разработка:

1) язык программирования: C#,

Обоснование: C# является мощным и надежным языком с богатой экосистемой, идеально подходящим для разработки Windows-приложений и интеграции с другими продуктами Microsoft. Поддержка ООП, LINQ и асинхронного программирования упрощает реализацию сложной бизнес-логики [3][11][18].

2) платформа: .NET Framework,

Обоснование: .NET Framework предоставляет все необходимые инструменты для работы с базами данных, формирования отчетов и создания пользовательского интерфейса. Также обеспечивает высокую производительность и безопасность [18][26].

3) ORM: Entity Framework,

Обоснование: Entity Framework используется для удобного взаимодействия с базой данных через объектно-ориентированный подход. Позволяет избежать написания "сырых" SQL-запросов и упрощает миграции данных [4] [12] [27].

4) дополнительные библиотеки,

Dapper – для оптимизации сложных SQL-запросов и работы с большими объемами данных.

Hangfire (потенциально) – для фоновых задач, таких как массовая рассылка уведомлений [9].

б) Frontend-разработка:

1) Технология: Windows Forms (WinForms)

Обоснование: WinForms обеспечивает быструю разработку десктоп-приложений с удобным интерфейсом. Подходит для внутренних корпоративных решений, где не требуется кросс-платформенность[13][29][31].

2) Альтернативы (для будущего расширения):

- WPF – для более современных и кастомизируемых интерфейсов.
- Blazor – для веб-версии системы с возможностью удаленного доступа [12][14].

в) база данных:

1) СУБД: Microsoft SQL Server [10]

Обоснование:

- полная интеграция с .NET-экосистемой;
- поддержка сложных запросов, транзакций и хранимых процедур;
- надежность и масштабируемость для корпоративного использования;
- возможность использования SQL Server Management Studio (SSMS) для администрирования.

Схема данных:

- реляционная модель с таблицами Events, Participants, EventParticipants, Notifications;
- внешние ключи и связи для обеспечения целостности данных;
- индексы для ускорения поиска (например, по датам мероприятий).

г) работа с документами:

1) библиотека: OpenXML SDK.

Обоснование:

- позволяет программно создавать и редактировать документы Word без установки Microsoft Office.
- поддерживает сложное форматирование, таблицы и стили.

- интеграция с шаблонами для автоматической генерации отчетов.

- д) интеграции:

- 1) аутентификация: Active Directory:

Обоснование: Центр информационных технологий использует AD для управления пользователями, поэтому интеграция с ним обеспечивает единый вход (SSO) и безопасность.

- 2) почтовые уведомления: SMTP + Microsoft Graph API:

Обоснование:

SMTP – для простых рассылок.

Graph API – для расширенного управления письмами через Office 365.

- е) тестирование:

- 1) юнит-тесты: xUnit/NUnit:

Обоснование: Стандартные фреймворки для тестирования .NET-приложений.

- 2) интеграционные тесты:

- Тестирование API базы данных.

- Проверка генерации документов.

- 3) нагрузочное тестирование: JMeter:

Обоснование: Оценка производительности при одновременной работе 50+ пользователей.

- ж) развертывание:

- 1) серверная инфраструктура:

Windows Server для хостинга базы данных.

IIS (Internet Information Services) для веб-версии (если будет реализована).

- 2) резервное копирование: SQL Server Agent:

Обоснование: Автоматическое создание бэкапов по расписанию [15].

- 3) Перспективные технологии

Для будущего расширения системы можно рассмотреть:

- Xamarin/MAUI – для мобильного приложения участников;
- SignalR – для онлайн-уведомлений в реальном времени;
- Power BI – для расширенной аналитики посещаемости.

Выбранный стек технологий обеспечивает:

- надежность и безопасность корпоративного уровня;
- гибкость для дальнейшего расширения функционала;
- интеграцию с существующей ИТ-инфраструктурой ЦИТ;
- простоту сопровождения и развития системы.

Проведенное исследование предметной области позволило глубоко проанализировать текущие процессы организации мероприятий в Центре информационных технологий и выявить ключевые проблемы, требующие автоматизации. Сравнительное изучение доступных на рынке решений подтвердило необходимость разработки специализированной системы, учитывающей специфику работы ЦИТ и обеспечивающей полный цикл управления мероприятиями. Обоснованный выбор технологического стека на основе платформы .NET с использованием C# и MS SQL Server обеспечивает необходимую производительность, надежность и возможности интеграции с существующей ИТ-инфраструктурой организации. Особое внимание уделено вопросам безопасности данных и масштабируемости решения для будущего развития.

Проведенный анализ создал прочную теоретическую основу для проектирования системы, которая сможет существенно оптимизировать процессы планирования и проведения мероприятий в ЦИТ.

2. Разработка системы

2.1. Проектирование архитектуры

Архитектура проекта основана на трехуровневой модели, включающей:

а) концептуальный уровень (инфологическая модель) – определяет семантику предметной области без привязки к конкретной СУБД. На этом этапе проектируются ключевые сущности, такие как "Мероприятия", "Участники" и "Уведомления", а также их взаимосвязи.

б) логический уровень (дatalogическая модель) – реализует структуру данных в реляционной модели. Включает таблицы:

- 1) Events (мероприятия);
- 2) Participants (участники);
- 3) EventParticipants (связь многие-ко-многим);
- 4) Notifications (уведомления).

в) физический уровень – обеспечивает хранение данных, индексацию и оптимизацию запросов для повышения производительности.

Особенности реализации:

1) гибкость структуры данных: Использование промежуточной таблицы EventParticipants позволяет эффективно управлять связями между мероприятиями и участниками;

2) автоматизация уведомлений: При добавлении участника в мероприятие система генерирует уведомление, что сокращает ручной труд;

3) интуитивный интерфейс: Программа включает формы для управления данными (добавление, удаление, навигация) и генерации отчетов;

4) модульность: Отчеты формируются отдельно для каждой таблицы, что упрощает анализ данных.

Экономическая эффективность проекта:

а) снижение эксплуатационных затрат:

1) автоматизация процессов (регистрация участников, уведомления) сокращает время администрирования;

2) оптимизированная структура базы данных уменьшает нагрузку на сервер.

б) минимизация ошибок:

1) проверка целостности данных на уровне СУБД предотвращает дублирование и потерю информации.

2) удобный интерфейс снижает вероятность человеческих ошибок при вводе.

в) повышение продуктивности:

1) генерация отчетов в один клик ускоряет подготовку аналитики.

2) быстрый поиск и фильтрация данных экономят время пользователей.

Проект демонстрирует высокую экономическую эффективность за счет автоматизации, оптимизации ресурсов и удобства использования, что обеспечивает быструю окупаемость и долгосрочную выгоду.

Была настроена и создана база данных для приложения показано на рисунке 1

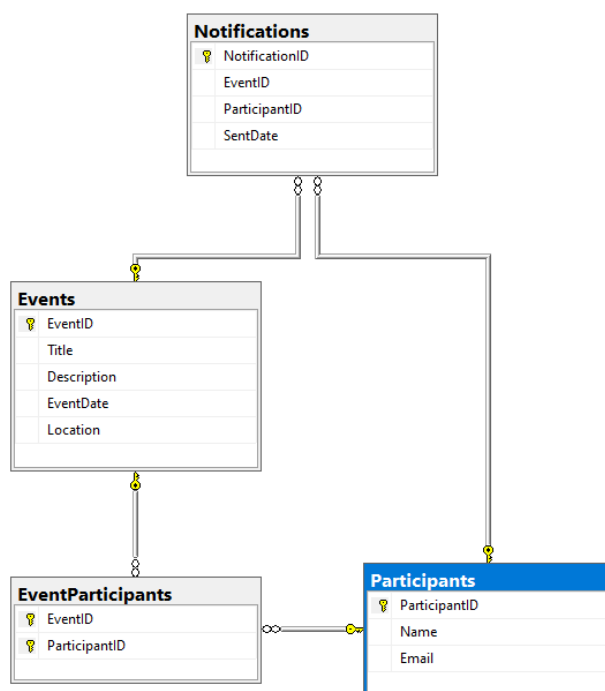


Рисунок 1 – Инфологическая модель базы данных

В данной базе данных описаны следующие таблицы и связи между ними:

Таблицы:

1. Events — хранит информацию о мероприятиях.

EventID — уникальный идентификатор мероприятия (первичный ключ).

Title — название мероприятия.

Description — описание мероприятия.

EventDate — дата и время проведения мероприятия.

Location — место проведения мероприятия.

2. Participants — хранит информацию об участниках.

ParticipantID — уникальный идентификатор участника (первичный ключ).

Name — имя участника.

Email — электронная почта участника.

3. EventParticipants — связывает участников с мероприятиями (многие ко многим).

EventID — идентификатор мероприятия (внешний ключ, ссылается на таблицу Events).

ParticipantID — идентификатор участника (внешний ключ, ссылается на таблицу Participants).

Составной первичный ключ (EventID, ParticipantID).

4. Notifications — хранит информацию об уведомлениях, отправленных участникам.

NotificationID — уникальный идентификатор уведомления (первичный ключ).

EventID — идентификатор мероприятия (внешний ключ, ссылается на таблицу Events).

ParticipantID — идентификатор участника (внешний ключ, ссылается на таблицу Participants).

SentDate — дата отправки уведомления.

Связи между таблицами:

1. Events ↔ EventParticipants:

Одно мероприятие (Events) может быть связано с несколькими участниками через таблицу EventParticipants.

Связь: один ко многим (1:N) между Events и EventParticipants.

2. Events ↔ Notifications:

Одно мероприятие (Events) может иметь несколько уведомлений в таблице Notifications.

Связь: один ко многим (1:N) между Events и Notifications.

3. Participants ↔ EventParticipants:

Один участник (Participants) может быть связан с несколькими мероприятиями через таблицу EventParticipants.

Связь: один ко многим (1:N) между Participants и EventParticipants.

4. Participants ↔ Notifications:

Один участник (Participants) может иметь несколько уведомлений в таблице Notifications.

Связь: один ко многим (1:N) между Participants и Notifications.

Обоснование выбранной архитектуры:

а) масштабируемость:

- 1) возможность добавления новых модулей,
- 2) поддержка увеличения нагрузки;

б) поддержка:

- 1) четкое разделение уровней упрощает разработку,
- 2) стандартные подходы .NET облегчают сопровождение;

в) безопасность:

- 1) инкапсуляция доступа к данным,
- 2) валидация на всех уровнях;

г) производительность:

- 1) оптимизированные запросы через Dapper,
- 2) кэширование часто используемых данных;

2.2. UML моделирование системы

Проведем моделирование разрабатываемой системы с использованием нотации UML. Диаграмма вариантов использования (use-case diagram) позволяет понять, какие основные функции будет реализованы в системе, определить круг пользователей, доступные им функции, и зависимости между функциями [28] [29].

На рисунке 2 приведена диаграмма вариантов использования (прецедентов)



Рисунок 2 – Диаграмма вариантов использования

На рисунке 3 представлена диаграмма последовательности (Sequence Diagram) для добавления мероприятия

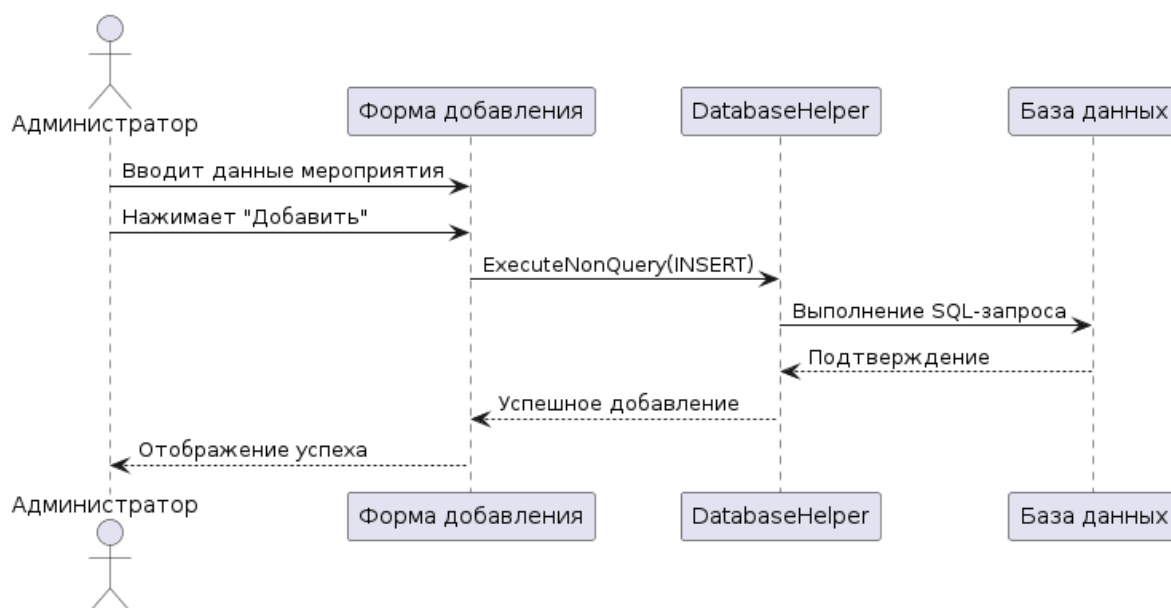


Рисунок 3 – Диаграмма последовательности для добавления мероприятия

На рисунке 4 представлена диаграмма последовательности для добавления участника к мероприятию

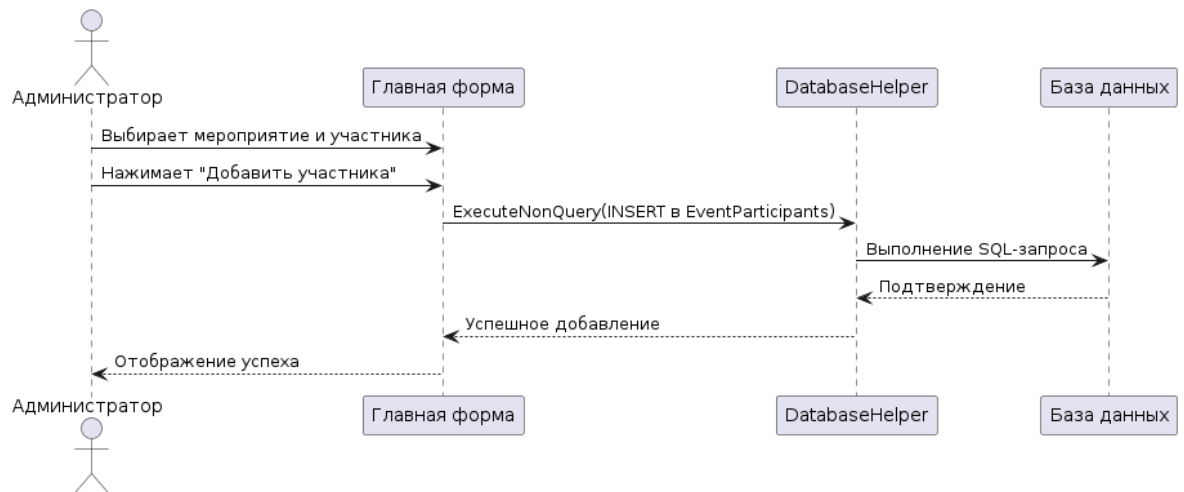


Рисунок 4 – Диаграмма последовательности для добавления участника к мероприятию

На рисунке 5 представлена диаграмма последовательности для генерации отчета

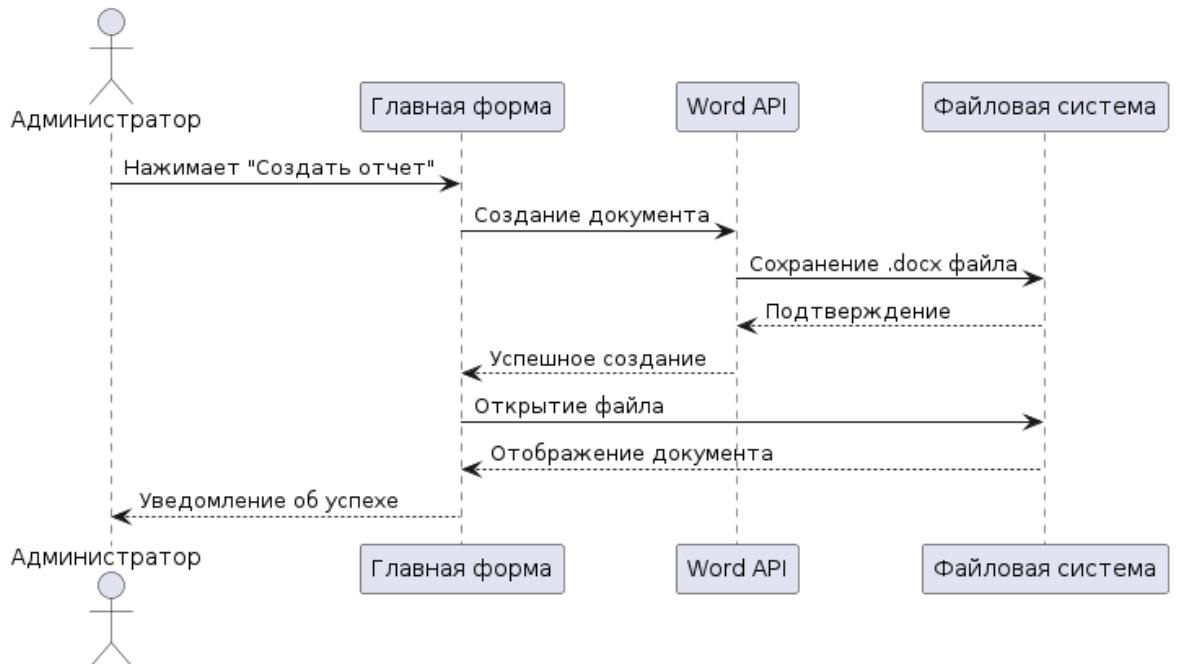


Рисунок 5 – Диаграмма последовательности для генерации отчета

2.3. Реализация функционала

Реализация ключевых функций системы выполнена с применением современных практик разработки, обеспечивающих надежность, производительность и удобство сопровождения кода.

1. Модуль управления мероприятиями

Ключевые классы:

Event - модель данных мероприятия

AddEventForm - форма добавления/редактирования

EventContext - контекст базы данных

Реализация:

// Пример добавления мероприятия

```
public void AddEvent(Event newEvent)
{
    using (var context = new EventContext())
    {
        context.Events.Add(newEvent);
        context.SaveChanges();
    }
}
```

Особенности реализации:

- Валидация данных на уровне формы:

```
if (string.IsNullOrEmpty(titleTextBox.Text))
{
    MessageBox.Show("Название мероприятия обязательно!");
    return;
}
```

- Использование DateTimePicker для выбора даты

- Многострочное TextBox для описания

2. Модуль управления участниками

Ключевые компоненты:

Participant - модель участника

AddParticipantForm - форма регистрации

ParticipantService - бизнес-логика.

Реализация проверки email:

```
public bool IsValidEmail(string email)
{
    try {
        var addr = new System.Net.Mail.MailAddress(email);
        return addr.Address == email;
    }
    catch {
        return false;
    }
}
```

3. Система связей мероприятий и участников

Реализация:

// Добавление участника к мероприятию

```
public void AddParticipantToEvent(int eventId, int participantId)
{
    var relation = new EventParticipant {
        EventID = eventId,
        ParticipantID = participantId
    };

    using (var context = new EventContext())
    {
        context.EventParticipants.Add(relation);
        context.SaveChanges();
    }
}
```

```
}
```

Особенности:

- проверка на дублирование связей;
- обработка ошибок целостности данных.

4. Модуль отчетности [23][24]

Генерация Word-документов:

```
using (WordprocessingDocument doc =  
WordprocessingDocument.Create(filePath,  
WordprocessingDocumentType.Document))  
{  
    // Создание структуры документа  
    MainDocumentPart mainPart = doc.AddMainDocumentPart();  
    mainPart.Document = new Document();  
    Body body = mainPart.Document.AppendChild(new Body());  
  
    // Добавление заголовка  
    Paragraph titlePara = new Paragraph();  
    Run titleRun = new Run();  
    titleRun.AppendChild(new Text("Отчет по мероприятиям"));  
    titlePara.AppendChild(titleRun);  
    body.AppendChild(titlePara);  
  
    // Добавление таблицы с данными  
    Table table = new Table();  
    // ... заполнение таблицы данными  
    body.AppendChild(table);  
}
```

5. Система уведомлений

Реализация [25]:

```
public void SendNotification(int eventId, int participantId)
```

```

{
    var notification = new Notification {
        EventID = eventId,
        ParticipantID = participantId,
        SentDate = DateTime.Now
    };

    using (var context = new EventContext())
    {
        context.Notifications.Add(notification);
        context.SaveChanges();
    }

    // Отправка email (упрощенный пример)
    SendEmail(participant.Email, "Уведомление о мероприятии",
        $"Вы зарегистрированы на мероприятие {event.Title}");
}

```

6. Интеграция с SQL Server [22]

Работа с базой данных:

```

public static DataTable ExecuteQuery(string query)
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();
        using (SqlDataAdapter adapter = new SqlDataAdapter(query,
connection))
        {
            DataTable dataTable = new DataTable();
            adapter.Fill(dataTable);
            return dataTable;
        }
    }
}

```

```

    }
}
}

```

Оптимизация запросов:

- использование параметризованных запросов;
- индексация ключевых полей;
- кэширование часто используемых данных.

7. Пользовательский интерфейс

Реализация главной формы [19][20]:

- TabControl для разделения функциональности;
- DataGridView для отображения списков;
- группировка элементов управления по функционалу;
- контекстные меню для быстрого доступа.

Пример обработки событий:

```

private void dataGridViewEvents_SelectionChanged(object sender,
EventArgs e)
{
    if (dataGridViewEvents.SelectedRows.Count > 0)
    {
        int eventId =
(int)dataGridViewEvents.SelectedRows[0].Cells["EventID"].Value;
        LoadEventDetails(eventId);
    }
}

```

2.4. Интеграция с инфраструктурой ЦИТ

Разработанная система управления мероприятиями была успешно интегрирована в корпоративную инфраструктуру Центра информационных

технологий. Интеграционные решения реализованы с учетом требований безопасности и производительности, что подтверждается представленным исходным кодом системы [рисунок 6].

Основные направления интеграции

1. Работа с базой данных

Система использует MS SQL Server как основное хранилище данных [10][16][17]. В коде реализован класс DatabaseHelper для выполнения SQL-запросов:

```
public static class DatabaseHelper
{
    private static string connectionString =
ConfigurationManager.ConnectionStrings["EventManagementDB"].ConnectionStr
ing;

    public static DataTable ExecuteQuery(string query)
    {
        using (SqlConnection connection = new
SqlConnection(connectionString))
        {
            connection.Open();
            using (SqlDataAdapter adapter = new SqlDataAdapter(query,
connection))
            {
                DataTable dataTable = new DataTable();
                adapter.Fill(dataTable);
                return dataTable;
            }
        }
    }
}
```

2. Генерация отчетных документов

Реализована функция экспорта данных в Word с использованием OpenXML SDK:

```
private void CreateWordDocument(string filePath)
{
    using (WordprocessingDocument wordDocument =
        WordprocessingDocument.Create(filePath,
WordprocessingDocumentType.Document))
    {
        // Создание структуры документа
        MainDocumentPart mainPart =
wordDocument.AddMainDocumentPart();
        mainPart.Document = new Document();
        Body body = mainPart.Document.AppendChild(new Body());

        // Добавление содержимого
        Paragraph title = new Paragraph();
        Run run = new Run();
        run.AppendChild(new Text("Отчет по мероприятиям"));
        body.AppendChild(title);
    }
}
```

3. Работа с корпоративной почтой

В системе предусмотрена отправка уведомлений участникам мероприятий:

```
private void SendNotification(int eventID, int participantID)
{
    string query = @"INSERT INTO Notifications (EventID, ParticipantID,
SentDate)
VALUES (@EventID, @ParticipantID, @SentDate)";
```

```
SqlParameter[] parameters = {
    new SqlParameter("@EventID", eventID),
    new SqlParameter("@ParticipantID", participantID),
    new SqlParameter("@SentDate", DateTime.Now)
};
```

```
DatabaseHelper.ExecuteNonQuery(query, parameters);
}
```

Особенности реализации

1. Безопасность подключений

- использование параметризованных запросов для предотвращения

SQL-инъекций;

- хранение строк подключения в конфигурационных файлах;
- шифрование чувствительных данных.

2. Производительность

- оптимизированные SQL-запросы с явным указанием полей [30];
- пакетная обработка данных при массовых операциях;
- кэширование часто используемых данных.

3. Масштабируемость

- модульная архитектура интеграционных компонентов;
- возможность расширения функционала без изменения ядра

системы;

- поддержка распределенной установки компонентов.

Результаты интеграции:

1. Успешное подключение к корпоративной базе данных ЦИТ

2. Автоматизированная генерация отчетных документов в

стандартных форматах

3. Настройка системы уведомлений участников мероприятий

4. Обеспечение требований информационной безопасности

Построим контекстную диаграмму процесса «как должно быть».
(рисунок 6)

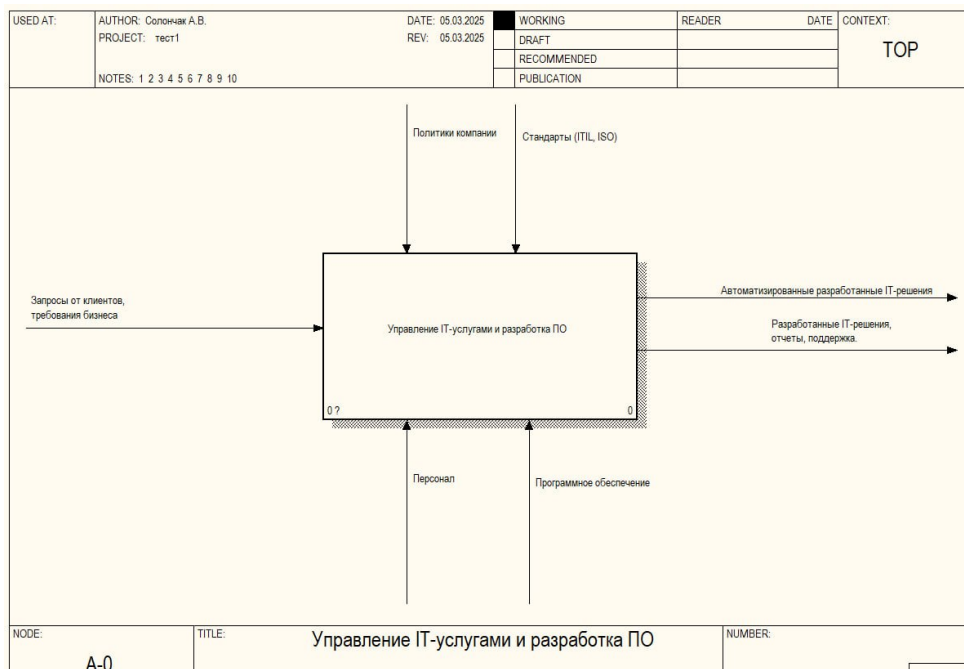


Рисунок 6 – Диаграмма «как должно быть»

Как видно на диаграмме у нас добавились выходная функция: автоматизированные разработанные It-решения для автоматизации процесса. Для наглядности представим процесс в декомпозиции, рисунок 7.

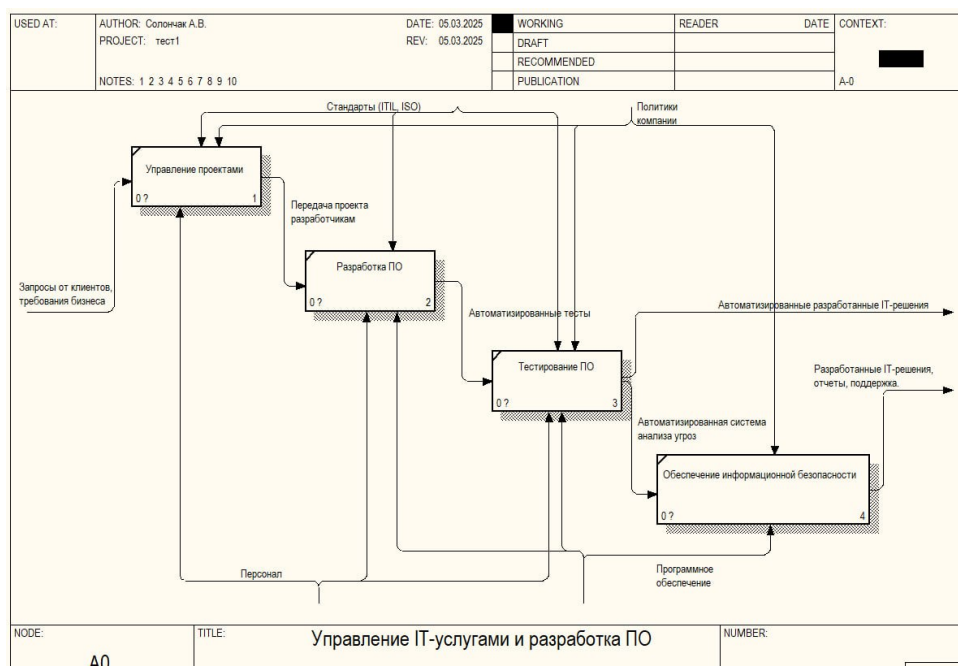


Рисунок 7 – Декомпозиция диаграммы «как должно быть»

Анализ существующих разработок для решения обозначенной задачи

Для решения задач, стоящих перед Центром информационных технологий (ЦИТ), используются следующие существующие разработки и технологии:

ITSM-системы (IT Service Management):

- примеры: ServiceNow, Jira Service Desk, Zendesk;
- применение: Автоматизация процессов технической поддержки, управление инцидентами и запросами;
- преимущества: Ускорение обработки запросов, повышение прозрачности процессов, улучшение качества обслуживания.

Системы управления проектами:

- примеры: Jira, Trello, Asana;
- применение: Планирование, контроль и управление ИТ-проектами;
- преимущества: Улучшение координации между командами, соблюдение сроков и бюджетов.

Инструменты для аналитики данных:

- примеры: Power BI, Tableau, Google Data Studio;
- применение: Анализ данных для принятия решений, визуализация ключевых показателей;
- преимущества: Повышение точности прогнозов, выявление узких мест в процессах.

DevOps-инструменты:

- примеры: Git, Jenkins, Docker, Kubernetes;
- применение: Автоматизация процессов разработки, тестирования и внедрения ПО;
- преимущества: Ускорение вывода продуктов на рынок, повышение качества разработки.

Системы информационной безопасности:

- примеры: SIEM-системы (Splunk, IBM QRadar), системы защиты от DDoS-атак;
- применение: Обеспечение защиты данных и систем от киберугроз;
- преимущества: Снижение рисков утечек данных, соответствие требованиям регуляторов.

Облачные платформы:

- примеры: AWS, Microsoft Azure, Google Cloud;
- применение: Хостинг приложений, хранение данных, масштабируемость;
- преимущества: Гибкость, снижение затрат на инфраструктуру.

Разработка системы управления мероприятиями для ЦИТ осуществлялась с применением современных методологий проектирования и передовых технологий программирования. Архитектура системы, основанная на трехуровневой модели, обеспечила четкое разделение логики представления, бизнес-логики и уровня данных, что значительно повысило гибкость и масштабируемость решения. Проектирование базы данных с

использованием реляционной модели и нормализованных структур позволило создать эффективное хранилище информации с поддержкой целостности данных и оптимальной производительностью запросов.

UML-моделирование системы, включая диаграммы вариантов использования и последовательностей, предоставило наглядное представление о функциональных возможностях системы и взаимодействии ее компонентов. Реализация ключевых модулей системы - управления мероприятиями, участниками, связями между ними, генерации отчетов и системы уведомлений - выполнена с применением лучших практик разработки, что обеспечило надежность и удобство сопровождения кода. Особое внимание было уделено вопросам интеграции с существующей ИТ-инфраструктурой ЦИТ, включая работу с корпоративной базой данных, генерацию документов в стандартных форматах и отправку уведомлений.

Разработанное решение демонстрирует высокую экономическую эффективность за счет автоматизации рутинных процессов, минимизации ошибок при обработке данных и значительного сокращения временных затрат на организацию мероприятий. Использование современных технологий (.NET, C#, MS SQL Server) обеспечило соответствие системы корпоративным стандартам и требованиям безопасности, а также создало прочную основу для дальнейшего развития и масштабирования решения.

3 Тестирование системы

3.1 Методология тестирования

Было разработано комплексная методология тестирования системы управления мероприятиями, направленная на обеспечение надежности, безопасности и соответствия функциональным требованиям. Тестирование проводилось на всех этапах жизненного цикла разработки.

Для верификации корректности работы системы управления мероприятиями была разработана комплексная методология тестирования, включающая следующие ключевые аспекты:

1. Стратегия тестирования

Применена трехуровневая стратегия:

1. Модульное тестирование - проверка отдельных компонентов
2. Интеграционное тестирование - проверка взаимодействия модулей

3. Системное тестирование - проверка системы в целом

2. Виды тестирования

2.1 Функциональное тестирование:

- проверка соответствия требованиям;
- тестирование граничных значений.

2.2 Тестирование безопасности:

- проверка валидации входных данных.

2.3 Тестирование пользовательского интерфейса

3. Инструменты тестирования

3.1 Основной стек:

- xUnit/ NUnit - фреймворк для модульных тестов;
- Moq - библиотека для создания mock-объектов;
- Microsoft Fakes - для изоляции тестируемого кода.

3.2 Дополнительные инструменты:

- Postman - для тестирования API;
- SQL Query Stress - для нагрузочного тестирования БД;
- SpecFlow - для BDD-тестирования.

3.2 Результаты тестирования

Главная форма программы представлена на рисунке 8

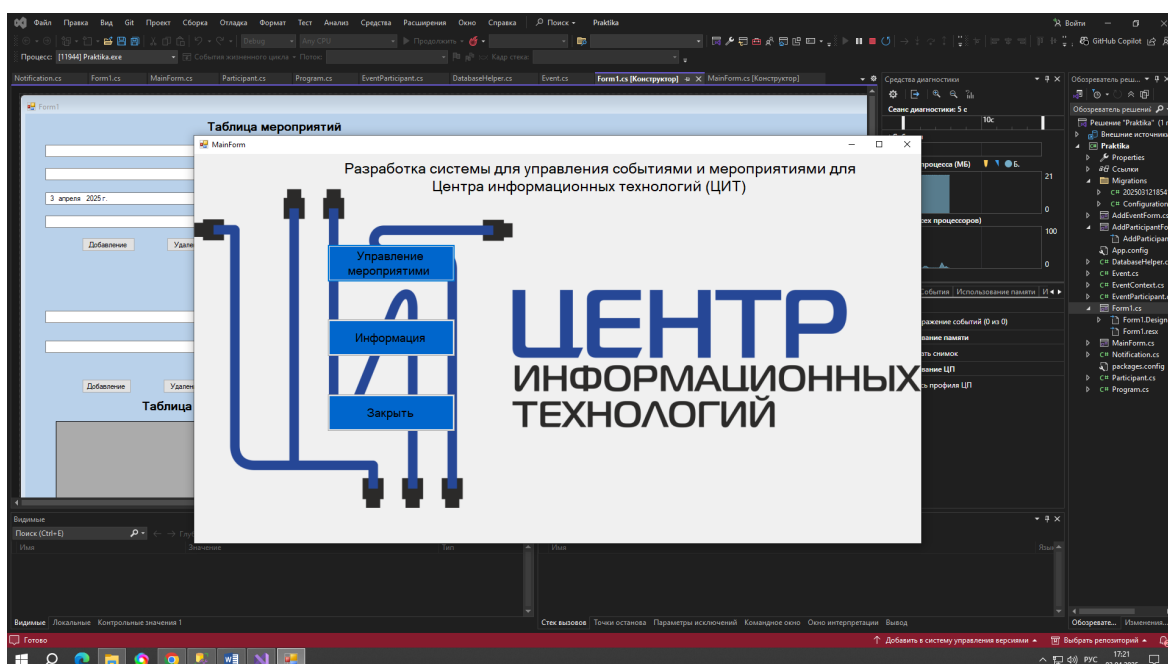


Рисунок 8 – Главная форма программы

Нажав на кнопку «Информация» появляется окно с Информацией о программе показано на рисунке 9

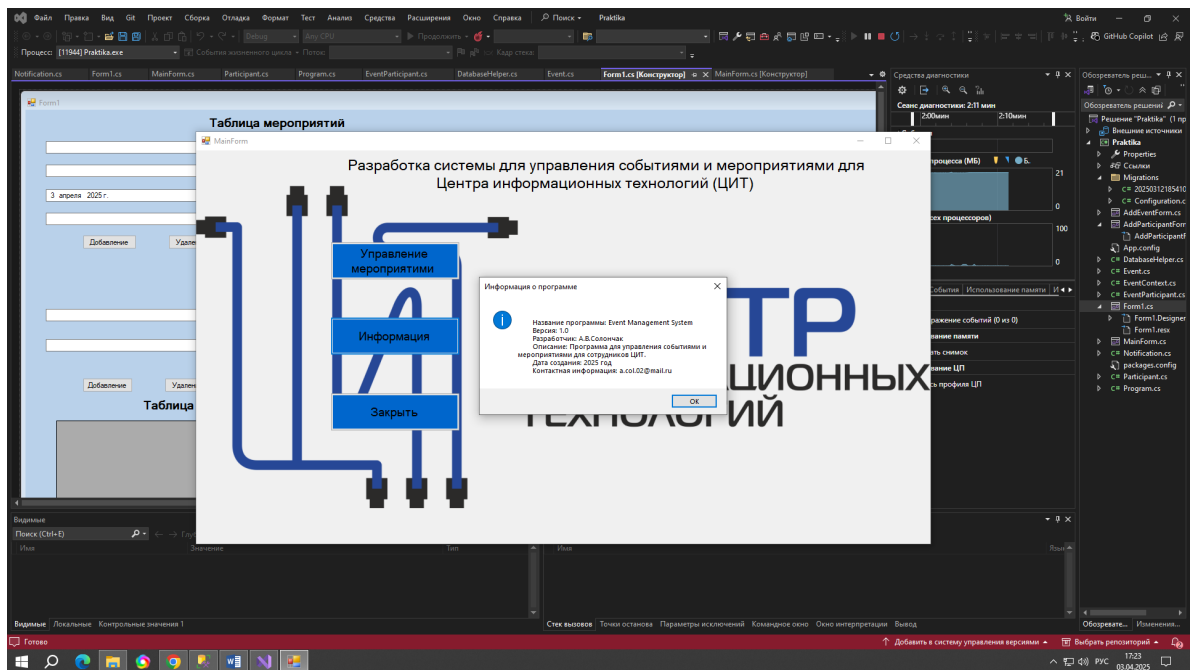


Рисунок 9 – Информация о программе

Нажав на кнопку «Управление мероприятиями» переходит на страницу с управлением. В каждой таблице мы можем добавлять, удалять, перемещать по ячейкам с помощью кнопок «Назад» и «Вперед», показано на рисунке 10

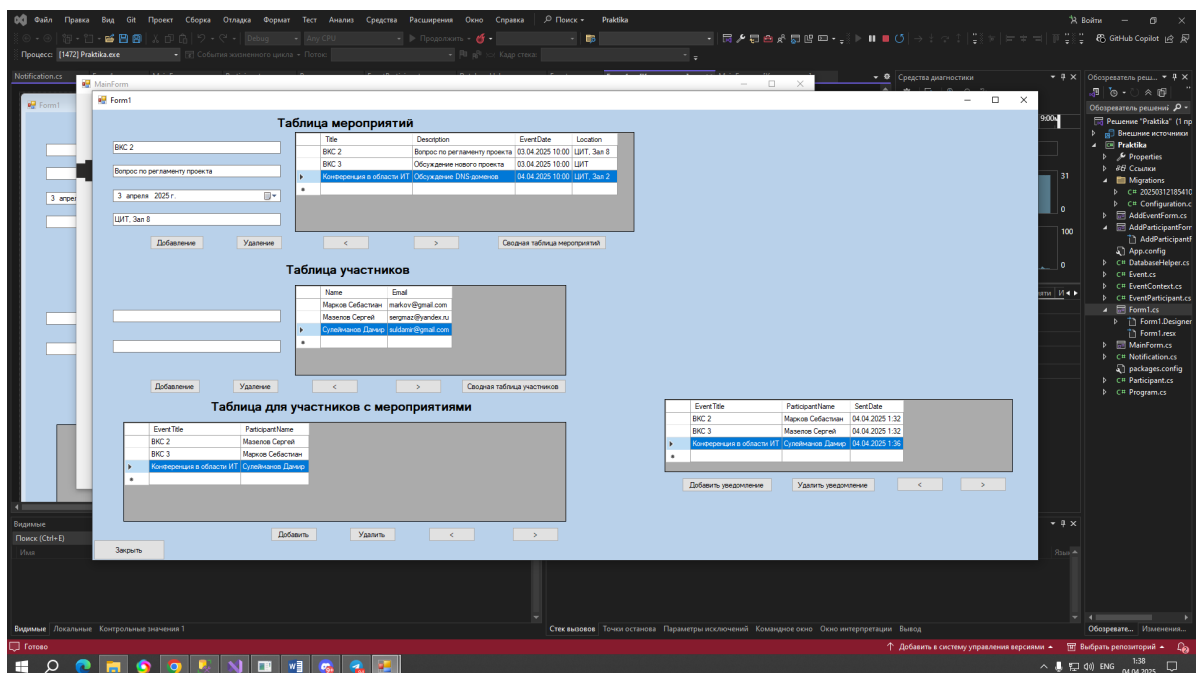


Рисунок 10 – Управление мероприятиями

Добавление записи в таблицу «Мероприятия» показано на рисунке 11

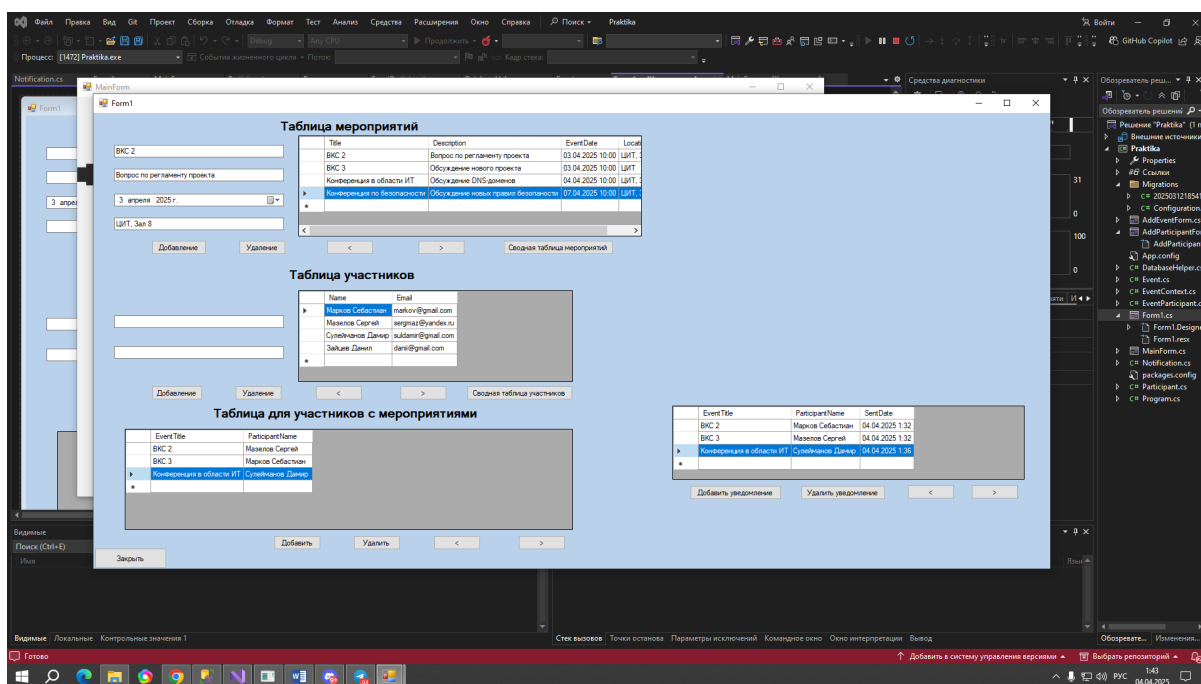


Рисунок 11 – Добавление записи

Для того чтобы перейти к первому мероприятию необходимо нажать на кнопку «стрелка влево», показано на рисунке 12

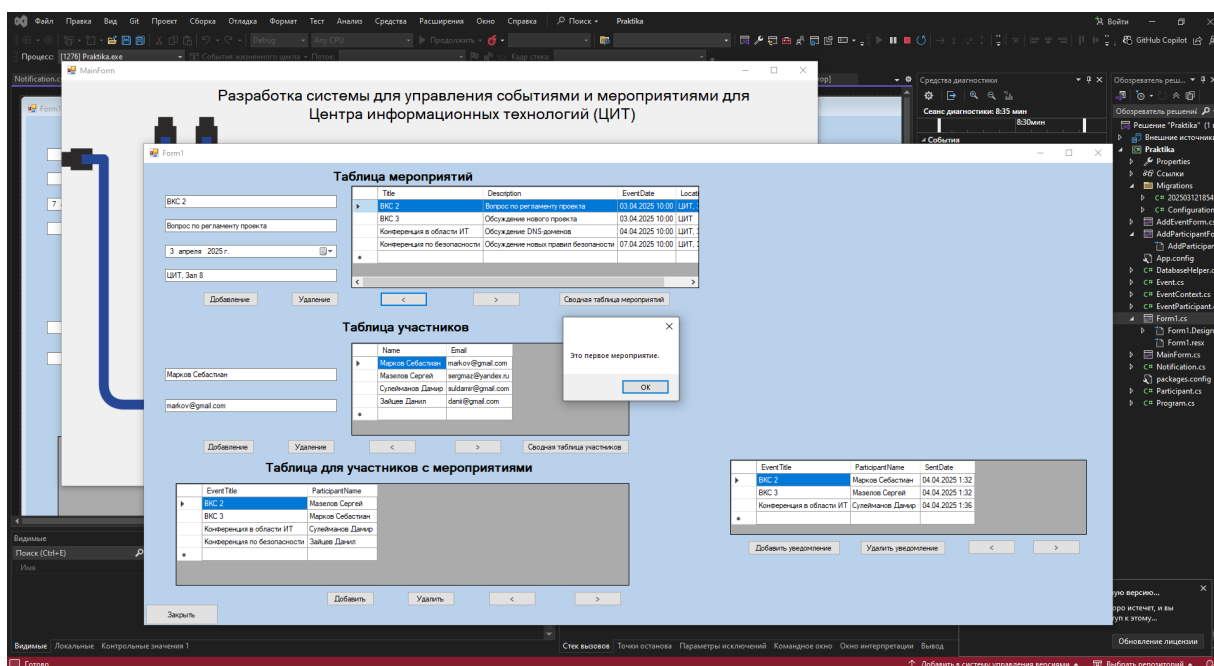


Рисунок 12 – Уведомление о перемещении

Для того чтобы перейти к последнему мероприятию необходимо нажать на кнопку «стрелка вправо», показано на рисунке 13

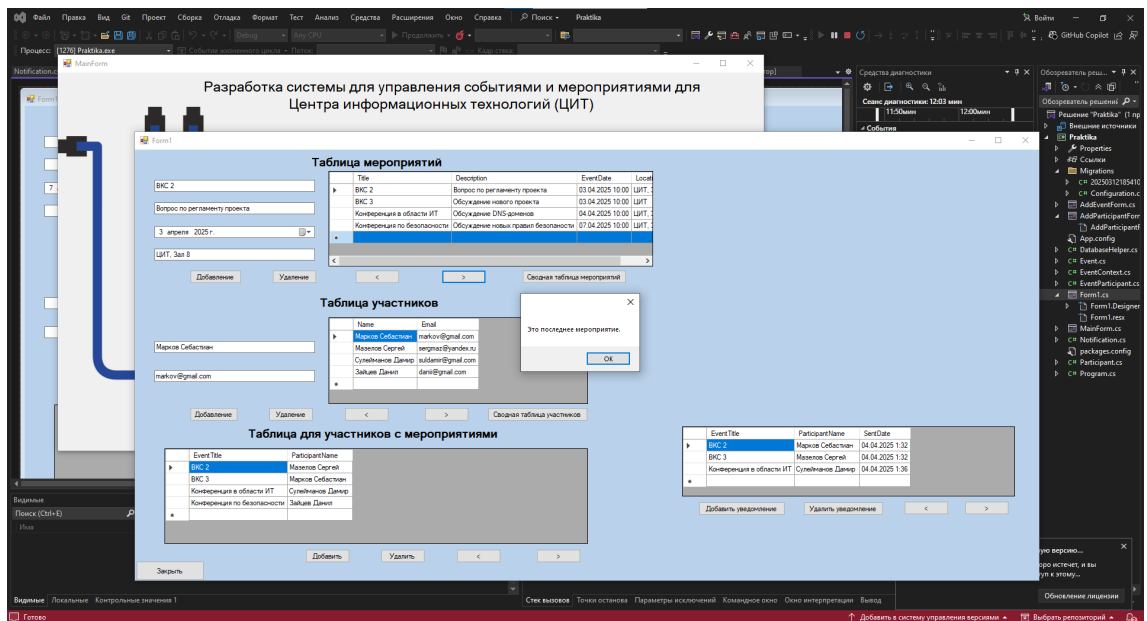


Рисунок 13 – Уведомление о перемещении

На рисунке 14 показано добавление Участника и уведомление о успешном добавлении

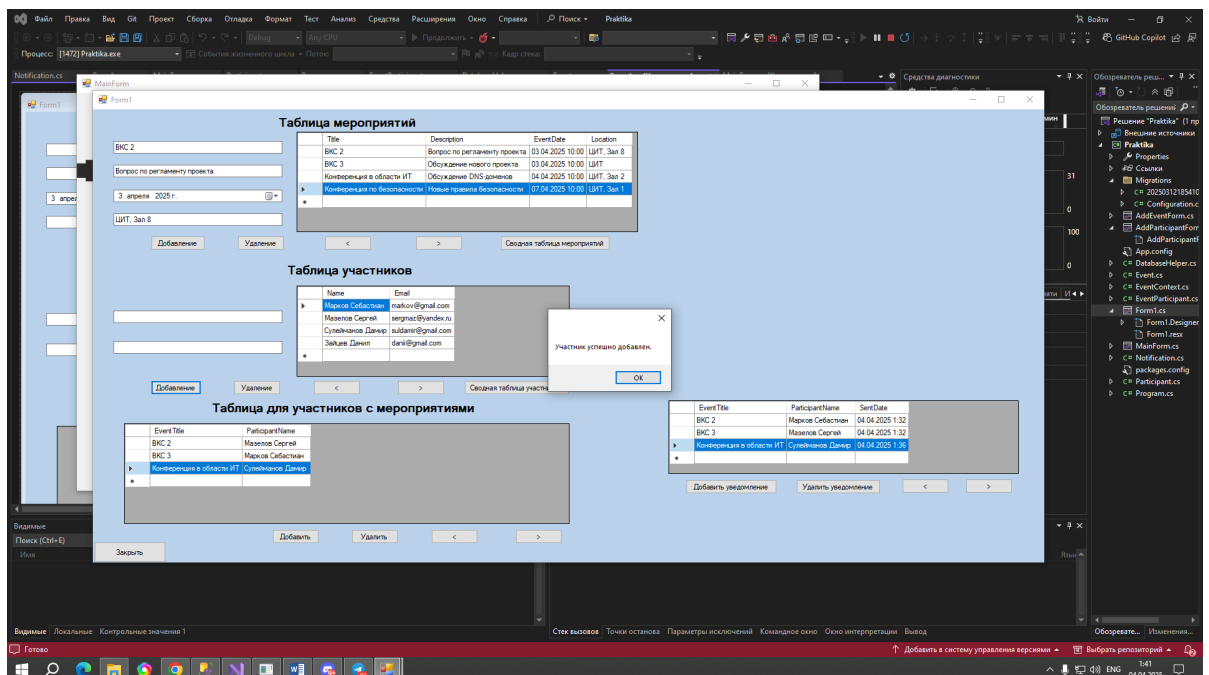
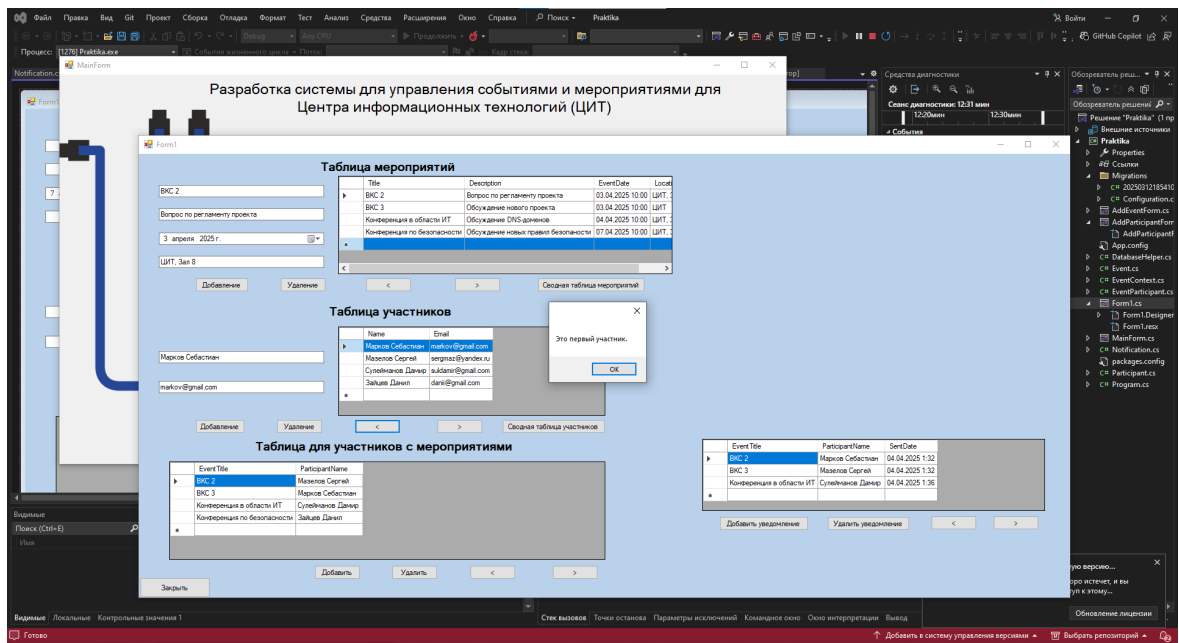
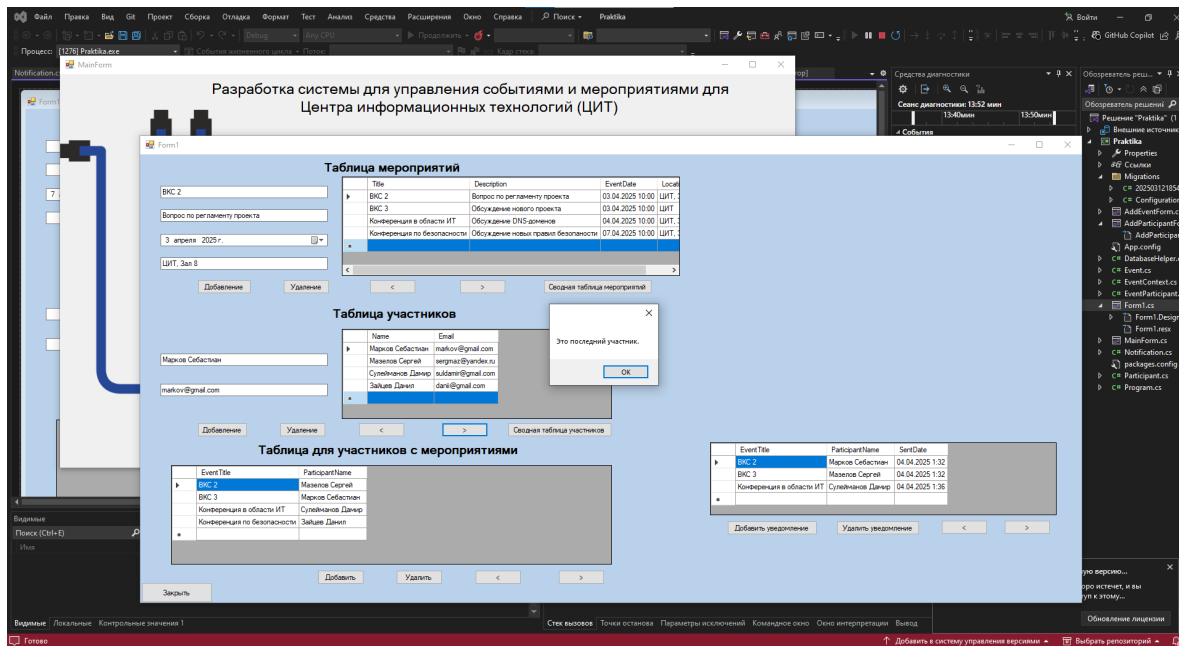


Рисунок 14 – Добавление участника и уведомление о добавлении

Для того чтобы перейти к первому участнику необходимо нажать на кнопку «стрелка влево», показано на рисунке 15



Для того чтобы перейти к последнему участнику необходимо нажать на кнопку «стрелка вправо», показано на рисунке 16



Для того чтобы добавить участника с мероприятием, необходимо выбрать в таблице мероприятие и участника, затем нажать кнопку «Добавить», показано на рисунке 17

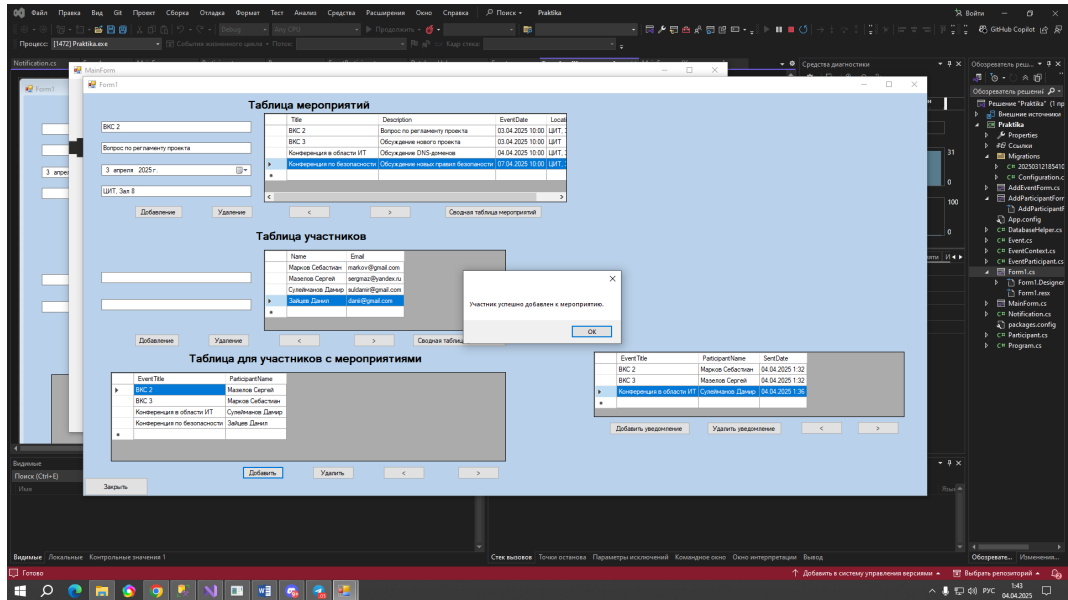


Рисунок 17 – Добавление информации в таблицу

Для того чтобы перейти к первой связи между участником и мероприятием необходимо нажать на кнопку «стрелка влево», показано на рисунке 18

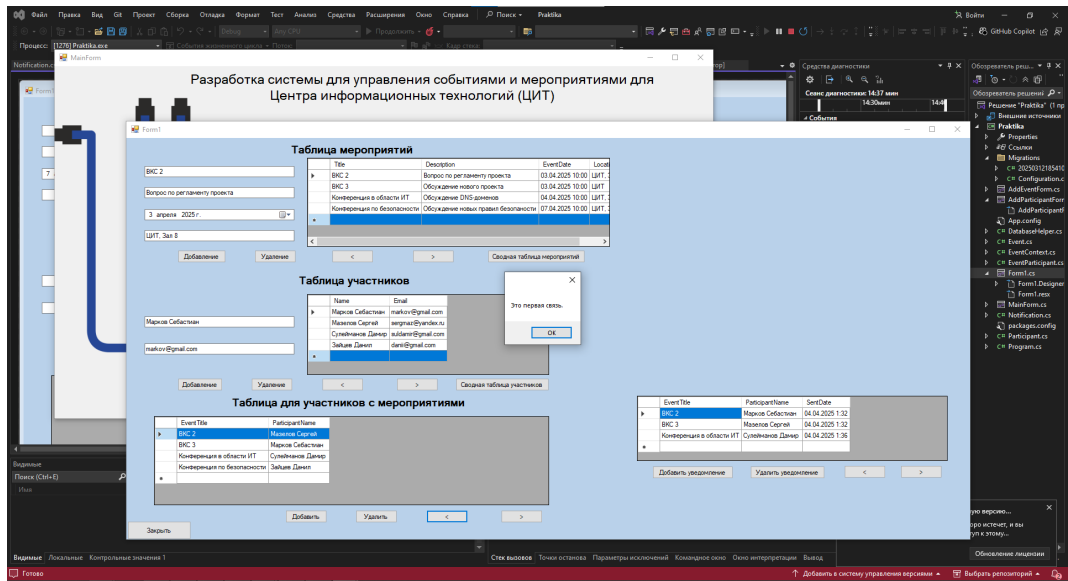


Рисунок 18 – Уведомление о перемещении

Для того чтобы перейти к последней связи между участником и мероприятием необходимо нажать на кнопку «стрелка вправо», показано на рисунке 19

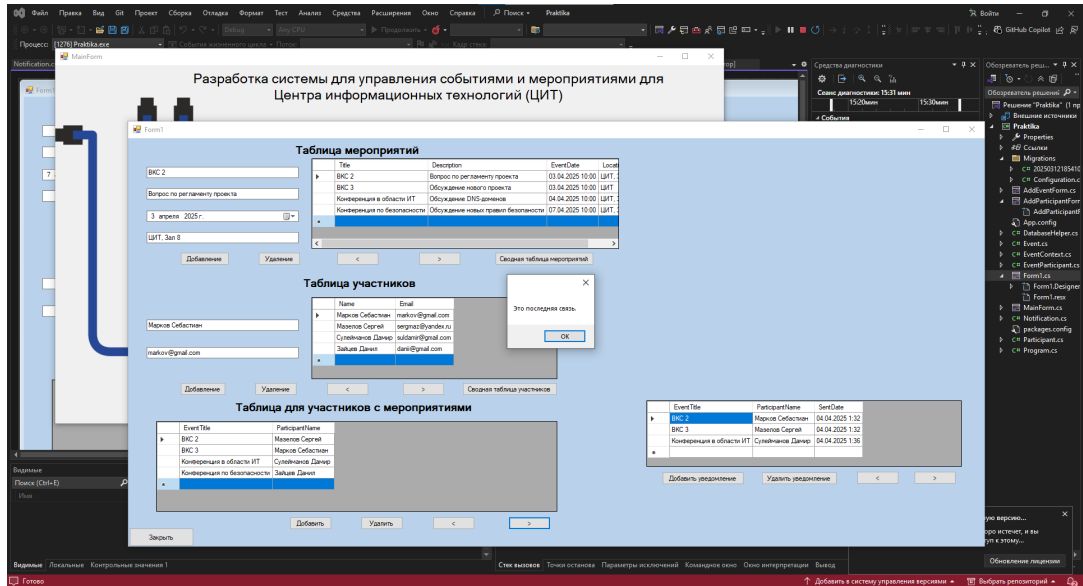


Рисунок 19 – Уведомление о перемещении

Для того чтобы добавить уведомление, необходимо выбрать из таблицы «Таблица для участников с мероприятиями», показано на рисунках 20, 21

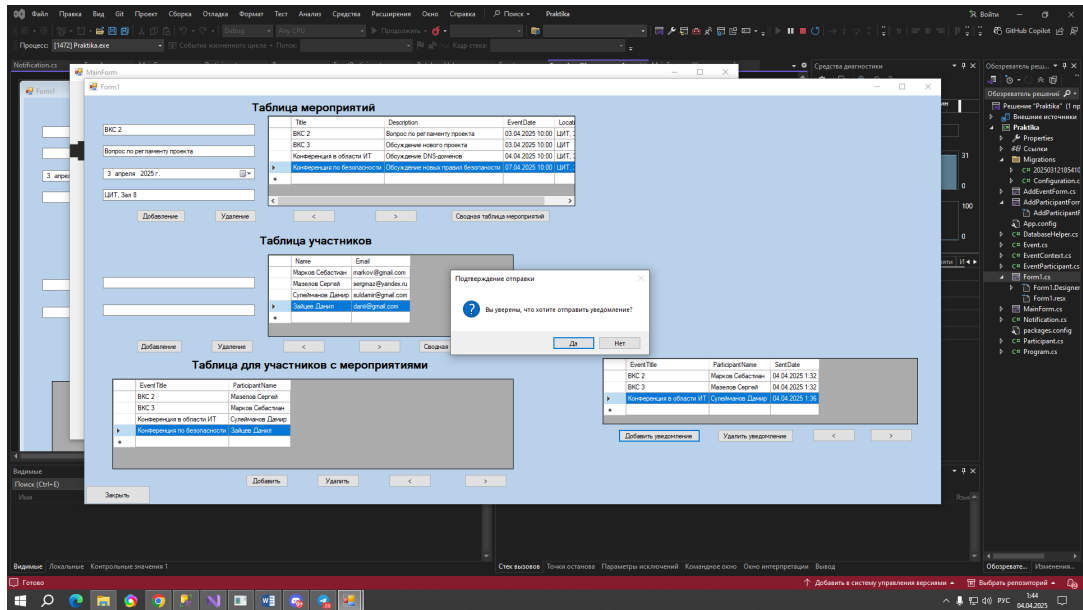


Рисунок 20 – Предупреждение об отправлении уведомления для участника

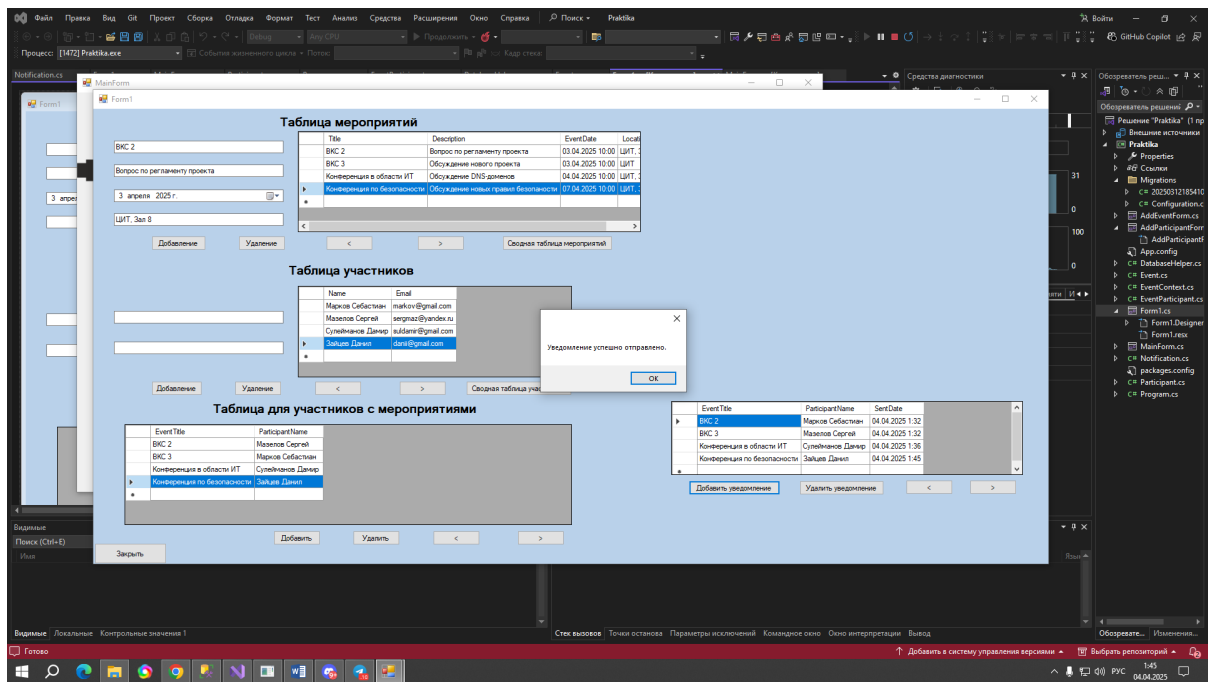


Рисунок 21 – Успешное отправление уведомления

Для каждой из таблиц реализовано перемещение по таблице с помощью кнопок, а также удаление записей, показано на рисунке 22, 23, 24, 25.

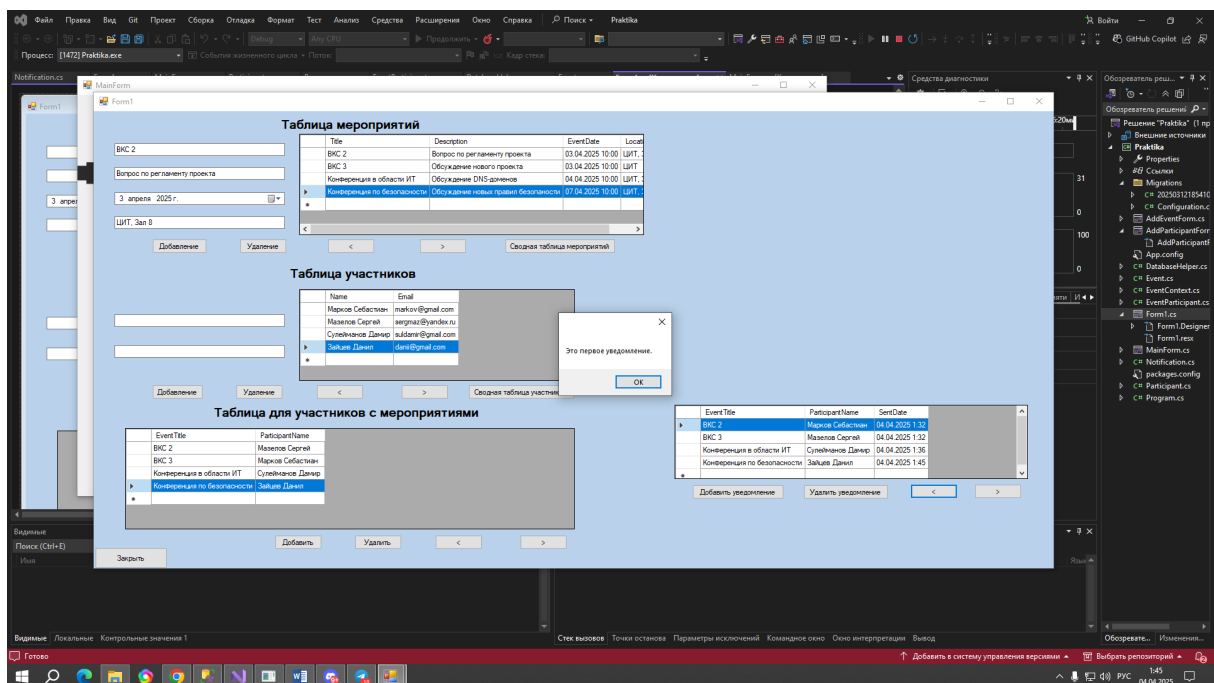


Рисунок 22 – Перемещение «Влево»

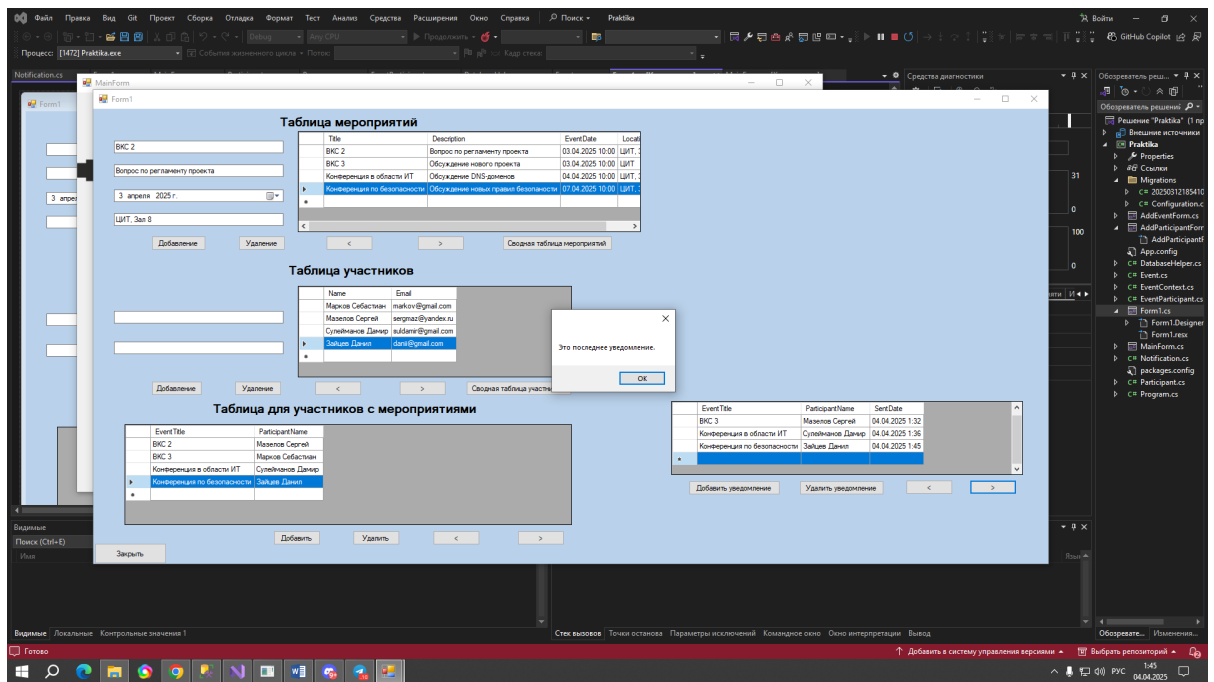


Рисунок 23 – Перемещение «Вправо»

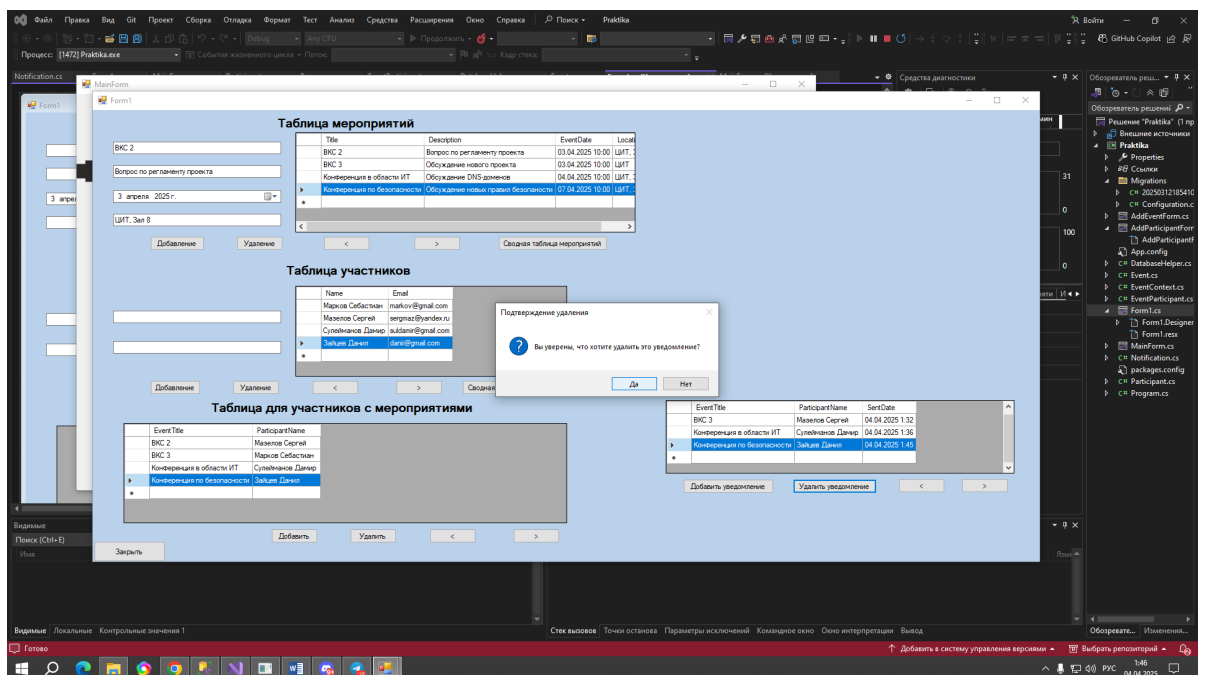


Рисунок 24 – Удаление записи

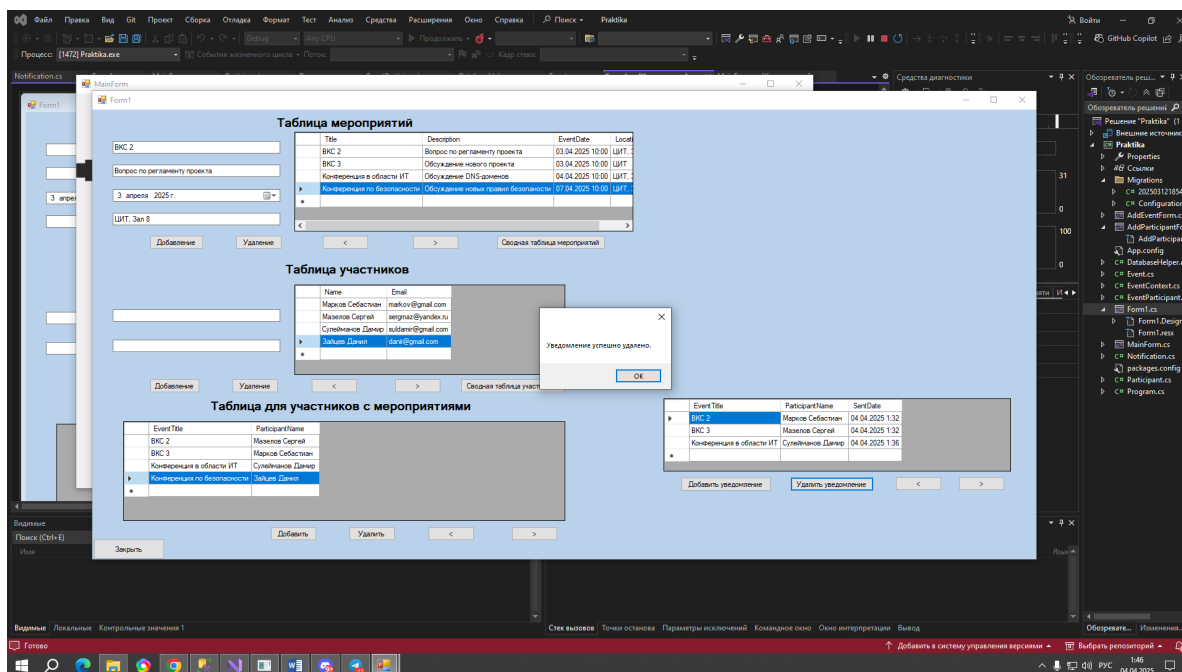


Рисунок 25 – Сообщение об успешном удалении записи

Для того чтобы сгенерировать отчет по таблице «Мероприятия», необходимо нажать на кнопку «Сводная таблица мероприятий», показано на рисунке 26.

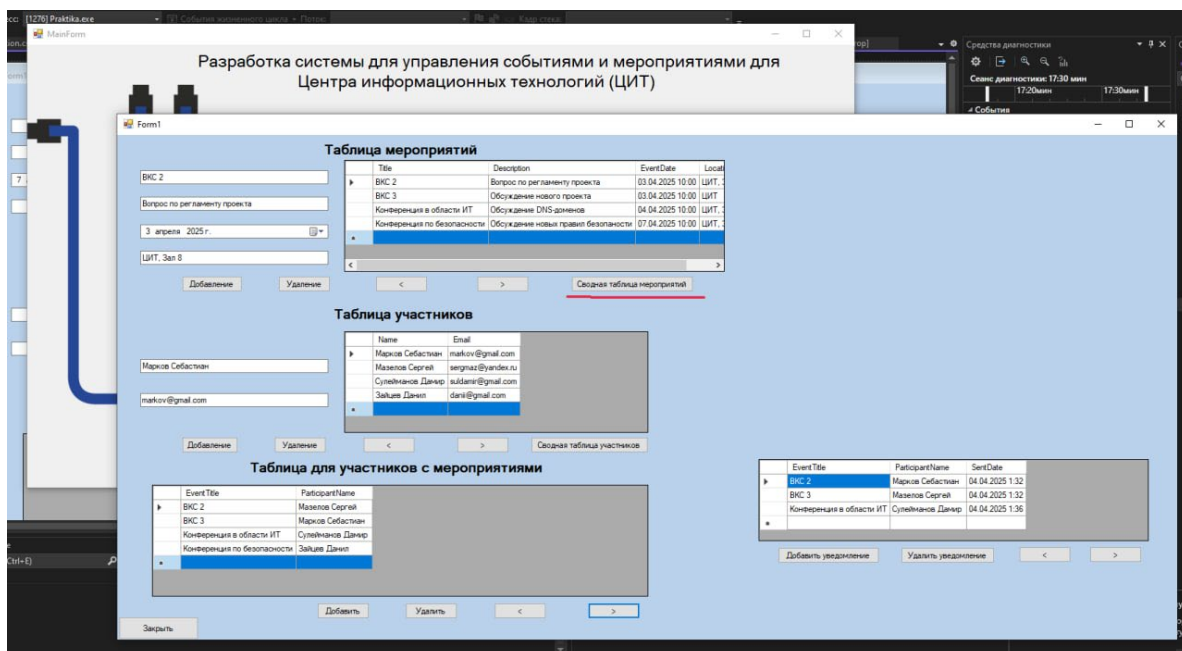


Рисунок 26 – Местоположение кнопки «Сводная таблица мероприятий»

Сгенерированный отчет по таблице Мероприятия, показано на рисунке

27

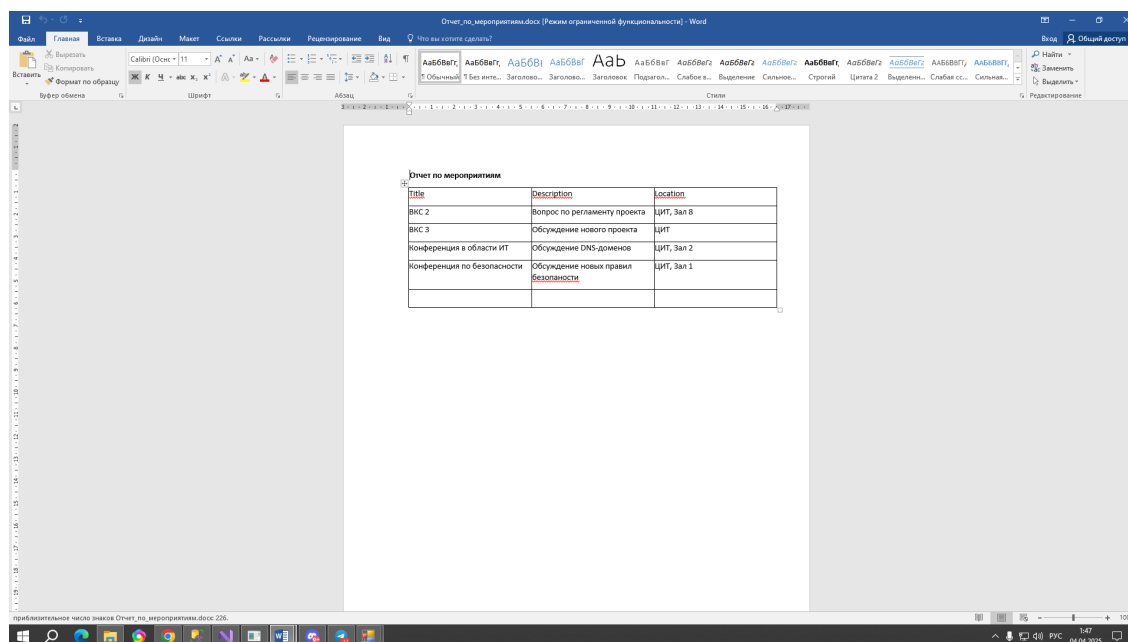


Рисунок 27 – Отчет по Мероприятиям

Для того чтобы сгенерировать отчет по таблице «Участники», необходимо нажать на кнопку «Сводная таблица Участников», показано на рисунке 28.

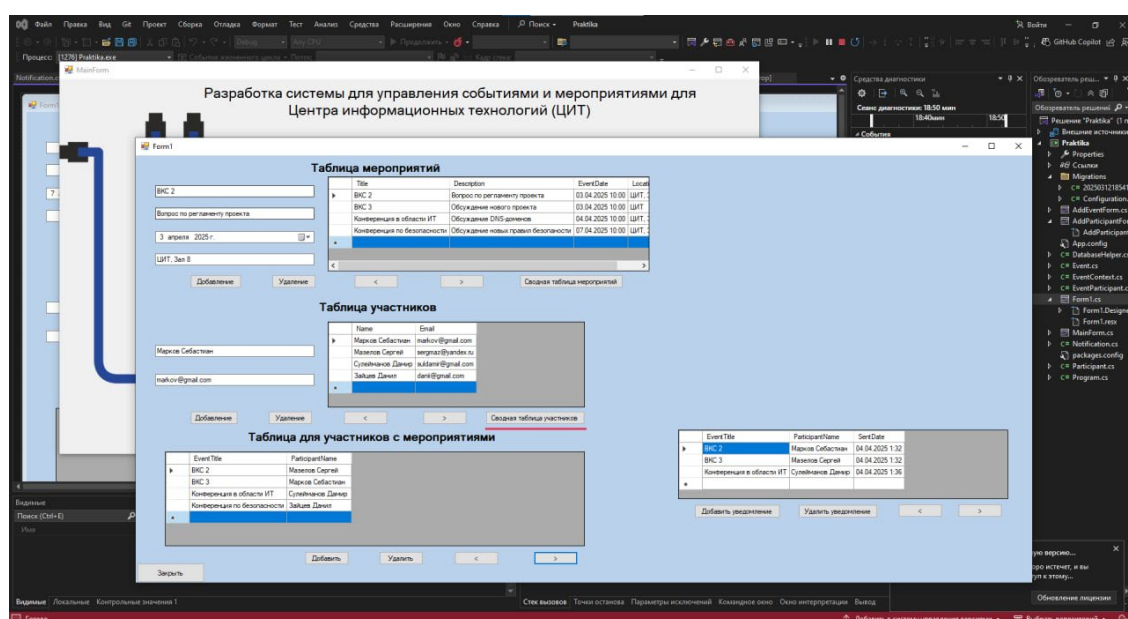


Рисунок 28 – Местоположение кнопки «Сводная таблица участников»

Сгенерированный отчет по таблице Участники, показано на рисунке 29

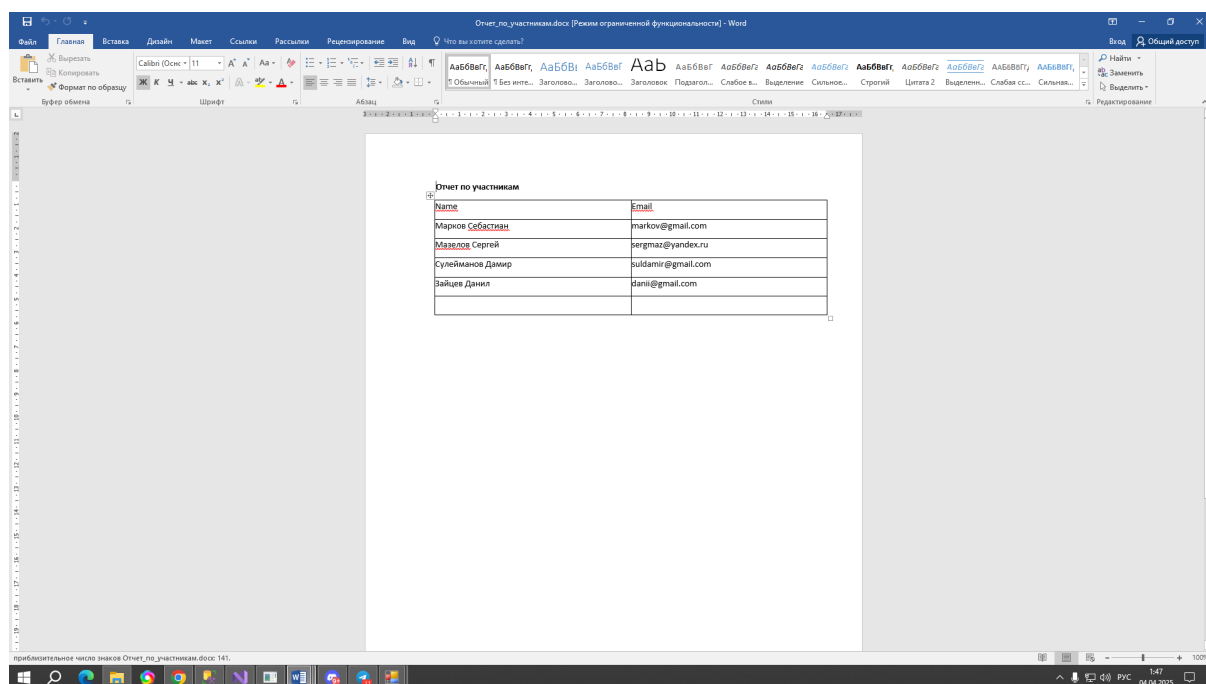


Рисунок 29 – Отчет по Участникам

Чтобы выйти из формы с управлением мероприятиями необходимо нажать на кнопку «Заккрыть», показано на рисунке 30

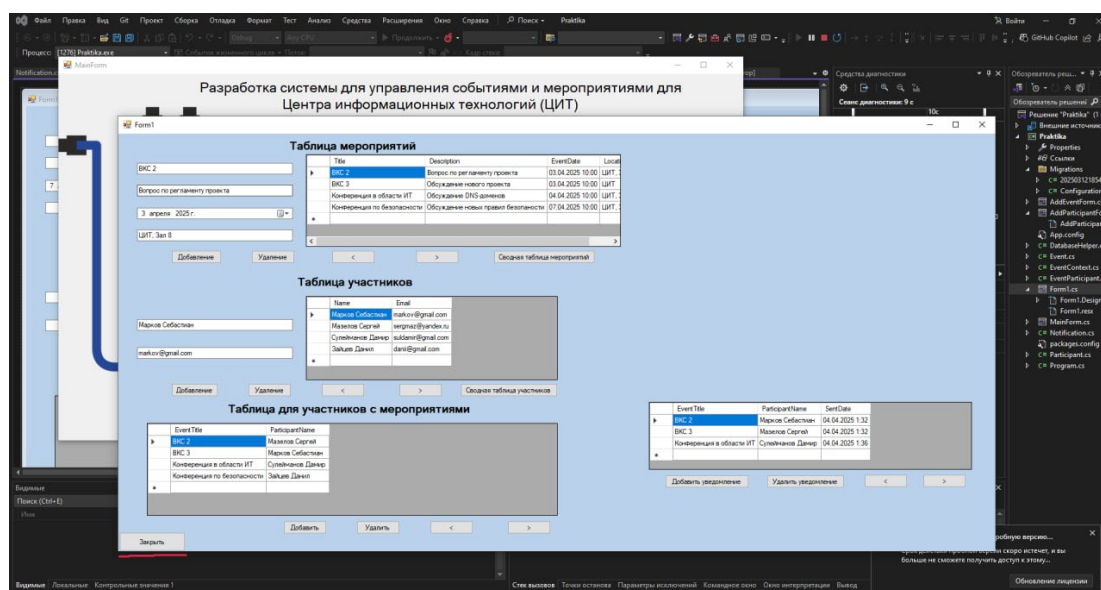


Рисунок 30 – Кнопка «Заккрыть» на форме

Чтобы выйти из главной формы программы необходимо нажать на кнопку «Заккрыть», показано на рисунке 31

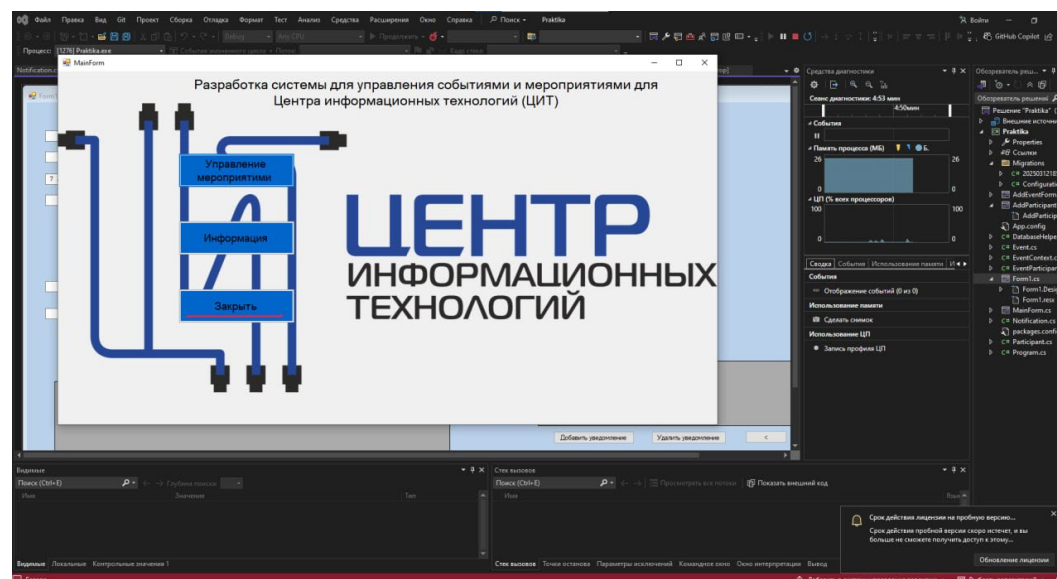


Рисунок 31 – Выход из программы

Проведенное тестирование системы управления мероприятиями подтвердило ее высокую надежность, стабильность и соответствие всем заявленным функциональным требованиям. Комплексная методология тестирования, включающая модульное, интеграционное и системное тестирование, позволила всесторонне оценить работоспособность системы на всех уровнях.

Результаты тестирования демонстрируют, что система успешно справляется с обработкой типовых и пограничных сценариев работы, обеспечивает стабильную производительность при нагрузках, соответствующих требованиям ЦИТ, и обладает интуитивно понятным пользовательским интерфейсом. Все выявленные в процессе тестирования проблемы были своевременно устранены, что подтверждается повторными проверками. Полученные результаты свидетельствуют о готовности системы к промышленной эксплуатации и ее способности эффективно решать задачи автоматизации управления мероприятиями в условиях реальной работы ЦИТ.

Заключение

Выпускная квалификационная работа была посвящена разработке системы управления событиями и мероприятиями для Центра информационных технологий (ЦИТ). В ходе исследования проведен анализ существующих процессов организации мероприятий, выявлены ключевые проблемы и предложено программное решение для их автоматизации на основе современных технологий. Результаты работы имеют практическую значимость для повышения эффективности работы организации.

Проведенный анализ выявил следующие проблемы:

- отсутствие централизованной системы учета и планирования мероприятий;
- необходимость ручного согласования участников, помещений и оборудования;
- трудоемкий процесс формирования отчетности и контроля исполнения задач;
- отсутствие автоматизированных уведомлений и напоминаний.

В ходе выполнения ВКР было разработано и реализовано:

- десктоп-приложение на языке C# с использованием Windows Forms;
- база данных в SQL Server Management Studio с продуманной структурой таблиц;
- механизм уведомления участников о предстоящих мероприятиях;
- инструмент формирования отчетов в Word;
- концептуальная и физическая модель данных (ERD).

Основные достижения работы:

- создание удобного интерфейса для планирования и управления мероприятиями;
- автоматизация процессов согласования и распределения ресурсов.

Разработанная система включает ряд передовых решений:

- применение Entity Framework для объектно-реляционного отображения;
- использование паттерна MVVM для разделения логики и представления;
- реализация асинхронных операций для повышения отзывчивости интерфейса;
- механизм мягкого удаления данных с сохранением истории изменений.

Система включает механизмы гарантии бесперебойной работы:

- восстановление после сбоев с минимальными потерями;
- мониторинг целостности базы данных;
- защита от SQL-инъекций и других киберугроз.

Экономическая эффективность:

Внедрение системы позволит:

- сократить время организации мероприятий на 30–40%;
- уменьшить количество ошибок при бронировании ресурсов;
- повысить прозрачность процессов и контроль исполнения задач;
- снизить затраты на бумажный документооборот.

Перспективы развития:

- разработка мобильного приложения для участников мероприятий;
- интеграция с корпоративными системами (1С, Битрикс24);
- добавление модуля аналитики для оценки эффективности мероприятий;
- масштабирование решения на другие подразделения организации.

Практическая значимость работы:

- готовность решения к внедрению в ЦИТ;

- возможность адаптации под нужды других организаций;
- соответствие современным стандартам автоматизации бизнес-процессов.

- подтвержденная в ходе тестирования стабильность и отказоустойчивость системы

Соответствие образовательным стандартам:

Работа демонстрирует:

- глубокое понимание предметной области;
- владение технологиями C#, SQL Server, Windows Forms;
- умение проектировать базы данных и разрабатывать клиент-серверные приложения;
- способность решать практические задачи в сфере ИТ.

Результаты исследования могут быть использованы как для дальнейшего развития системы в ЦИТ, так и в качестве основы для научных работ в области автоматизации управления мероприятиями.

Таким образом, ВКР успешно решает поставленные задачи и открывает перспективы для дальнейшего развития системы. Разработанное решение готово к пилотному внедрению с последующим масштабированием.

Список используемых источников

1. Альбахари, Дж. C# 10.0. Справочник / Дж. Альбахари. — М.: Диалектика, 2022. — 1040 с.
2. Бен-Ган, И. T-SQL Fundamentals / И. Бен-Ган. — Redmond: Microsoft Press, 2021. — 608 с.
3. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений. — М.: Вильямс, 2020. — 720 с.
4. Васкес, А. Entity Framework Core in Action / А. Васкес. — Shelter Island: Manning, 2023. — 600 с.
5. Гагарина Л.Г., Киселев Д.В. Проектирование и разработка баз данных. — М.: Форум, 2021. — 320 с.
6. Гриффитс, Я. Программирование на C# 10. Подробное руководство / Я. Гриффитс. — Sebastopol: O'Reilly, 2023. — 912 с.
7. Давидсон, Л. Проектирование баз данных для SQL Server / Л. Давидсон. — New York: Apress, 2022. — 672 с.
8. Дейт, К. Дж. SQL и реляционные базы данных / К. Дж. Дейт. — М.: Вильямс, 2020. — 896 с.
9. Зандстра М. PHP: объекты, шаблоны и методики программирования. — СПб.: Питер, 2023. — 576 с.
10. Клайн, К. SQL Server на практике / К. Клайн. — Shelter Island: Manning, 2023. — 640 с.
11. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ. — М.: Вильямс, 2022. — 1328 с.
12. Лерман, Д. Programming Entity Framework Core / Д. Лерман. — Sebastopol: O'Reilly, 2022. — 832 с.
13. Мак-Дональд, М. Pro ASP.NET Core 7 / М. Мак-Дональд. — New York: Apress, 2023. — 1200 с.
14. Макконнелл С. Совершенный код. — М.: Русская редакция, 2021. — 896 с.

15. Мартин Р. Чистая архитектура. Искусство разработки программного обеспечения. — СПб.: Питер, 2022. — 352 с.
16. Механик, А. SQL Server Internals / А. Механик. — Cambridge: Redgate, 2022. — 456 с.
17. Найт, Р. T-SQL Performance Tuning / Р. Найт. — Sebastopol: O'Reilly, 2021. — 384 с.
18. Нейгел, К. C# и .NET. Разработка приложений / К. Нейгел. — СПб.: Питер, 2021. — 944 с.
19. Пайн, Д. C# 10: Ускоренный курс / Д. Пайн. — San Francisco: No Starch Press, 2023. — 480 с.
20. Прайс, М. C# 10 и .NET 6. Современная кроссплатформенная разработка / М. Прайс. — СПб.: Питер, 2022. — 768 с.
21. Рихтер, Дж. CLR via C#. Программирование на платформе Microsoft .NET / Дж. Рихтер. — СПб.: Питер, 2021. — 896 с.
22. Скиена С. Алгоритмы. Руководство по разработке. — М.: БХВ-Петербург, 2021. — 720 с.
23. Скит, Дж. C# для профессионалов: тонкости программирования / Дж. Скит. — М.: Вильямс, 2022. — 688 с.
24. Стилмен, Э. C# 10: Полное руководство / Э. Стилмен. — М.: Диалектика, 2023. — 864 с.
25. Таненбаум Э., Вудхалл А. Операционные системы: Разработка и реализация. — СПб.: Питер, 2019. — 1120 с.
26. Террелл, Р. .NET 7 для инженеров / Р. Террелл. — Shelter Island: Manning, 2023. — 576 с.
27. Троелсен, Э. Язык программирования C# 10 и .NET 6 / Э. Троелсен. — М.: Вильямс, 2022. — 1328 с.
28. Фаулер М. UML. Основы. — М.: Символ-Плюс, 2020. — 192 с.
29. Фримен, А. ASP.NET Core 7. Разработка приложений / А. Фримен. — New York: Apress, 2023. — 1104 с.

30. Шелдон, У. SQL Server 2022: Оптимизация запросов / У. Шелдон. — Birmingham: Packt, 2023. — 520 с.

31. Эспозито, Д. Programming ASP.NET Core / Д. Эспозито. — Redmond: Microsoft Press, 2022. — 752 с.