

СОДЕРЖАНИЕ

ОПРЕДЕЛЕНИЯ	5
ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ	6
РЕФЕРАТ	7
ВВЕДЕНИЕ	8
1 Аналитическая часть	9
1.1 Анализ предметной области	9
1.2 Анализ известных решений	9
1.3 Формализация данных	11
1.4 Формализация ролей и типов	13
1.5 Анализ моделей баз данных	14
1.5.1 Дореляционная модель	15
1.5.2 Реляционная модель	16
1.5.3 Постреляционная модель	17
1.6 Формализация задачи	17
1.7 Вывод	17
2 Конструкторская часть	18
2.1 Требования к программе	18
2.2 Описание сущностей базы данных	19
2.3 Ролевая модель	23
2.4 Формализация мероприятия	23
2.4.1 Определение мероприятия	23
2.4.2 N-мерные случаи	25
2.4.3 Базис мероприятия	25

2.4.4	Уравнения баланса	28
2.4.5	Фундаментальная цена	29
2.4.6	Получение прибыли	31
2.4.7	Известные теоремы	31
2.5	Используемые триггеры	36
2.6	Архитектура приложения	37
2.6.1	Диаграмма потока данных	38
2.6.2	Диаграмма компонентов	38
2.6.3	Диаграмма классов	38
2.7	Вывод	38
3	Технологическая часть	40
3.1	Выбор системы управления базами данных	40
3.2	Выбор средств реализации приложения	41
3.3	Структура приложения и базы данных	41
3.4	Интерфейс приложения	42
3.5	Тестирование	49
3.6	Вывод	50
4	Исследовательская часть	51
ЗАКЛЮЧЕНИЕ		52
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ		53
ПРИЛОЖЕНИЕ А		55

ОПРЕДЕЛЕНИЯ

В данной расчетно-пояснительной записке применяется следующая терминология:

ER-модель (*от англ. Entity-Relationship model*) – модель данных, позволяющая описывать концептуальные схемы предметной области [5].

Диаграмма прецедентов – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей [9].

VIP-персона (*от англ. Very important person*) – человек, имеющий персональные привилегии, льготы из-за своего высокого статуса, популярности или капитала [6].

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В данной расчетно-пояснительной записке применяются следующие сокращения и обозначения:

БД – база данных

СУБД – система управления базами данных

$\mathbb{R}_{\geq 0}$ – множество неотрицательных вещественных чисел

РЕФЕРАТ

ВВЕДЕНИЕ

Целью данной курсовой работы является разработка приложения для внесения, хранения, изменения и обработки данных о мероприятиях.

Чтобы достичь данной цели, необходимо выполнить следующие задачи:

- проанализировать известные форматы представления информации и определить оптимальный вариант;
- проанализировать известные подходы к хранению информации и системы управления базами данных и выбрать наиболее подходящие;
- спроектировать базу данных, определив основные сущности и связи между ними;
- выбрать наиболее подходящие алгоритмы для достижения поставленной цели;
- спроектировать архитектуру и графический интерфейс приложения;
- выбрать средства реализации приложения и реализовать его;
- провести исследование зависимости цены посещения мероприятия от количества человек на нем и других.

1 Аналитическая часть

В этой части рассматриваются анализ предметной области, известных решений и моделей баз данных, формализации задачи, данных, ролей и типов.

1.1 Анализ предметной области

Организация мероприятий – это многогранный и трудоемкий процесс, который требует внимания к деталям и учета множества факторов. От выбора подходящей локации до составления списка гостей, от планирования бюджета до подготовки меню – каждый этап организации требует тщательной проработки. В ходе подготовки организаторы сталкиваются с рядом типичных вопросов, таких как: «Что необходимо приобрести для мероприятия?», «Какое количество гостей ожидается?» и «Какова цена участия?». Эти вопросы, хотя и кажутся простыми, но требуют значительных временных и организационных затрат, особенно если мероприятие масштабное или включает множество участников [1].

Для упрощения этого процесса были разработаны специализированные инструменты – планировщики мероприятий. Эти приложения предназначены для того, чтобы объединить все этапы организации мероприятия в единую систему, сделать процесс планирования более структурированным и прозрачным. Планировщик мероприятий позволяет организаторам:

- 1) систематизировать задачи – разбить процесс организации на этапы и подзадачи;
- 2) координировать участников – вести список гостей, учитывать их предпочтения и информировать о деталях мероприятия;
- 3) управлять бюджетом – учитывать расходы и планировать финансы, чтобы избежать непредвиденных затрат;
- 4) контролировать сроки – устанавливать дедлайны для каждой задачи и отслеживать их выполнение.

1.2 Анализ известных решений

Организация мероприятий всегда была важной и востребованной сферой, потому для упрощения этого процесса были разработаны специализированные

информационные системы, которые автоматизируют множество задач. Наиболее популярными являются:

- 1) Eventbrite – платформа для организации мероприятий, которая позволяет создавать страницы событий, продавать билеты онлайн, собирать данные о посетителях и управлять регистрациями [2];
- 2) Cvent – профессиональная платформа для организации мероприятий, которая предлагает комплексные решения для планирования, управления гостями, бюджетирования и аналитики [3];
- 3) Trello – инструмент для управления проектами и задачами, который можно адаптировать для планирования мероприятий [4].

Критерии сравнения известных решений и результаты их сравнительного анализа представлены в таблицах 1.1 и 1.2 соответственно.

Таблица 1.1 — Критерии сравнения известных решений

Критерий	Описание
Аккаунт	Возможность иметь аккаунт
Бесплатный доступ	Бесплатный доступ ко всем возможностям приложения
Привилегии участников	Возможность выдавать участникам роли с привилегиями
Рейтинг	Формирование оценки мероприятия по оставленным отзывам

Таблица 1.2 — Результаты сравнительного анализа известных решений

Критерий	Eventbrite	Cvent	Trello
Аккаунт	+	+	+
Бесплатный доступ	-	-	+
Привилегии участников	-	+	-
Рейтинг	+	+	-

Ни одно из рассматриваемых решений не обеспечивает пользователя всеми необходимыми функциями для организации мероприятий.

1.3 Формализация данных

Исходя из анализа предметной области, можно выделить следующие ключевые группы данных:

- локация;
- мероприятие;
- день мероприятия;
- участник мероприятия;
- меню мероприятия;
- предмет меню;
- отзыв;
- пользователь.

Группы данных и сведения о них представлены в таблице 1.3

Таблица 1.3 — Группы данных и их сведения

Группа данных	Сведения
Локация	Название, описание, цена аренды, вместимость
Мероприятие	Название, описание, дата, количество участников, количество дней, локация, рейтинг
День мероприятия	Название, порядковый номер, описание, цена посещения, меню
Отзыв	Комментарий, рейтинг, участник мероприятия
Пользователь	Имя, телефон, гендер, роль, права доступа, пароль
Участник мероприятия	Имя, тип, факт оплаты
Меню мероприятия	Название, стоимость
Предмет меню	Название, тип, цена

Диаграмма «сущность-связь» в нотации Чена представлена на рисунке 1.1.

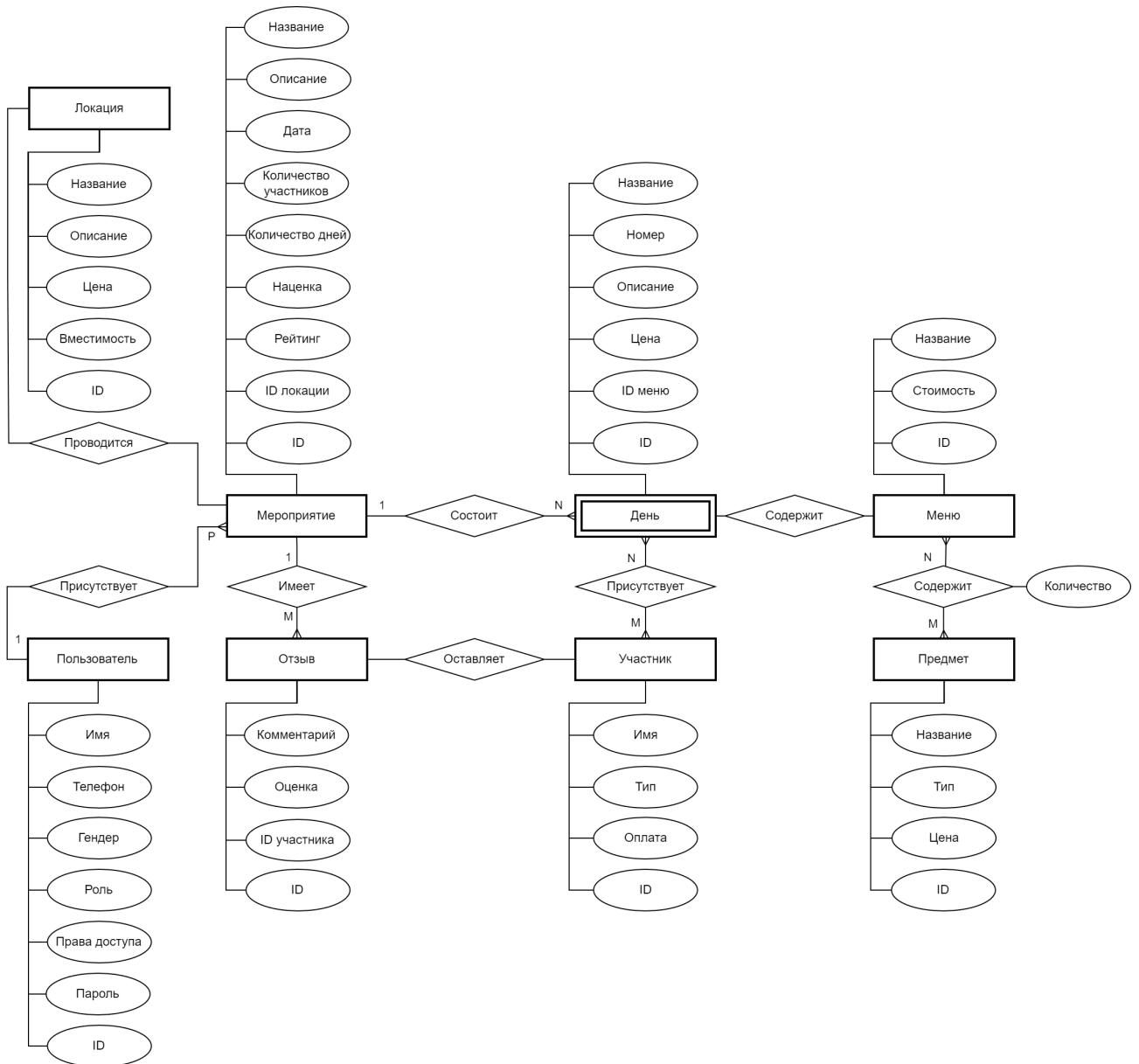


Рисунок 1.1 — Диаграмма «сущность-связь» в нотации Чена

1.4 Формализация ролей и типов

Множества значений перечисляемых сведений представлены в таблице 1.4.

Таблица 1.4 — Множества значений перечисляемых сведений

Группа данных	Сведение	Множество значений
Пользователь	Роль	Гость, зарегистрированный пользователь, администратор
	Гендер	Мужской, женский
Участник мероприятия	Тип	Простой участник, VIP, организатор
Предмет меню	Тип	Однодневный, многодневный

Роли пользователей, выделяемые в разрабатываемой системе, представлены в таблице 1.5.

Таблица 1.5 — Категории пользователей

Роль	Описание
Гость	Незарегистрированный пользователь, который может пройти регистрацию и войти в систему
Зарегистрированный пользователь	Может просматривать данные о мероприятиях и сдавать их, подавать и удалять заявки на участие, а также оставлять и удалять отзывы
Администратор	Обладает правами на просмотр, добавление и изменение данных о мероприятиях

Типы участников, выделяемые в рамках мероприятия, представлены в таблице 1.6.

Таблица 1.6 — Типы участников

Тип	Описание
Простой участник	Не имеет специальных прав или привилегий
VIP-персона	Присутствует на мероприятии бесплатно
Организатор	Имеет права на просмотр и изменение данных о мероприятии и присутствует на нём бесплатно

Типы предметов, выделяемые в рамках меню, представлены в таблице 1.7.

Таблица 1.7 — Типы предметов

Тип	Описание
Однодневный	Требуется только для одного дня мероприятия
Многодневный	Необходим на протяжении всего мероприятия

1.5 Анализ моделей баз данных

База данных – самодокументированное собрание интегрированных записей. Она служит основой для хранения, обработки и анализа информации, необходимой для функционирования бизнеса, научных исследований или других сфер деятельности. В зависимости от типа задач, базы данных могут быть классифицированы на два основных типа [7, 8]:

- 1) OLAP (англ. *Online Analytical Processing*) – технология для многомерного анализа больших объемов данных, используемая для быстрого выполнения сложных запросов;
- 2) OLTP (англ. *Online Transaction Processing*) – система для обработки транзакций в реальном времени, ориентированная на быструю запись и чтение небольших объемов данных.

Система управления базами данных – приложение, обеспечивающее создание, хранение, обновление и поиск информации в базах данных [8].

Основные функции СУБД:

- управление данными во внешней памяти;
- управление буферами оперативной памяти;
- управление транзакциями;
- журнализация;
- поддержка языка или языковых пакетов;
- обеспечение целостности и безопасности базы данных.

Модель данных – это структурное представление элементов данных, их отношений и ограничений в системе управления базами данных. Системы управления базами данных классифицируются по модели данных, которая определяет их архитектуру, структуры данных и методы обработки.

Выделяются три категории моделей [7, 8]:

- дореляционные;
- реляционные;
- постреляционные.

1.5.1 Дореляционная модель

Дореляционные базы данных представляют собой ранние системы управления данными, которые широко использовались до появления реляционной модели. Эти системы, такие как иерархические и сетевые СУБД, были разработаны для организации и хранения данных в сложных структурах, которые отражали специфические взаимосвязи между объектами. Однако, несмотря на свою функциональность, они обладают рядом ограничений, которые делают их менее гибкими и удобными по сравнению с современными реляционными базами данных [7].

Иерархические базы данных используют дерево, где каждый элемент данных (узел) имеет строгую связь «родитель-потомок». Такая организация эффективна для данных, которые естественным образом поддаются иерархическому представлению, например, файловые системы или организационные структуры компаний. Однако главным недостатком иерархических моделей является их неэффективность: доступ к данным часто требует обхода всей структуры, что усложняет выполнение запросов.

Сетевые базы данных являются развитием иерархических моделей, устраняя некоторые их ограничения. В сетевых моделях данные организовываются в виде графа, где узлы могут иметь множественные связи друг с другом. Это позволяет более гибко представлять сложные взаимосвязи между объектами, что делает их подходящими для задач, где данные имеют множество пересекающихся связей. Однако, несмотря на большую универсальность, сетевые модели остаются сложными в проектировании и использовании.

Хотя дореляционные модели хорошо справляются с управлением памятью, их применение осложняется зависимостью от физической структуры данных, что делает процесс внесения изменений в базу данных более трудоёмким.

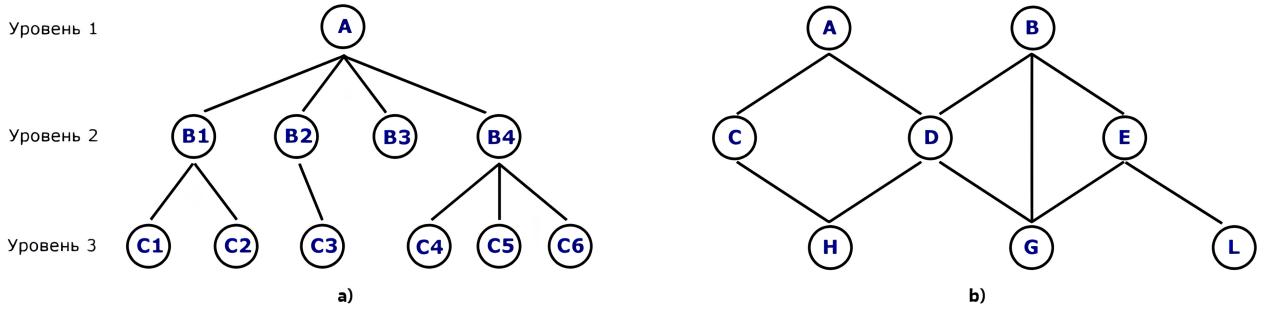


Рисунок 1.2 — Структуры дореляционных моделей: иерархическая (а), сетевая (б)

1.5.2 Реляционная модель

Реляционная модель данных включает в себя три части [8]:

- 1) структурная часть – данные представляются в виде таблиц, называемых отношениями. Каждая таблица состоит из строк (кортежей) и столбцов (атрибутов). Атрибуты имеют определённые домены – допустимые значения, которые они могут принимать. Ключи (первичные и внешние) обеспечивают уникальность записей и связь между таблицами. Схемы определяют структуру базы данных, включая таблицы, атрибуты и связи между ними;
- 2) целостная часть – определяет правила, которые гарантируют корректность и согласованность данных:
 - целостность сущностей: каждый кортеж в таблице должен иметь уникальный идентификатор (первичный ключ), что исключает дублирование записей;
 - целостность ссылок: внешние ключи в одной таблице должны соответствовать первичным ключам в другой, что обеспечивает согласованность данных между таблицами. Это требование поддерживается нормализацией отношений, которая устраняет избыточность данных.

- 3) манипуляционная часть – включает инструменты для работы с данными, такие как реляционная алгебра и реляционное исчисление. Реляционная алгебра основана на теории множеств и предоставляет набор операций (например, выборка, проекция, объединение, разность) для манипуля-

ции данными. Реляционное исчисление, основанное на логике предикатов, позволяет формулировать запросы на декларативном уровне.

Модель отличается простотой, независимостью данных и удобством реализации, но имеет недостатки, такие как сложность описания иерархических связей.

1.5.3 Постреляционная модель

Классическая реляционная модель предполагает, что данные в полях таблиц должны быть неделимыми, что может ограничивать эффективность некоторых приложений. Постреляционная модель устраняет это ограничение, допуская многозначные поля и вложенные таблицы, что упрощает описание сложных структур. Основное преимущество постреляционной модели – возможность объединять несколько связанных таблиц в одну, что повышает наглядность и ускоряет обработку данных.

1.6 Формализация задачи

Необходимо спроектировать и реализовать базу данных, которая будет хранить данные о пользователях, мероприятиях и отзывах. Также требуется разработать приложение с функционалом для просмотра, добавления, редактирования и удаления данных.

1.7 Вывод

В аналитической части работы были проведены анализы предметной области, известных решений и моделей баз данных, проведены формализации задачи, данных и ролей. Был сделан выбор в пользу реляционной модели базы данных.

2 Конструкторская часть

В этой части представляются требования к программе, описание существенных базы данных, ролевая модель, формализация мероприятия, используемые триггеры и архитектура приложения.

2.1 Требования к программе

Программа должна обладать графическим интерфейсом, который позволит пользователю:

- регистрировать новую учетную запись;
- авторизоваться в системе;
- просматривать список всех доступных мероприятий;
- просматривать детальную информацию о мероприятии;
- создавать новые мероприятия;
- редактировать созданные мероприятия;
- отмечать участие в мероприятиях;
- настраивать распределение участника мероприятия по дням;
- оставлять отзывы о мероприятии.

Разработанная программа должна выполнять следующие требования:

- пароли пользователей должны храниться в зашифрованном виде;
- система должна обрабатывать некорректные вводимые данные.

Диаграмма прецендентов представлена на рисунке 2.1.

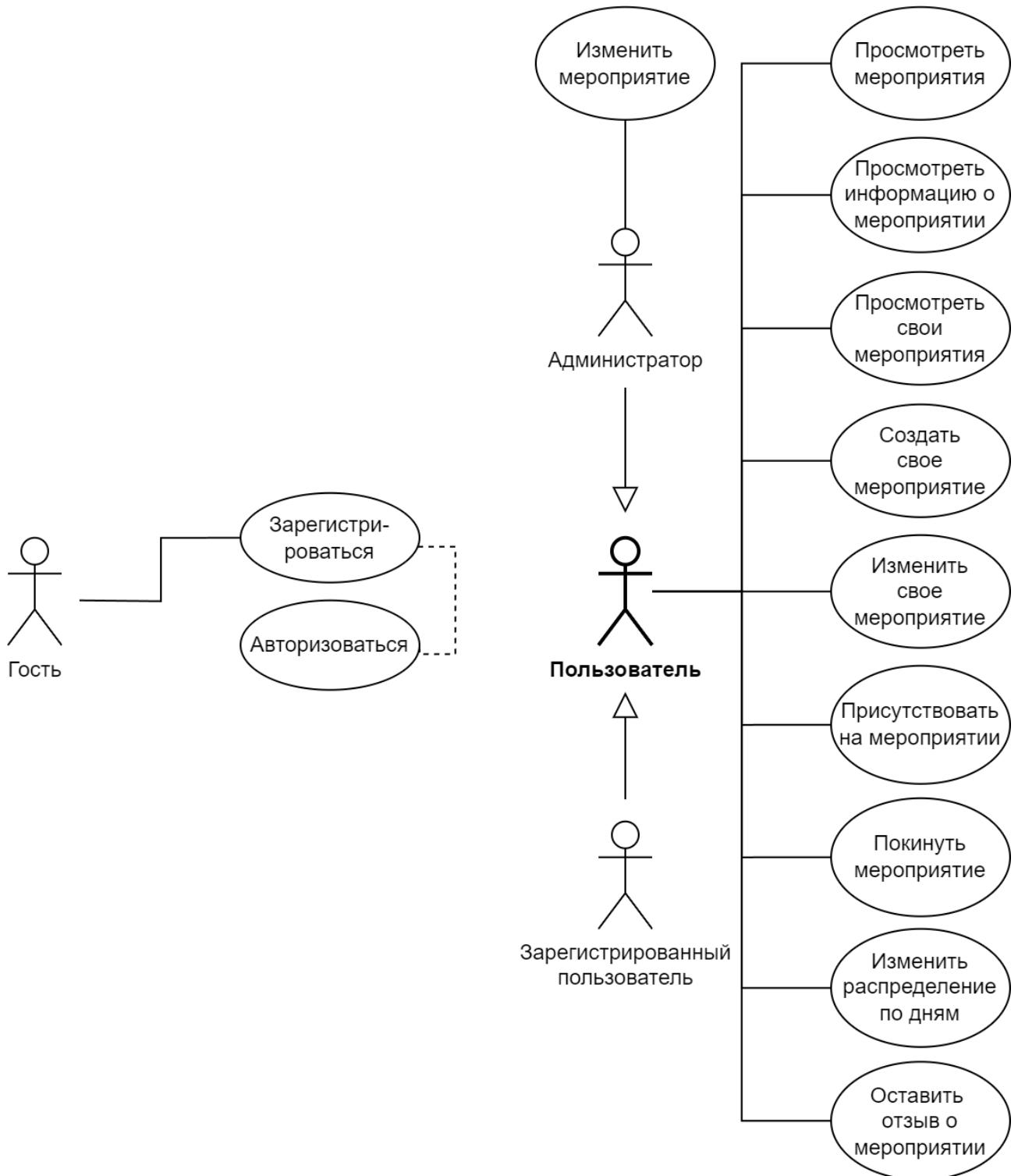


Рисунок 2.1 — Диаграмма прецендентов

2.2 Описание сущностей базы данных

На основе данных, представленных в таблице 1.3, можно определить таблицы, которые должны быть включены в базу данных:

- 1) locations – таблица локаций;
- 2) events – таблица мероприятий;
- 3) persons – таблица участников мероприятий;
- 4) days – таблица дней мероприятий;
- 5) menu – таблица меню дней мероприятий;
- 6) items – таблица предметов меню;
- 7) feedbacks – таблица отзывов участников;
- 8) users – таблица пользователей.

На основе информации о выбранном типе базы данных и диаграммы «сущность-связь» на рисунке 1.1 можно определить структуры столбцов, их типы и ограничения для каждой таблицы, которые представлены в таблицах 2.1-2.8.

Таблица 2.1 — Информация о таблице локаций

Атрибут	Тип данных	Ограничения	Сведение
location_id	UUID	NOT NULL, PRIMARY KEY	Идентификатор локации
name	Строчный	NOT NULL	Название
description	Строчный	NOT NULL	Описание
price	Вещественный	NOT NULL, CHECK (price >= 0)	Цена аренды на 1 день
capacity	Целочисленный	NOT NULL, CHECK (capacity >= 0)	Вместимость

Таблица 2.2 — Информация о таблице мероприятий

Атрибут	Тип данных	Ограничения	Сведение
event_id	UUID	NOT NULL, PRIMARY KEY	Идентификатор мероприятия
name	Строковой	NOT NULL	Название
description	Строковой	NOT NULL	Описание
date	Дата	NOT NULL	Дата
person_count	Целочисленный	NOT NULL, CHECK (person_count >= 0)	Количество участников
days_count	Целочисленный	NOT NULL, CHECK (days_count > 0)	Количество дней
percent	Вещественный	NOT NULL, CHECK (percent >= 0)	Наценка на посещение в процентах
rating	Вещественный	NOT NULL, CHECK (rating BETWEEN 0 AND 10)	Рейтинг

Таблица 2.3 — Информация о таблице дней мероприятий

Атрибут	Тип данных	Ограничения	Сведение
day_id	UUID	NOT NULL, PRIMARY KEY	Идентификатор дня мероприятия
name	Строковой	NOT NULL	Название
sequence_number	Целочисленный	NOT NULL, CHECK (sequence_number > 0)	Порядковый номер
description	Строковой	NOT NULL	Описание
price	Вещественный	NOT NULL, CHECK (price >= 0)	Цена посещения

Таблица 2.4 — Информация о таблице участников мероприятий

Атрибут	Тип данных	Ограничения	Сведение
person_id	UUID	NOT NULL, PRIMARY KEY	Идентификатор участника
name	Строковой	NOT NULL	Имя
type	Перечисляемый	NOT NULL	Тип
paid	Логический	NOT NULL	Факт оплаты

Таблица 2.5 — Информация о таблице меню дней мероприятий

Атрибут	Тип данных	Ограничения	Сведение
menu_id	UUID	NOT NULL, PRIMARY KEY	Идентификатор меню
name	Строковой	NOT NULL	Название
cost	Вещественный	NOT NULL, CHECK (cost >= 0)	Стоимость

Таблица 2.6 — Информация о таблице предметов меню

Атрибут	Тип данных	Ограничения	Сведение
item_id	UUID	NOT NULL, PRIMARY KEY	Идентификатор предмета
name	Строковой	NOT NULL	Название
type	Перечисляемый	NOT NULL	Тип
price	Вещественный	NOT NULL, CHECK (price >= 0)	Цена

Таблица 2.7 — Информация о таблице отзывов

Атрибут	Тип данных	Ограничения	Сведение
feedback_id	UUID	NOT NULL, PRIMARY KEY	Идентификатор отзыва
event_id	UUID	NOT NULL, FOREIGN KEY	Идентификатор мероприятия
person_id	UUID	NOT NULL, FOREIGN KEY	Идентификатор участника
comment	Строковой	NOT NULL	Комментарий
rating	Вещественный	NOT NULL, CHECK (rating BETWEEN 0 AND 10)	Рейтинг

Таблица 2.8 — Информация о таблице пользователей

Атрибут	Тип данных	Ограничения	Сведение
user_id	UUID	NOT NULL, PRIMARY KEY	Идентификатор пользователя
phone	Строковой	NOT NULL	Телефон
gender	Перечисляемый	NOT NULL	Гендер
password	Строковой	NOT NULL	Пароль
role	Перечисляемый	NOT NULL	Роль

Диаграмма базы данных представлена на рисунке 2.2.

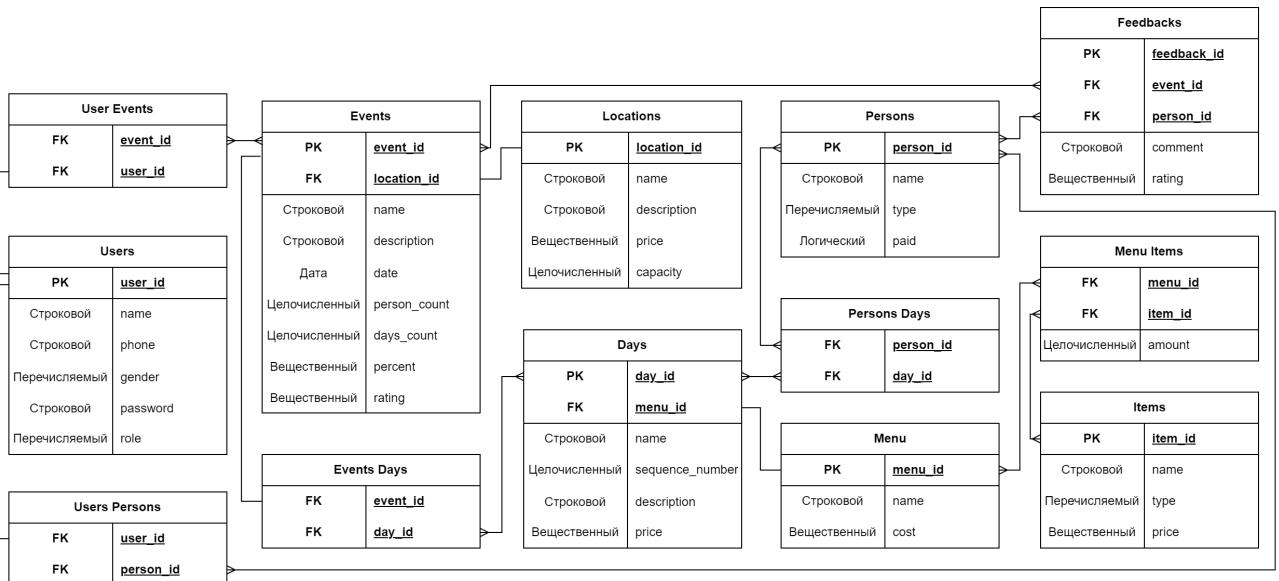


Рисунок 2.2 — Диаграмма базы данных

2.3 Ролевая модель

Ролевая модель в СУБД распределяет права доступа между ролями, ограничивая их действия для обеспечения безопасности:

- гость может выполнять SELECT и INSERT для таблицы пользователей;
- авторизованный пользователь может выполнять SELECT, INSERT, UPDATE и DELETE для всех таблиц;
- администратор имеет все права доступа.

2.4 Формализация мероприятия

Мероприятие рассматривается как система, состоящая из участников, дней, предметов, меню и других элементов.

2.4.1 Определение мероприятия

Множеством участников называется группа людей, участвующих в мероприятии и обозначается:

$$P = \{p_1, p_2, \dots, p_m\} \quad (2.1)$$

где m – общее количество участников.

Множеством дней называются временные интервалы, на которые разбито мероприятие и обозначается:

$$D = \{d_1, d_2, \dots, d_n\} \quad (2.2)$$

где n – общее количество дней.

Общим множеством предметов называется набор материальных ресурсов необходимых для проведения и обозначается:

$$O = \{o_1, o_2, \dots, o_k\} \quad (2.3)$$

где k – общее количество предметов.

Общим множеством меню называется множество наборов предметов и обозначается:

$$M = \{m_1, m_2, \dots, m_p\} \quad (2.4)$$

где $\forall m_i \in M : m_i \subseteq O$ и p – общее количество меню.

Множеством меню называется множество наборов предметов, привязанных к конкретным дням и обозначается:

$$DM = \{(d, m) : (d, m) \in D \times M\} \quad (2.5)$$

где $d \in D$ – день мероприятия, $m \in M$ – меню, соответствующее этому дню и $\forall d \in D \exists! m \in M : (d, m) \in DM$.

Множеством посещений называется множество связей участников с днями их присутствия и обозначается:

$$PD = \{(p, c) : (p, c) \in P \times 2^D\} \quad (2.6)$$

где $p \in P$ – участник, $c \in 2^D$ – подмножество дней, которые участник p планирует посетить, и $\forall p \in P \exists! c \in 2^D : (p, c) \in PD$.

Мероприятием называется кортеж, объединяющий все перечисленные множества и обозначается:

$$E = (P, D, O, M, PD, DM) \quad (2.7)$$

2.4.2 N-мерные случаи

Одномерным случаем называется ситуация выбора участником $p \in P$ ровно одного дня для посещения мероприятия, то есть:

$$p \in P \exists!(p, c) \in PD : |c| = 1 \quad (2.8)$$

Общим одномерным случаем называется ситуация выбора всеми участниками $p \in P$ ровно одного дня для посещения мероприятия, то есть:

$$\forall p \in P \exists!(p, c) \in PD : |c| = 1 \quad (2.9)$$

N-мерным случаем называется ситуация выбора участником $p \in P$ ровно n дней для посещения мероприятия, то есть:

$$p \in P \exists!(p, c) \in PD : |c| = n \quad (2.10)$$

Общим n-мерным случаем называется ситуация выбора всеми участниками $p \in P$ от 1-го до n дней для посещения мероприятия, то есть:

$$\forall p \in P \exists!(p, c) \in PD : |c| \in \{1, \dots, n\} \quad (2.11)$$

Из определения следует, что общий одномерный случай является частным примером общего n -мерного случая.

2.4.3 Базис мероприятия

Базисом мероприятия называется множество функций, обрабатывающих его.

Функции базиса можно классифицировать следующим образом:

- 1) первого порядка – функции, рассматривающие одномерный случай;
- 2) n -го порядка – функции, рассматривающие n -мерный случай.

Из определения следует, что функции первого порядка являются частным примером функций n -го порядка.

Функции стоимости

Функцией стоимости C называется функция, ставящая своему объекту-аргументу в соответствие его денежную стоимость. Областью определения функции C является $D(C) = E \cup D \cup M \cup O$, а областью значений $E(C) = \mathbb{R}_{\geq 0}$.

Функция стоимости предмета обозначается как:

$$C_O : O \rightarrow \mathbb{R}_{\geq 0} \quad (2.12)$$

Функция стоимости меню обозначается как $C_M : M \rightarrow \mathbb{R}_{\geq 0}$ и определяется как сумма всех стоимостей предметов, входящих в меню:

$$C_M(m) = \sum_{o_i \in m} C_O(o_i) \quad (2.13)$$

Функция стоимости дня первого порядка обозначается как $C_D : D \rightarrow \mathbb{R}_{\geq 0}$ и определяется как стоимость меню, соответствующего этому дню:

$$C_D(d) = C_M(m), \text{ где } (d, m) \in DM \quad (2.14)$$

Функция стоимости дня n-го порядка обозначается как $C_D : D^n \rightarrow \mathbb{R}_{\geq 0}$ и определяется как сумма всех стоимостей меню, соответствующих набору-аргументу из n дней:

$$C_D(d_1, \dots, d_n) = \sum_{d_i \in (d_1, \dots, d_n)} C_M(m), \text{ где } (d_i, m) \in DM \quad (2.15)$$

Функция стоимости мероприятия обозначается как $C_E : E \rightarrow \mathbb{R}_{\geq 0}$ и определяется как сумма стоимостей всех дней мероприятия:

$$C_E(E) = \sum_{d_i \in D} C_D(d_i) \quad (2.16)$$

Функции цены

Функцией цены P называется функция, ставящая своему объекту-аргументу в соответствие цену его посещения участником $p \in P$. Областью определения

функции P является $D(P) = E \cup D$, а областью значений $E(P) = \mathbb{R}_{\geq 0}$.

Функция цены дня первого порядка обозначается как:

$$P_D : D \rightarrow \mathbb{R}_{\geq 0} \quad (2.17)$$

Функция цены дня n -го порядка обозначается как:

$$P_D : D^n \rightarrow \mathbb{R}_{\geq 0} \quad (2.18)$$

Функция цены мероприятия обозначается как $P_E : E \rightarrow \mathbb{R}_{\geq 0}$ и определяется как сумма цен посещения всех его дней:

$$P_E(E) = \sum_{d_i \in D} P_D(d_i) \quad (2.19)$$

Вспомогательные функции

Функция текущих комбинаций дней обозначается как $H : E \rightarrow 2^D$ и определяется как множество текущих комбинаций дней $c \in 2^D$, выбранных участниками P :

$$H(E) = \{c : p \in P \exists! c : (p, c) \in PD\} \quad (2.20)$$

Функция количества человек дня первого порядка обозначается как $N_D : D \rightarrow \mathbb{N}$ и определяется как мощность множества участников конкретного дня:

$$N_D(d) = |\{p : (p, d) \in PD\}| \quad (2.21)$$

Функция количества человек дня n -го порядка обозначается как $N_D : D^n \rightarrow \mathbb{N}$ и определяется как мощность множества участников конкретного набора из n дней:

$$N_D(d_1, \dots, d_n) = |\{p : (p, (d_1, \dots, d_n)) \in PD\}| \quad (2.22)$$

Функция строгого количества человек дня первого порядка обозначается как $N_D^* : D \rightarrow \mathbb{N}$ и определяется как мощность множества участников,

посещающих только конкретный день:

$$N_D^*(d) = |\{p : (p, d) \in PD \wedge \forall d_i \in D \setminus d : (p, d_i) \notin PD\}| \quad (2.23)$$

Функция строгого количества человек дня n-го порядка обозначается как $N_D^* : D^n \rightarrow \mathbb{N}$ и определяется как мощность множества участников, посещающих только дни конкретной комбинации:

$$N_D^*(c) = |\{p : (p, c) \in PD \wedge \forall d_i \in D \setminus c : (p, d_i) \notin PD\}| \quad (2.24)$$

Функция количества человек мероприятия обозначается как $N_E : E \rightarrow \mathbb{N}$ и определяется как мощность множества участников:

$$N_E(E) = |P| \quad (2.25)$$

Функция коэффициента дня первого порядка обозначается как $A : D \rightarrow \mathbb{R}_{\geq 0}$ и определяется как отношение стоимости дня к минимальной стоимости:

$$A(d) = \frac{C_D(d)}{\min_{d_i \in D} C_D(d_i)} \quad (2.26)$$

Функция коэффициента дня n-го порядка обозначается как $A : D^n \rightarrow \mathbb{R}_{\geq 0}$ и определяется как сумма отношений стоимости каждого дня из набора к минимальной стоимости:

$$A(d_1, \dots, d_n) = \frac{1}{\min_{d_i \in D} C_D(d_i)} \sum_{d_i \in \{d_1, \dots, d_n\}} C_D(d_i) \quad (2.27)$$

2.4.4 Уравнения баланса

Общий одномерный случай

Чтобы все расходы на конкретный день были покрыты, необходимо, чтобы доходы от участников этого дня были им равны:

$$C_D(d) = P_D(d) \cdot N_D(d) \quad (2.28)$$

Из определения 2.16 и уравнения 2.28 имеем:

$$C_E(E) = \sum_{d_i \in D} C_D(d_i) = \sum_{d_i \in D} P_D(d_i) \cdot N_D(d_i) \quad (2.29)$$

Общий n -мерный случай

Чтобы все расходы на конкретную комбинацию дней были покрыты, необходимо, чтобы доходы от участников этой комбинации были им равны:

$$C_D(d_1, \dots, d_n) = P_D(d_1, \dots, d_n) \cdot N_D^*(d_1, \dots, d_n) \quad (2.30)$$

В n -мерном случае участники оплачивают присутствие не на конкретных днях, а на их комбинациях, потому, чтобы покрыть расходы на всё мероприятие, необходимо покрыть расходы на все комбинации:

$$C_E(E) = \sum_{c_i \in H(E)} P_D(c_i) \cdot N_D^*(c_i) \quad (2.31)$$

Уравнения 2.29 и 2.31 называются *уравнениями баланса* для общих одномерного и n -мерного случаев соответственно.

2.4.5 Фундаментальная цена

Фундаментальная цена – это формальное решение уравнения баланса, которое используется для расчета цены посещения каждого дня мероприятия. Она определяется таким образом, чтобы цена каждого дня мероприятия была пропорциональна его стоимости.

Цена посещения дня выражается через фундаментальную цену P_0 и коэффициент дня:

- для одномерного случая:

$$P_D(d) = A(d) \cdot P_0(E) \quad (2.32)$$

- для n -мерного случая:

$$P_D(d_1, \dots, d_n) = A(d_1, \dots, d_n) \cdot P_0(E) \quad (2.33)$$

В силу линейности уравнения баланса, если фундаментальная цена существует, она единственна:

$$P_0 = P_0(E) \quad (2.34)$$

Общий одномерный случай

Подставив уравнение 2.32 в уравнение 2.28, имеем:

$$C_D(d) = A(d) \cdot P_0 \cdot N_D(d) \quad (2.35)$$

Подставив уравнение 2.35 в уравнение баланса 2.29, имеем:

$$C_E(E) = \sum_{d_i \in D} A(d_i) \cdot P_0 \cdot N_D(d_i) = P_0 \cdot \sum_{d_i \in D} A(d_i) \cdot N_D(d_i) \quad (2.36)$$

Из уравнения 2.36 имеем:

$$P_0(E) = \frac{C_E(E)}{\sum_{d_i \in D} A(d_i) \cdot N_D(d_i)} \quad (2.37)$$

Общий n-мерный случай

Подставив уравнение 2.33 в уравнение 2.30, имеем:

$$C_D(d_1, \dots, d_n) = A(d_1, \dots, d_n) \cdot P_0 \cdot N_D^*(d_1, \dots, d_n) \quad (2.38)$$

Подставив уравнение 2.38 в уравнение баланса 2.31, имеем:

$$C_E(E) = \sum_{c_i \in H(E)} A(c_i) \cdot P_0 \cdot N_D^*(c_i) = P_0 \cdot \sum_{c_i \in H(E)} A(c_i) \cdot N_D^*(c_i) \quad (2.39)$$

Из уравнения 2.39 имеем:

$$P_0(E) = \frac{C_E(E)}{\sum_{c_i \in H(E)} A(c_i) \cdot N_D^*(c_i)} \quad (2.40)$$

2.4.6 Получение прибыли

Наличие прибыли – это финансовый результат, при котором доходы от мероприятия превышают его затраты. В рамках уравнения баланса условие наличия прибыли можно обозначить как:

$$C_E(E) < \sum_{c_i \in H(E)} P_D(c_i) \cdot N_D^*(c_i) \Leftrightarrow \sum_{c_i \in H(E)} P_D(c_i) \cdot N_D^*(c_i) - C_E(E) > 0 \quad (2.41)$$

Прибыль определяется как положительная разница между совокупным доходом от участников мероприятия и общими затратами на его организацию:

$$\Pi = \sum_{c_i \in H(E)} P_D(c_i) \cdot N_D^*(c_i) - C_E(E) \quad (2.42)$$

Прибыль может быть реализована через процентную наценку на фундаментальную цену. Пусть $\alpha \geq 0$ – коэффициент, увеличивающий фундаментальную цену P_0 . Тогда функция цены посещения конкретного дня примет вид:

$$P'_D(c_i) = (1 + \alpha) \cdot A(c_i) \cdot P_0 \quad (2.43)$$

Прибыль в этом случае будет равна:

$$\Pi = \sum_{c_i \in H(E)} P'_D(c_i) \cdot N_D^*(c_i) - C_E(E) \quad (2.44)$$

$$\Pi = (1 + \alpha) \cdot \sum_{c_i \in H(E)} P_D(c_i) \cdot N_D^*(c_i) - C_E(E) \quad (2.45)$$

$$\Pi = (1 + \alpha) \cdot C_E(E) - C_E(E) = \alpha \cdot C_E(E) \quad (2.46)$$

2.4.7 Известные теоремы

Теорема об условии существования решения

Уравнение баланса для общего n -мерного случая имеет решение, только если:

$$\begin{cases} \forall d \in D : C_D(d) > 0 \\ \forall d \in D : N_D(d) > 0 \end{cases} \quad (2.47)$$

Доказательство: для существования решения уравнения баланса необходимо и достаточно: а) чтобы существовала фундаментальная цена; б) чтобы были покрыты все затраты на мероприятие.

а) из формулы 2.40 имеем, что для существования фундаментальной цены требуется:

$$\sum_{c_i \in H(E)} A(c_i) \cdot N_D(c_i) > 0 \quad (2.48)$$

Условие 2.48 не выполняется, если:

- $\exists c_i \in H(E) : A(c_i)$ не определено, то есть $\min_{d_i \in D} C_D(d_i) = 0$;
- $\exists c_i \in H(E) : N_D(c_i) = 0$, то есть $H(E) = \emptyset \Leftrightarrow PD = \emptyset$.

Следовательно, для выполнения условия 2.48 необходимо и достаточно:

$$\begin{cases} \forall d \in D : C_D(d) > 0 \\ PD \neq \emptyset \end{cases} \quad (2.49)$$

б) если $\exists d \in D : N_D(d) = 0$, то затраты на этот день в случае их наличия не покрываются, так как $P_D(d) \cdot N_D(d) = 0$. Следовательно, фундаментальная цена P_0 не является решением уравнения баланса. Значит, для покрытия затрат на всё мероприятие необходимо следующее условие:

$$\forall d \in D : C_D(d) > 0 \Rightarrow N_D(d) > 0 \quad (2.50)$$

Добавив условие 2.50 в систему 2.49, имеем:

$$\begin{cases} \forall d \in D : C_D(d) > 0 \\ PD \neq \emptyset \\ \forall d \in D : C_D(d) > 0 \Rightarrow N_D(d) > 0 \end{cases} = \begin{cases} \forall d \in D : C_D(d) > 0 \\ PD \neq \emptyset \\ \forall d \in D : N_D(d) > 0 \end{cases} \quad (2.51)$$

Заметив, что $\forall d \in D : N_D(d) > 0 \Rightarrow PD \neq \emptyset$, в итоге имеем:

$$\begin{cases} \forall d \in D : C_D(d) > 0 \\ \forall d \in D : N_D(d) > 0 \end{cases} \quad (2.52)$$

Теорема об инвариантности к масштабу

Если все стоимости $C_D(d)$ возрастают в k раз, то фундаментальная цена P_0 так же возрастет в k раз, при этом:

$$\frac{P_D(d)}{P'_D(d)} = \frac{1}{k} \quad (2.53)$$

где $P'_D(d)$ – изменённая функция цены.

Доказательство: пусть $C'_D(d) = k \cdot C_D(d)$, тогда:

- $A'(d) = \frac{C'_D(d)}{\min_{d_i \in D} C'_D(d_i)} = \frac{k \cdot C_D(d)}{k \cdot \min_{d_i \in D} C_D(d)} = A(d);$
- $C'_E(E) = \sum_{d_i \in D} C'_D(d_i) = k \cdot \sum_{d_i \in D} C_D(d_i) = k \cdot C_E(E).$

Следовательно, имеем:

$$P'_0 = \frac{C'_E(E)}{\sum_{c_i \in H(E)} A'(c_i) \cdot N_D(c_i)} = \frac{k \cdot C_E(E)}{\sum_{c_i \in H(E)} A(c_i) \cdot N_D(c_i)} = k \cdot P_0 \quad (2.54)$$

При этом $P'_D(d) = A'(d) \cdot P'_0 = A(d) \cdot k \cdot P_0 = k \cdot P_D(d)$, то есть:

$$\frac{P_D(d)}{P'_D(d)} = \frac{1}{k} \quad (2.55)$$

Теорема о равномерном распределении

При общем одномерном случае, если участники распределены по дням равномерно и все их стоимости одинаковы, то и цены их посещения равны.

Доказательство: по условию имеем

$$\forall d \in D : \begin{cases} N_D(d) = N \\ C_D(d) = C \end{cases} \quad (2.56)$$

Следовательно:

$$\forall d \in D : A(d) = \frac{C_D(d)}{\min_{d_i \in D} C_D(d_i)} = \frac{C}{C} = 1 \quad (2.57)$$

Тогда при условии $|D| = n$:

$$P_0 = \frac{C_E(E)}{\sum_{d_i \in D} A(d_i) \cdot N_D(d_i)} = \frac{n \cdot C}{\sum_{d_i \in D} N_D(d_i)} = \frac{n \cdot C}{n \cdot N} = \frac{C}{N} \quad (2.58)$$

Имеем:

$$\forall d \in D : P_D(d) = A(d) \cdot P_0 = 1 \cdot \frac{C}{N} = \frac{C}{N} \quad (2.59)$$

Теорема о количестве участников

Мощность множества участников равна сумме количеств участников каждой текущей комбинации:

$$|P| = \sum_{c_i \in H(E)} N_D(c_i) \quad (2.60)$$

Доказательство: из определения PD следует, что каждый участник $p \in P$ выбирает ровно одну комбинацию дней $c \subseteq D$. То есть:

$$\forall p \in P \exists! c \in H(E) : (p, c) \in PD \quad (2.61)$$

Множество участников P можно разбить на непересекающиеся множества P_{c_i} , где каждое P_{c_i} содержит участников, выбравших комбинацию c_i :

$$P = \bigcup_{c_i \in H(E)} P_{c_i}, \text{ где } P_{c_i} = \{p : (p, c_i) \in PD\} \quad (2.62)$$

Поскольку все P_{c_i} попарно не пересекаются, то мощность их объединения равна сумме их мощности:

$$|P| = \sum_{c_i \in H(E)} |P_{c_i}| \quad (2.63)$$

Из определения имеем $N_D(c_i) = |P_{c_i}|$. Следовательно:

$$|P| = \sum_{c_i \in H(E)} N_D(c_i) \quad (2.64)$$

Теорема о критическом количестве участников

Для покрытия расходов мероприятия с стоимостью $C_E(E)$ и максимальной ценой посещения P_{max} необходимо:

$$|P| \geq \frac{C_E(E)}{P_{max}} \quad (2.65)$$

Доказательство: из условия имеем:

$$P_{max} = \max_{c_i \in H(E)} P_D(c_i) \quad (2.66)$$

Рассмотрим уравнение баланса:

$$C_E(E) = \sum_{c_i \in H(E)} P_D(c_i) \cdot N_D(c_i) \leq \max_{c_i \in H(E)} P_D(c_i) \cdot \sum_{c_i \in H(E)} N_D(c_i) \quad (2.67)$$

Применив теорему о количестве участников, неравенство 2.67 можно переписать следующим образом:

$$C_E(E) \leq P_{max} \cdot |P| \Leftrightarrow |P| \geq \frac{C_E(E)}{P_{max}} \quad (2.68)$$

Теорема об ограничении наценки

Коэффициент наценки α при условии $P_D(c_i) \leq P_{max}$ для всех комбинаций дней $c_i \in H(E)$ ограничен сверху:

$$\alpha \leq \min_{c_i \in H(E)} \left(\frac{P_{max}}{A(c_i) \cdot P_0} - 1 \right) \quad (2.69)$$

Доказательство: по определению цены комбинации дней с наценкой:

$$P_D(c_i) = (1 + \alpha) \cdot A(c_i) \cdot P_0 \quad (2.70)$$

Условие $\forall c_i \in H(E) : P_D(c_i) \leq P_{max}$ может быть записано следующим образом:

$$(1 + \alpha) \cdot A(c_i) \cdot P_0 \leq P_{max} \Leftrightarrow \alpha \leq \frac{P_{max}}{A(c_i) \cdot P_0} - 1 \quad (2.71)$$

Чтобы неравенство выполнялось для всех c_i , выберем минимальное значение правой части:

$$\alpha \leq \min_{c_i \in H(E)} \left(\frac{P_{max}}{A(c_i) \cdot P_0} - 1 \right) \quad (2.72)$$

Теорема о самом дешевом дне

Если участники концентрируются на дне с минимальной стоимостью $C_D(d)$, то фундаментальная цена P_0 возрастает.

Доказательство: пусть $d' \in D$ – день с минимальной стоимостью, то есть $C_D(d') = \min_{d_i \in D} C_D(d_i)$. Тогда имеем по определению коэффициента дня:

$$A(d') = \frac{C_D(d')}{\min_{d_i \in D} C_D(d_i)} = \frac{\min_{d_i \in D} C_D(d_i)}{\min_{d_i \in D} C_D(d_i)} = 1 \quad (2.73)$$

Если $N_D(d')$ увеличивается при $N_E(E) = const$, тогда существует день $d \in D, d \neq d'$ такой, что $N_D(d)$ уменьшается. Следовательно,

$\sum_{c_i \in H(E)} A(c_i) \cdot N_D(c_i)$ будет уменьшаться, так как $\forall d \in D : A(d') \leq A(d)$, и P_0 будет возрастать.

2.5 Используемые триггеры

В базе данных реализованы следующие триггеры, каждый из которых автоматизирует управление связанными данными мероприятий, включая создание и удаление связанных сущностей, синхронизацию количества дней, обновление количества участников, динамическое ценообразование, пересчёт стоимости меню и рейтинга мероприятия:

- 1) Триггер создания связанных данных мероприятия – автоматически создаёт связанные дни и меню при добавлении нового мероприятия;
- 2) Триггер синхронизации количества дней – поддерживает актуальное количество дней мероприятия при изменении связей, увеличивает или уменьшает число дней в соответствии с заданным значением, удаляя или добавляя необходимые сущности;
- 3) Триггер обновления количества участников – пересчитывает общее число участников мероприятия при изменении данных о посещении дней, учитывает добавление, удаление или изменение записей о присутствии;

- 4) Триггер динамического ценообразования – автоматически корректирует цены дней при изменении влияющих параметров;
- 5) Триггер удаления связанных данных – обеспечивает каскадное удаление всех связанных сущностей при удалении мероприятия, гарантируя целостность данных;
- 6) Триггер обновления стоимости меню – пересчитывает стоимость меню при изменении его состава;
- 7) Триггер расчёта рейтинга мероприятия – обновляет средний рейтинг мероприятия при добавлении, изменении или удалении отзывов.

2.6 Архитектура приложения

Структура данного приложения основана на принципах чистой архитектуры. Данная архитектура направлена на создание систем, которые легко модифицировать и тестировать, а также которые обеспечивают высокую степень независимости от внешних факторов, таких как фреймворки, базы данных и пользовательские интерфейсы [10].



Рисунок 2.3 — Слои приложения согласно принципам чистой архитектуры

В рамках этой архитектуры приложение структурировано таким образом, что бизнес-логика отделена от деталей реализации, что позволяет сосредоточиться на функциональности без необходимости беспокоиться о том, как имен-

но функции будут реализованы в различных средах. Это достигается за счёт четкого разделения на уровни, где каждый уровень отвечает за свою область ответственности [10].

2.6.1 Диаграмма потока данных

Поток данных в приложении инициируется взаимодействием пользователя с графическим интерфейсом, передающим действия в контроллеры слоя представления. Контроллеры перенаправляют запросы в слой бизнес-логики, где данные проходят валидацию и обработку в соответствии с установленными правилами. Бизнес-логика взаимодействует с адаптерами для выполнения операций с базой данных, после чего обработанные данные возвращаются через слои приложения в графический интерфейс для отображения.

Диаграмма потока данных представлена на рисунке 2.4.



Рисунок 2.4 — Диаграмма потока данных

2.6.2 Диаграмма компонентов

Диаграмма компонентов представлена на рисунке 2.5.

2.6.3 Диаграмма классов

Диаграмма классов представлена на рисунке 2.6.

2.7 Вывод

В конструкторской части работы были сформулированы требования к программе, разработаны диаграммы прецедентов и базы данных, определены структуры таблиц с типами данных и ограничениями, формализована модель мероприятия, описаны триггеры для автоматизации процессов, а также представлена архитектура приложения с диаграммами потоков данных, компонентов и классов.

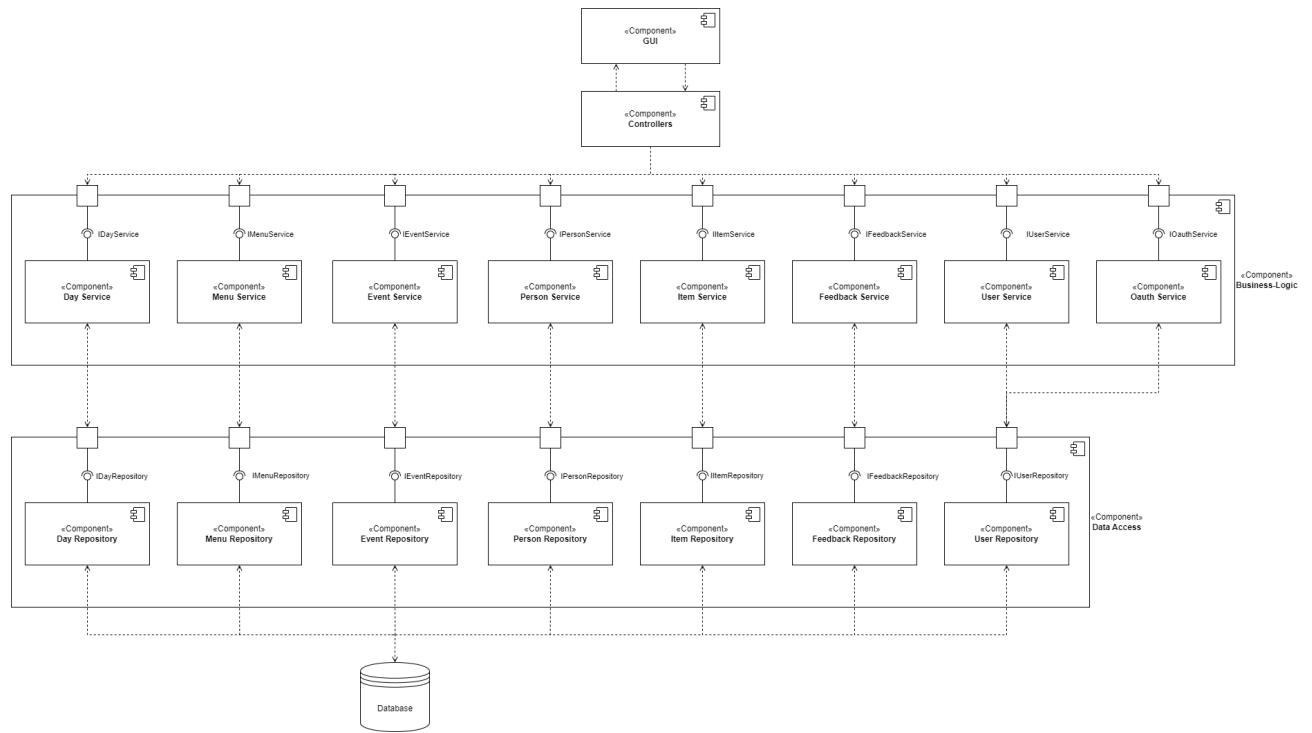


Рисунок 2.5 — Диаграмма компонентов

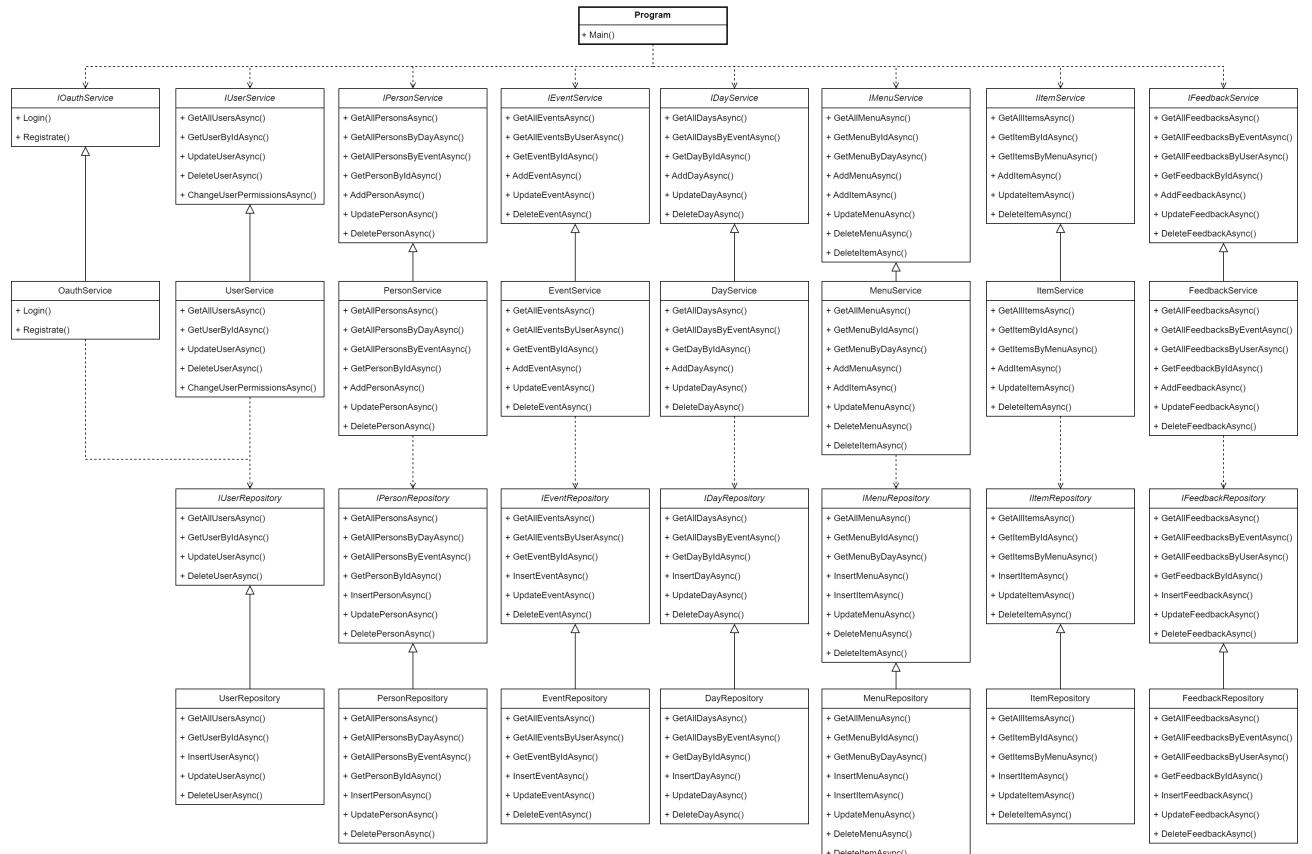


Рисунок 2.6 — Диаграмма классов

3 Технологическая часть

В данной части представляются выбор системы управления базами данных и средств реализации приложения, структуры приложения и базы данных, интерфейс приложения и тестирование.

3.1 Выбор системы управления базами данных

Наиболее распространёнными системами управления реляционными базами данных являются [11]:

- Oracle [12];
- MySQL [13];
- Microsoft SQL Server [14];
- PostgreSQL [15].

Критерии сравнения систем управления реляционными базами данных представлены в таблице 3.1.

Таблица 3.1 — Критерии сравнения систем управления реляционными базами данных

Критерий	Описание	Значения
Бесплатная полная версия	Наличие полностью функциональной версии СУБД без ограничений, доступной для использования без оплаты лицензии	Да, нет
Возможность создавать роли, пользователей и управлять правами	Поддержка механизмов создания пользователей, ролей и тонкой настройки прав доступа к объектам БД на уровне SQL-запросов	Да, нет
Активная поддержка	Наличие официальной технической поддержки от разработчиков или активного сообщества, обеспечивающего регулярные обновления, исправления ошибок и документацию	Да, нет
Полное соответствие стандарту SQL	Строгое соблюдение стандартов SQL без использования расширений или отклонений от спецификаций	Да, нет

Результаты сравнительного анализа систем управления реляционными базами данных представлен в таблице 3.2.

Таблица 3.2 — Сравнительный анализ систем управления реляционными базами данных

Критерий	Oracle	MySQL	Microsoft SQL Server	PostgreSQL
Бесплатная полная версия	Нет	Да	Нет	Да
Возможность создавать роли, пользователей и управлять правами	Да	Да	Да	Да
Активная поддержка	Да	Да	Да	Да
Полное соответствие стандарту SQL	Нет	Нет	Нет	Да

Для достижения поставленной цели была выбрана PostgreSQL благодаря её широкому набору инструментов, который отвечает всем необходимым критериям.

3.2 Выбор средств реализации приложения

В качестве языка программирования для реализации приложения был выбран C# по следующим причинам:

- поддержка объектно-ориентированной методологии программирования;
- присутствие в стандартной библиотеке языка необходимых типов данных, структур и классов, выбранных по результатам проектирования;
- возможность выполнения запросов к базе данных на уровне языка благодаря механизму LINQ [16].

3.3 Структуры приложения и базы данных

Структура приложения соответствует спроектированной логике, отраженной на диаграммах 2.4, 2.5 и 2.6.

Сопоставление типов данных к типам данных выбранной СУБД представлено в таблице 3.3.

Таблица 3.3 — Сопоставление типов данных к типам выбранной СУБД

Тип данных	Тип данных СУБД
Целочисленный	INT
Строковой	VARCHAR(255), TEXT
Вещественный	NUMERIC(14,2)
Логический	BOOL
Перечисляемый	ENUM
Дата	DATE

Структура базы данных соответствует спроектированной логике, отраженной на диаграмме 3.1.

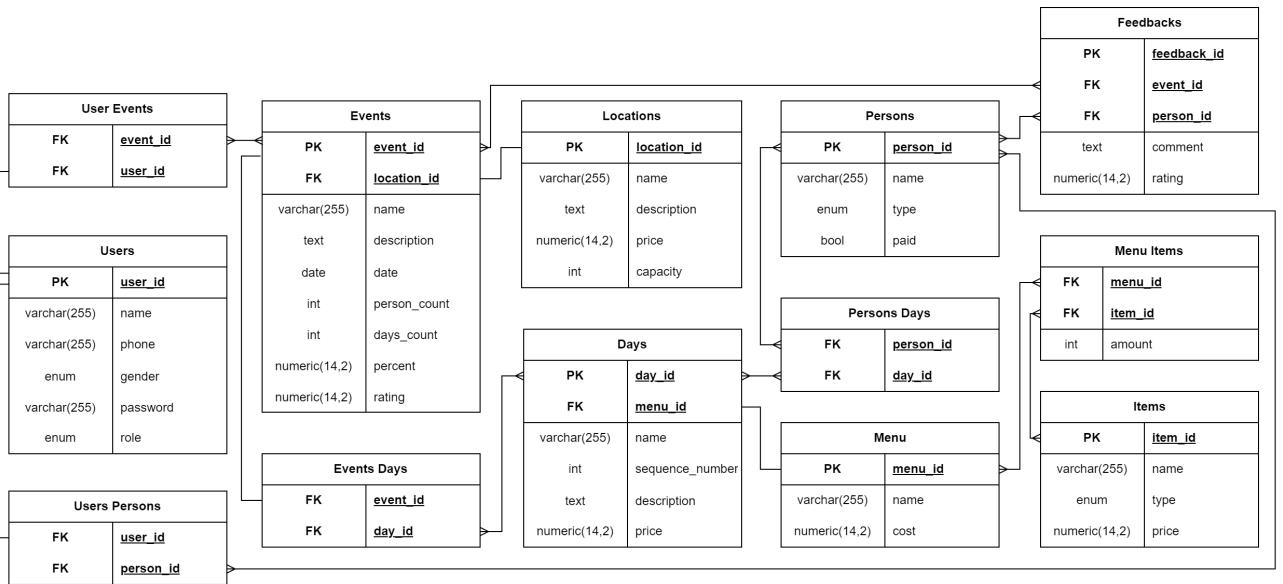


Рисунок 3.1 — Диаграмма базы данных (типы данных СУБД)

Исходный код реализованных приложения и базы данных доступен в открытом сетевом репозитории [17, 18].

3.4 Интерфейс приложения

При запуске приложения пользователю демонстрируется окно «Вход в систему», предназначенную для регистрации и авторизации в системе. Текстовые поля обеспечивают возможность заполнить личные данные.

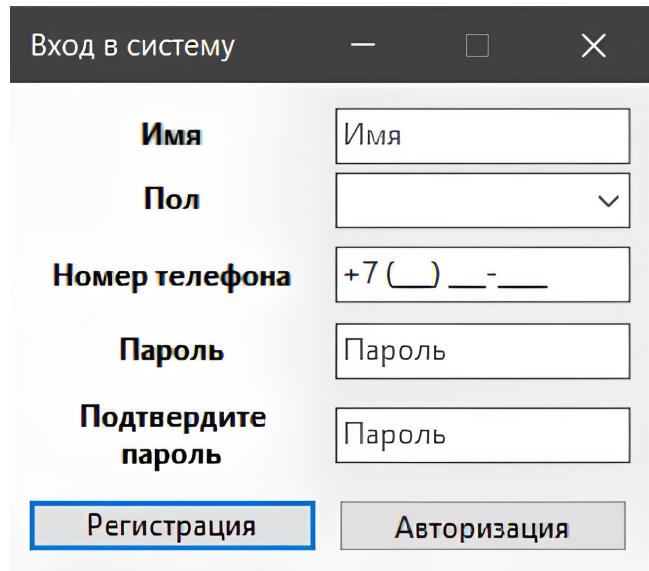


Рисунок 3.2 — Интерфейс окна «Вход в систему»

После успешного входа в систему пользователю демонстрируется главное окно приложения с панелью, содержащую вкладки «Профиль», «Мероприятия» и «Организация». Вкладка «Профиль» содержит данные о пользователе и позволяет редактировать их. Вкладка «Мероприятия» содержит доступные пользователю мероприятия для участия. Вкладка «Организация» содержит организованные пользователем мероприятия.

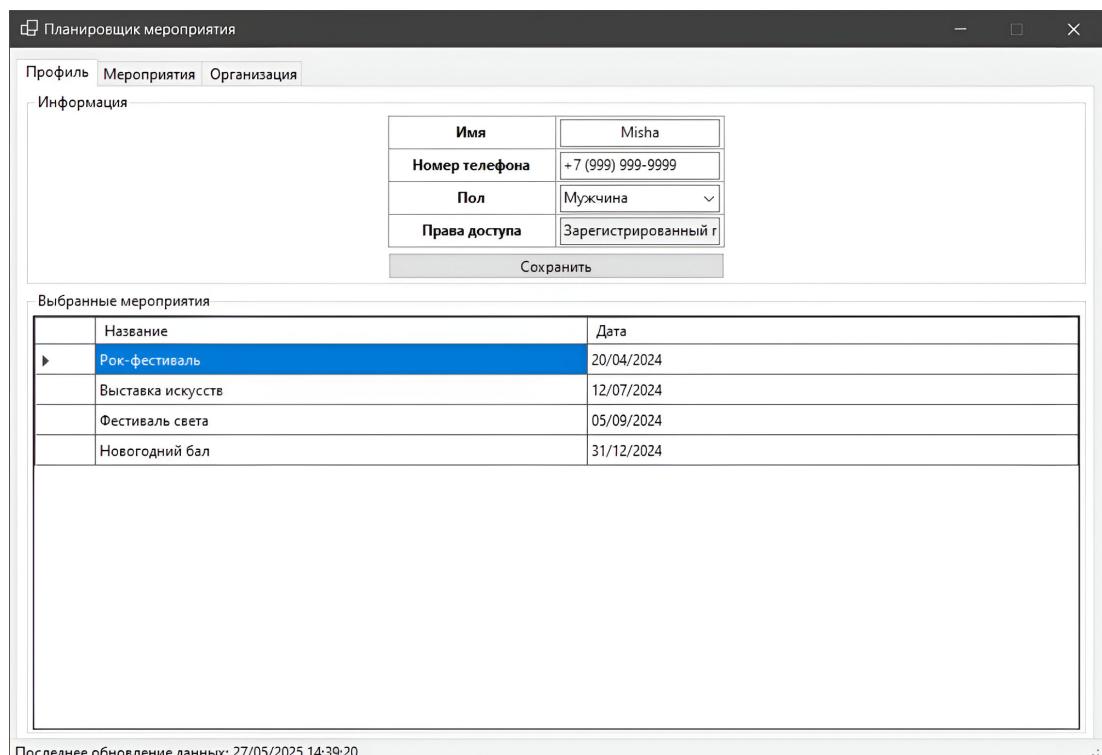


Рисунок 3.3 — Интерфейс главного окна приложения (часть 1)

Планировщик мероприятия

Профиль Мероприятия Организация

Доступные мероприятия

	Название	Описание	Дата	Количество человек	Количество дней	Рейтинг
▶	Технологическая кон...	Конференция по ИИ ...	15/03/2024	6	3	10
	Рок-фестиваль	Крупнейший рок-фе...	20/04/2024	4	2	10
	Гастрономический м...	Мастер-класс от ми...	10/05/2024	4	1	10
	Кинофестиваль	Премьеры мирового...	01/06/2024	5	4	10
	Выставка искусств	Современное искусс...	12/07/2024	4	2	10
	Бизнес-форум	Нетворкинг и инвест...	22/08/2024	5	3	10
	Фестиваль света	Световые инсталляц...	05/09/2024	5	1	10
	Марафон	Городской благотво...	10/10/2024	4	1	10
	Книжная ярмарка	Встречи с авторами ...	18/11/2024	4	2	10
	Новогодний бал	Торжественное меро...	31/12/2024	6	1	10

Последнее обновление данных: 27/05/2025 14:52:05

Рисунок 3.4 — Интерфейс главного окна приложения (часть 2)

Планировщик мероприятия

Профиль Мероприятия Организация

Организованные мероприятия

	Название	Локация	Описание	Дата	Количество человек	Количество дней	Наценка	Рейтинг
▶	Рок-фестиваль	Летняя терраса	Крупнейший р...	20/04/2024	4	2	15	10

Создать мероприятие

Последнее обновление данных: 27/05/2025 14:54:39

Рисунок 3.5 — Интерфейс главного окна приложения (часть 3)

При нажатии на данные о мероприятии, если пользователь не является администратором или организатором мероприятия, ему демонстрируется окно «Информация о мероприятии», содержащее информацию о связанном мероприятии и позволяющее взаимодействовать с ней.

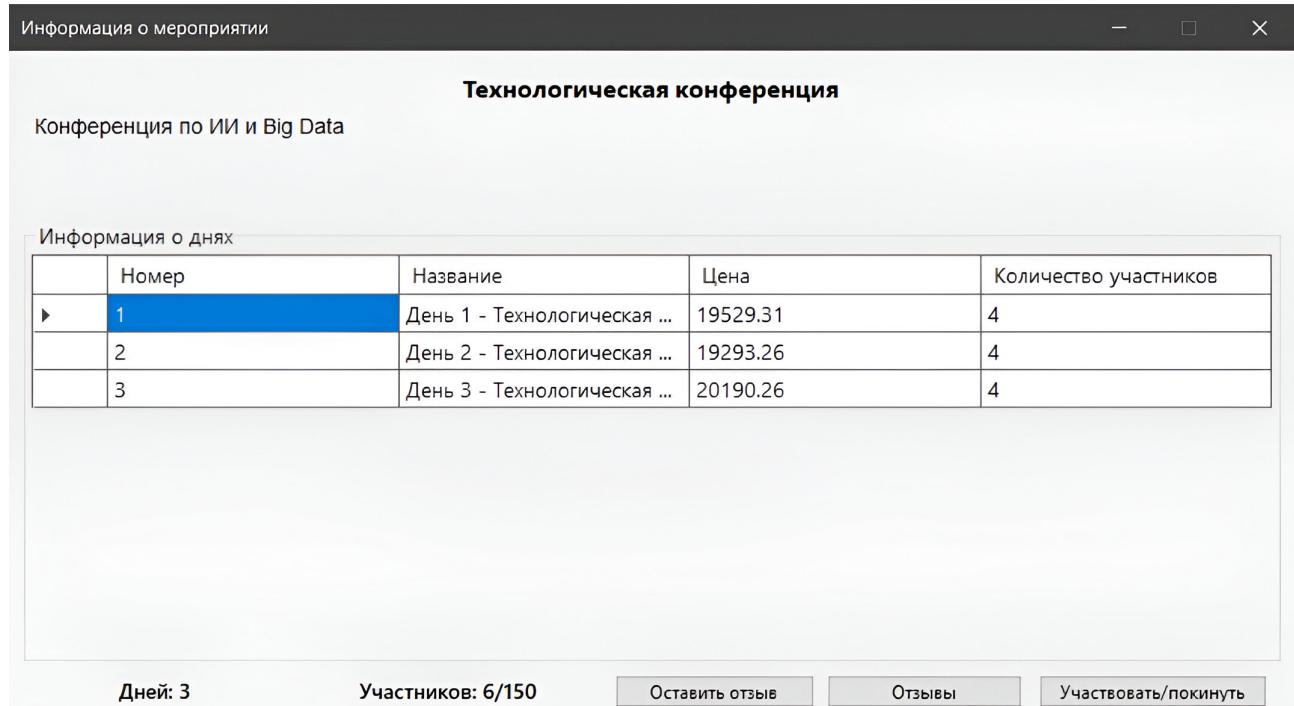


Рисунок 3.6 — Интерфейс окна «Информация о мероприятии»

При нажатии на текстовое поле содержащее количество участников мероприятия пользователю демонстрируется окно «Участники мероприятия», содержащую список участников мероприятия.

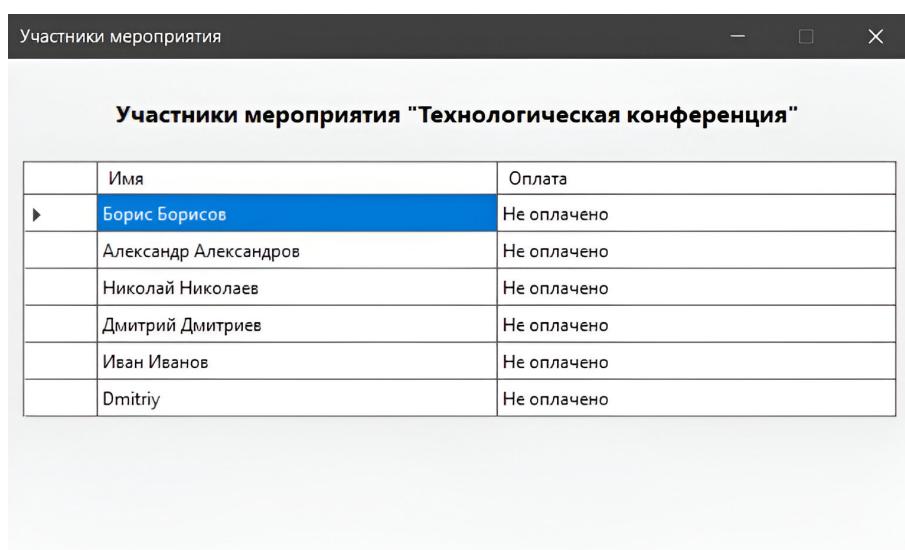


Рисунок 3.7 — Интерфейс окна «Участники мероприятия»

При нажатии на кнопку «Оставить отзыв» пользователю демонстрируется окно «Отзыв», позволяющее создать отзыв о мероприятии.

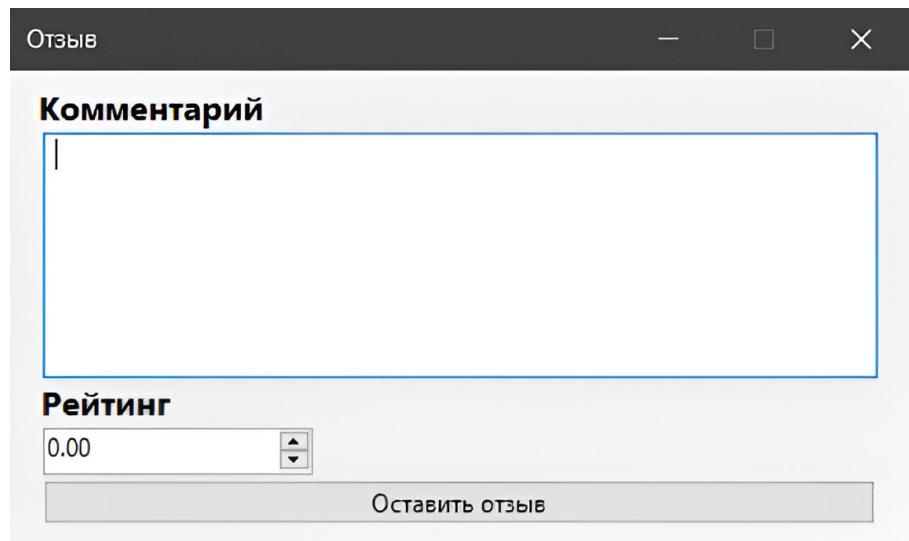


Рисунок 3.8 — Интерфейс окна «Отзыв»

При нажатии на кнопку «Отзывы» пользователю демонстрируется окно «Отзывы мероприятия», содержащее список всех отзывов мероприятия.

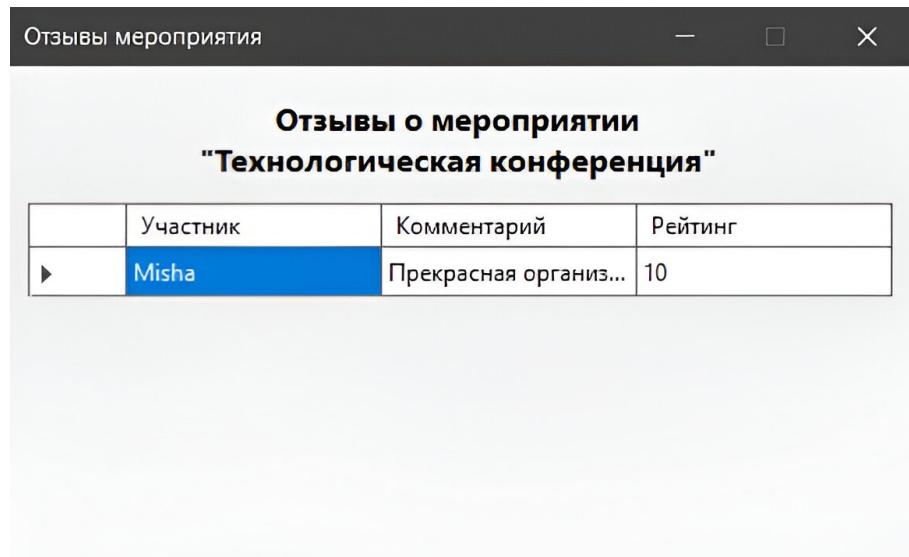


Рисунок 3.9 — Интерфейс окна «Отзывы мероприятия»

При нажатии на кнопку «Участвовать/покинуть» пользователю демонстрируется окно «Дни присутствия», позволяющее участвовать в мероприятии, покинуть его или изменить дни присутствия.

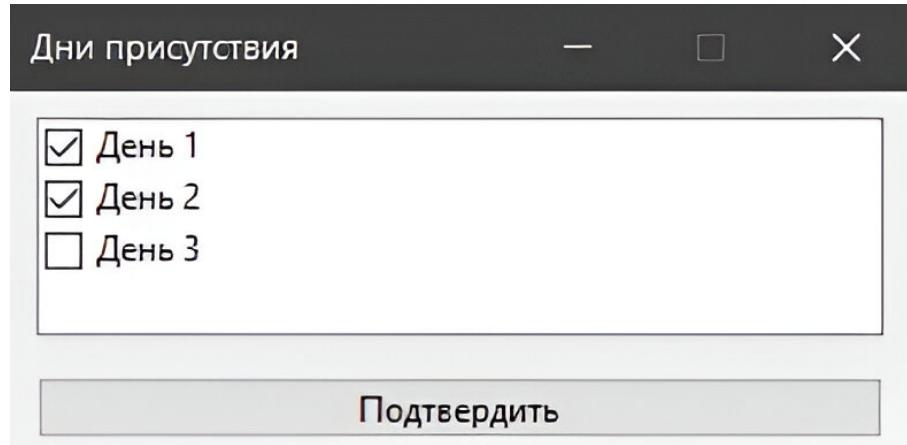


Рисунок 3.10 — Интерфейс окна «Дни присутствия»

При нажатии на данные о дне мероприятия пользователю демонстрируется окно «Информация о дне мероприятия», содержащее всю информацию о связанном дне мероприятия.

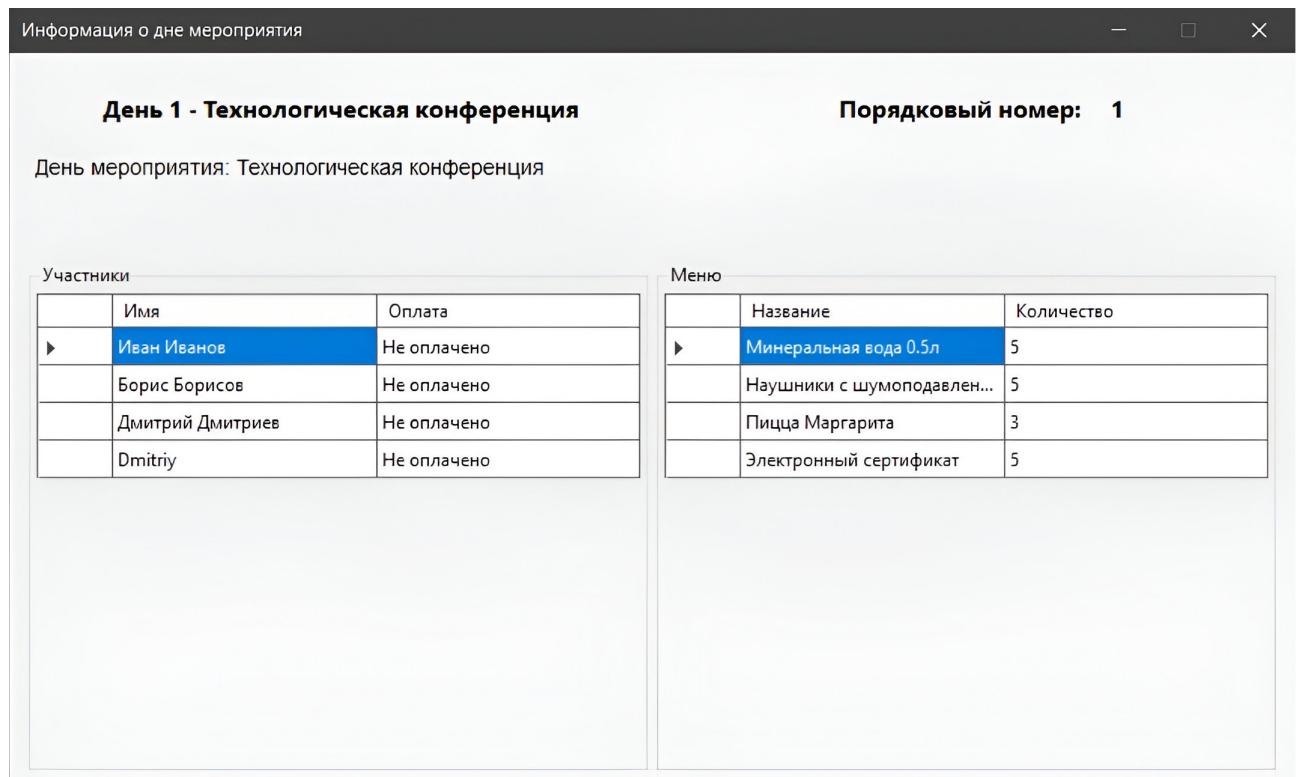


Рисунок 3.11 — Интерфейс окна «Информация о дне мероприятия»

При нажатии на данные о мероприятии, если пользователь является администратором или организатором мероприятия, ему демонстрируется окно «Организация мероприятия», содержащее всю информацию о связанном мероприятии и позволяющее взаимодействовать с ней.

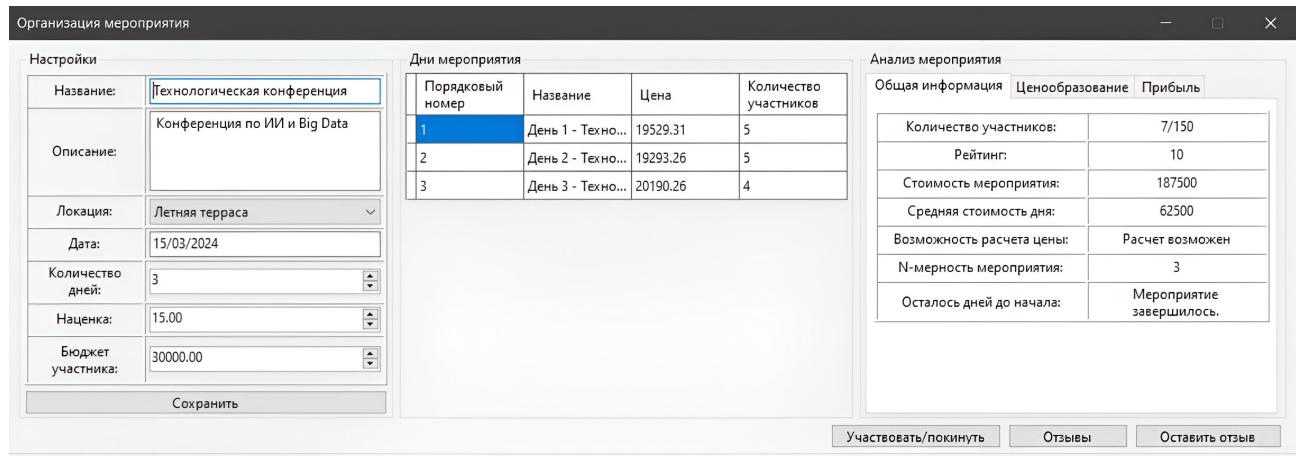


Рисунок 3.12 — Интерфейс окна «Организация мероприятия»

The figure shows three tabs of the 'Analysis of Event' section:

- a) Общая информация (General Information):**

Количество участников:	7/150
Рейтинг:	10
Стоимость мероприятия:	187500
Средняя стоимость дня:	62500
Возможность расчета цены:	Расчет возможен
N-мерность мероприятия:	3
Осталось дней до начала:	Мероприятие завершилось.
- b) Ценообразование (Pricing):**

Фундаментальная цена:	13160.53
Фундаментальная цена (с привилегиями):	16707.25
Относительная разница фундаментальных цен:	26.95%
Средняя цена дня:	15430.88
Средняя цена дня (с привилегиями):	19589.45
Цена мероприятия:	46292.65
Цена мероприятия (с привилегиями):	58768.35
- c) Прибыль (Profit):**

Расходы:	187500
Доход:	215625
Фактическая прибыль:	28125
Теоретическая прибыль:	28125
Максимальная наценка:	71.59%
Минимальное количество участников:	7

Рисунок 3.13 — Вкладки панели в группе «Анализ мероприятия»: общая информация (а), ценообразование (б), прибыль (с)

При нажатии на данные о дне мероприятия пользователю демонстрируется окно «Организация дня мероприятия», содержащее всю информацию о связанном дне мероприятия и позволяющее взаимодействовать с ней.

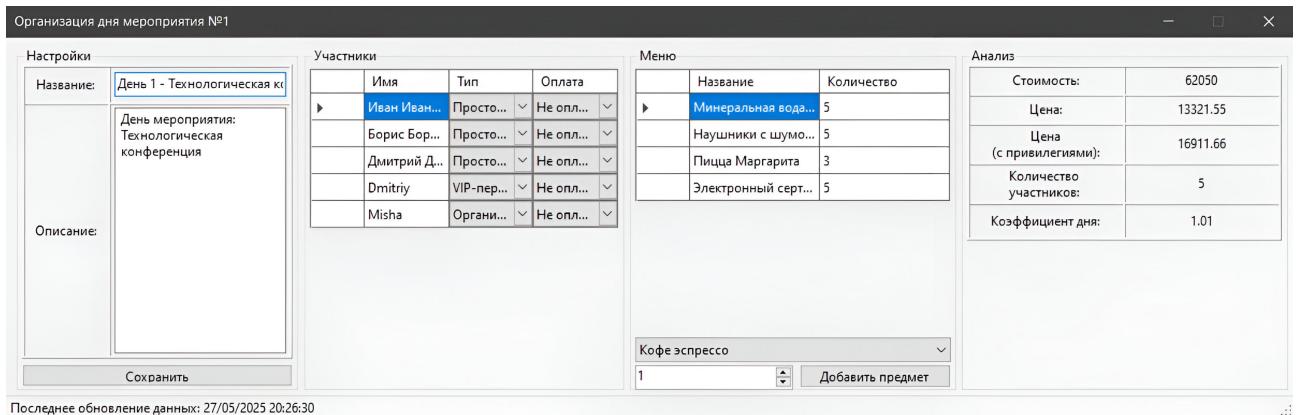


Рисунок 3.14 — Интерфейс окна «Организация дня мероприятия»

3.5 Тестирование

Классы эквивалентности для входа в систему представлены в таблицах 3.4, 3.5, 3.6 и 3.7.

Таблица 3.4 — Классы эквивалентности для поля «Имя»

Класс	Примеры	Ожидаемый результат
Непустая строка	Иван, Anna-Maria, 123	Успешная проверка
Пустая строка	Пустое поле	Ошибка: «Имя обязательно»

Таблица 3.5 — Классы эквивалентности для поля «Пол»

Класс	Примеры	Ожидаемый результат
Корректный пол	Мужской, женский	Успешная проверка
Пустое значение	Не выбран пол	Ошибка: «Пол обязательен»

Таблица 3.6 — Классы эквивалентности для поля «Номер телефона»

Класс	Примеры	Ожидаемый результат
Корректный формат	+7 (999) 123-45-67	Успешная проверка формата
Некорректный формат	+7 (999) 123, пустая строка	Ошибка: «Некорректный номер»
Занятый номер	+7 (919) 345-56-7	Ошибка: «Номер занят»
Несуществующий номер	+7 (919) 345-56-79	Ошибка: «Пользователь не найден»
Уже зарегистрированный номер	Использовать существующий в базе данных номер	Успешная авторизация

Таблица 3.7 — Классы эквивалентности для поля «Пароль»

Класс	Примеры	Ожидаемый результат
Верный пароль	Password123	Успешная авторизация
Пустой пароль	Пустая строка	Ошибка: «Пароль обязательен»
Неверный пароль	WrongPassword	Ошибка: «Неверный пароль»

Все тесты были успешно пройдены.

3.6 Вывод

В данной части были представлены выбор средств реализации и исходный код программы, описаны организация классов в программе и её интерфейс.

4 Исследовательская часть

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Румянцев Д., Франкель Н. Event-маркетинг. Все об организации и продвижении событий [Текст] / Румянцев Д., Франкель Н. — 1-е изд. — Санкт-Петербург: СПб, 2019 — 320 с.
2. Eventbrite [Электронный ресурс]. — URL: <https://www.eventbrite.com/> (дата обращения 15.03.2025)
3. Cvent [Электронный ресурс]. — URL: <https://www.cvent.com/> (дата обращения 15.03.2025).
4. Trello [Электронный ресурс]. — URL: <https://trello.com> (дата обращения 15.03.2025).
5. Peter Pin-Shan Chen. The Entity-Relationship Model — Toward a Unified View of Data / Peter Pin-Shan Chen [Электронный ресурс] // Association for Computing Machinery : [сайт]. — URL: http://portal.acm.org/ft_gateway.cfm?id=320440&type=pdf&coll=GUIDE&dl=GUIDE&CFID=72010946&CFTOKEN=54542835 (дата обращения: 17.03.2025).
6. Мюллер В. К., Англо-русский словарь. 53000 слов. [Текст] / В. К. Мюллер — 1-е изд. — Москва: Рус. яз., 1981 — 887 с.
7. Аврунев О. Е., Стасышин В. М. Модели баз данных: учебное пособие [Текст] / Аврунев О. Е., Стасышин В. М. — 1-е изд. — Новосибирск: НГТУ, 2018 — 123 с.
8. Дейт К. Дж. Введение в системы баз данных [Текст] / Дейт К. Дж. — 8-е изд. — Москва: Издательский дом «Вильяме», 2005 — 1328 с.
9. Мартин Фаулер. Основы UML [Текст] / Мартин Фаулер — 3-е изд. — Санкт-Петербург: Символ-Плюс, 2004 — 192 с.

10. Роберт Мартин. Чистая архитектура. Искусство разработки программного обеспечения [Текст] / Роберт Мартин — 1-е изд. — Санкт-Петербург: Питер, 2024 — 352 с.
11. DB-Engines Ranking / [Электронный ресурс] // <https://db-engines.com> : [сайт]. — URL: <https://db-engines.com/en/ranking> (дата обращения: 25.05.2025).
12. Реляционная СУБД Oracle / [Электронный ресурс] // <https://www.oracle.com> : [сайт]. — URL: <https://www.oracle.com/cis/database/> (дата обращения: 25.05.2025).
13. Реляционная СУБД MySQL / [Электронный ресурс] // <https://www.mysql.com/> : [сайт]. — URL: <https://www.mysql.com/> (дата обращения: 25.05.2025).
14. Реляционная СУБД Microsoft SQL Server / [Электронный ресурс] // <https://www.microsoft.com/> : [сайт]. — URL: <https://www.microsoft.com/ru-ru/sql-server> (дата обращения: 25.05.2025).
15. Реляционная СУБД PostgreSQL / [Электронный ресурс] // <https://www.postgresql.org/> : [сайт]. — URL: <https://www.postgresql.org/> (дата обращения: 25.05.2025).
16. Обзор LINQ / [Электронный ресурс] // <https://learn.microsoft.com> : [сайт]. — URL: <https://learn.microsoft.com/ru-ru/dotnet/standard/linq/> (дата обращения: 25.05.2025).
17. Исходный код приложения / [Электронный ресурс] // <https://github.com> : [сайт]. — URL: https://github.com/Kori-Tamashi/bmstu/tree/course_work_db/course_works/databases/code/Eventor (дата обращения: 25.05.2025).
18. Исходный код базы данных / [Электронный ресурс] // <https://github.com> : [сайт]. — URL: https://github.com/Kori-Tamashi/bmstu/tree/course_work_db/course_works/databases/code/Database (дата обращения: 25.05.2025).

ПРИЛОЖЕНИЕ А