Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Казанский (Приволжский) федеральный университет»

*Институт вычислительной математики и информационных технологий*

**Техническое задание**

Литературные вечера: Проведение литературных вечеров и чтений

Выполнили

обучающиеся 1 курса факультета «Прикладной математики»

Балобанова Арина, Лисюткина Руслана, Назарова Дарья

Казань, 2025

Содержание

[Введение 1](#_Toc195710105)

[Цель 3](#_Toc195710106)

[Краткая сводка возможностей 3](#_Toc195710107)

[Определения, акронимы и сокращения 3](#_Toc195710108)

[Краткое содержание 3](#_Toc195710109)

[Обзор системы 4](#_Toc195710110)

[Определение типов пользователей 4](#_Toc195710111)

[Окружение системы 4](#_Toc195710112)

[Обзор вариантов пользователей 4](#_Toc195710113)

[Нефункциональные требования 5](#_Toc195710114)

[Детальные требования 5](#_Toc195710115)

[Функциональные требования 5](#_Toc195710116)

[Требования к пользовательским интерфейсам 9](#_Toc195710117)

[Требования к логической структуре БД 14](#_Toc195710118)

# Введение

## Цель

Цель документа – предоставить информацию о полном функционале разрабатываемого приложения для организации литературных вечеров и чтений. В документе содержится информация о возможностях системы, сценариях использования (пользовательских сценариях) и структуре системы.

## Краткая сводка возможностей

## Приложение разработано для управления и отслеживании проводимых мероприятий, связанных с литературой. В приложении предоставлены следующие функциональные возможности:

- Добавление новых событий

- Редактирование уже существующих событий

- Фильтрация событий по выбранным категориям

- Удаление событий

- Хранение информации о всех событиях в базе данных

- Генерирование отчетов о проведенных событиях

## Определения, акронимы и сокращения

БД – база данных

API - Интерфейс программирования приложений

.NET – среда разработки приложения

SQL (Structured Query Language) – реляционная база данных

UML (**U**nified **M**odeling **L**anguage) – диаграмма вариантов использования системы

Event – событие, мероприятие

## Краткое содержание

Глава 2 содержит информацию о типах пользователей, окружения системы, обзор вариантов использования системы и нефункциональных требованиях. Глава 3 содержит подробное описание пользовательского интерфейса, информация и структуре базы данных и ее содержании.

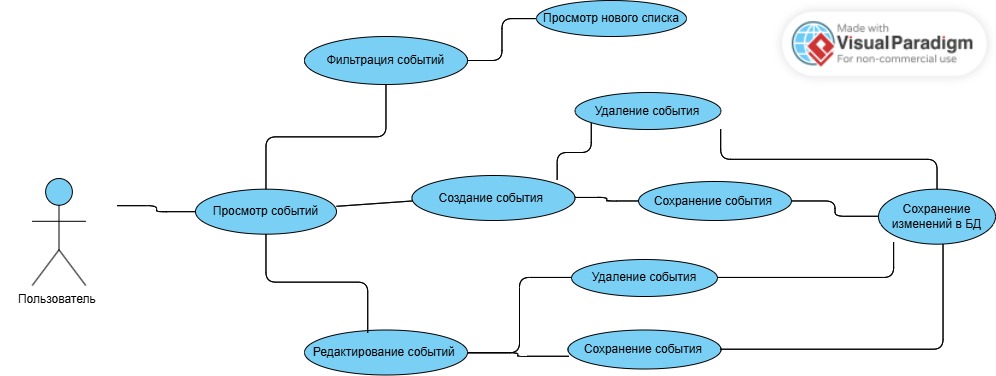
# Обзор системы

## 2.1. Определение типов пользователей

Пользователь: доступ к созданию, редактированию, удалению, просмотру, фильтрации событий и возможность запроса генерации отчетов о мероприятиях.

## 2.2. Окружение системы

UML – диаграмма вариантов использования приложения:



## 2.3. Обзор вариантов пользователей

User Story 1. Фильтрация событий:

Пользователь хочет иметь возможность фильтровать список событий по выбранной категории, чтобы просматривать только интересуемые мероприятия. Например, только литературные вечера, или только чтения.

User Story 2. Создание событий:

Пользователь хочет иметь возможности создавать новые мероприятия, чтобы организовывать литературные мероприятия и собирать компанию единомышленников.

User Story 3. Удаление событий:

Пользователь хочет иметь возможность удалять собственное мероприятие, так как могли возникнуть обстоятельства, препятствующие проведению мероприятия. Например, неожиданная поездка или болезнь. В таком случае мероприятие проводится не будет.

User Story 4. Просмотр событий:

Пользователь хочет иметь возможность просматривать все существующие мероприятия, чтобы участвовать в них или быть зрителем/слушателем на литературных вечерах.

User Story 5. Редактирование событий:

Пользователь хочет иметь возможность редактировать созданное мероприятие, чтобы изменять данные о мероприятии, если это необходимо. К примеру, событие переносится на другую дату или помялась программа.

User Story 6. Генерация отчетов:

Пользователь хочет иметь возможность скачивать отчеты по выбранным категориям в формате excel, чтобы анализировать данные по событиям и лучше понимать интересы аудитории.

## 2.4. Нефункциональные требования

Приложение написано на языке программирования C#.

Используемые инструменты: Visual Studio, Windows Forms, .NET.

Приложение работает с реляционной базой данных SQL.

# Детальные требования

## 3.1. Функциональные требования

User Case 1. Фильтрация событий:

Исполнитель: пользователь

Цель: просмотреть только интересующие события

Сценарий:

1. Пользователь открывает список событий.
2. Система отображает все доступные мероприятия.
3. Пользователь выбирает категорию (например, литературные вечера).
4. Система фильтрует события и показывает только выбранную категорию.



User Case 2. Создание событий:

Исполнитель: пользователь

Цель: создание нового события  
Сценарий:

1. Пользователь нажимает «Создать событие».
2. Система открывает форму для ввода данных.
3. Пользователь заполняет название, описание, дату, категорию и тд.
4. Пользователь нажимает кнопку «Сохранить».
5. Система сохраняет событие и добавляет его в базу данных.



User Case 3. Удаление событий:

Исполнитель: пользователь

Цель: Удаление события, если оно отменено по каким-либо причинам

Сценарий:

1. Пользователь открывает список своих событий.
2. Система показывает созданные мероприятия.
3. Пользователь выбирает событие.
4. Система открывает окно существующего события.
5. Пользователь нажимает кнопку «Удалить событие»
6. Система удаляет событие и вносит соответствующие изменения в базу данных.



User Case 4. Просмотр событий:

Исполнитель: пользователь

Цель: Просмотр всех доступных мероприятий

Сценарий:

1. Пользователь открывает приложение.
2. Система берет из базы данных все существующие мероприятия.
3. Система загружает список всех мероприятий.
4. Пользователь просматривает события.



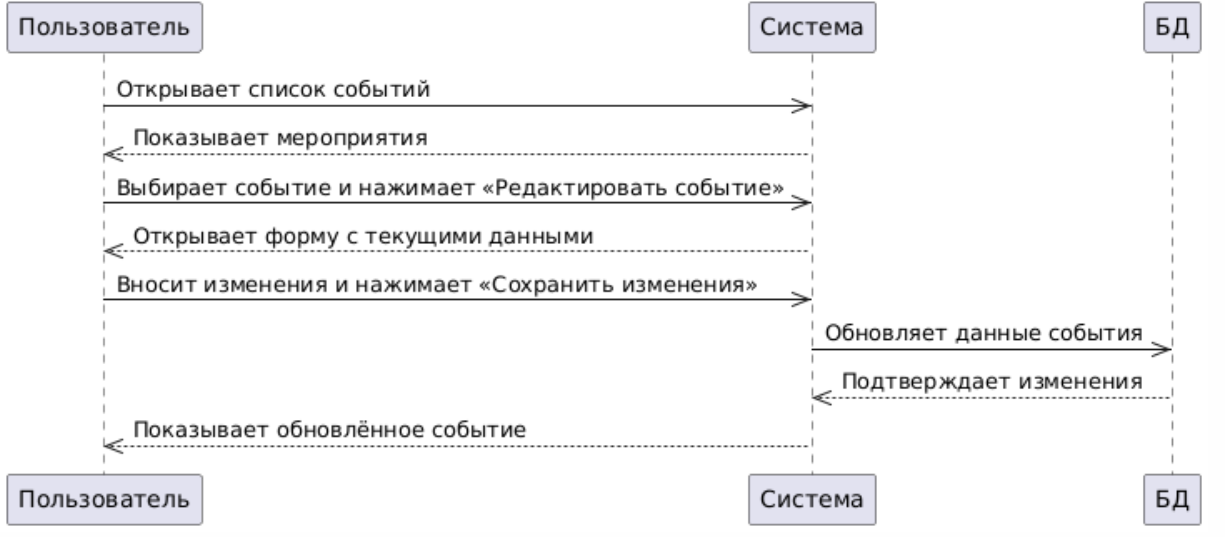
User Case 5. Редактирование событий:

Исполнитель: пользователь

Цель: корректирование информации об уже существующем событии

Сценарий:

1. Пользователь открывает список событий.
2. Система показывает мероприятия.
3. Пользователь выбирает событие и нажимает "Редактировать событие".
4. Система открывает форму с текущими данными.
5. Пользователь вносит изменения и нажимает кнопку «Сохранить изменения».
6. Система сохраняет поправки и вносит изменения в базу данных.



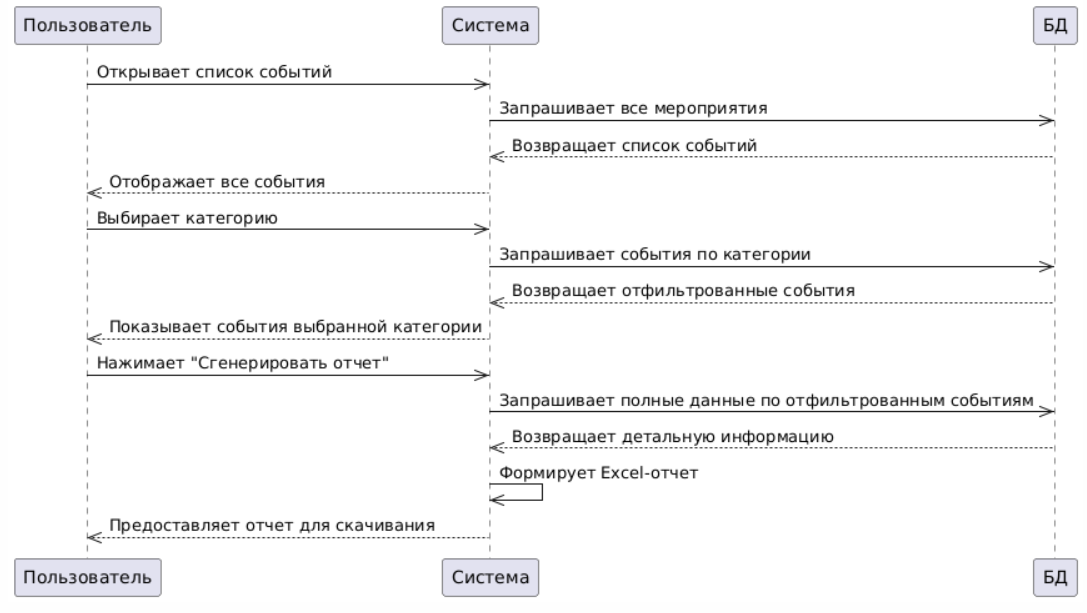
User Case 6. Генерация отчетов:

Исполнитель: пользователь

Цель: скачать отчет о событиях по выбранной категории

Сценарий:

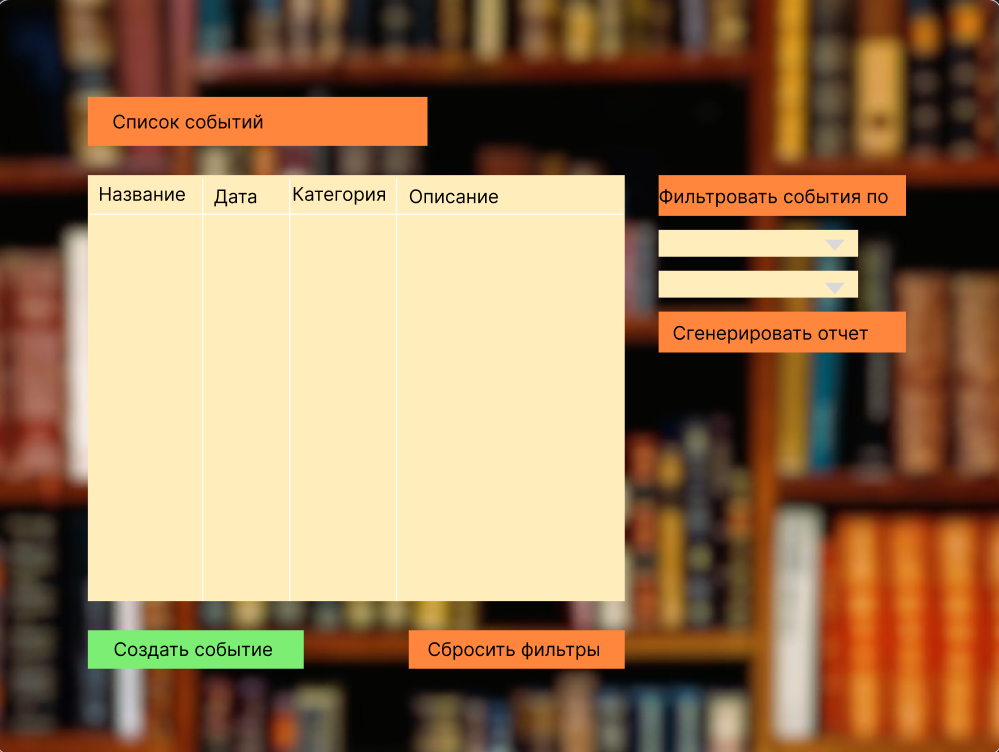
1. Пользователь открывает список событий.
2. Система отображает все доступные мероприятия.
3. Пользователь выбирает нужную ему категорию.
4. Система фильтрует события и показывает только выбранную категорию.
5. Пользователь нажимает кнопку «Сгенерировать отчет»
6. Система берет из БД все данные о требуемых событиях.
7. Система создает документ в формате excel с отчетом о событиях по выбранной категории.



## 3.2. Требования к пользовательским интерфейсам

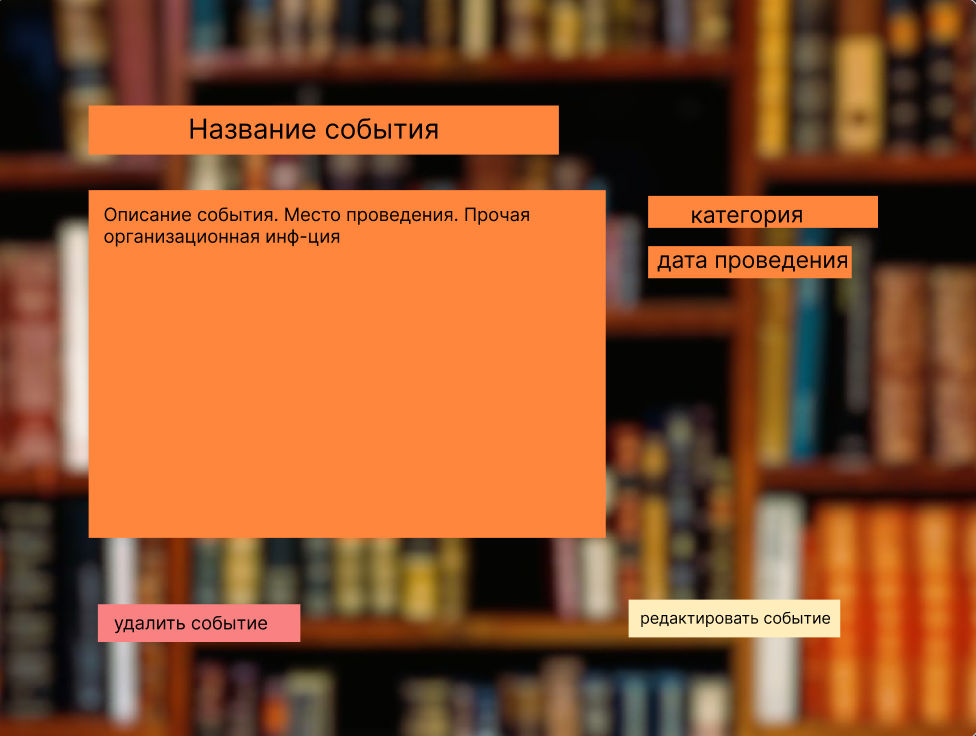
1. Главный экран:

Главный экран – это окно, которое изначально видит пользователь, заходя в приложение. На главном экране представлен список всех событий, который можно отфильтровать, используя кнопку «Фильтровать события по». События можно отфильтровать по категории и по дате. Система отображает нужный пользователю список событий. Кнопка «Сбросить фильтры» отменяет выбранную функцию для сортировки и показывает общий список событий. Кнопка «Сгенерировать отчет» дает возможность пользователю приложения скачивать в формате excel отчеты по событиям, в том числе и отчеты, отфильтрованные по выбранной категории. События в списке кликабельные для возможности просмотреть всю информацию по мероприятию. Кнопка «Создать событие» предоставляет пользователю форму для создания нового мероприятия с указанием всей информации о нем.



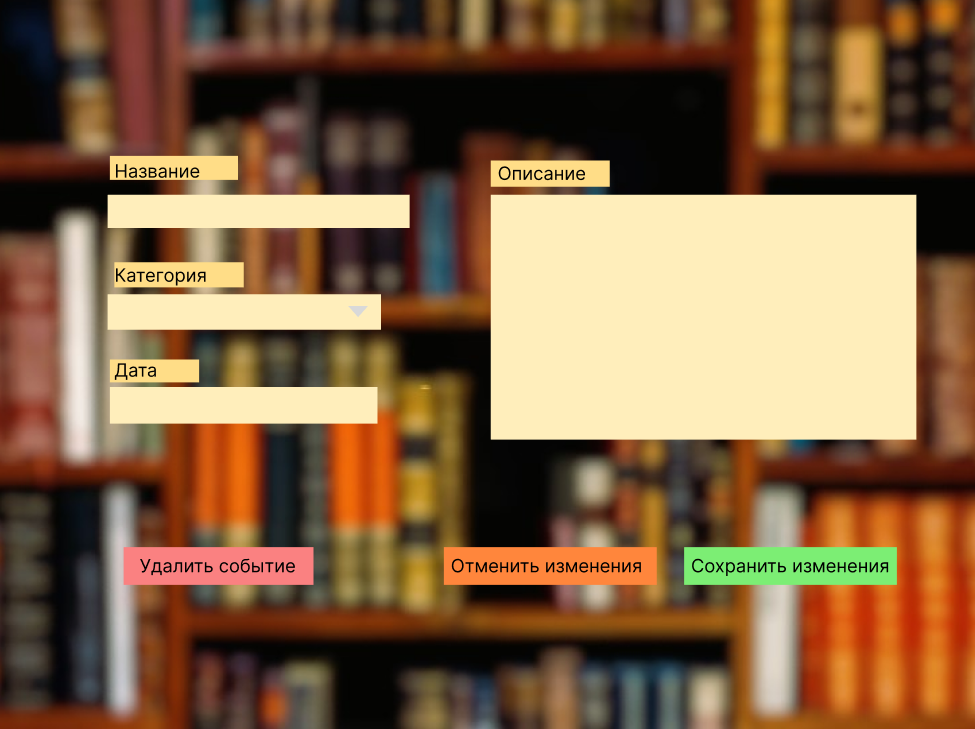
1. Окно существующего события:

Данное окно отображает всю информацию о выбранном событии: название, дату проведения, категорию, описание. При нажатии на кнопку «Редактировать событие» система откроет окно для редактирования события, что позволяет вносить изменения. Удаление существующего события происходит при нажатии кнопки «Удалить событие», которая удаляет событие из списка событий на главном экране.



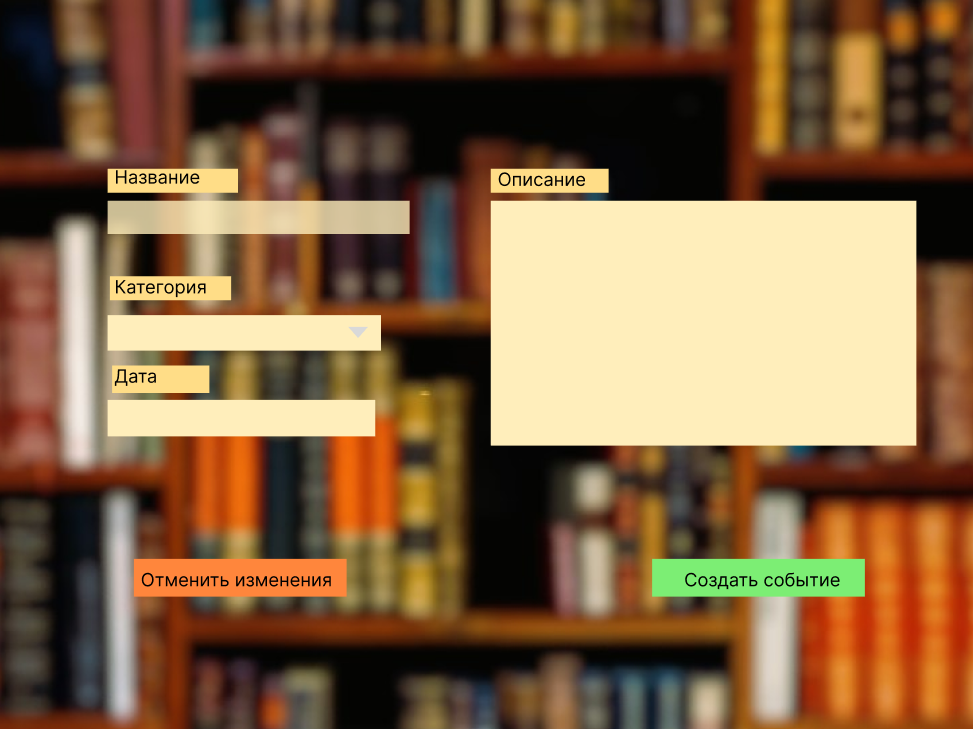
Окно редактирования событий:

Окно редактирования дает возможность корректировать информацию о событии. Например, изменилась дата проведения или участники. Пользователь может внести нужные ему изменения по мероприятию. При нажатии кнопки «Сохранить изменения» все исправления сохраняются и список событий обновляется, отображая актуальную информацию. Также в этом окне можно «Удалить событие», если мероприятие отменилось. Кнопка «Отменить изменения» позволяет пользователю оставить первоначальную информацию о событии без внесенных корректировок.



1. Окно создания события:

Окно создание события открывается при нажатии кнопки «Создать событие» на главном экране. Пользователь должен корректно заполнить все поля, выбрав категорию нового мероприятия и нажать кнопку «Создать событие», чтобы новое событие отобразилось в списке событий на главном экране. Кнопка «Отменить изменения» позволяет отменить создание нового события, если пользователь передумал или нажал кнопку «Создать событие» на главном экране по ошибке.



## Требования к логической структуре БД

|  |  |  |
| --- | --- | --- |
| event\_id | INT (PK) | Уникальный идентификатор. |
| title | TEXT | Название события. |
| description | TEXT | Описание события. |
| date | DATE | Дата события. |
| time | TIME | Время события. |
| category | VARCHAR | Категория события. |
| created\_at | DATE/TIME | Хранит дату и время создания записи в таблице. |
| updated\_at | DATE/TIME | Хранит дату и время последнего обновления записи. |
|  |  |  |