**ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«ТОМСКИЙ ЭКОНОМИКО-ПРОМЫШЛЕННЫЙ КОЛЛЕДЖ»**

***ТЕХНИЧЕСКОЕ ЗАДАНИЕ***

*СПЕЦИАЛЬНОСТЬ 09.02.07 ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ПРОГРАММИРОВАНИЕ*

Выполнили:

Студент группы 1912с

\_\_\_\_\_\_\_\_\_\_\_\_ / Волков Д.П.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.

Проверил:

Руководитель ДП

\_\_\_\_\_\_\_\_\_\_\_\_ / Уляхин В. А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.

Томск 2025

Оглавление

[**ТЕХНИЧЕСКОЕ ЗАДАНИЕ** 3](#_Toc195814487)

[Описание предметной области 3](#_Toc195814488)

[Общее описание проекта 4](#_Toc195814489)

[Ролевая модель 4](#_Toc195814490)

[**Проектирование и разработка базы данных** 5](#_Toc195814491)

[Концептуальная модель данных 5](#_Toc195814492)

[Логическая модель данных 6](#_Toc195814493)

[Физическая модель данных 7](#_Toc195814494)

[Диаграмма прецедентов (Диаграмма вариантов использования) 9](#_Toc195814495)

[Диаграмма деятельности (Диаграмма активностей) 11](#_Toc195814496)

[Диаграмма последовательности 13](#_Toc195814497)

# **ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

## Описание предметной области

Предметная область охватывает деятельность виртуальной библиотеки, предоставляющей доступ к материалам, посвящённым искусственному интеллекту (ИИ) и его эффективному применению. Основная цель библиотеки — предоставить список ИИ и предоставить пользователям возможность обмениваться знаниями, а именно, как практично и эффективно пользоваться ИИ в различных сферах деятельности.

Библиотека объединяет ИИ с различным назначением, обсуждения и промпты написанные пользователями. Важной функцией веб-сайта является организация информации, позволяющая пользователям быстро находить нужные материалы и обмениваться опытом в рамках сообщества.

**Цель**

Разработать и спроектировать веб-сайт "Библиотека ИИ и его эффективное использование", обеспечивающий удобный доступ материалам для изучения применения технологий ИИ, а также предоставляющий платформу для взаимодействия пользователей и обмена опытом.

**Задачи**

1. **Анализ предметной области и требований**
   * Исследовать существующие решения и определить ключевые потребности целевой аудитории.
2. **Проектирование архитектуры веб-сайта**
   * Создать информационную структуру, обеспечивающую интуитивную навигацию.
   * Разработать дизайн пользовательского интерфейса (UI).
3. **Реализация функционала веб-сайта**
   * Разработать разделы веб-сайта, включая каталог ИИ, личный кабинет пользователя, статьи, обсуждения.
4. **Тестирование и доработка веб-сайта**
   * Провести автоматизированное тестирование со стороны сервера на соответствие требованиям, выявить и устранить возможные ошибки.
   * Провести ручное тестирование веб-сайта на соответствие требованиям, выявить и устранить возможные ошибки.

## Общее описание проекта

Все пользователи системы подразделяются на три группы:

1. Администраторы;
2. Пользователи;
3. Гости

Гости могут авторизироваться и регистрироваться. Авторизация производится по токенам, генерируемые при авторизации и отзываемых при выходе. Авторизация на другом устройстве не должна вызывать «вылет» из аккаунта на всех устройствах.

Администратор форума имеет возможность удалить комментарии пользователя, но **не изменять.** Свободно взаимодействовать со статьями форума и обсуждениями (т.е. изменять, удалять информацию), а также создавать новую информацию.

Пользователи получают доступ к некоторым функциям только после успешной авторизации.

Функционал администратора:

1. CRUD операции статей;
2. CRUD операции обсуждений;
3. CRUD операции записей библиотеки ИИ;
4. CRUD операции комментариев (Изменение только своих комментариев).

Функционал пользователя:

1. Редактирование профиля;
2. Просмотр статей;
3. Просмотр обсуждений;
4. Просмотр библиотеки ИИ.
5. Создание обсуждений и смена их статуса
6. Удаление и изменение своих комментариев.

Функционал гостя:

1. Авторизация;
2. Просмотр статей;
3. Просмотр обсуждений;
4. Просмотр библиотеки ИИ.

## Ролевая модель

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Раздел** | **Фичи** | **Админ** | **Пользователь** | **Гость** |
| Навигация | Раздел статьи | да | да | да |
| Раздел обсуждения | да | да | да |
| Раздел библиотека ИИ | да | да | да |
| Профиль | да | да | нет |
| Авторизация | Вход | нет | нет | да |
| Регистрация | нет | нет | да |
| Выход | да | да | нет |
| Библиотека ИИ | Просмотр | да | да | да |
| Удаление | да | нет | нет |
| Редактирование | да | да нет | нет |
| Статьи | Просмотр | да | да | да |
| Удаление | да | нет | нет |
| Редактирование | да | нет | нет |
| Создание | да | нет | нет |
| Создание комментария | да | да | нет |
| Обсуждения | Просмотр | да | да | да |
| Удаление | да | нет | нет |
| Редактирование | да | да | нет |
| Создание | да | да | нет |
| Редактирование (статуса) | да | да | нет |
| Создание комментария | да | да | нет |
| Профиль | Смена фото | да | да | нет |
| Просмотр профиля | да | да | да (не свой) |

# **Проектирование и разработка базы данных**

## Концептуальная модель данных

Концептуальная модель — это формализованное описание предметной области, не связанное с какими-либо компьютерными средствами. Сущности (таблицы) обозначаются прямоугольниками; Атрибуты (поля таблиц) — овалы, которые могут быть объедены в скруглённые прямоугольники; Связи с обозначением мощности — стрелками (в направлении, где мощность равна многим — двойная стрелка, а со стороны, где она равна единице — одинарная).

Анализируя предметную область, можно выделить следующие сущности с их атрибутами (см. рисунок 1):

1. Сущность «Роль» с атрибутами «Название», «Код».
2. Сущность «Пользователь» с атрибутами «Токен», «Почта», «Аватар», «Логин», «Пароль».
3. Сущность «Статья» с атрибутами «Название», «Текст», «Фото», «Описание».
4. Сущность «Категория» (Статья) с атрибутами «Название», «Код».
5. Сущность «Задачи» с атрибутами «Название», «Код».
6. Сущность «Задачи ИИ».
7. Сущность «ИИ» с атрибутами «Название», «Название», «Платный», «Триал», «Преобразование из», «Преобразование в»
8. Сущность «Обсуждение» с атрибутами «»
9. Сущность «Комментарий» с атрибутами «»
10. Сущность «Категория» (Обсуждение) с атрибутами «Название», «Код».

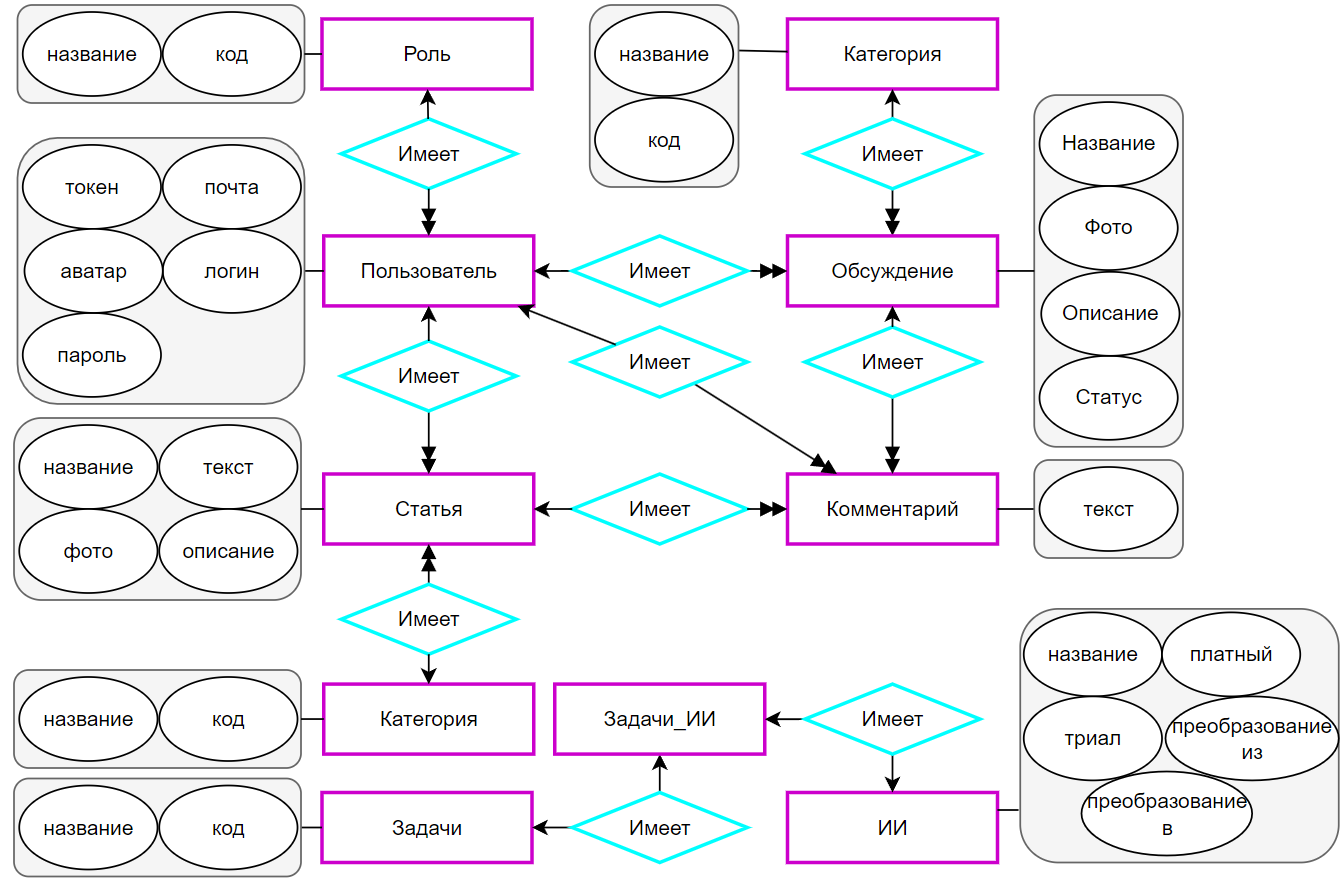


Рисунок 1. Концептуальная модель данных

## Логическая модель данных

Логическая модель базы данных — схема базы данных, выраженная в понятиях модели данных. Этим отличается от концептуальной модели, описывающей семантику предметной области без указания технологии (конкретных методов реализации), и от физической модели, которая описывает конкретные физические механизмы, применяемые для хранения данных в накопителях.

Используя нашу концептуальную модель данных и выполнив нормализацию данных до 3 нормальной формы включительно, построим логическую модель данных (см.рисунок 2):

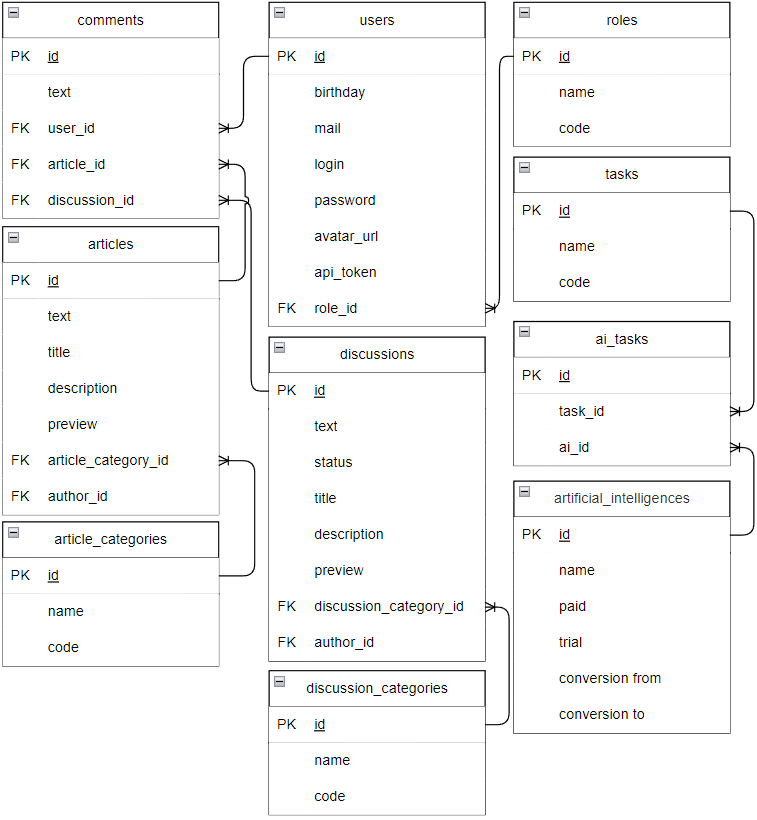


Рисунок 2. Логическая модель данных

## Физическая модель данных

Физическая модель данных — это модель данных, описанная с помощью средств конкретной системы управления базами данных (СУБД). Физическая модель данных строится на базе логической путем добавления особенностей конкретной СУБД. К таким особенностям могут относиться поддерживаемые СУБД типы данных, соглашения о присвоении имен таблицам, атрибутам и т.д. Физическая модель данных фактически является готовым заданием на создание БД, имея которое можно реализовать БД в выбранной СУБД.

Выбранной СУБД является MySQL. Используя логическую модель, построим физическую модель данных (см. рисунок 3):

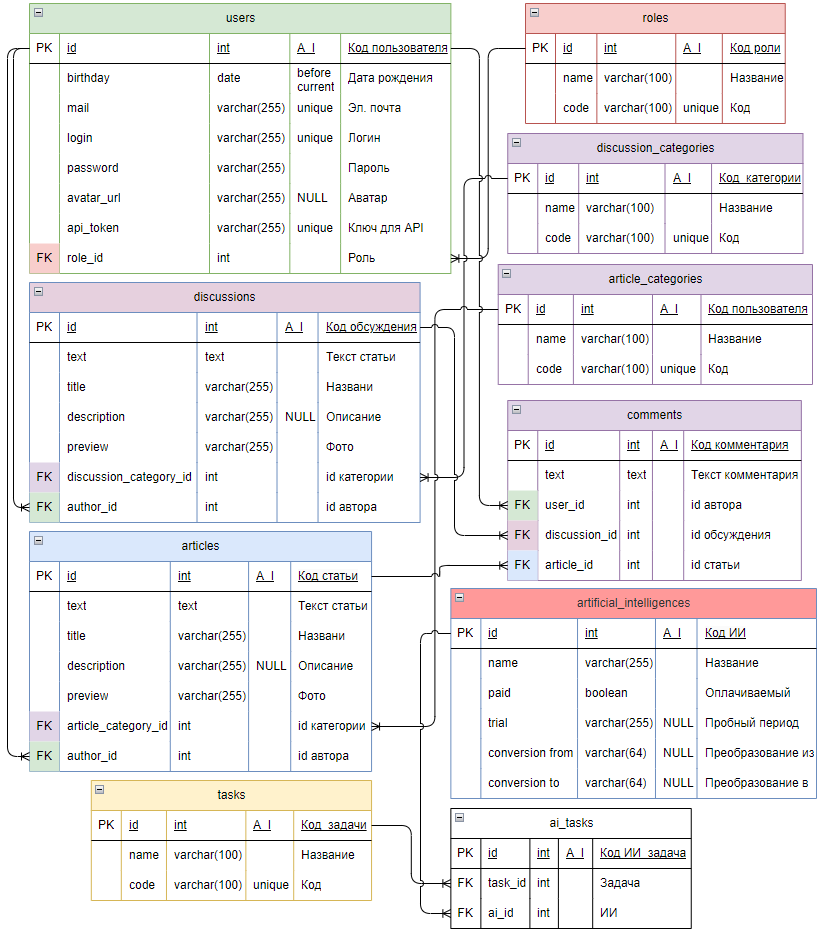


Рисунок 3. Физическая модель данных

## Диаграмма прецедентов (Диаграмма вариантов использования)

Диаграмма прецедентов — это визуальное представление взаимодействия пользователей (акторов) с системой. Она показывает основные функции системы (прецеденты) и то, как акторы их используют. Диаграмма помогает понять, какие действия доступны для каждой роли и как они связаны между собой (см. рисунок 4).

Ключевые элементы и их взаимодействие:

1. Акторы (роли):
   * Гость: Неавторизованный пользователь.
   * Пользователь: Авторизованный пользователь.
   * Администратор: Управляющий системой.
2. Прецеденты (функции):
   * Для гостя: авторизация, регистрация, просмотр библиотеки ИИ, статей и обсуждений.
   * Для пользователя: создание обсуждения, изменение статуса обсуждения, создание комментария и его редактирование, просмотр профиля, смена аватара профиля, выход из системы.
   * Для администратора: управление статьями, обсуждениями, библиотекой ИИ, комментариями, профилем, выход из системы.
3. Взаимодействие:
   * Акторы взаимодействуют с системой через прецеденты.
   * Используются связи include (обязательное включение одного прецедента в другой) и extend (дополнительное расширение функционала).

Гость может просматривать статьи и библиотеку ИИ, но для создания обсуждения или комментария ему нужно стать пользователем через авторизацию или регистрацию.

Администратор может выполнять все функции пользователя, а также управлять контентом (статьями, обсуждениями, записями ИИ).

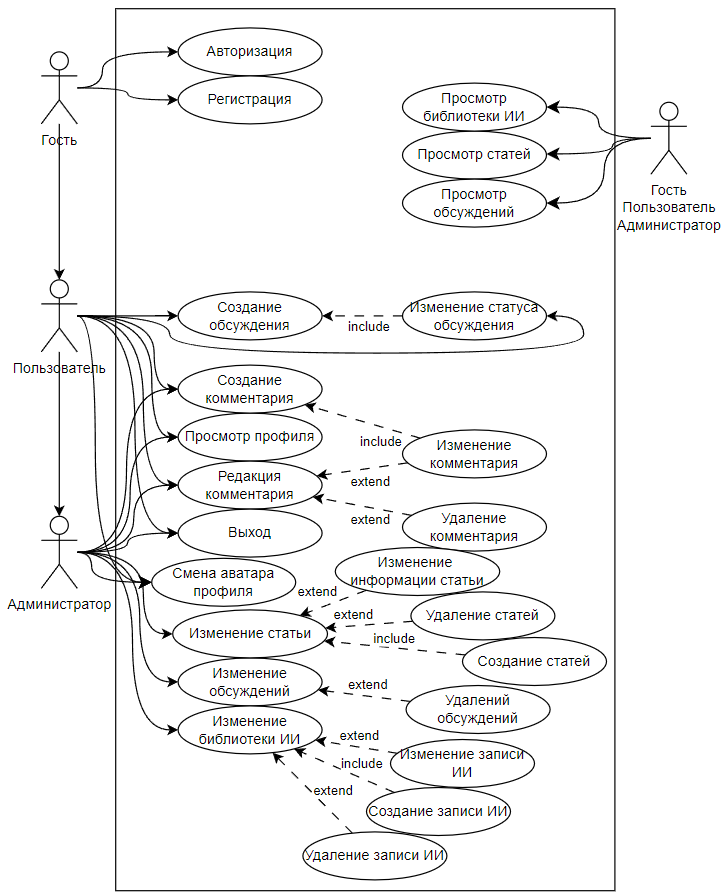


Рисунок 4. Диаграмма прецедентов

## Диаграмма деятельности (Диаграмма активностей)

Диаграмма деятельности (Диаграмма активностей) — это визуальное представление процессов или алгоритмов в системе, которое показывает последовательность действий и поток управления между ними. Она используется для моделирования рабочих процессов, бизнес-процессов или логики выполнения операций в системе.

Диаграммы показывают основные процессы взаимодействия пользователя и системы на сайте. Каждая диаграмма отражает этапы выполнения действий пользователя и отклик системы. Рассмотрим ключевые моменты (см. рисунок 5):

1. **Регистрация**:

* **Пользователь** заполняет необходимые поля в форме регистрации.
* **Система** проверяет валидность введённых данных.
  + Если данные невалидны, выводится сообщение об ошибке.
  + Если данные валидны, система выдаёт токен и перенаправляет на главную страницу.

1. **Авторизация:**

* Схожа с регистрацией, но используется форма авторизации.
* При успешной проверке данных пользователь получает токен и переходит на главную страницу.
* При ошибке выводится уведомление.

1. **Выход:**

* Пользователь нажимает кнопку выхода.
* **Система**:
  + Очищает токен.
  + Перенаправляет на главную страницу.
  + Показывает уведомление о завершении операции.

1. **Создание:**

* Пользователь заполняет необходимые данные для создания объекта (например, книги, записи).
* Если данные валидны, система уведомляет о создании.
* В случае ошибки отображается соответствующее сообщение.

1. **Удаление:**

* Пользователь нажимает на кнопку удаления.
* Система удаляет объект и сообщает об успешном завершении.

1. **Редактирование:**

* Пользователь вводит новые данные для редактирования объекта.
* Если данные валидны, происходит успешное обновление, и система уведомляет об этом.
* В случае ошибки пользователь видит сообщение об ошибке.

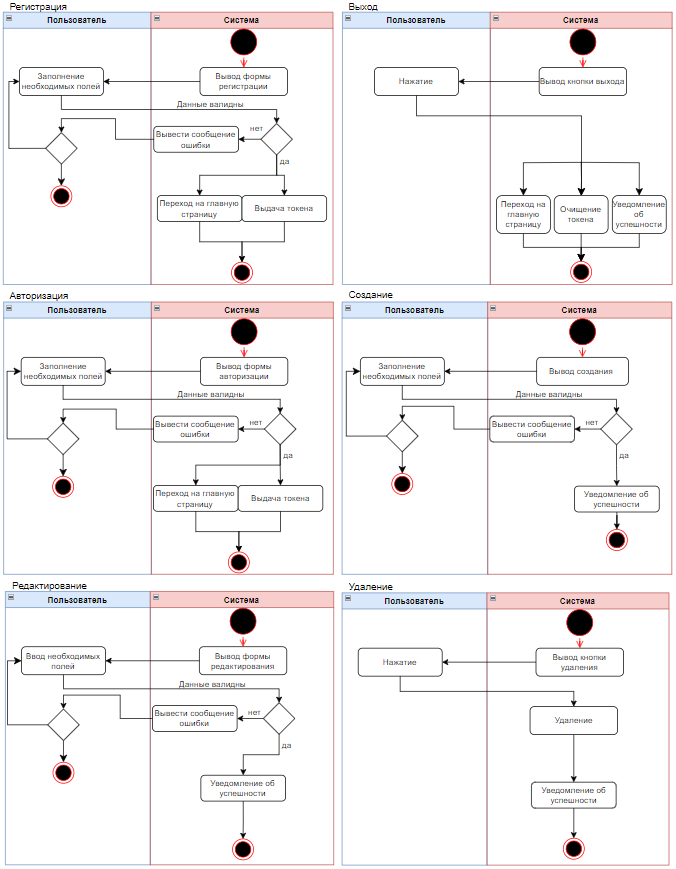


Рисунок 5. Диаграмма деятельности

## Диаграмма последовательности

Диаграмма последовательности — это вид UML-диаграммы, который показывает взаимодействие объектов в системе в рамках конкретного сценария. Она визуализирует порядок вызовов методов и обмен сообщениями между объектами с течением времени.

Диаграммы представляют процессы взаимодействия пользователя, системы и базы данных на сайте "Библиотека ИИ". Каждая диаграмма иллюстрирует последовательность действий и обмен сообщениями между участниками (пользователь, система, компонент, база данных) (см. рисунок 6).

* + - 1. **Регистрация (Диаграмма 1):**
* Пользователь отправляет запрос на регистрацию.
* **Система**:
  + Отображает форму для ввода данных.
  + Проверяет валидность данных через компонент авторизации.
  + Если данные корректны:
    - Генерируется токен, данные пользователя сохраняются в базе данных.
    - Возвращается успешный ответ (201).
  + Если данные некорректны:
    - Выдаётся сообщение об ошибке (422).

1. **Авторизация (Диаграмма 2):**

* Пользователь отправляет запрос на вход с логином и паролем.
* **Система**:
  + Передаёт данные на проверку в компонент авторизации.
  + Проверка успешна:
    - Пользователь аутентифицирован, токен сохраняется, отправляется успешный ответ (200).
  + Если ошибка в данных:
    - Возвращаются коды ошибок (401 — неверный пароль, 422 — некорректный формат данных).

1. **Выход (Диаграмма 3):**

* Пользователь инициирует запрос на выход.
* **Система**:
  + Удаляет токен пользователя из базы данных.
  + Перенаправляет пользователя на главную страницу.
  + Отправляется успешный ответ (200).

1. **Создание объекта (Диаграмма 4):**

* Пользователь отправляет запрос на создание нового объекта (например, книги).
* **Система**:
  + Проверяет валидность данных.
  + Если данные корректны:
    - Объект создаётся, данные сохраняются в базе.
    - Отправляется успешный ответ (201).
  + Если данные некорректны:
    - Отправляется сообщение об ошибке (422).

1. **Редактирование объекта (Диаграмма 5):**

* Пользователь отправляет запрос на редактирование существующего объекта.
* **Система**:
  + Проверяет доступ к объекту.
  + Если пользователь имеет доступ и данные валидны:
    - Изменения сохраняются в базе.
    - Отправляется успешный ответ (200).
  + Если доступ запрещён или данные некорректны:
    - Выдаётся ошибка (403 или 422).

1. **Удаление объекта (Диаграмма 6):**

* Пользователь инициирует запрос на удаление объекта.
* **Система**:
  + Проверяет права доступа.
  + Если доступ разрешён:
    - Объект удаляется из базы данных.
    - Отправляется успешный ответ (204).
  + Если доступ запрещён:
    - Выдаётся ошибка (403).

Ключевые моменты:

1. **Роли участников:**
   * Пользователь инициирует действия.
   * Система обрабатывает запросы и выполняет операции с базой данных через компонент авторизации.
2. **Обработка данных:**
   * Валидация данных — ключевой этап перед сохранением или изменением информации.
   * Ошибки возвращаются с указанием кода (401, 403, 422 и др.).
3. **Права доступа:**
   * Для операций редактирования и удаления система проверяет права пользователя на объект.
4. **Коды ответов:**
   * Успешные операции возвращают коды 200, 201, 204.
   * Ошибки возвращают соответствующие коды (422, 403, 401).
5. **Повторное использование модулей:**
   * Базовые действия (создание, редактирование, удаление) построены универсально, что позволяет их многократное применение для различных объектов сайта.

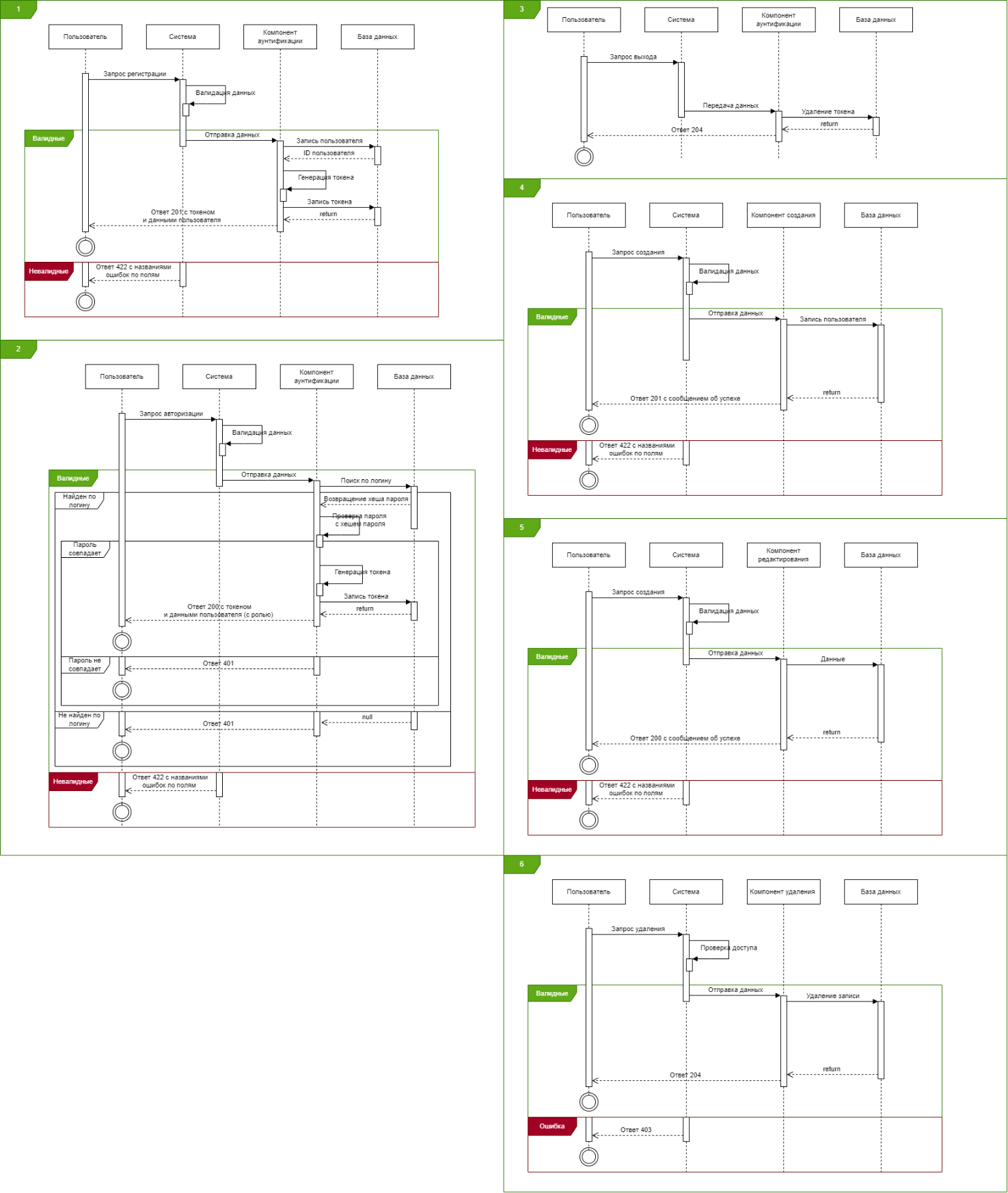


Рисунок 6. Диаграмма последовательности