|  |  |  |
| --- | --- | --- |
|  |  | УТВЕРЖДАЮ |
|  |  | Руководитель  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.  М.П. |

ТЕХНИЧЕСКИЙ ПРОЕКТ

сайта «Онлайн доска

2024

Содержание

[1 Уточнение структур данных 2](#_Toc179812259)

[2 Формы и их пердставления 5](#_Toc179812260)

[3 Разработка алгоритма решения задачи 8](#_Toc179812261)

[4 Определения языка 15](#_Toc179812262)

[5 Структура программы и требования к техническим средствам 16](#_Toc179812263)

# Уточнение структур данных

У сайта «Онлайн доска» должна быть база данных. База данных для сайта «Онлайн доска» изображена в логической ER диаграмме и физической ER диаграмме они в свою очередь изображены на рисунках 1 и 2.

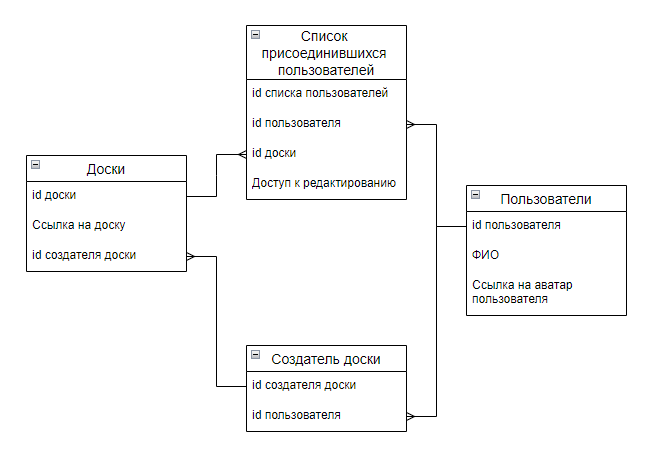


Рисунок 1 – Логическая ER диаграмма

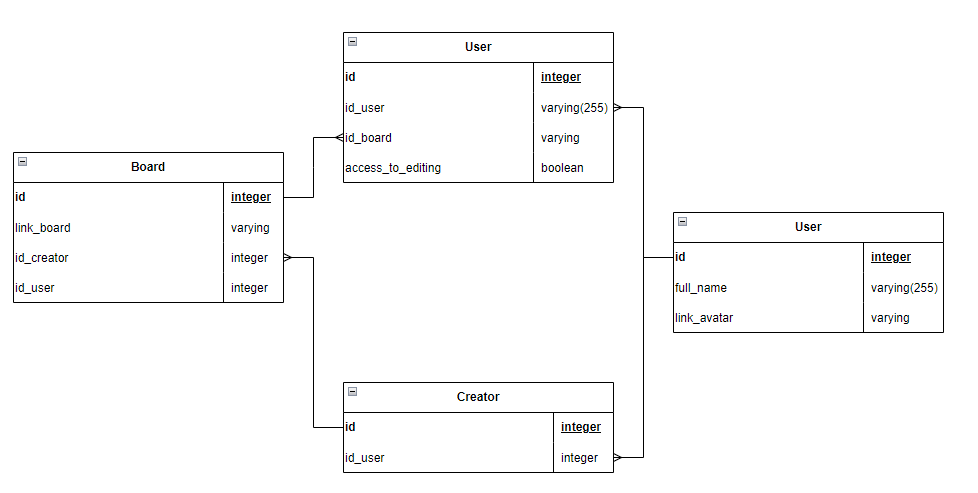


Рисунок 2 – Физическая ER диаграмма

На ER диаграмма показано что есть 3 таблицы: «Доски», «Создатель доски» и «Пользователи».

Таблица «Доски» содержит следующие данные:

* id доски: Идентификатор доски.
* Ссылка на доску: URL-адрес, по которому можно найти доску.
* id создателя доски: ссылается на пользователя, который создал доску (внешний ключ к таблице «Создатель доски»).
* id присоединившихся пользователей: список идентификаторов пользователей, которые присоединились к доске (внешний ключ к таблице «Пользователи»).

Таблица «Создатель доски»

* id создателя доски: Уникальный идентификатор создателя доски.
* id пользователя: Ссылка на пользователя, который создал доску (внешний ключ к таблице «Пользователи»).

Таблица «Пользователи»

* id пользователя: Уникальный идентификатор пользователя.
* ФИО: Полное имя пользователя.
* Ссылка на аватар пользователя: URL-адрес изображения аватара пользователя.

Таблица «Список присоединившихся пользователей»

* id списка пользователей: Уникальный идентификатор для списка пользователей, которые присоединились к конкретной доске.
* id пользователя: Ссылка на таблицу «Пользователи», которая идентифицирует присоединившегося пользователя.
* id доски: Ссылка на таблицу «Доски», указывающая на доску, к которой пользователь присоединился.
* Доступ к редактированию: разрешение, которое указывает, имеет ли пользователь права на редактирование доски.

Связи:

Связь между «Доски» и «Создатель доски»: Каждая доска связана с создателем через поле id создателя доски, которое является внешним ключом, ссылающимся на таблицу «Создатель доски».

Связь между «Создатель доски» и «Пользователи»: Таблица «Создатель доски» через поле id пользователя связана с таблицей «Пользователи», что позволяет идентифицировать, какой пользователь создал доску.

Связь между «Доски» и «Список присоединившихся пользователей»: Таблица «Список присоединившихся пользователей» через поле id доски ссылается на таблицу «Доски», что позволяет отследить, к какой доске присоединился пользователь.

Связь между «Список присоединившихся пользователей» и «Пользователи»: Поле id пользователя в таблице «Список присоединившихся пользователей» ссылается на таблицу «Пользователи», что позволяет определить, кто присоединился к конкретной доске.

# Формы и их пердставления

Прототип главной страницы изображена на рисунке 3.

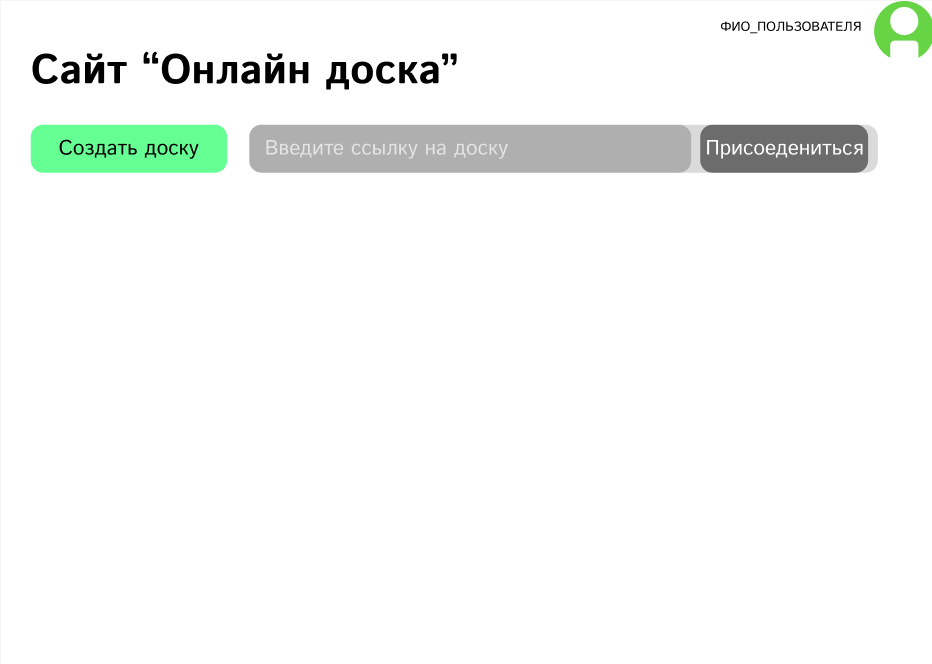


Рисунок 3 – Прототип главной страницы

На этом прототипе показано что пользователь может создать доску или же присоединиться к уже созданной доске.

Прототип доски указан изображен на рисунке 4.

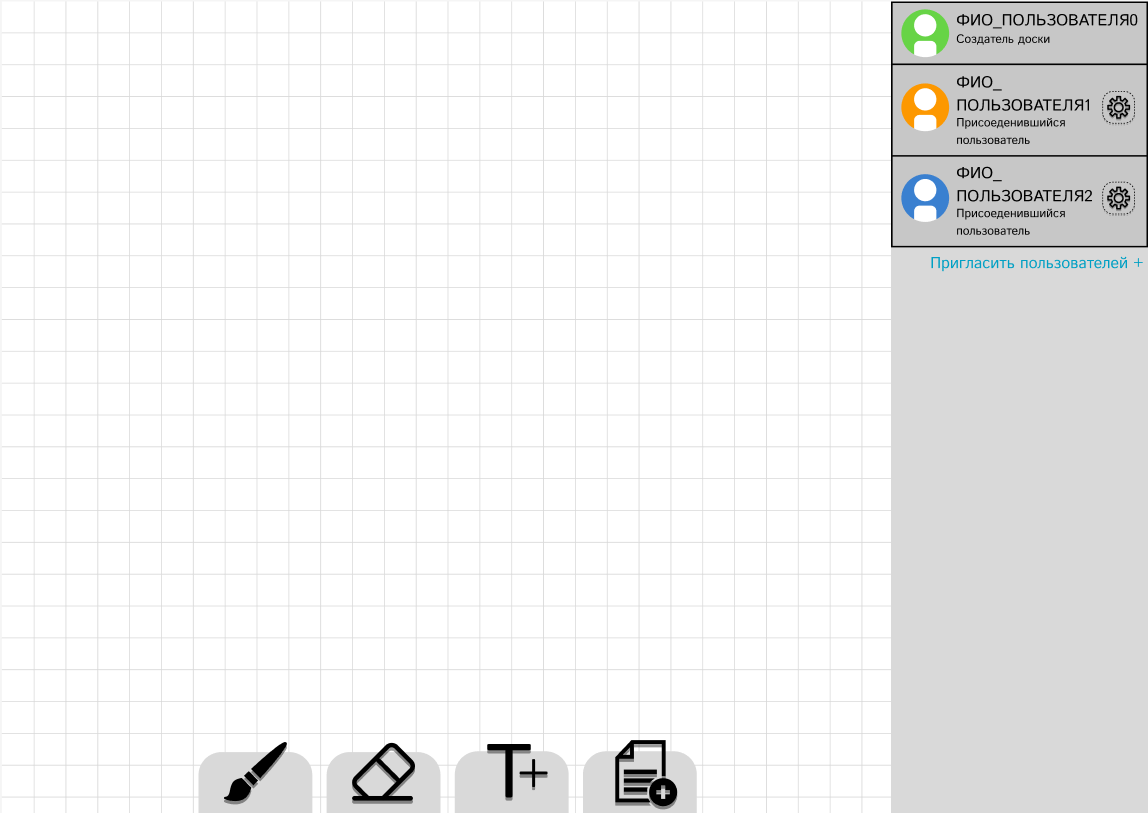


Рисунок 4 – Прототип доски

На это прототипе показано что пользователь может выбрать следующие инструменты для редактирования доски:

* «Кисть» - для рисования кривых линий
* «Ластик» - для стирания уже нарисованного материала
* «Добавление текста» - для добавления текста на доску
* «Добавление изображения» - для добавления изображения формата «.PNG» и «.JPG» на доску

Так же создатель доски может пригласить пользователей на свою доску, это показано на прототипе приглашения пользователей он же в свою очередь изображен на рисунке 5.

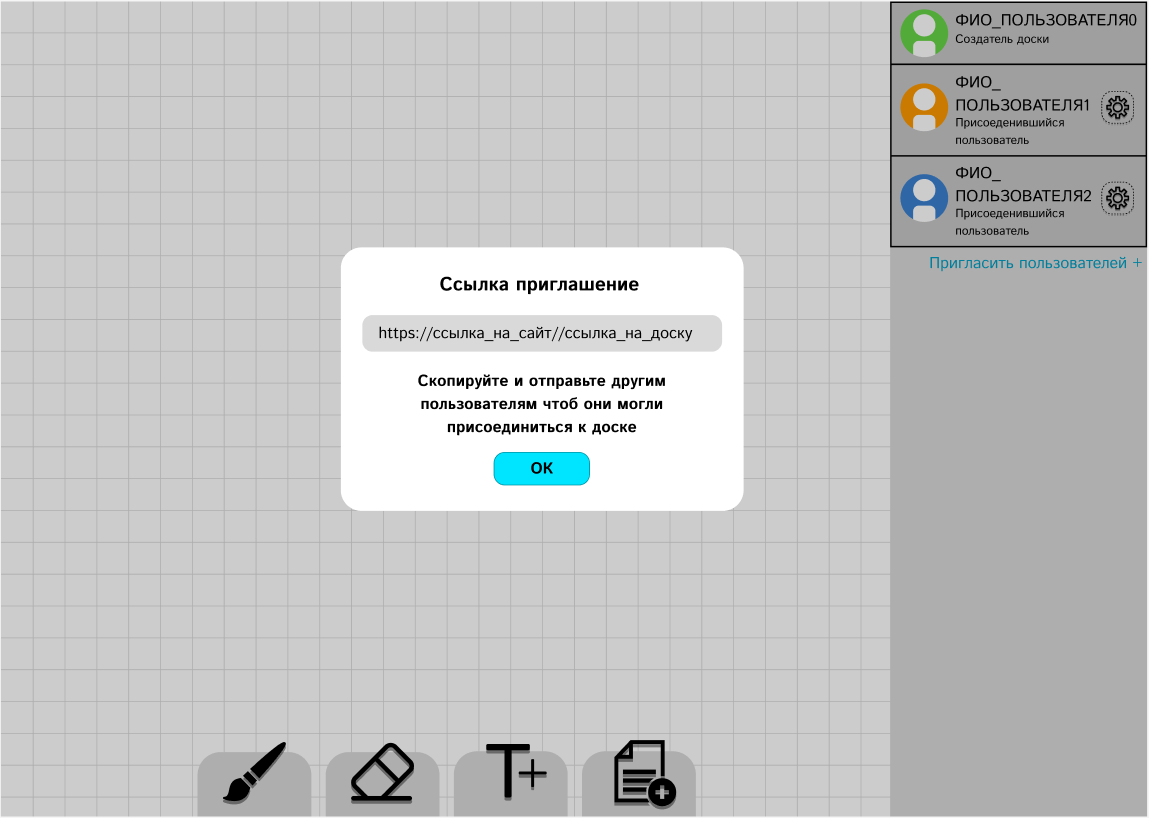


Рисунок 5 – Прототип приглашения пользователей

Так же создатель доски может управлять доступом другим пользователям, это показано на прототипе управление пользователями который изображен на рисунке 6.

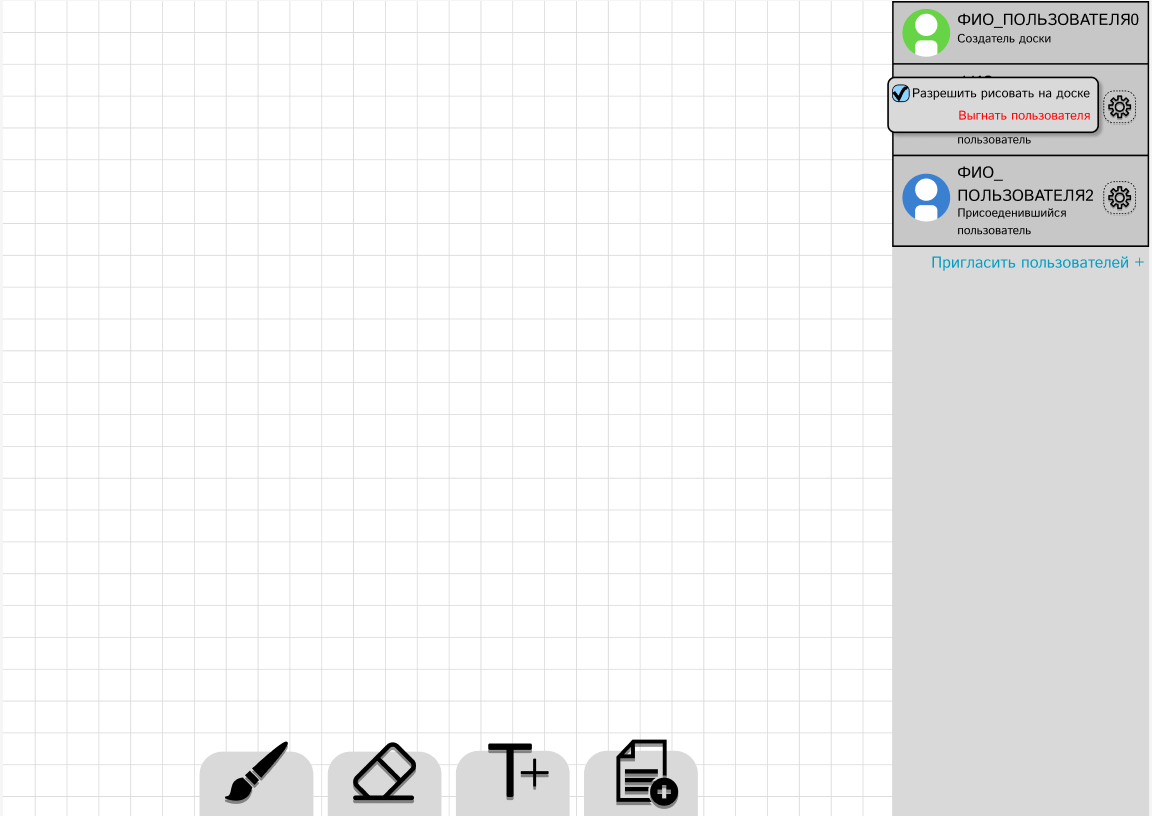


Рисунок 6 – Прототип управление пользователями

# Разработка алгоритма решения задачи

Алгоритм: **Создание новой доски**

Входные данные:

* Название доски
* Настройки доступа (публичная или приватная)

Выходные данные:

* Идентификатор созданной доски
* URL для доступа к доске

Шаги алгоритма:

* Получить данные о новой доске (название, настройки доступа) от пользователя.
* Сгенерировать уникальный идентификатор доски.
* Сохранить информацию о доске в базе данных.
* Создать начальное состояние доски (пустой холст).
* Вернуть идентификатор доски и URL для доступа к ней.

Обработка ошибок:

* Если название доски отсутствует или некорректно, вернуть ошибку валидации.
* При ошибке сохранения в базе данных, вернуть ошибку сервера.

Алгоритм: **Управление доступом к доске**

Входные данные:

* Идентификатор доски
* Список пользователей и их ролей

Выходные данные:

* Обновленная матрица доступа

Шаги алгоритма:

* Получить запрос на изменение прав доступа от создателя доски.
* Проверить права создателя на управление доступом.
* Обновить матрицу доступа для указанных пользователей и ролей.
* Сохранить изменения в базе данных.
* Отправить уведомления затронутым пользователям об изменении их прав доступа.

Обработка ошибок:

* Если создатель не имеет прав на управление доступом, вернуть ошибку доступа.
* При ошибке сохранения в базе данных, вернуть ошибку сервера.

Алгоритм: **Синхронизация состояния доски в реальном времени**

Входные данные:

* Идентификатор доски
* Обновления состояния доски от клиентов

Выходные данные:

* Обновленное состояние доски для всех клиентов

Шаги алгоритма:

* Получить обновления состояния доски от клиентов.
* Проверить права пользователей на внесение изменений.
* Объединить полученные обновления в единое состояние доски.
* Сохранить обновленное состояние в базе данных.
* Отправить обновленное состояние всем подключенным клиентам.

Обработка ошибок:

* Если пользователь не имеет прав на внесение изменений, игнорировать его обновления.
* При конфликте обновлений, применить стратегию разрешения конфликтов (например, на основе временных меток).
* При ошибке сохранения в базе данных, повторить попытку сохранения.

Алгоритм: **Рисование на доске**

Входные данные:

* Координаты точек (x, y)
* Параметры инструмента (цвет, толщина)

Выходные данные:

* Обновленное состояние доски

Шаги алгоритма:

* Получить координаты точек и параметры инструмента от клиента.
* Проверить, имеет ли пользователь права на рисование.
* Если права есть, добавить точки в буфер рисования.
* Отправить обновленное состояние буфера всем подключенным клиентам.
* Сохранить изменения в базе данных.
* Вернуть обновленное состояние доски клиенту.

Обработка ошибок:

* Если пользователь не имеет прав на рисование, вернуть ошибку доступа.
* Если координаты точек некорректны, игнорировать их.
* При сбое сети, повторить отправку обновлений клиентам.

Алгоритм: **Стирание на доске**

Входные данные:

* Координаты точек (x, y)
* Размер ластика

Выходные данные:

* Обновленное состояние доски

Шаги алгоритма:

* Получить координаты точек и размер ластика от клиента.
* Проверить, имеет ли пользователь права на стирание.
* Если права есть, определить область стирания на основе координат и размера ластика.
* Удалить все точки и линии, попадающие в область стирания.
* Отправить обновленное состояние доски всем подключенным клиентам.
* Сохранить изменения в базе данных.
* Вернуть обновленное состояние доски клиенту.

Обработка ошибок:

* Если пользователь не имеет прав на стирание, вернуть ошибку доступа.
* Если координаты точек некорректны, игнорировать их.
* При сбое сети, повторить отправку обновлений клиентам.

Алгоритм: **Добавление текста на доску**

Входные данные:

* Координаты точки (x, y)
* Текстовое содержимое
* Параметры текста (шрифт, размер, цвет)

Выходные данные:

* Обновленное состояние доски

Шаги алгоритма:

* Получить координаты точки, текстовое содержимое и параметры текста от клиента.
* Проверить, имеет ли пользователь права на добавление текста.
* Если права есть, создать текстовый элемент с заданными параметрами.
* Добавить текстовый элемент на доску в указанной точке.
* Отправить обновленное состояние доски всем подключенным клиентам.
* Сохранить изменения в базе данных.
* Вернуть обновленное состояние доски клиенту.

Обработка ошибок:

* Если пользователь не имеет прав на добавление текста, вернуть ошибку доступа.
* Если координаты точки некорректны, вернуть ошибку валидации.
* Если текстовое содержимое пустое или слишком длинное, вернуть ошибку валидации.
* При сбое сети, повторить отправку обновлений клиентам.

Алгоритм: **Выгнать пользователя с доски**

Входные данные:

* Идентификатор доски
* Идентификатор пользователя, которого нужно выгнать

Выходные данные:

* Статус операции (успех или ошибка)

Шаги алгоритма:

* Получить идентификатор доски и идентификатор пользователя.
* Проверить, является ли текущий пользователь создателем доски или имеет ли он права на выгон других пользователей.
* Если у текущего пользователя нет прав, вернуть ошибку доступа.
* Найти пользователя с указанным идентификатором на доске.
* Если пользователь не найден, вернуть ошибку.
* Удалить пользователя из списка активных пользователей доски.
* Отправить пользователю уведомление о том, что он был выгнан с доски.
* Вернуть статус операции (успех).

Обработка ошибок:

* Если идентификатор доски или идентификатор пользователя отсутствует или некорректен, вернуть ошибку валидации.
* При ошибке доступа, вернуть ошибку доступа.
* При ошибке удаления пользователя из списка активных пользователей, вернуть ошибку сервера.

Алгоритм: **Добавить картинку к доске**

Входные данные:

* Идентификатор доски
* Файл картинки
* Координаты расположения картинки на доске

Выходные данные:

* Статус операции (успех или ошибка)

Шаги алгоритма:

* Получить идентификатор доски, файл картинки и координаты расположения.
* Проверить, имеет ли текущий пользователь права на добавление картинок к доске.
* Если у текущего пользователя нет прав, вернуть ошибку доступа.
* Проверить, является ли загруженный файл допустимым форматом картинки (например, PNG, JPEG).
* Если формат файла некорректен, вернуть ошибку валидации.
* Сохранить картинку на сервере и сгенерировать уникальный идентификатор для нее.
* Добавить информацию о картинке (идентификатор, координаты) к состоянию доски.
* Отправить всем активным пользователям доски обновление с новой картинкой.
* Вернуть статус операции (успех).

Обработка ошибок:

* Если идентификатор доски отсутствует или некорректен, вернуть ошибку валидации.
* При ошибке доступа, вернуть ошибку доступа.
* При ошибке сохранения картинки на сервере, вернуть ошибку сервера.
* При ошибке добавления информации о картинке к состоянию доски, вернуть ошибку сервера.

Так же для понимания как реагирует система был составлен пользовательский сценарий, который указан в таблице Н.

Таблица Н – Пользовательские сценарий

|  |  |  |
| --- | --- | --- |
| № функции | Пользователь | Система |
| 1 | Создание новой доски | 1. Отображает форму для ввода названия доски и выбора настроек доступа.  2. После ввода данных и нажатия кнопки "Создать", создает новую доску с указанными параметрами.  3.Перенаправляет пользователя на страницу созданной доски. |
| 2 | Рисование на доске | 1. Отображает выбранный инструмент рисования и его параметры.  2. При перемещении курсора с удерживаемой левой кнопкой мыши, отображает рисунок на доске в реальном времени.  3. Синхронизирует изменения на доске для всех участников. |
| 3 | Стирание на доске | 1. Отображает выбранный инструмент "Ластик" и его размер.  2. При перемещении курсора с удерживаемой левой кнопкой мыши над областью рисунка, удаляет рисунок в указанной области.  3. Обновляет доску для всех участников, отображая изменения. |
| 4 | Добавление текста на доску | 1. Отображает поле ввода текста в указанном месте на доске.  2. После ввода текста и настройки параметров, добавляет текстовый элемент на доску.  3. Обновляет доску для всех участников, отображая добавленный текст. |
| 5 | Управление доступом к доске | 1. Отображает список пользователей и их текущие роли доступа.  2. При выборе роли для каждого пользователя и нажатии кнопки «Разрешить доступ к редактированию доски», обновляет права доступа для выбранных пользователей.  3. Система изменяет права доступ для выбранных пользователей |

# Определения языка

Для реализации проекта "Онлайн доска", будут использоваться следующие языки программирования и технологии:

* HTML (HyperText Markup Language). Язык разметки, используемый для структуры и содержания веб-страниц. В проекте HTML будет применяться для создания каркаса пользовательского интерфейса, включая элементы управления доской, такие как кнопки, панели инструментов и формы.
* CSS (Cascading Style Sheets). Таблицы стилей, используемые для оформления и визуальной презентации веб-страниц. CSS обеспечит адаптивный и современный дизайн интерфейса доски, а также поддержит кросс-браузерную совместимость.
* JavaScript (JS). Основной язык программирования, ответственный за интерактивные элементы на стороне клиента. JavaScript будет управлять функционалом доски, включая возможность рисования, управления доступом пользователей (например, запрет или разрешение на редактирование), и обновления данных в реальном времени без перезагрузки страницы.
* Node.js. Среда выполнения JavaScript на серверной стороне. Node.js будет использоваться для обработки запросов от клиентов, управления сеансами работы, обеспечения безопасности и синхронизации действий между пользователями в режиме реального времени.
* Библиотеки и фреймворки. В проекте будут использоваться специализированные библиотеки и фреймворки, которые обеспечат создание функционала онлайн-доски. В том числе: Socket.IO для реализации взаимодействия пользователей в режиме реального времени. Express.js для управления серверной логикой. Paper.js для работы с растровой графикой, он же будет использоваться для редактирования досок.

# Структура программы и требования к техническим средствам

Функциональные требования:

* Регистрация и авторизация пользователей
* Создание и управление досками (создание, редактирование, удаление)
* Управление доступом к доскам (разрешение/запрет рисования для других пользователей)
* Рисование на доске (различные инструменты: карандаш, ластик, цвета, толщина линии и т.д.)
* Сохранение и загрузка досок
* Возможность приглашения других пользователей на доску по ссылке

Нефункциональные требования:

* Удобство использования: интуитивно понятный и простой интерфейс, адаптивный дизайн для различных устройств
* Надежность: система должна быть стабильной и доступной большую часть времени

Взаимодействие с внешними системами:

Интеграция с системами аутентификации:

* Возможность входа через VK ID

Интеграция с облачными хранилищами:

* Возможность сохранения досок в облачных хранилищах, таких как Google Drive, Dropbox и др.