**Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Владимирский государственный университет**

**имени Александра Григорьевича и Николая Григорьевича Столетовых» (ВлГУ)**

Колледж инновационных технологий и предпринимательства

**кафедра физики и прикладной математики**

**ОТЧЕТ**

**ПО производственной ПРАКТИКЕ**

**ПО МОДУЛЮ ПМ.02 «** **Участие в интеграции программных модулей»**

Студента \_\_Руденко Александры Максимовны

Специальность 09.02.07 Информационные системы и программирование

Группа ИПсп-121

Время прохождения практики с «13» ноября 2024 г. по «24» декабря 2024 г.

Руководитель от ВлГУ

преподаватель КИТП (кафедра ФиПМ)

*\_\_\_\_\_\_\_\_\_\_\_\_\_*М.Ю. Кабанова

Руководитель от предприятия:

Инженер-программист 2 категории

+79065627313

*\_\_\_\_\_\_\_\_\_\_\_\_\_* К.А. Финаков

Владимир, 2024

|  |  |
| --- | --- |
|  | Утверждаю  Зав. кафедрой ФиПМ\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |
|
|

**ЗАДАНИЕ**

на производственную практику по модулю ПМ.02 «Участие в интеграции программных модулей»

студента \_\_*Руденко Александры Максимовны*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

**4** курса, направления 09.02.07 Информационные системы и программирование\_\_\_\_\_\_\_\_\_\_\_\_\_\_

группы **ИПсп-121**

Предприятие\_ООО Фирма "Инрэко ЛАН", г. Владимир\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Последовательность прохождение практики получить задание, пройти инструктаж по технике безопасности и охране труда, выполнить все задания, оформить отчет согласно требованиям.

За время прохождения практики необходимо:

1. Изучить вопросы, предусмотренные программой по всем разделам.

2. Изучить и исследовать:

*2.1. На языке UML построить необходимые диаграммы для выбранного приложения. 2.2. Развернуть клиентскую часть системы контроля версий Git.*

*2.3. Поместить в репозиторий исходные файлы разработанного приложения.*

*2.4. Изучить принципы использования компилятора вне интегрированной среды разработки.*

*2.5. Разработать тест-кейсы, на их основании протестировать приложение.*

3. Получить навыки

*3.1. оформления отчета по результатам практики согласно требованиям к оформлению документации.*

4. Задание по охране труда, технике безопасности и охране окружающей среды:

*4.1. изучить правила техники безопасности при работе за компьютером, вопросы охраны труда на месте прохождения практики.*

Отчет по практике составить к \_\_20.12.2024\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Задание выдал:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.Ю. Кабанова

(Ф.И.О, руководителя практики от университета)

Задание получил:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.М. Руденко

(подпись студента, дата)

СОДЕРЖАНИЕ

[Введение 2](#_Toc181892991)

[1 Разработка приложения 3](#_Toc181892992)

[1.1 Описание предметной области 3](#_Toc181892993)

[2 проектирование приложения 4](#_Toc181892994)

[2.1 Проектирование процессов функционирования 4](#_Toc181892995)

[3 разработка приложения 5](#_Toc181892996)

[3.1 Инструментальные средства разработки приложения 5](#_Toc181892997)

[3.2 Разработка интерфейса 5](#_Toc181892998)

[4 система контроля версий 6](#_Toc181892999)

[5 тестирование 7](#_Toc181893000)

[5.1 Тест-требования 7](#_Toc181893001)

[5.2 Тест-план 7](#_Toc181893002)

[5.3 Результаты тестирования 7](#_Toc181893003)

[заключение 8](#_Toc181893004)

[список использованных источников 9](#_Toc181893005)

[приложение а 10](#_Toc181893006)

введение

Целью прохождения производственной практики, которая проходит с 13.11.24 по 24.12.24, является получение теоретических и практических знаний, умений и навыков в рамках профессионального модуля ПМ.02 «Участие в интеграции программных модулей». Для реализации на практике, полученных теоретических знаний, необходимо выполнить задания, охватывающие все предусмотренные программой практики разделы, а именно:

* изучить основные принципы и методы разработки веб-приложений, включая архитектурные паттерны и технологии;
* ознакомиться с концепцией канбан-досок и их применением в управлении проектами и организацией рабочего процесса;
* разработать проектную документацию, включая техническое задание, архитектурные схемы и пользовательские сценарии;
* реализовать функционал канбан-доски, включая создание, редактирование и удаление задач, а также управление их статусами;
* провести тестирование разработанного веб-приложения на предмет функциональности, производительности и безопасности;

1. разработка приложения

1.1 Описание предметной области

В рамках данной работы предусмотрена разработка веб-приложения, которое будет служить для организации рабочего процесса с использованием канбан-досок. Основная цель приложения — предоставить пользователям удобный инструмент для управления задачами и проектами в команде, что позволит повысить эффективность и прозрачность рабочего процесса.

В процессе разработки веб-приложения будут реализованы следующие функциональные возможности:

1. Регистрация и авторизация
   1. Пользователь сможет зарегистрироваться в системе, указав свой адрес электронной почты.
   2. После регистрации пользователю будет доступна функция авторизации с использованием электронной почты и пароля.
2. Работа с досками.
   1. Пользователь получит возможность создавать новые канбан-доски для управления задачами.
   2. Реализованная функциональность позволит изменять названия созданных досок, а также удалять их при необходимости.
3. Работа с задачами.
   1. Веб-приложение позволит пользователям добавлять задачи на доски, а также редактировать их, включая такие параметры, как название, описание, ответственный, срок завершения, статус и приоритет.
   2. Кроме того, в систему будет добавлена функция удаления и восстановления задач.
4. Совместная работа.
   1. Создатели досок смогут приглашать других пользователей для совместной работы над проектами.

В рамках веб-приложения будут предусмотрены две ключевых роли пользователей:

* Создатель доски — это пользователь, который создает канбан-доску. Он имеет полный контроль над доской, включая возможность добавления, редактирования и удаления колонок и задач. Создатель также отвечает за добавление других участников, настраивая их уровни доступа.
* Редактор — пользователь, которому создатель доски предоставляет право редактирования. Этой роли доступны функции внесения изменений в существующие задачи и колонки, а также добавления новых. Однако редакторы не могут удалять доску или управлять доступом других пользователей.

Входные параметры:

* Электронная почта и пароль для регистрации и авторизации пользователя.
* Информация о создаваемых досках и задачах (названия, описания и т. д.).

Выходные параметры:

* Успешное создание учетной записи, доски, колонки и задачи.
* Сообщения об ошибках при регистрации или авторизации.
* Измененные данные о досках и задачах после редактирования.

Таким образом, разработка данного веб-приложения обеспечит пользователей мощным инструментом для эффективного управления рабочими процессами и коллективной работы.

2 проектирование приложения

2.1 Проектирование процессов функционирования

Диаграмма прецедентов — это диаграмма, позволяющая описать систему на концептуальном уровне, отражающая отношения между актерами и прецедентами и является составной частью модели прецедентов.

Диаграмма прецедентов для приложения представлена на рисунке 2.1.

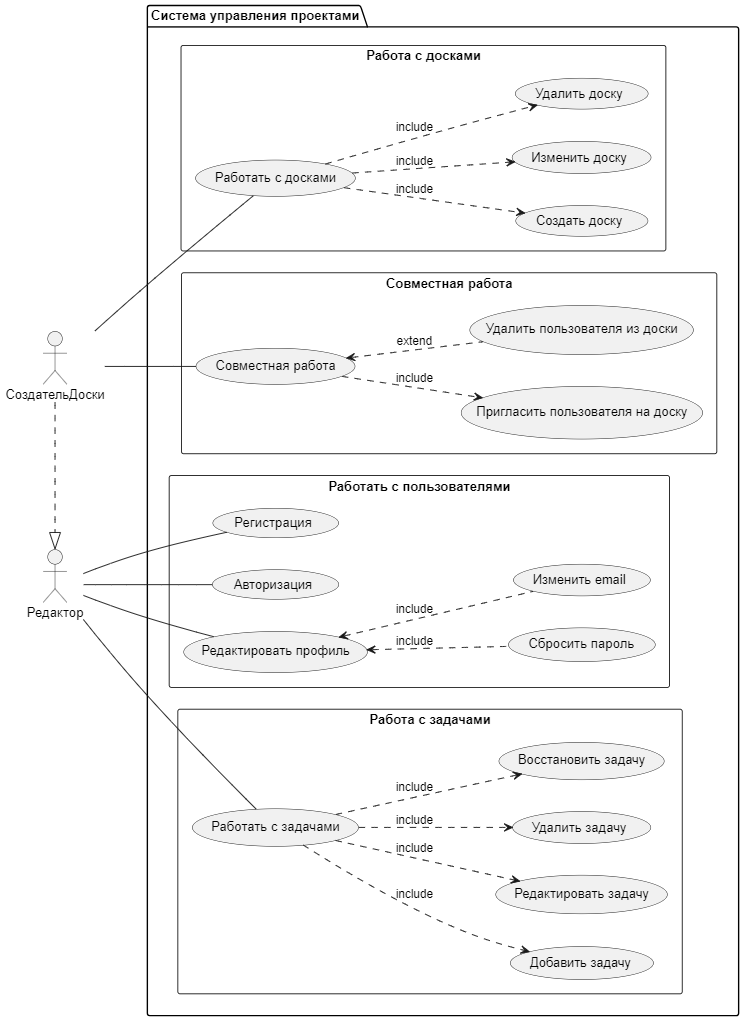


Рисунок 2.1 – Диаграмма прецедентов

Диаграмма деятельности — UML-диаграмма, на которой показаны действия, состояния которых описаны на диаграммах состояний. Под деятельностью понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов — вложенных видов деятельности и отдельных действий, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого.

Диаграмма деятельности для регистрации представлена на рисунке 2.2.

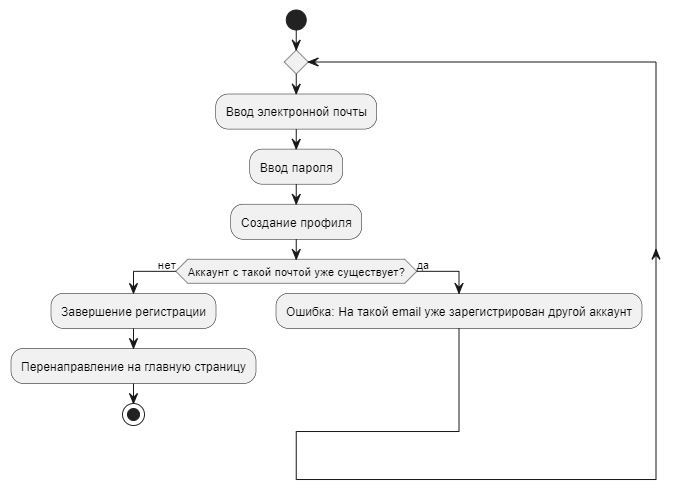


Рисунок 2.2 – Диаграмма деятельности регистрации

Диаграмма деятельности для авторизации представлена на рисунке 2.3.

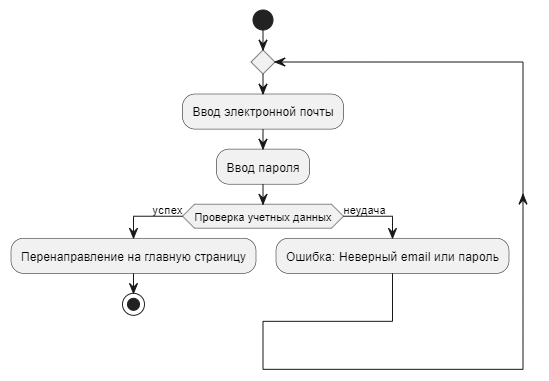


Рисунок 2.2 – Диаграмма деятельности авторизации

Диаграмма деятельности для работы с досками представлена на рисунке 2.4.

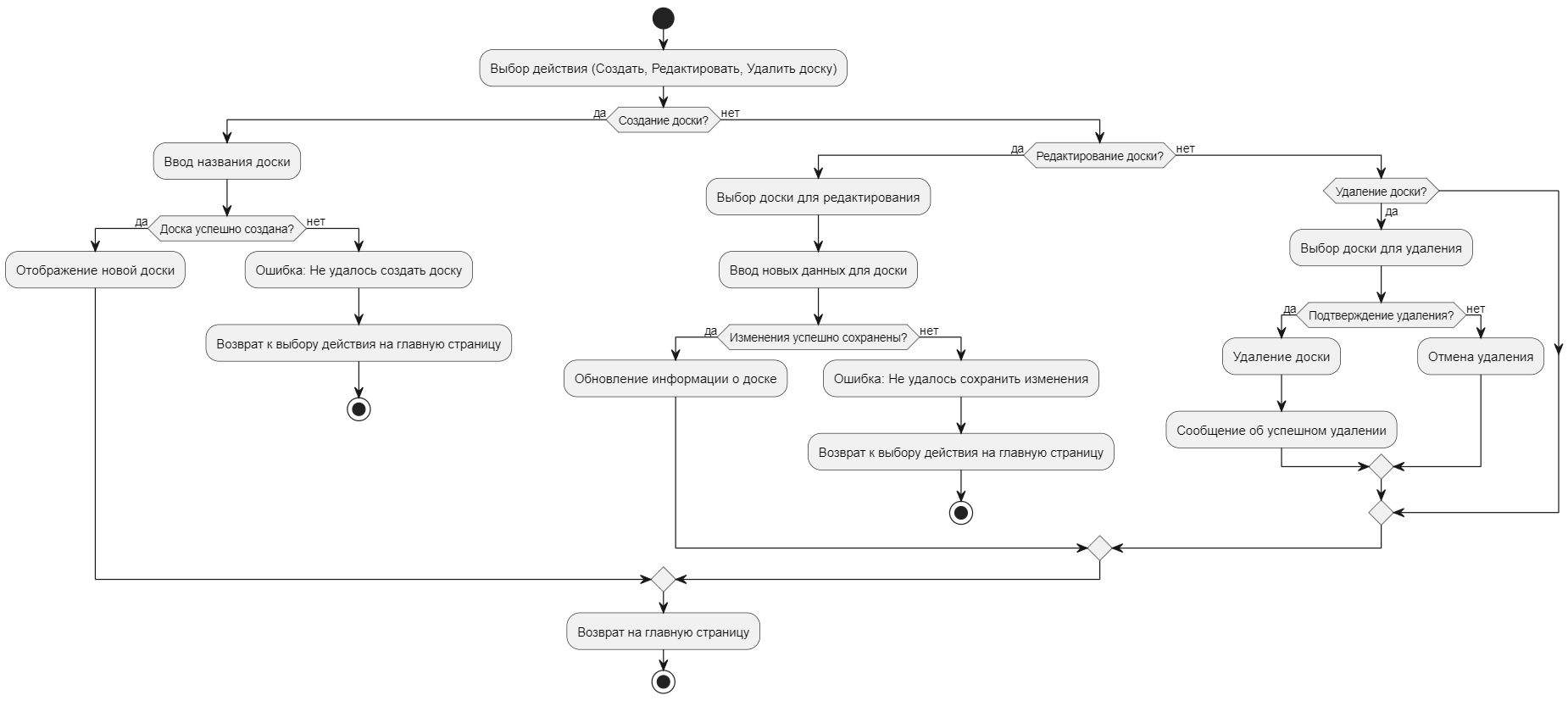


Рисунок 2.4 – Диаграмма деятельности работы с досками

Диаграмма деятельности для работы с задачами представлена на рисунке 2.5.

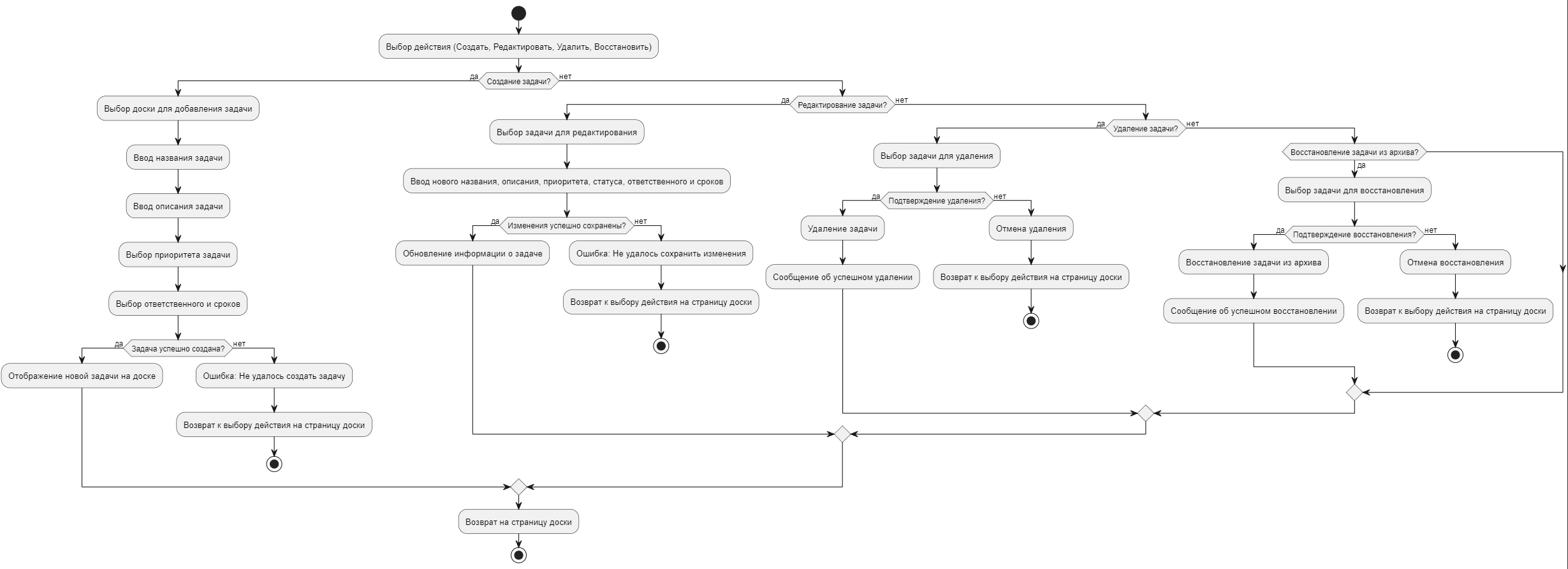


Рисунок 2.5 – Диаграмма деятельности работы с задачами

Диаграмма деятельности для создания приглашения представлена на рисунке 2.6.

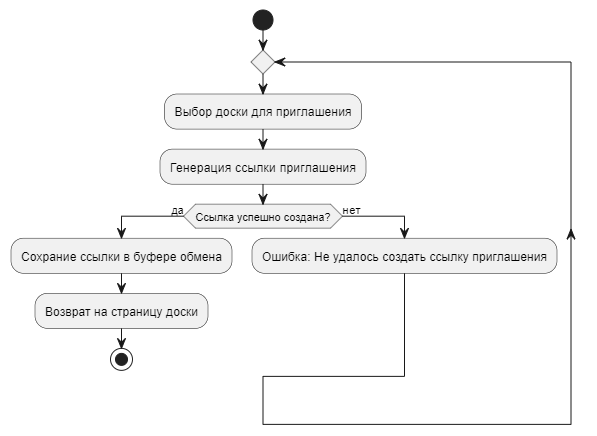


Рисунок 2.6 – Диаграмма деятельности создания приглашения на доску

Диаграмма деятельности для управления участниками доски представлена на рисунке 2.7.

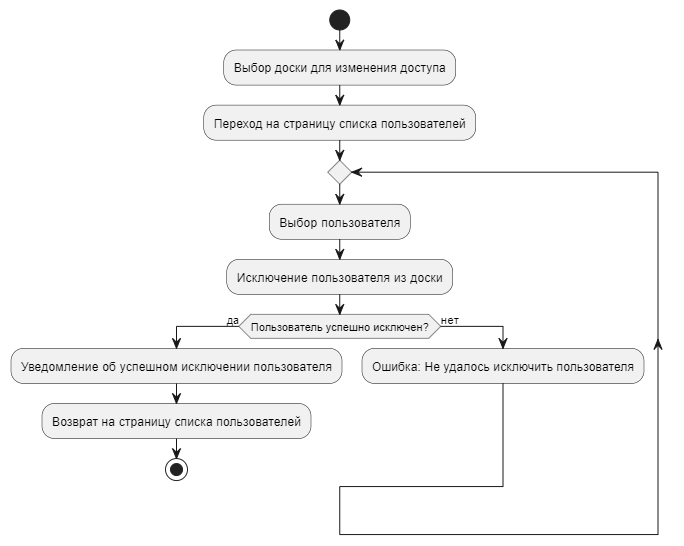


Рисунок 2.7 – Диаграмма деятельности управления участниками доски

Таким образом, разработанная диаграмма прецедентов отображает взаимодействие между пользователями и системой, позволяя четко выделить основные функции и сценарии использования. Диаграммы деятельности, в свою очередь, обеспечивают визуализацию процессов и последовательности шагов в рамках системы. Они помогают проанализировать и понять логику выполнения задач, условия переходов и взаимодействие между различными компонентами, что способствует лучшему пониманию рабочих процессов и их оптимизации.

3 разработка приложения

3.1 Инструментальные средства разработки приложения

Для разработки веб-приложения канбан-досок была выбрана следующая технологическая среда:

* Система контроля версий Git – это распределенная система, которая позволяет отслеживать изменения в коде, управлять версиями и эффективно работать в команде.
* Платформа для удаленных репозиториев GitHub – популярная платформа для хостинга Git-репозиториев, предлагающая функции для совместной работы, управления задачами и анализа кода.
* Язык разработки C# – современный объектно-ориентированный язык от Microsoft, подходящий для веб-приложений, обеспечивающий высокую производительность и простоту оформления кода.
* Технологии и платформы ASP.NET и Razor Pages. Мощная платформа для создания масштабируемых веб-приложений. Razor Pages упрощает разработку динамических страниц с удобным синтаксисом.
* СУБД SQLite – легковесная реляционная база данных, идеальная для малых и средних проектов. Проста в использовании и не требует сложной настройки.

3.2 Разработка интерфейса

При запуске приложения, а также выходе из аккаунта открывается страница для авторизации, представленная на рисунке 3.1.

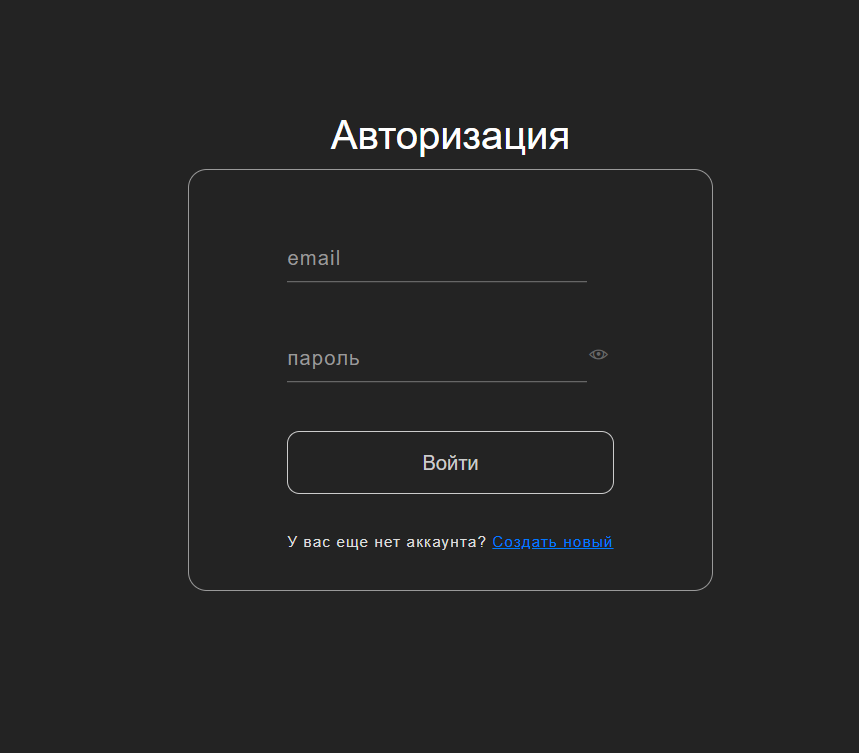


Рисунок 3.1 – Страница авторизации

Если пользователь еще не зарегистрирован в системе, при нажатии на ссылку «Создать новый» в окне авторизации открывается страница для регистрации, представленная на рисунке 3.2.

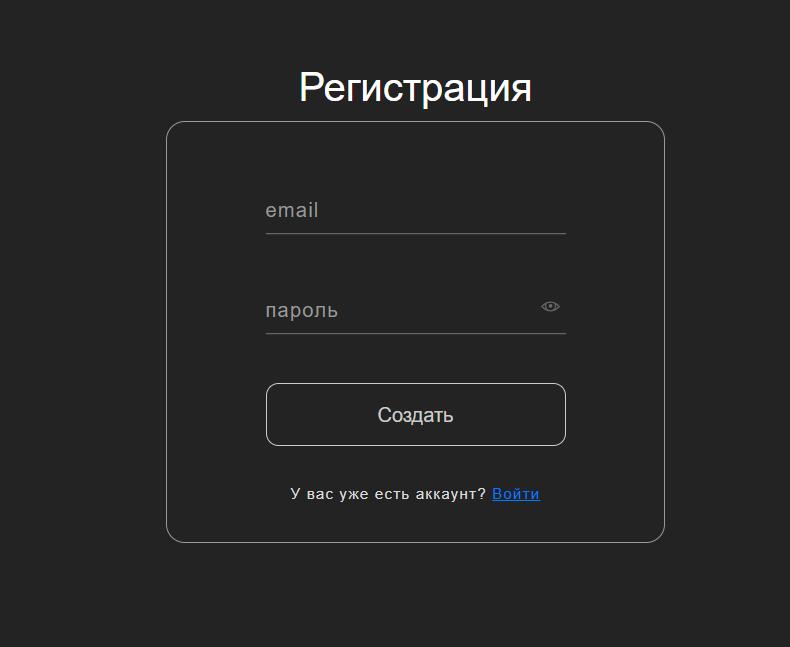


Рисунок 3.2 – Страница регистрации

После успешной авторизации пользователя переходит на главную страницу, на которой отображаются доступные ему доски, а также кнопка «Добавить доску». Скриншот главной страницы представлен на рисунке 3.3.

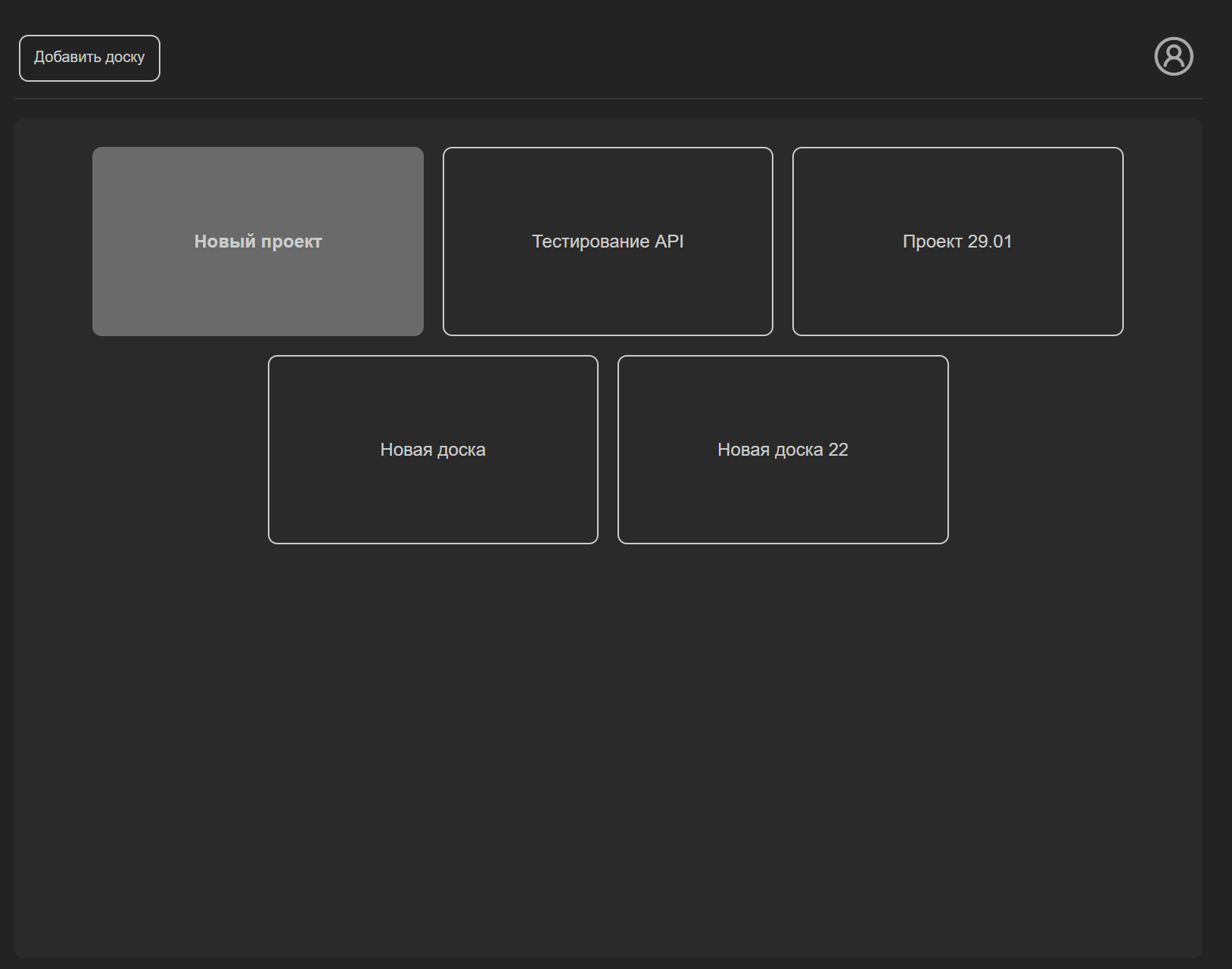


Рисунок 3.3 – Главная страница

При нажатии на кнопку «Добавить доску» на главной странице, открывается окно с полем для ввода названия создаваемой таблицы, представленное на рисунке 3.4.

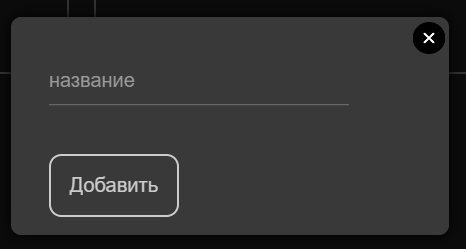


Рисунок 3.4 – Окно добавления доски

При нажатии на доску открывается страница с содержимым самой доски. На странице находятся кнопки удаления, генерации приглашения и отображения списка участников доски. Также есть кнопки добавления задачи и колонки. Страница доски представлена на рисунке 3.5.

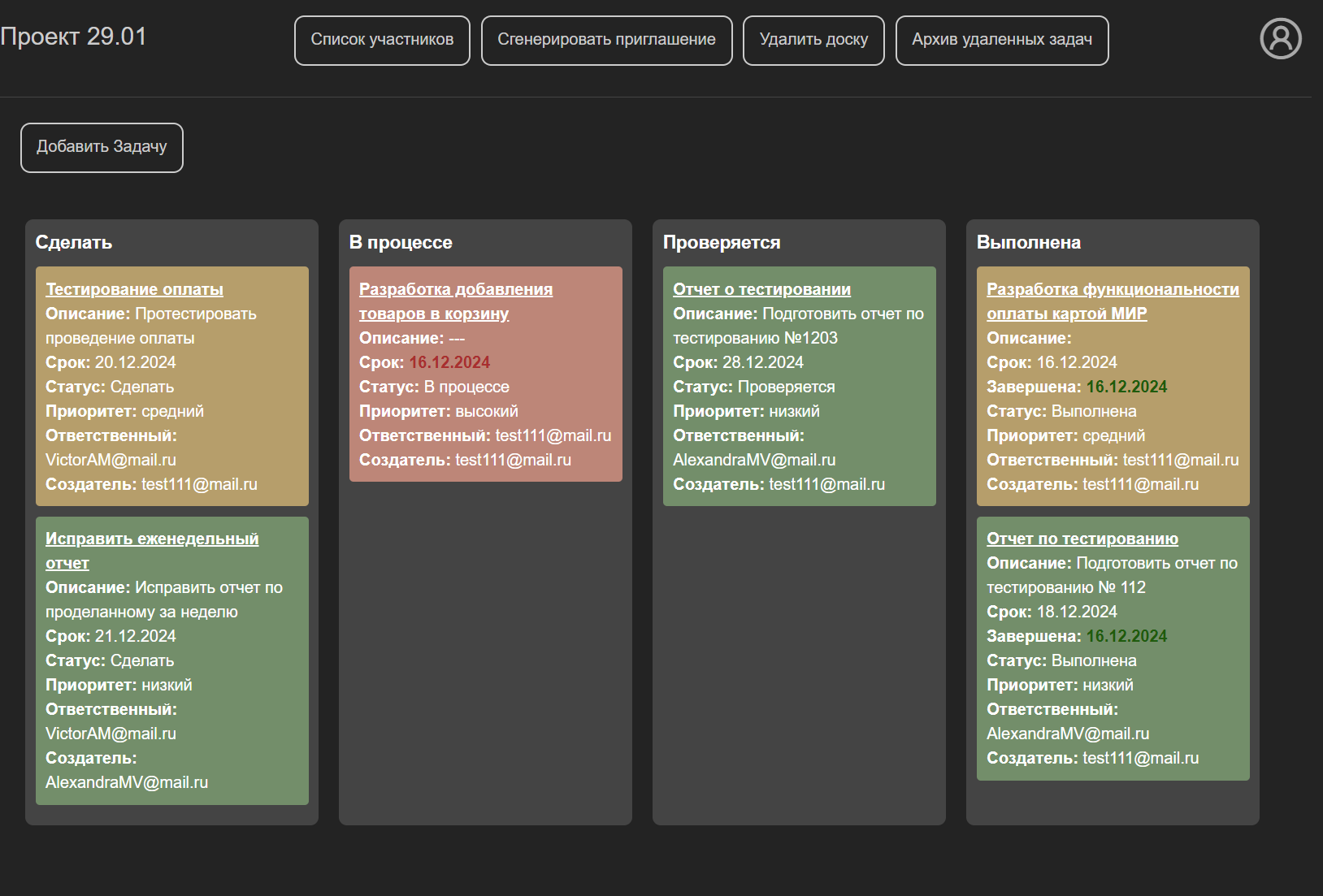


Рисунок 3.5 – Страница доски

При нажатии на кнопку «Добавить задачу» открывается окно с полями для ввода и выпадающими списками, для заполнения данных о задаче. Окно добавления задачи представлено на рисунке 3.6.

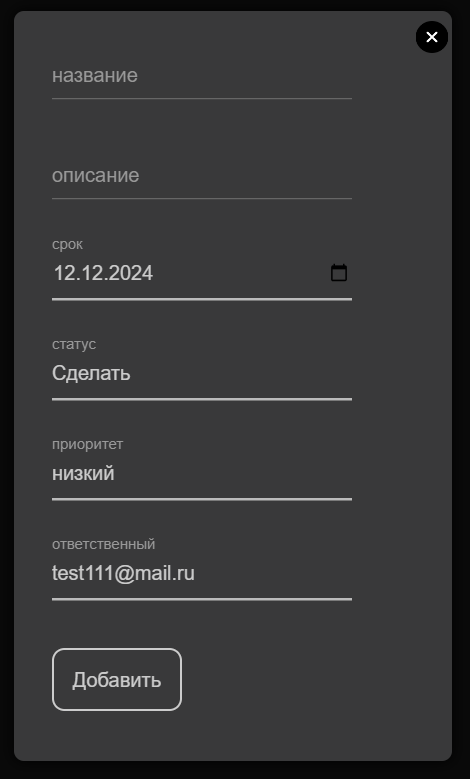


Рисунок 3.6 – Окно добавления задачи

При нажатии на кнопку «Удалить доску» открывается окно с подтверждением удаления доски, представленное на рисунке 3.7.

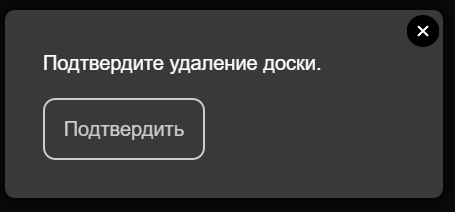


Рисунок 3.7 – Окно удаления доски

При нажатии на кнопку «Сгенерировать приглашение» открывается окно с возможностью выбора роли для которой будет генерироваться ссылка-приглашение. Окно для генерации ссылки представлено на рисунке 3.8.

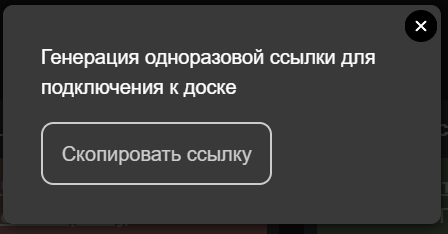


Рисунок 3.8 – Окно генерации приглашения на доску

При нажатии на кнопку «Список участников» открывается страница со списком всех участников доски. Отдельным участникам можно поменять роль на доске или удалить их. Страница списка участников представлена на рисунке 3.9.

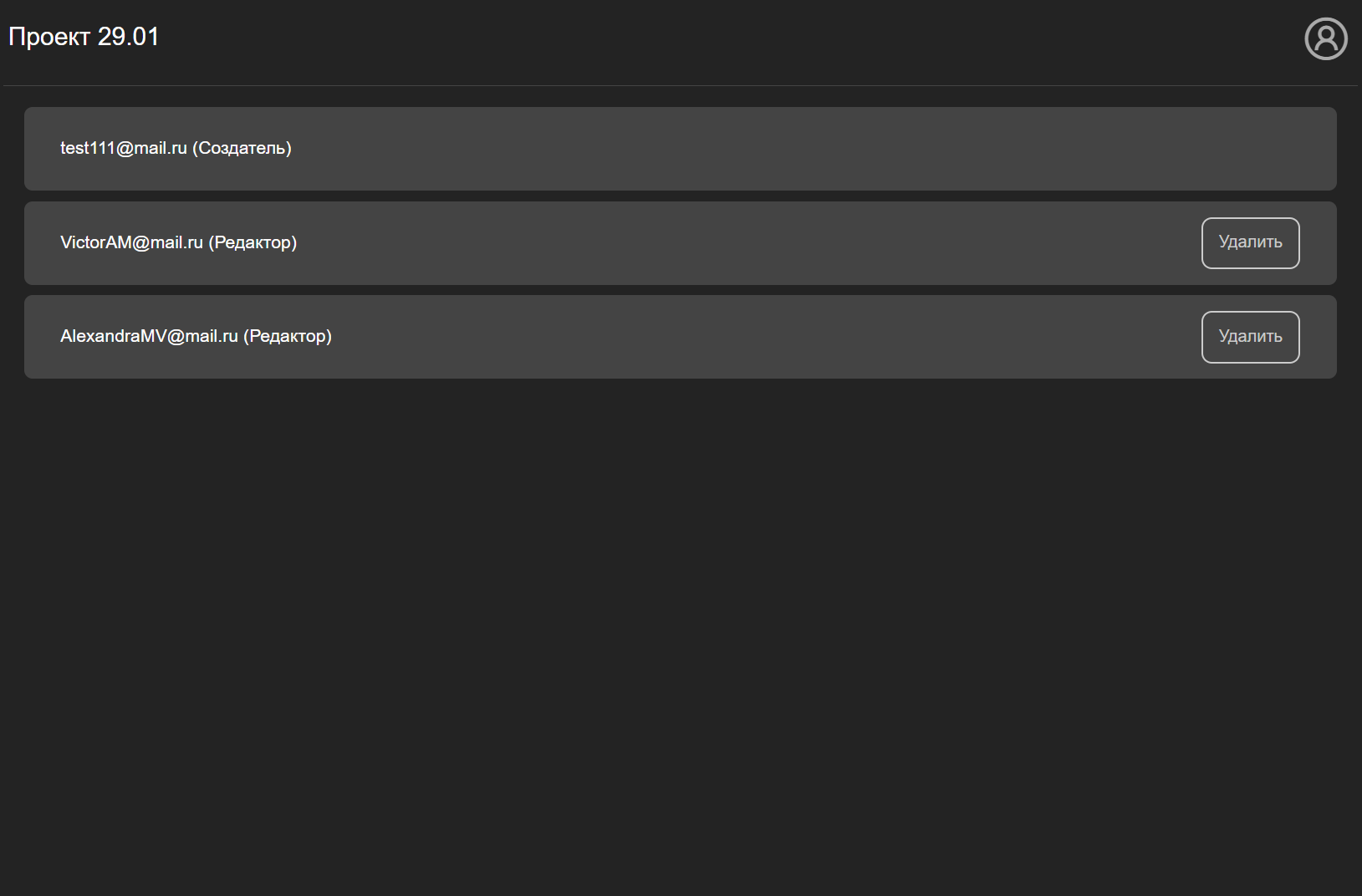


Рисунок 3.9 – Страница списка участников доски

При нажатии на кнопку «Удалить» возле участника открывается окно с подтверждением удаления участника из доски, представленное на рисунке 3.10.

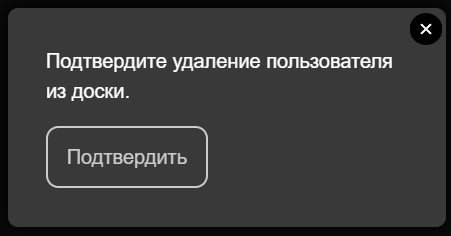


Рисунок 3.10 – Окно удаления участника из доски

При нажатии на кнопку «Архив удаленных задач» открывается страница со списком всех задач доски, которые были удалены, представленная на рисунке 3.11.

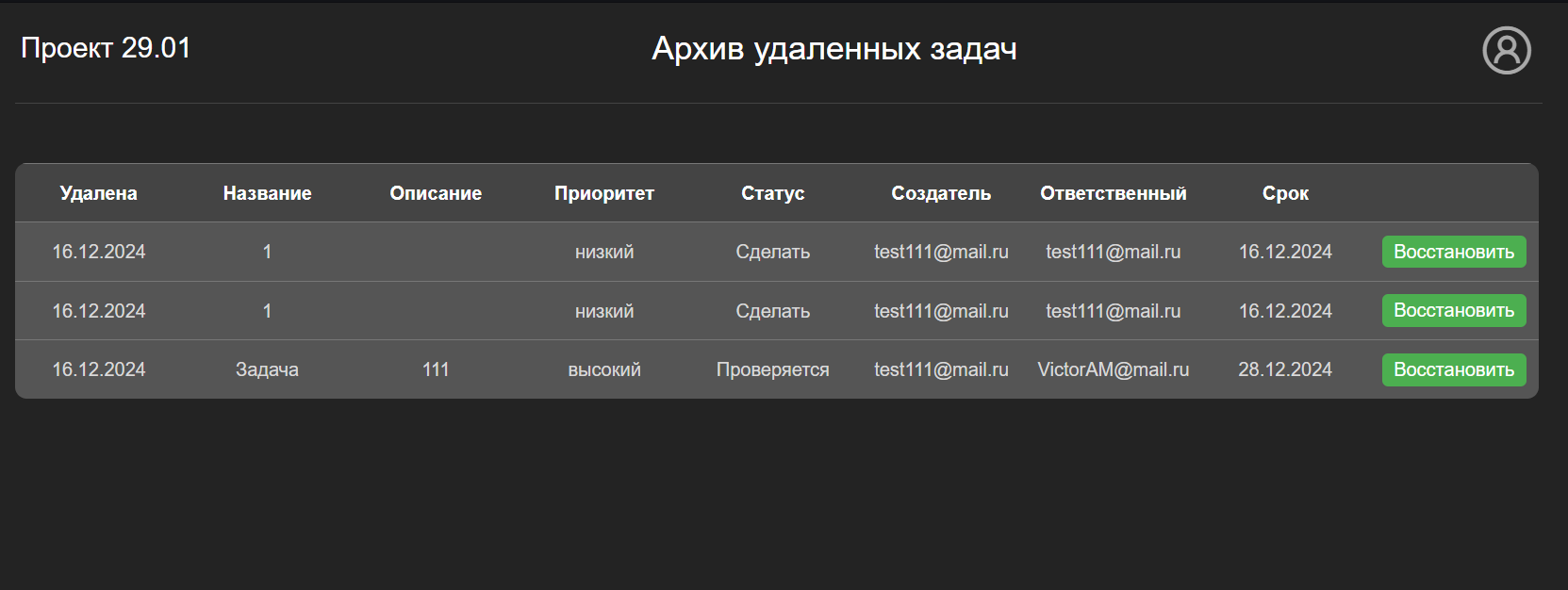


Рисунок 3.11 – Страница архива задач

На многих страницах в верхнем правом углу присутствует иконка профиля пользователя, при нажатии на которую открывается страница профиля пользователя, представленная на рисунке 3.12.

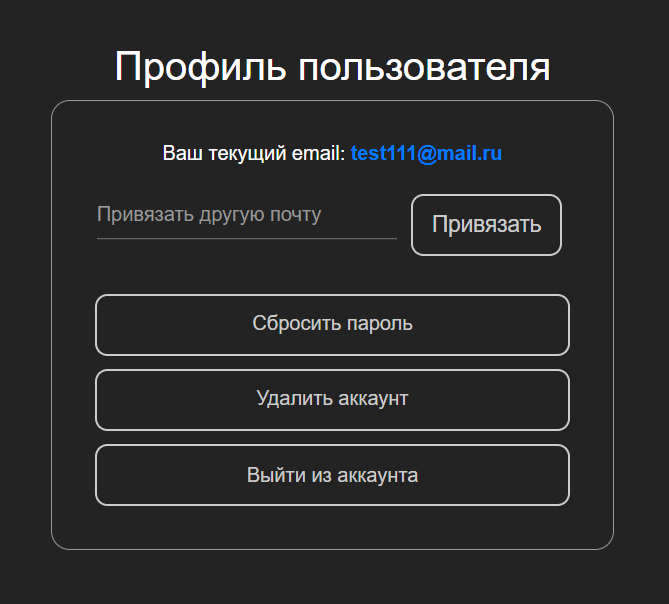


Рисунок 3.12 – Страница профиля пользователя

При нажатии на кнопку «Сбросить пароль» открывается окно, в котором можно ввести новый пароль. Окно сброса пароля представлено на рисунке 3.13.

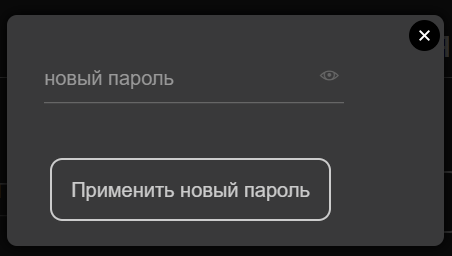


Рисунок 3.13 – Окно изменения пароля пользователя

Таким образом, был разработан простой и интуитивно понятный интерфейс для приложения, который реализует все предусмотренные возможности в ранее представленных диаграммах активностей и прецедентов.

4 система контроля версий

Git — это распределённая система контроля версий. Она позволяет следить за происходящими изменениями, а также работать над одним заданием одновременно нескольким разработчикам. Например, контроль версий помогает отслеживать исправления в текстовых документах или сохранять версии изменённых файлов. [1]

Для создания репозитория Git необходимо запустить в Visual Studio необходимый проект, в верхнем меню выбрать вкладку «Git» и пункт «Создать репозиторий Git». Затем откроется окно создания репозитория Git, где необходимо выбрать учетную запись GitHub в которую будет загружен репозиторий, а также убрать галочку с пункта «Частный репозиторий», чтобы другие пользователи могли просматривать репозиторий. Скриншот окна создания репозитория Git представлен на рисунке 4.1.

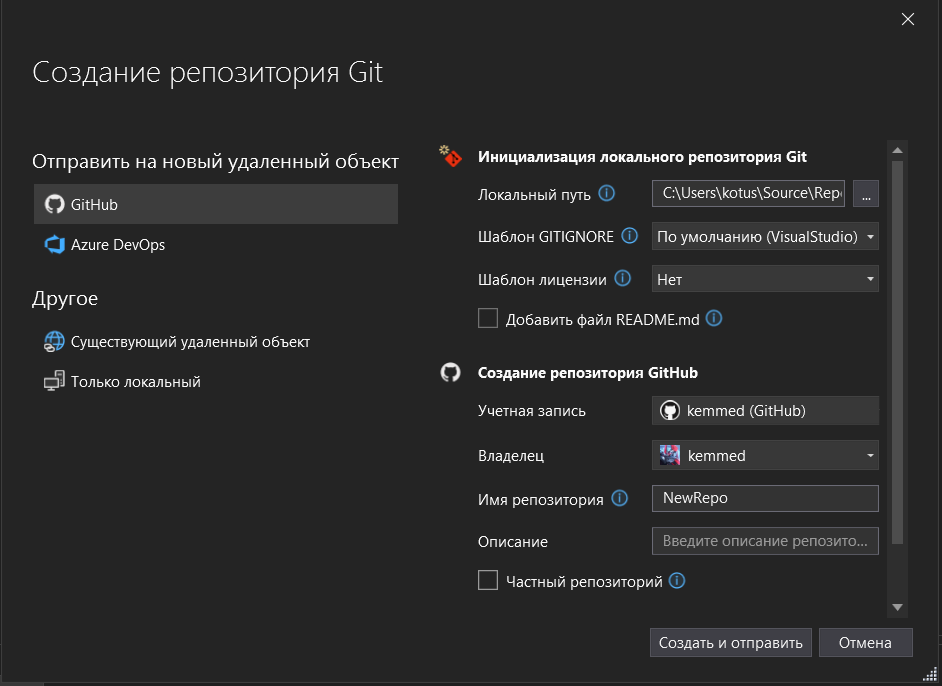


Рисунок 4.1 – Создание репозитория Git

После нажатия на кнопку «Создать и отправить» репозиторий будет создан и отправлен на GitHub. Скриншот успешно созданного репозитория на GitHub представлен на рисунке 4.2.

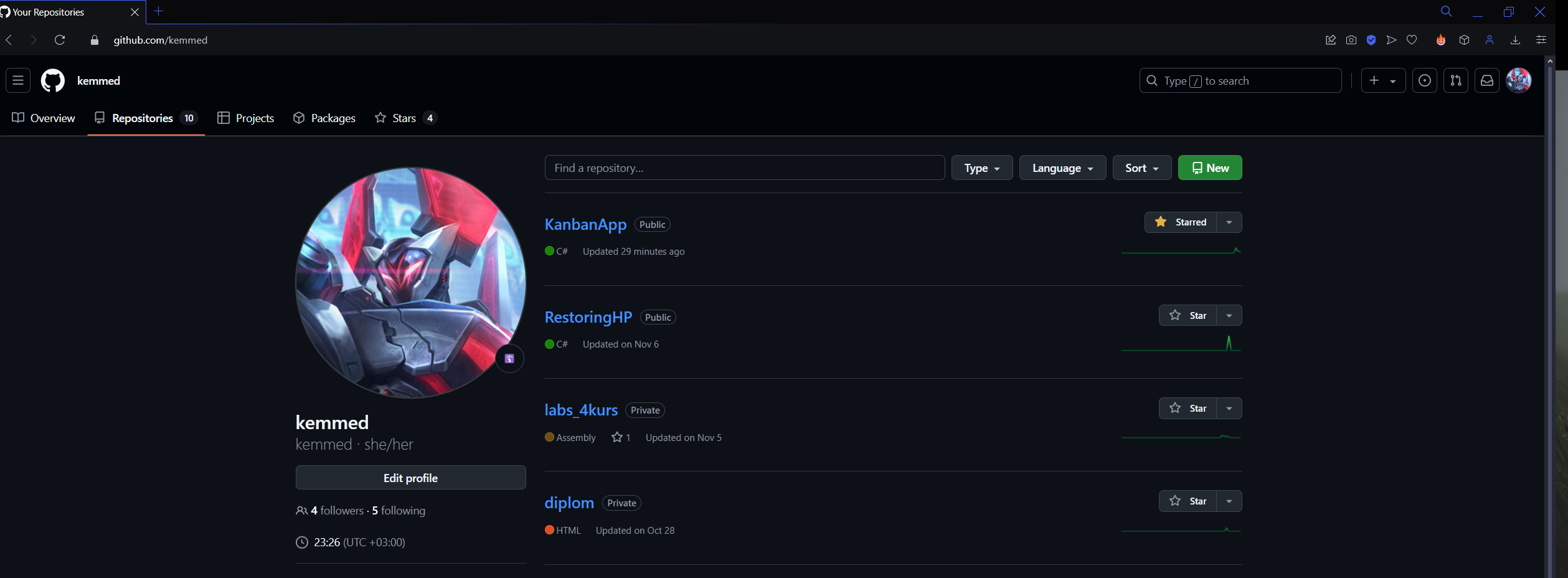


Рисунок 4.2 – Репозиторий на GitHub

Чтобы выполнить клонирование репозитория на устройство, необходимо открыть на GitHub страницу со структурой репозитория, представленную на рисунке 4.3.

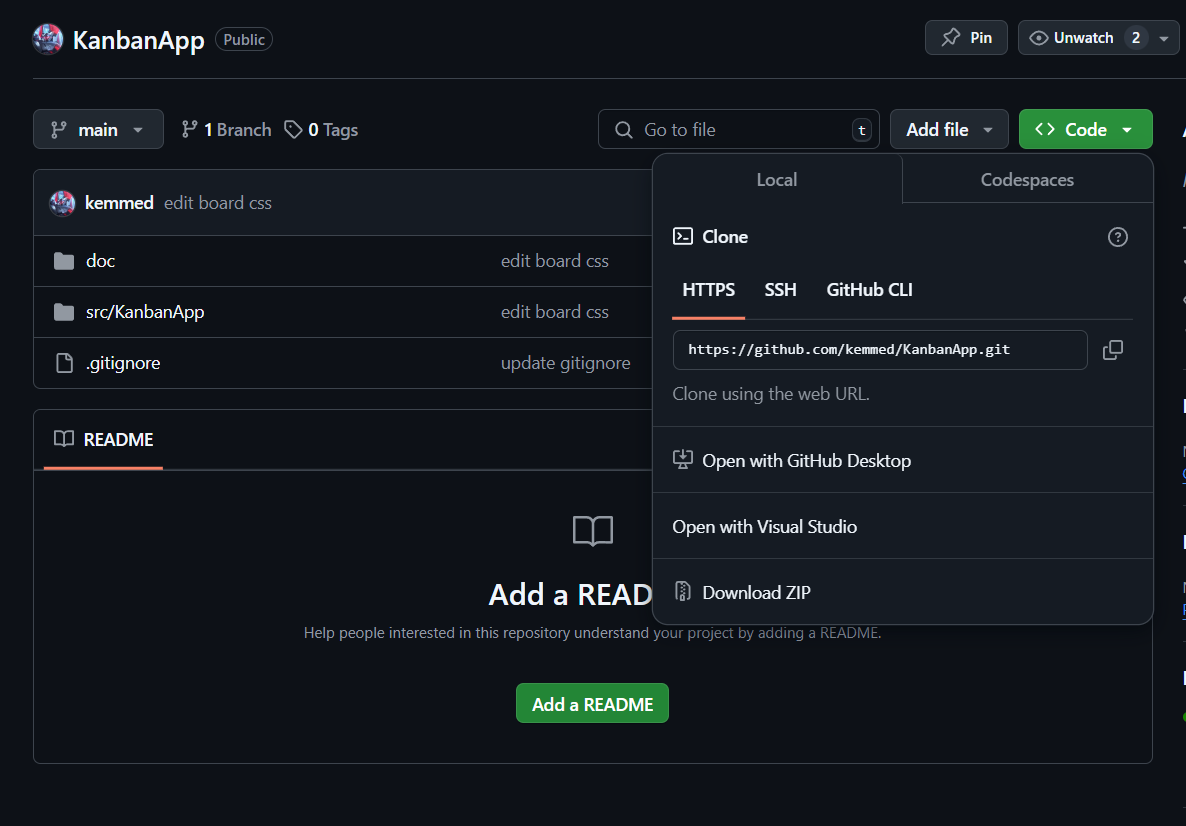


Рисунок 4.3 – Репозиторий на GitHub

После открытия страницы, необходимо нажать на кнопку «Code» и в открывшемся меню скопировать ссылку для клонирования репозитория. Затем необходимо открыть VisualStudio. Необязательно создавать новый проект, можно нажать на кнопку «Продолжить без кода». Затем в окне в верхнем меню выбрать необходимо вкладку «Git» и пункт «Клонировать репозиторий». Скриншот окна представлен на рисунке 4.4.

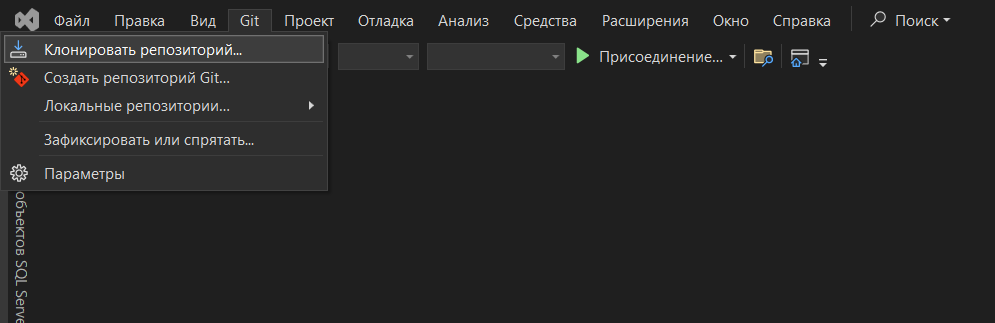


Рисунок 4.4 – Клонирование репозитория

В открывшемся окне, представленном на рисунке 4.5 в поле расположения репозитория необходимо вставить скопированную с GitHub ссылку и выбрать путь для клонирования репозитория, после чего нажать на кнопку «Клонировать».

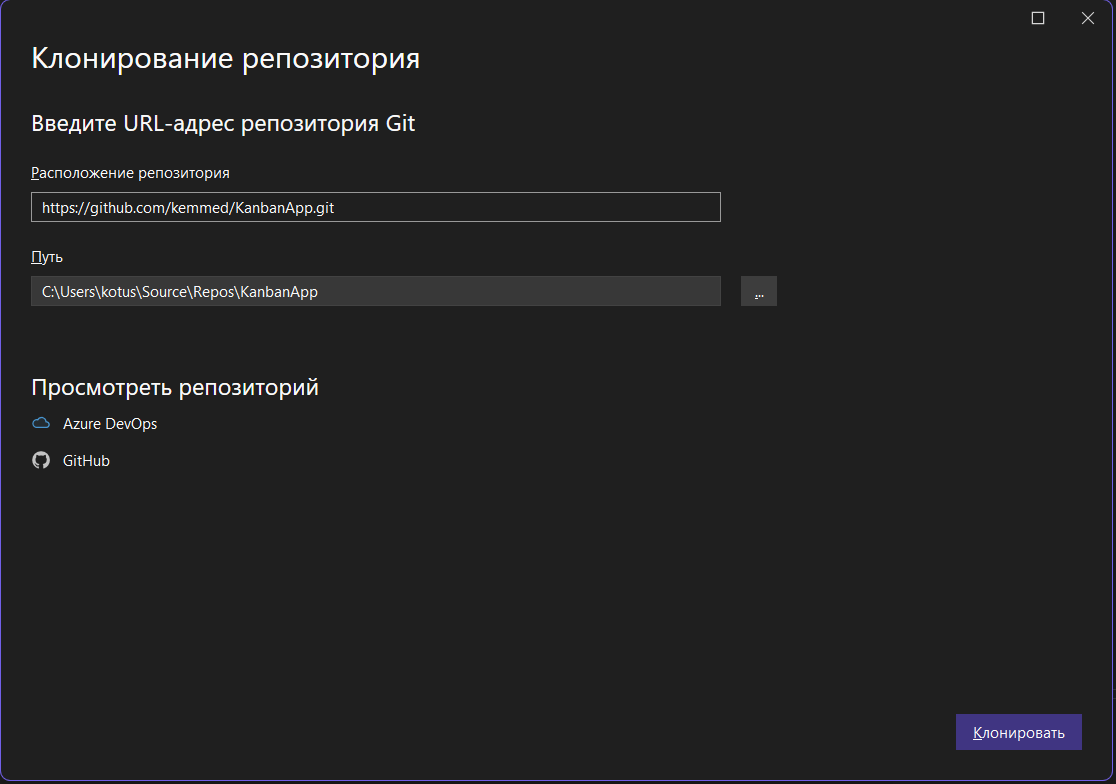


Рисунок 4.5 – Окно выбора расположения и пути репозитория

После этого все файлы проекта появятся в указанной папке и с программой можно будет работать как обычно.

5 тестирование

5.1 Тест-требования

Тест-требование 1. Проверка корректной регистрации.

Проверить, что регистрация проходит успешно при корректно введенных email и пароле.

Тест-требование 2. Проверка корректной авторизации.

Проверить, что авторизация с корректными учетными данными (email и пароль) предоставляет доступ к главной странице приложения.

Тест-требование 3. Проверка корректного добавления новой доски.

Проверить, что при добавлении новой доски, она отобразится в списке досок доступных пользователю.

Тест-требование 4. Проверка корректного добавления задачи.

Проверить, что при добавлении новой задачи, она корректно отобразится в соответствующей ее статусу колонке и будет выделена цветом, соответствующим ее приоритету.

Тест-требование 5. Проверка корректного редактирования профиля пользователя.

Проверить, что при смене email пользователя или его пароля, все изменения сохраняются успешно.

Тест-требование 6. Проверка корректного добавления пользователя на доску.

Проверить, что при генерации ссылки-приглашения на доску, она генерируется корректно и является рабочей.

5.2 Тест-план

Тестовый пример 1

Тест-требование: тест-требование 1

Описание теста: проверка регистрации с корректно введенными данными.

Входные данные:

* Email: user@example.com
* Пароль: Password123

Ожидаемые выходные данные: пользователь успешно зарегистрирован и перенаправлен на страницу подтверждения.

Сценарий:

* Открыть страницу регистрации.
* Ввести email и пароль.
* Нажать кнопку «Зарегистрироваться».
* Убедиться, что пользователь перенаправлен на страницу авторизации.

Тестовый пример 2

Номер тест-требования: тест-требование 1

Описание теста: проверка регистрации с email, на который уже зарегистрирован аккаунт.

Входные данные:

* Email: user@example.com
* Пароль: Password123

Ожидаемые выходные данные: появляется сообщение об ошибке регистрации с указанным email.

Сценарий:

* Открыть страницу регистрации.
* Ввести email и пароль.
* Нажать кнопку «Зарегистрироваться».
* Убедиться, что отображается сообщение об ошибке с указанием неправильных данных.

Тестовый пример 3

Тест-требование: тест-требование 2

Описание теста: проверка авторизации с корректно введенными данными.

Входные данные:

* Email: user@example.com
* Пароль: Password123

Ожидаемые выходные данные: пользователь успешно авторизован и перенаправлен на главную страницу приложения.

Сценарий:

* Открыть страницу авторизации.
* Ввести email и пароль.
* Нажать кнопку «Войти».
* Убедиться, что пользователь перенаправлен на главную страницу.

Тестовый пример 4

Тест-требование: тест-требование 2

Описание теста: проверка авторизации с некорректно введенными данными.

Входные данные:

* Email: wronguser@example.com
* Пароль: Password123

Ожидаемые выходные данные: появляется сообщение об ошибке авторизации.

Сценарий:

* Открыть страницу авторизации.
* Ввести email и пароль.
* Нажать кнопку «Войти».
* Убедиться, что отображается сообщение об ошибке.

Тестовый пример 5

Тест-требование: тест-требование 3

Описание теста: проверка корректного добавления новой доски.

Входные данные:

* Название доски: «Новая доска»

Ожидаемые выходные данные: новая доска отображается в списке доступных досок пользователя.

Сценарий:

* Перейти на главную страницу и нажать кнопку «Добавить доску».
* Ввести название доски и нажать кнопку «Добавить».
* Убедиться, что новая доска отображается в списке доступных досок.

Тестовый пример 6

Тест-требование: тест-требование 4

Описание теста: проверка корректного добавления задачи.

Входные данные:

* Название задачи: «Новая задача»
* Статус задачи: «В процессе»
* Приоритет задачи: «Высокий»
* Срок: текущая дата
* Ответственный: по умолчанию

Ожидаемые выходные данные: задача отображается в соответствующей колонке «В процессе» и выделена красным цветом.

Сценарий:

* Перейти на страницу доски и нажать на кнопку «Добавить задачу».
* Ввести название задачи, выбрать статус, приоритет, срок и ответственного.
* Нажать кнопку «Добавить задачу».
* Убедиться, что задача отображается в колонке «В процессе» и выделена красным цветом.

Тестовый пример 7

Тест -требование: тест-требование 5

Описание теста: проверка корректного редактирования профиля пользователя.

Входные данные:

* Новый email: newuser@example.com

Ожидаемые выходные данные: изменения сохраняются успешно, новый email отображается в профиле.

Сценарий:

* Перейти на страницу профиля пользователя.
* Ввести новый email и подтвердить изменения.
* Убедиться, что измененный email отображается в профиле пользователя.

Тестовый пример 8

Тест-требование: тест-требование 6

Описание теста: проверка корректной генерации ссылки-приглашения.

Входные данные: -

Ожидаемые выходные данные: генерируется и сохраняется в буфер обмена ссылка-приглашение для добавления пользователя на доску.

Сценарий:

* Перейти на страницу доски.
* Нажать на кнопку «Сгенерировать ссылку».
* В открывшемся окне нажать на кнопку «Скопировать ссылку».
* Убедиться, что ссылка сгенерирована корректно и сохранилась в буфер обмена.

5.3 Результаты тестирования

Для каждого тестового примера указать входные и выходные данные и подтвердить скриншотом

заключение

В процессе прохождения производственной практики по модулю ПМ.03 «Участие в интеграции программных модулей» была изучена предметная область, рассмотрена концепция канбан-досок и их применением в управлении проектами и организацией рабочего процесса

Также в ходе выполнения производственной практики были разработаны диаграммы активностей и прецедентов, реализован функционал работы с канбан-досками, включая создание, редактирование и удаление задач, а также совместную работу с ними. После разработки веб-приложения было проведено тестирование его функциональности.

Все задачи производственной практики были выполнены.

список использованных источников

1. Блог Productstar – Что такое Git и зачем нужен контроль версий [Электронный ресурс]. URL: https://blog.productstar.ru/pochemu-razrabotchiki-polzuyutsya-git/ (Дата обращения 29.05.24);

приложение а

Ссылка на репозиторий: https://github.com/kemmed/KanbanApp.git

Код главной формы