ГУАП

КАФЕДРА № 82

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доцент, кандидат техн. наук |  |  |  | Е.Л. Турнецкая |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2 |
| РАСПРЕДЕЛЕННЫЙ КОНТРОЛЬ ВЕРСИЙ GIT И GITHUB |
| по курсу: МЕТОДОЛОГИЯ И ТЕХНОЛОГИЯ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 8025М |  |  |  | Н.С. Калимов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2020

**Цель работы:**

Познакомиться с распределенной системой контроля версий Git и хостингом репозиториев GitHub.

**Ход работы:**

С GitHub и Git я знаком не так давно – с начала этого года. Когда только начал усиленно заниматься разработкой, весь наш код мы фиксировали именно на GitHub.

GitHub – это хостинг репозиториев, в котором можно хранить любые файлы. Ну, не совсем любые – Git лучше всего работает с файлами меньше 5 МБ. В противном случае, процесс отслеживания дельт файлов замедляется и Git проигрывает централизованным системам контроля версий. Для работы с большими файлами следует воспользоваться надстройкой Git LFS – Large File Storage.

GitHub – пристанище проектов с открытым исходным кодом и коллабораторов-разработчиков (хотя, думаю, и не только их). GitHub позволяет делиться своими наработками и курировать весь процесс коллаборации на своем сайте.

Мы, как пользователи, можем:

* Завести свой репозиторий или зайти на существующий (при наличии прав)
* Склонировать репозиторий себе и производить работу с файлами локально на своем ПК
* Загрузить изменения и составить Pull Request – запрос на вытягивание, который обращается к администраторам репозитория. В пул-реквесте они могут просмотреть предлагаемые изменения и либо отбросить их, либо слить в свою главную ветку.
* Общаться, документировать, делиться кодом, налаживать продолжительную поставку контента (CI/CD) и многое другое.

В ходе этой лабораторной работы я закрепил навыки работы с Git и GitHub, а также завел репозиторий для выполнения лабораторных работ по дисциплине МИТПИС. Для демонстрации результатов, я позвал своего одногруппника Александра Гловацкого и мы вместе проделали типичный порядок действий для коллаборации на гитхабе.

Первым делом, откроем главную страницу github.com.

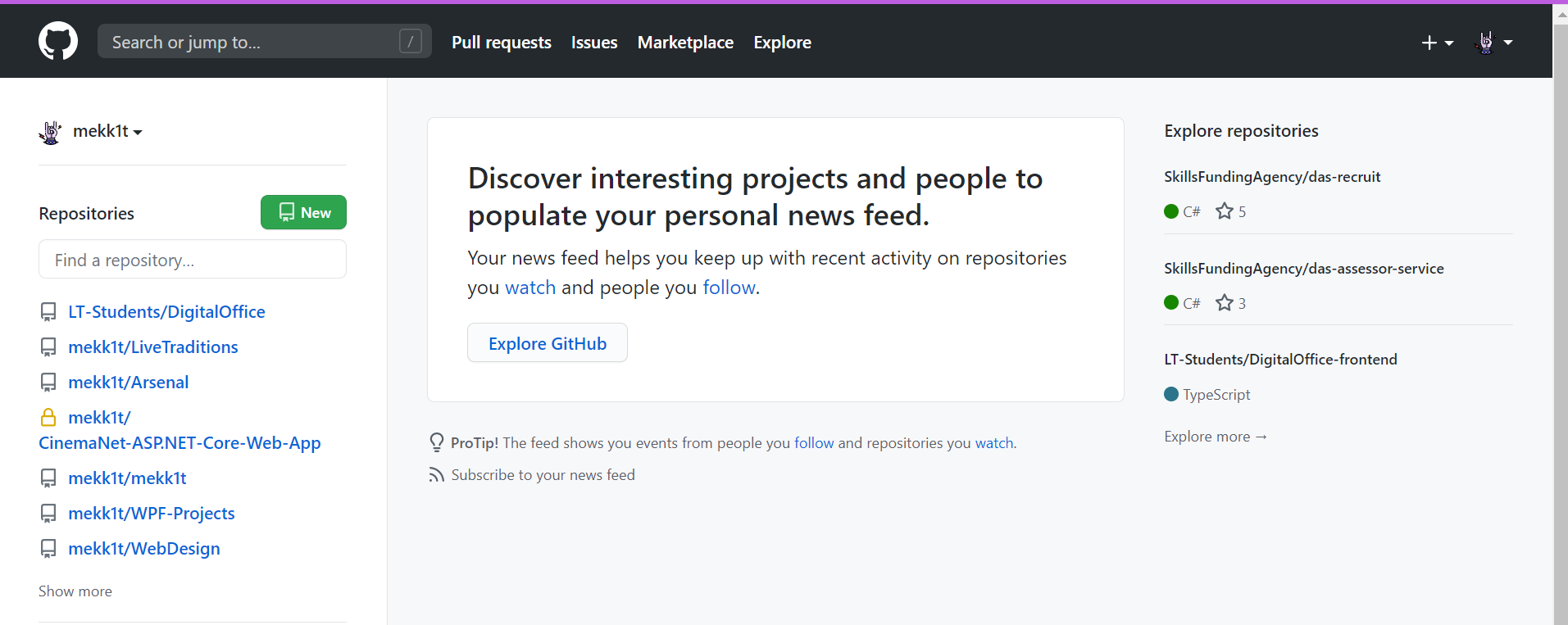


Рисунок 1 – главная страница GitHub

Затем зайдем в My Repositories и щелкнем New. Начнется процесс создания репозитория.

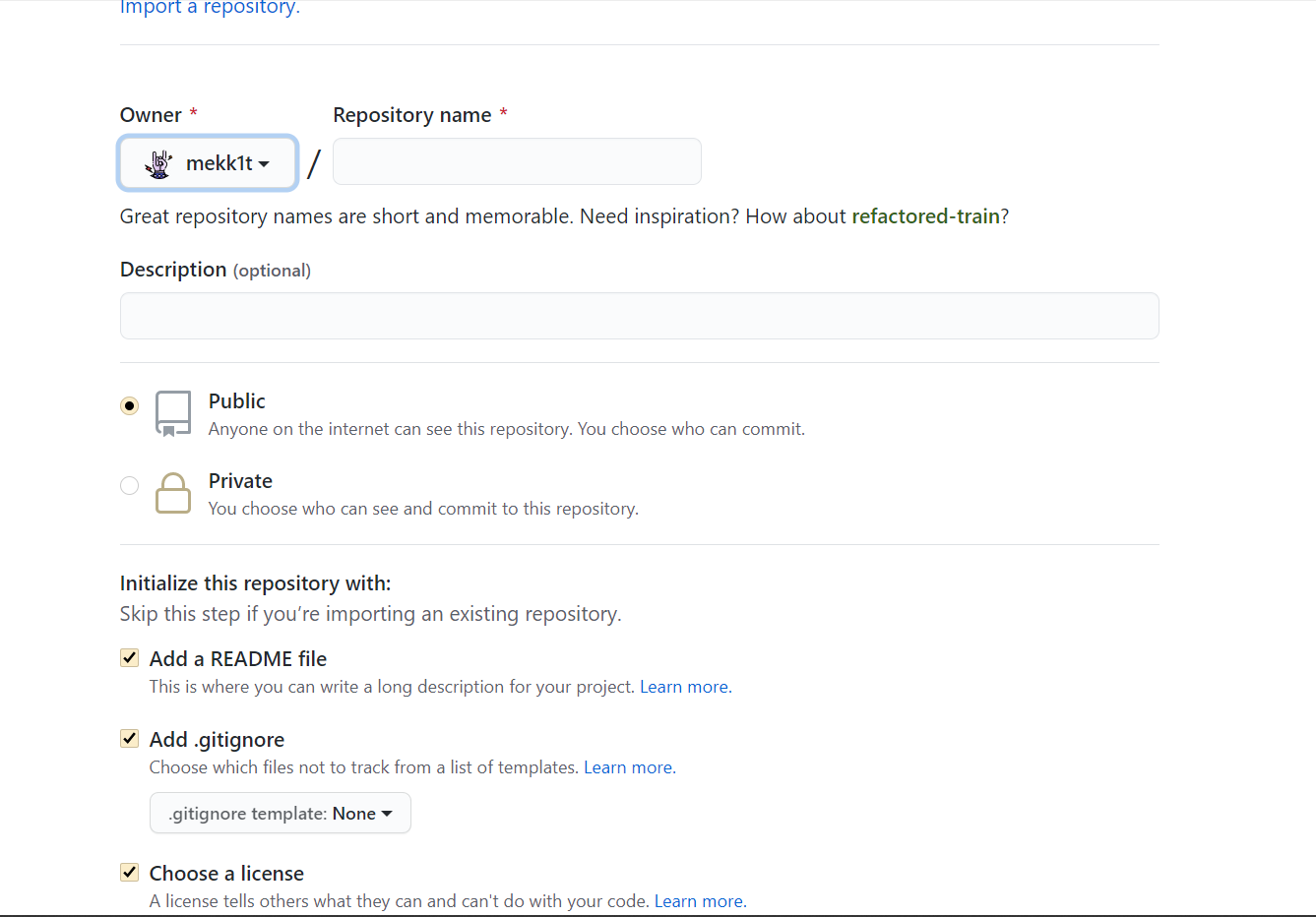


Рисунок 2 – создание репозитория

После создания репозитория, его необходимо склонировать к себе на ПК. Для этого я воспользуюсь интерфейсом GitHub Desktop:

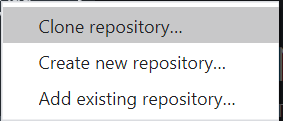


Рисунок 3 – клонирование репозитория

Теперь мне нужен соучастник! В этом мне поможет Александр Гловацкий, я приглашаю его через следующую форму:

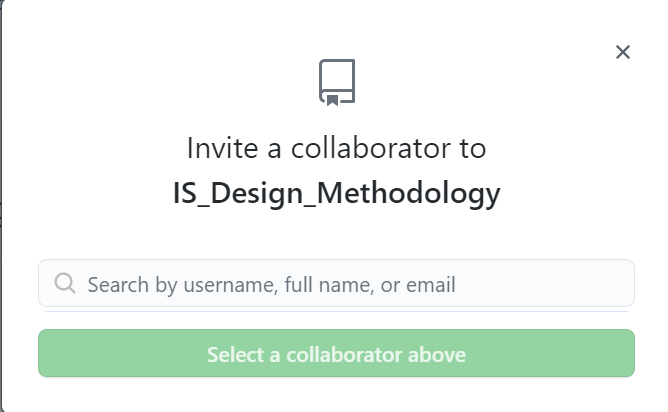


Рисунок 4 – приглашение людей для коллаборации

Открою GitHub Desktop. В нем пока открыт мой другой репозиторий – личный проект домашней книги рецептов.

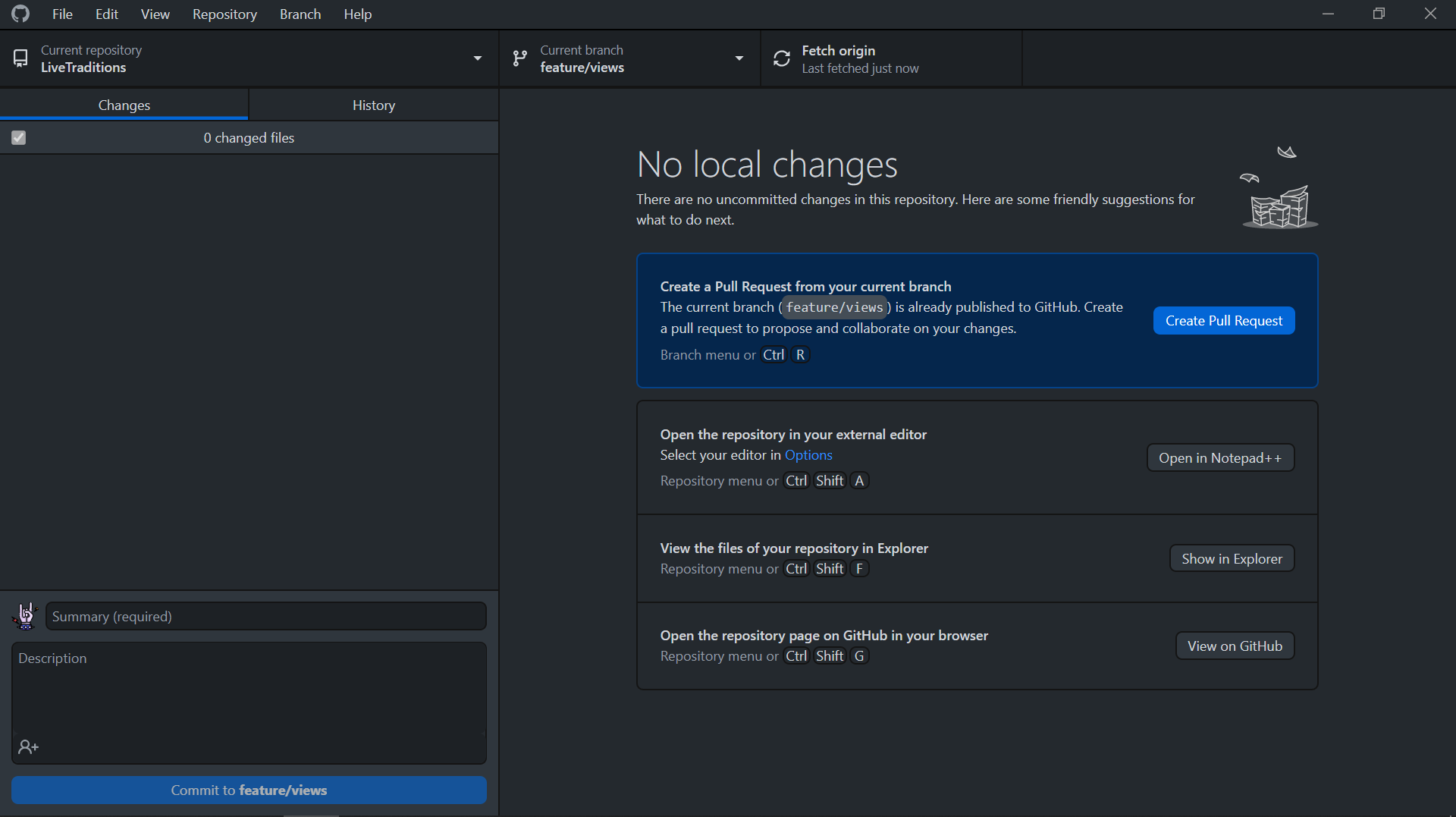


Рисунок 5 – клиент GitHub Desktop

Переключаюсь на свой репозиторий для выполнения лабораторных по дисциплине МИТПИС, создаю ветку lab2.

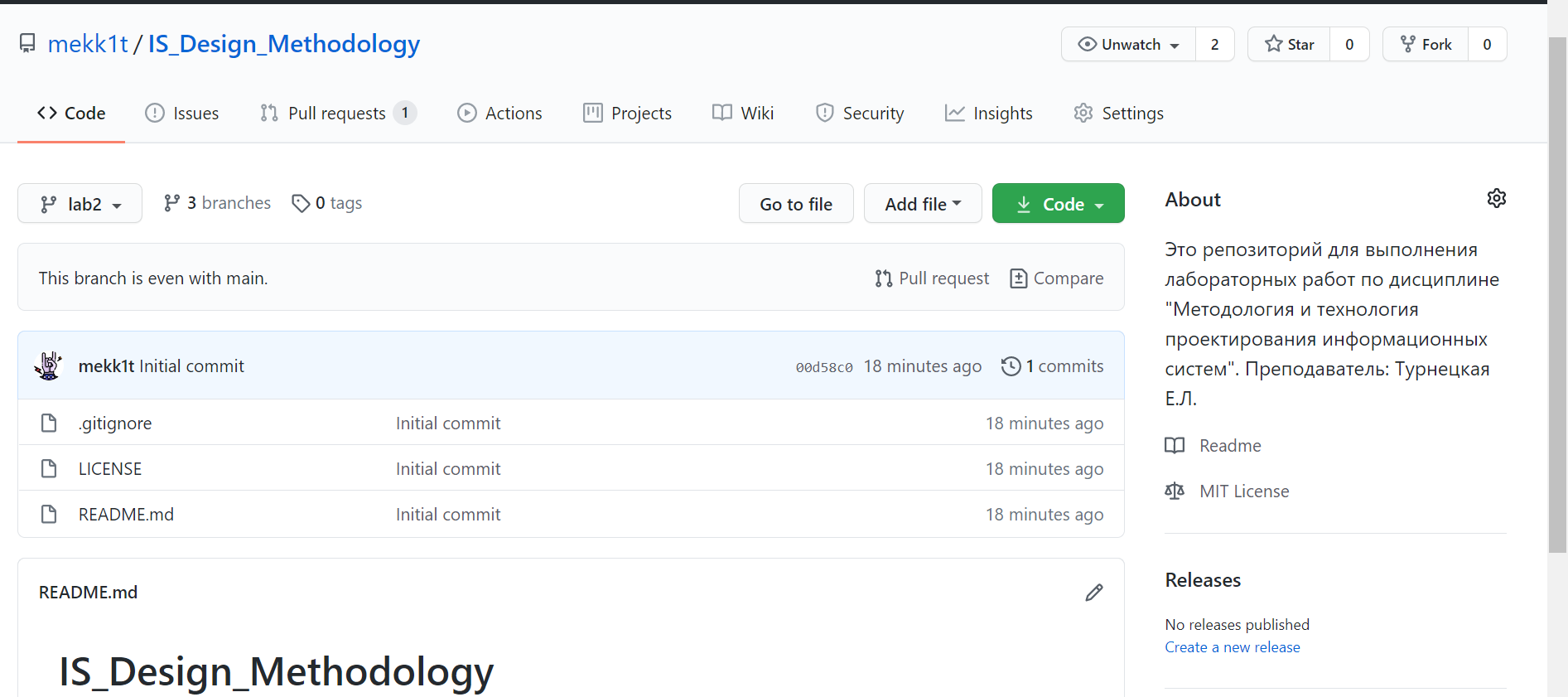


Рисунок 6 – ветка lab2 для выполнения лабораторной работы №2.

Затем прошу Александра внести какие-либо изменения в репозитории. Он предпочел создать текстовый файл в своей ветке lab2.Clone

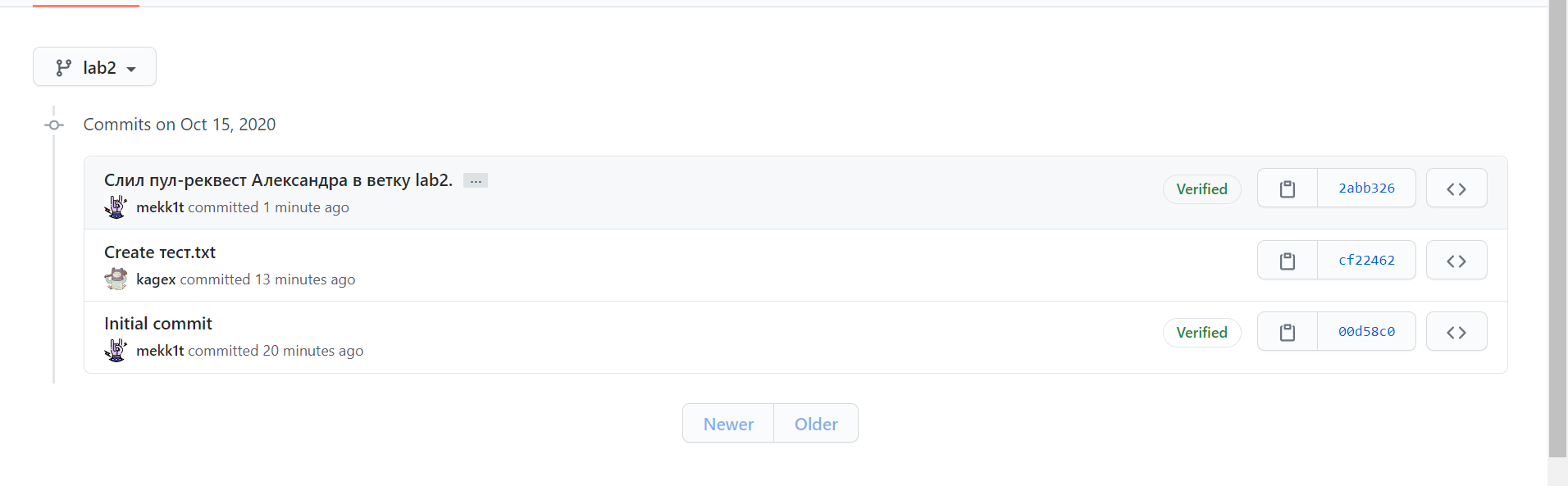


Рисунок 7 – коллаборация с Александром Гловацким под ником kagex

Я вижу этот Pull Request, отсматриваю предложенный Александром файл, комментирую эти изменения и сливаю их в ветку lab2. Затем удаляю ненужную ветку.

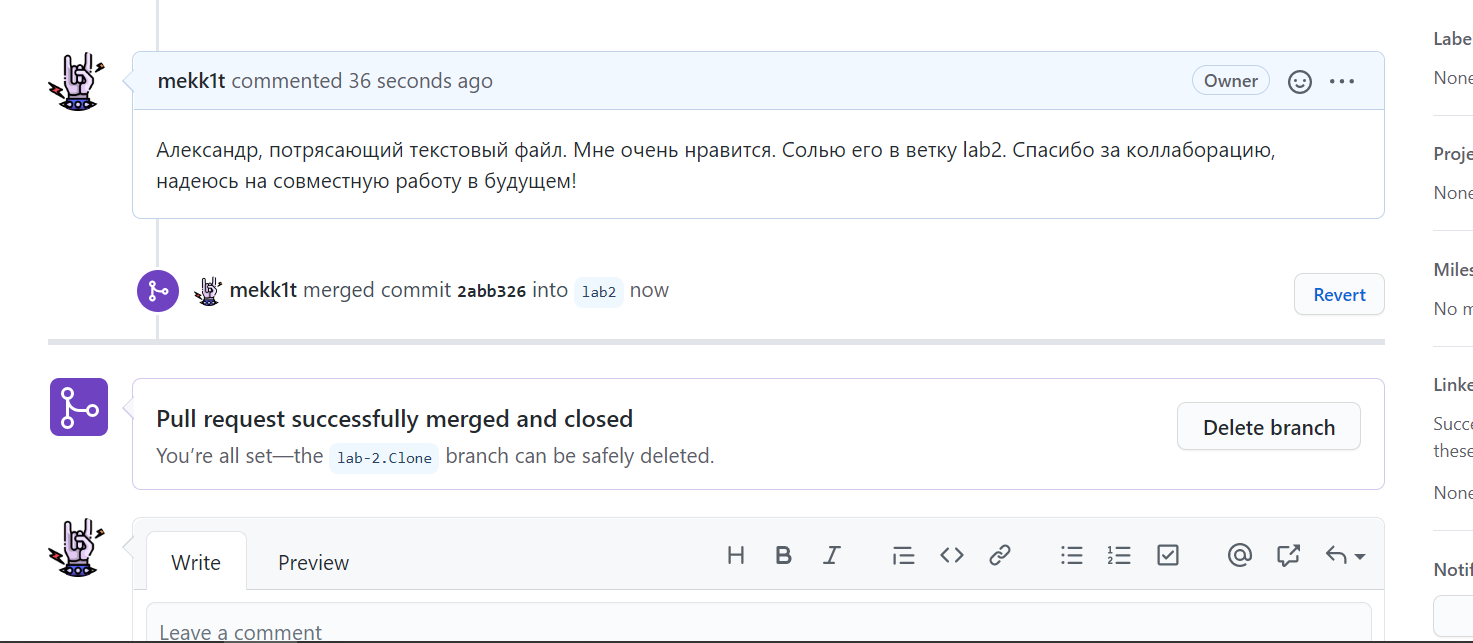


Рисунок 8 – слияние пул-реквеста из локальной ветки Александра в ветку lab2

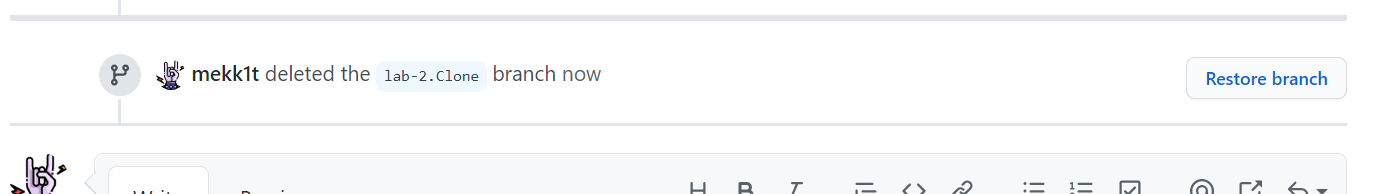


Рисунок 9 – удаление ненужной ветки

Я вижу, что файл Александра – пустой. Добавляю в него текст и повторяю процесс с пул-реквестом.

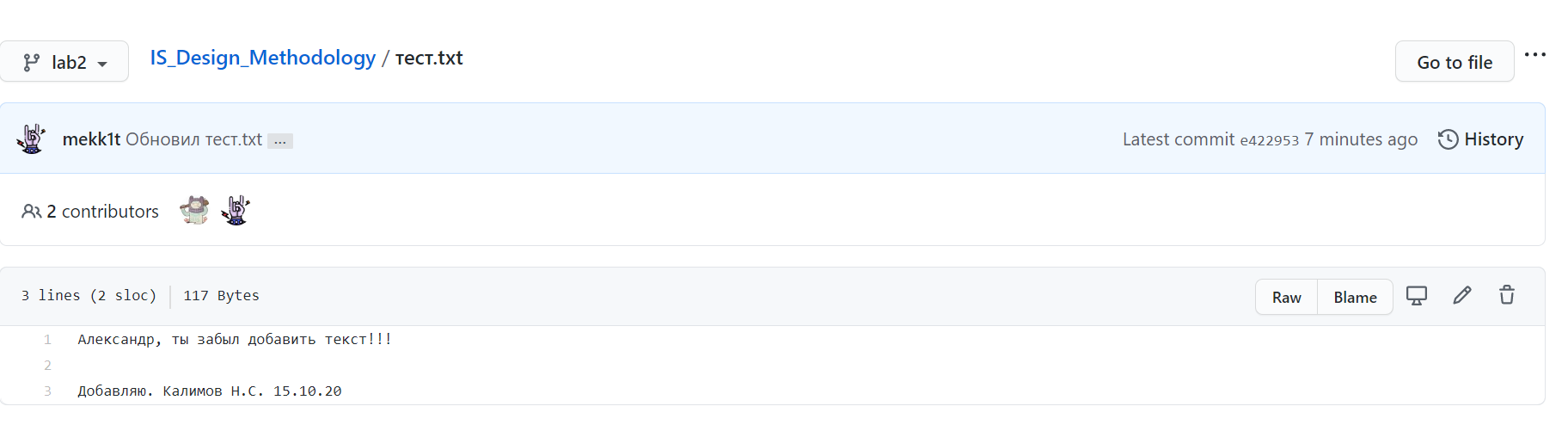


Рисунок 10 – моя очередь обновлять файл и сливать изменения

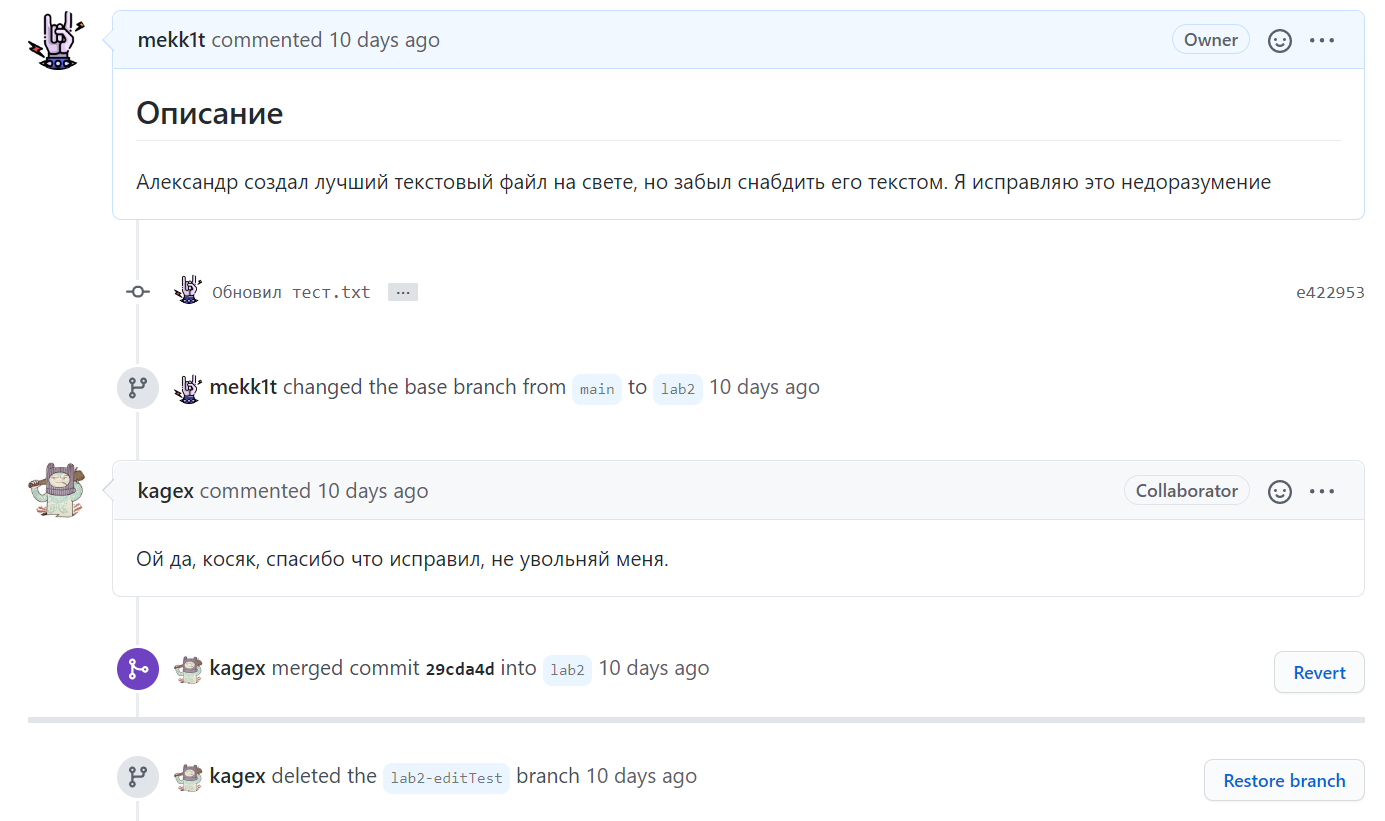


Рисунок 11 – демонстрация процесса слияния изменений

Вот так выглядит совместная работа на GitHub. Также для работы с гитом я предпочитаю иногда использовать GUI-клиенты: для визуализации состояния веток. Очень полезно, когда веток становится очень много. Например, GitKraken:

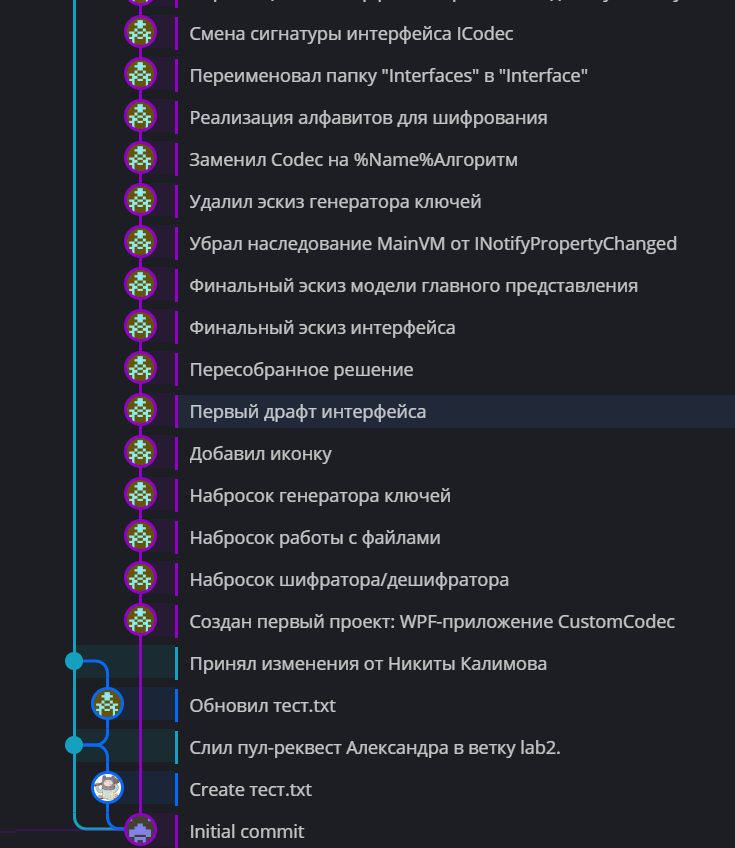


Рисунок 12 – визуализация ветвлений в GUI-клиенте GitKraken

Также здесь на рисунке 12 можно заметить параллельную ветку с изменениями – это ветка lab1, в которой я фиксировал историю изменений при разработке приложения шифратора / дешифратора.

**Вывод:**

Я закрепил навыки работы с Git и GitHub, а также завел репозиторий для хранения лабораторных работ.