

Федеральное государственное автономное
образовательное учреждение высшего
образования
«Национальный исследовательский университет
ИТМО»

Факультет Информационных технологий и программирования

Работа: Лабораторная работа по Git 3

Выполнил: Гонтарь Тимур Сергеевич
Проверил: Повышев Владислав Вячеславович

Санкт-Петербург
2022 г.

Отличия GitHub и GitLab

GitHub

GitHub – это онлайн-хостинг для репозиторий. Он представляет собой облачное хранилище файлов на удаленном сервере, куда сохраняются все новые и измененные версии программного кода. Таким образом, если Git является инструментом, то GitHub выступает сервисом, позволяющим использовать этот инструмент. Он считается наиболее крупной и известной в интернете платформой для хостинга IT-проектов и совместной работы над ними.

GitHub был разработан специально под Git, поэтому он имеет такие функции, как:

- Все функции системы распределенного контроля версий и гибкие возможности для управления исходным кодом;
- Система контроля доступа, возможности для управления задачами и ведения вики-страниц для проектов, опции багтрекинга;
- Доступ к репозиториям через внутренний интерфейс, командную строку и Git-команды;
- Ведение документации, история коммитов, рассылка уведомлений на email, запросы на принятие изменений, упоминания пользователей и графики;
- Создание форка (копии) удаленного репозитория на сервере GitHub и интерактивная подготовка (управление областью подготовленных файлов).

GitLab

GitLab является еще одним известным онлайн-сервисом для хранения git-репозиторий и работы с ними. В его основе находится система баз данных, что несколько усложняет установку GitLab, однако решить возможные проблемы пользователей помогут подробные инструкции.

Хранилище репозиторий GitLab можно развернуть как в облаке через официальный сайт проекта, так и на локальном сервере. Существует несколько вариантов установки БД: через установочный пакет Omnibus GitLab, из исходных файлов, при помощи облачного провайдера (Google Cloud Platform, AWS, OpenShift, Azure и т.д.) и ряд других способов.

Что касается функционала GitLab, то он предоставляет такие возможности:

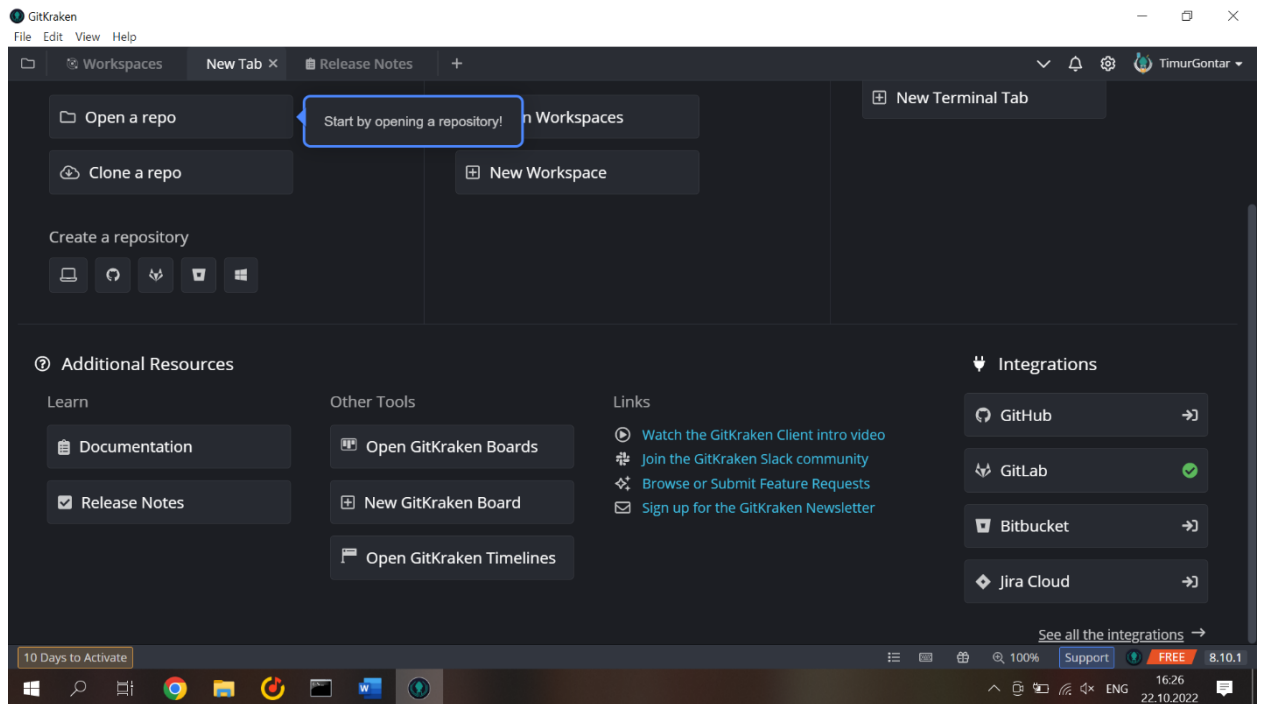
- Создание, хранение и управление публичными и приватными репозиториями;
- Организация совместного доступа к репозиториям, регулирование прав доступа пользователей и групп;
- Интеграция с CI-системами (Jenkins и т.д.);
- Отслеживание изменений, комментарии к проектам, создание вики-страниц, досок идей и задач, интеграция со сторонним ПО через API и т.д.;
- Инструменты аутентификации и авторизации, аналитика продуктивности и трекинг задач, мониторинг ошибок и контроль времени, управление исходным кодом и веб-консоль для его редактирования, поддержка непрерывной интеграции (CI), тестирование и проверка качества кода, управление репозиториями и контейнерами, управление уязвимостями и сканирование зависимостей.

В целом, различия GitHub vs GitLab резюмировали в этой таблице:

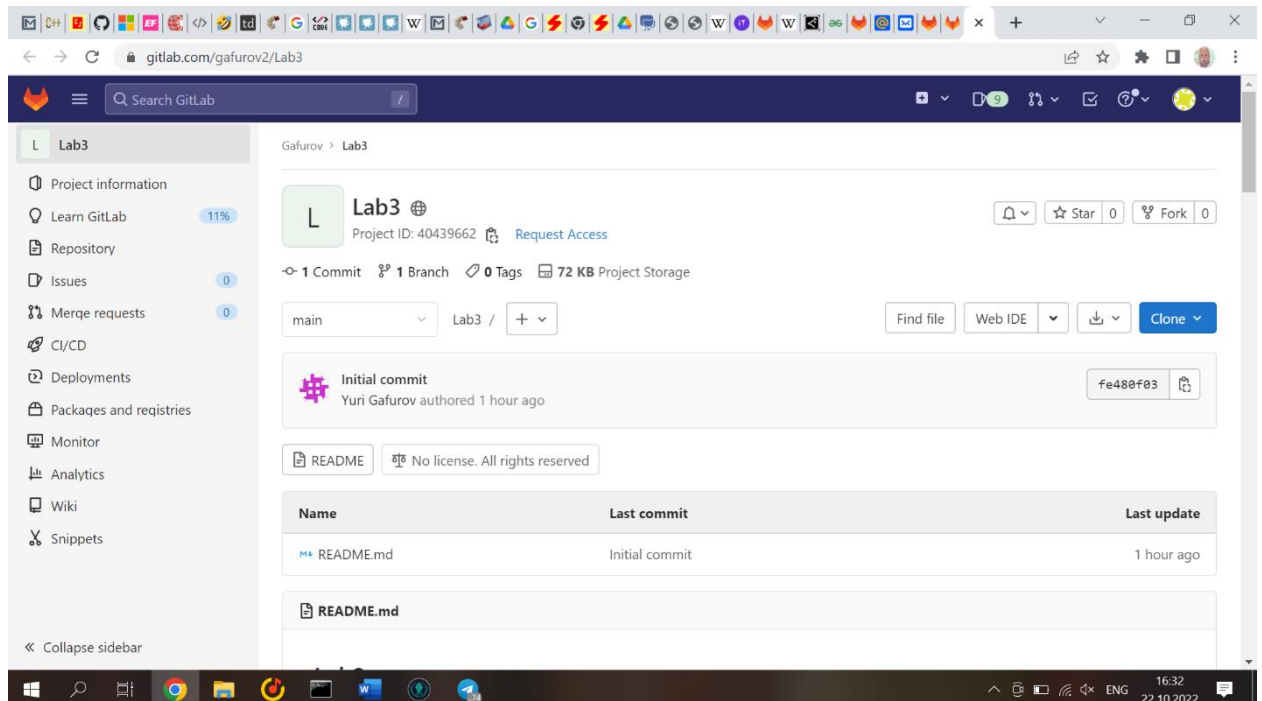
Характеристики	GitLab	GitHub
Создание	сентябрь 2011	Апрель 2008
Бесплатный план	Неограниченные публичные и частные репозитории	Бесплатно только для публичных репозиториях
Платные планы	От 19 долларов США за пользователя в год для премиум-плана. Или 99 долларов США на пользователя в год для Ultimate.	От 4 долларов США за пользователя в год для Team, 21 доллар США для Enterprise или больше для One.
Функции проверки кода	да	да
Wiki	да	да
Отслеживание ошибок и проблем	да	да
Private репозитории	да	да
Система сборки	да	да (со сторонним сервисом)
Импорт проектов	да	Нет
Экспорт объектов	да	Нет
Учет времени	да	Нет
веб хостинг	да	да
Самостоятельный хостинг	да	да (с бизнес-планом)
Популярность	546.000+ проектов	69.000.000+ проектов

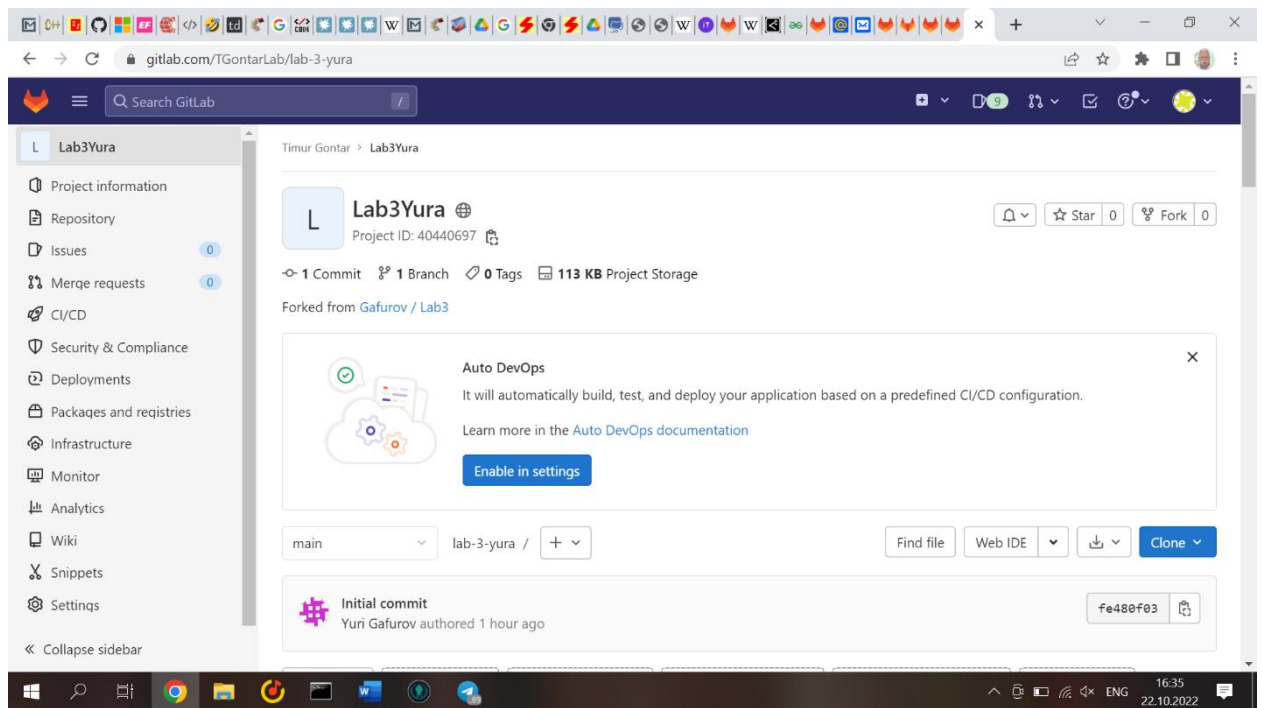
Работа с графическим интерфейсом

1) Запускаю Git Kraken, привязываю его к аккаунту GitLab

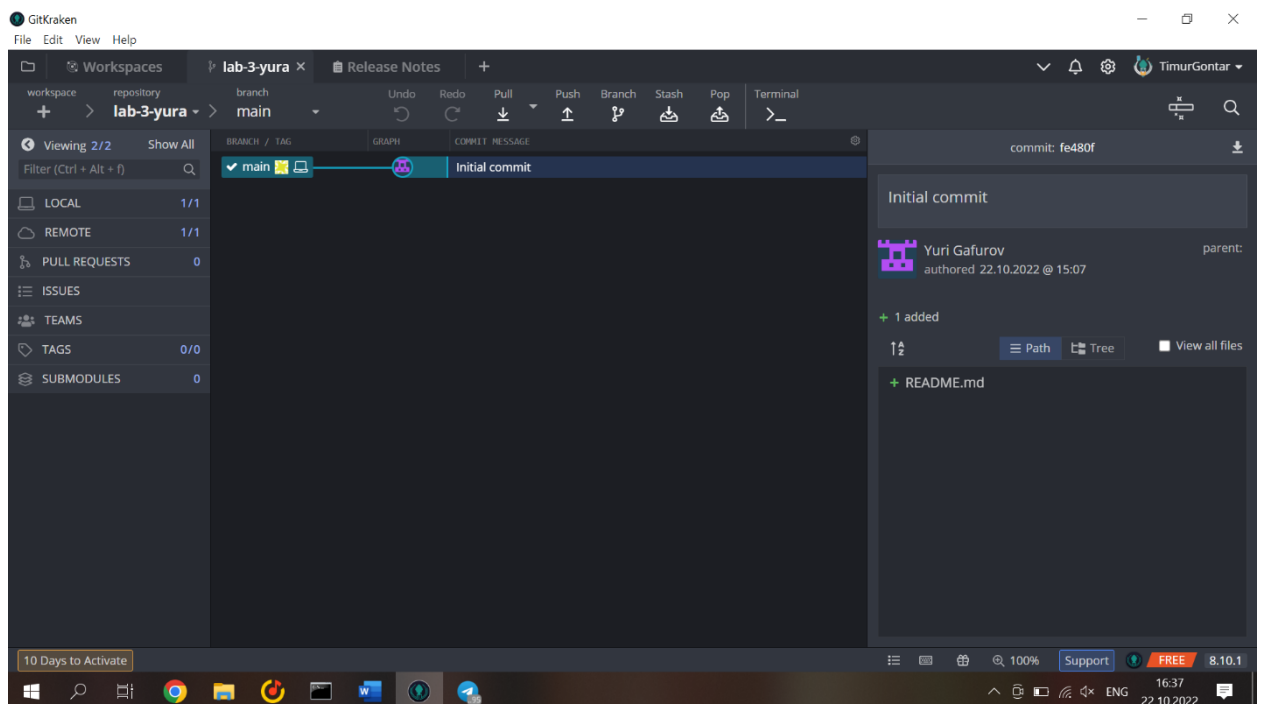


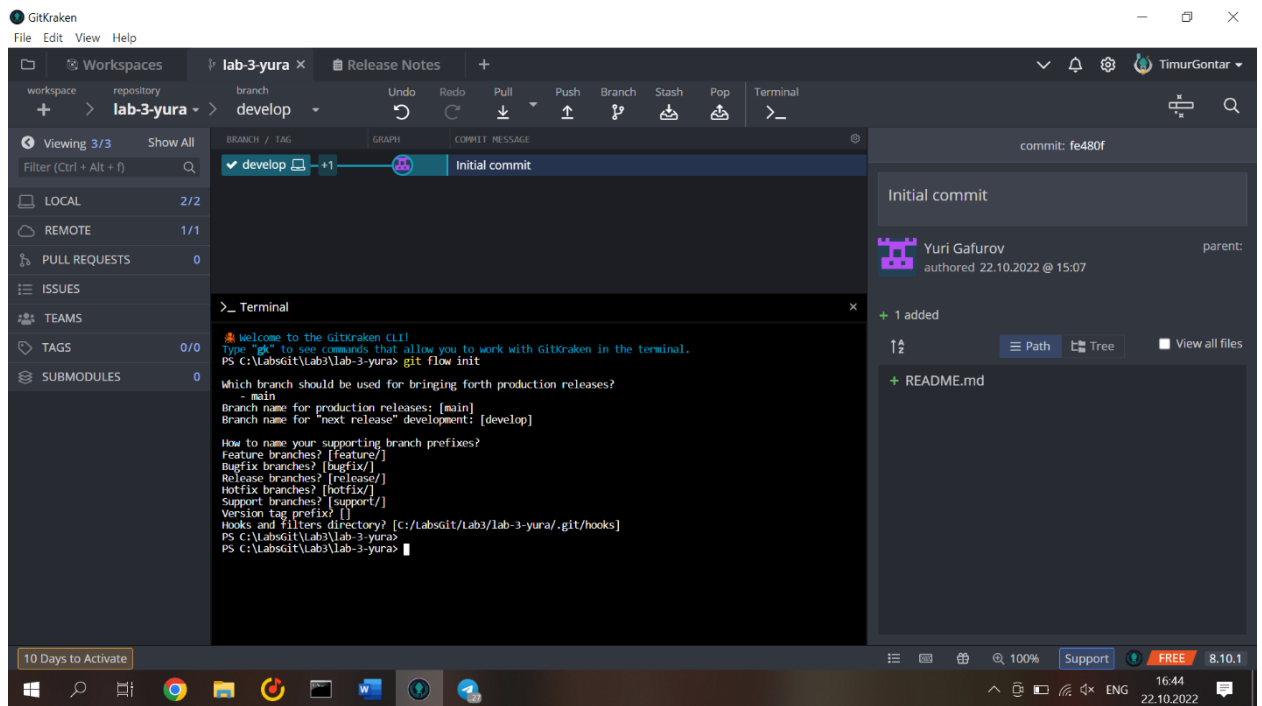
2) Напарник создаёт репозиторий, я беру оттуда Fork чтобы впоследствии сделать pull request



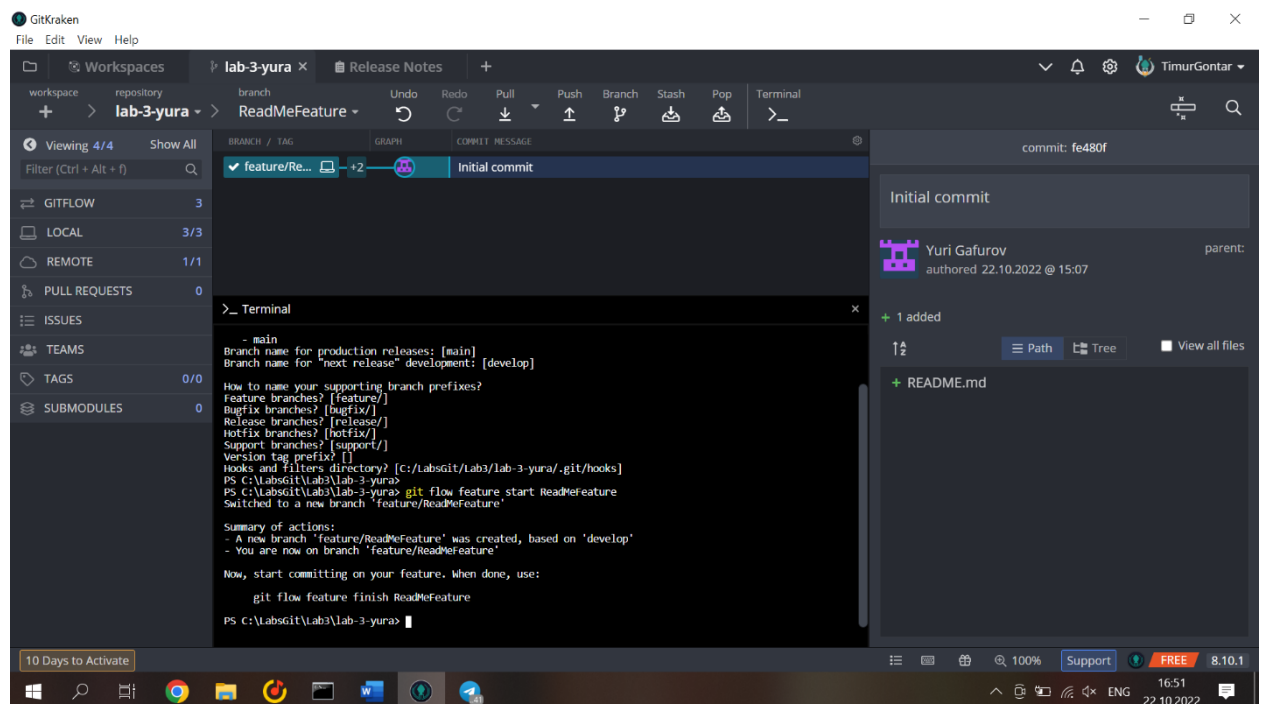


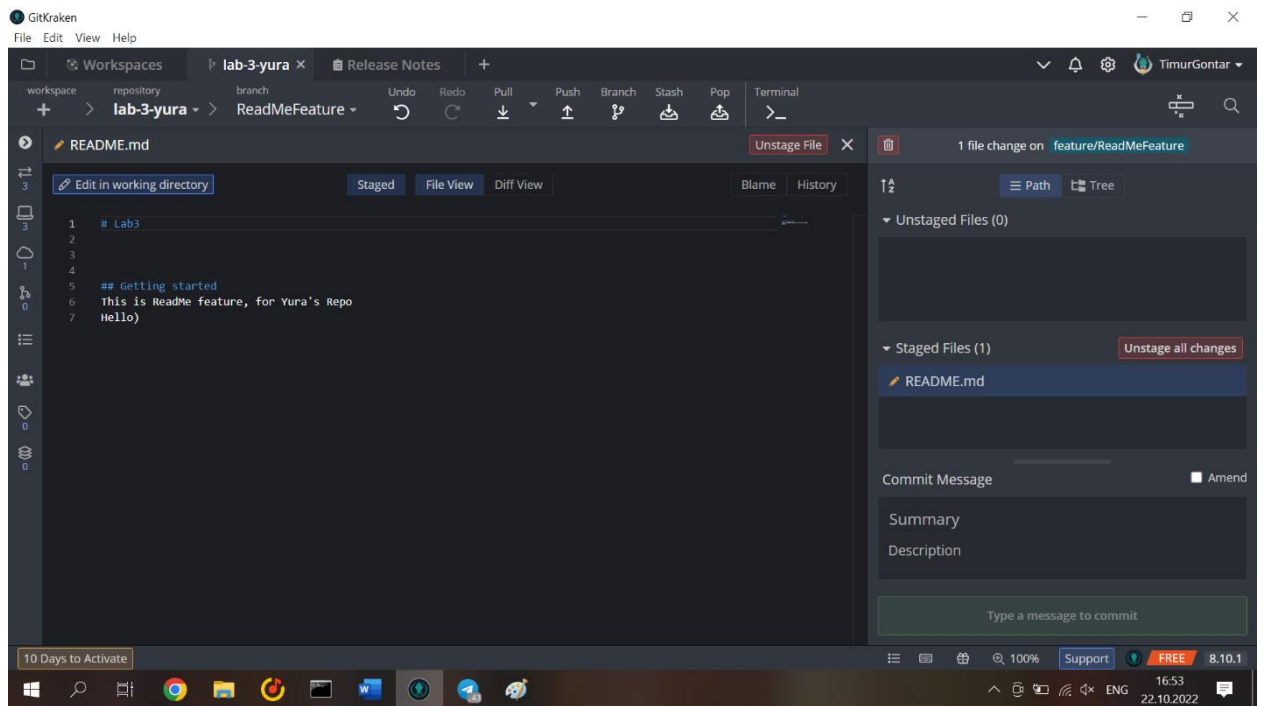
3) Клонировать репозиторий к себе, работать в Git Kraken. Инициализировать git flow в нём





- 4) Создаю новую фичу, делаю изменения в файле readme и коммичу это. Затем завершаю работу с фичей.





```
PS C:\LabsGit\Lab3\lab-3-yura> git flow feature finish ReadMeFeature
Switched to branch 'develop'
Updating fe480f0..141142c
Fast-forward
 README.md | 89 ++-----
 1 file changed, 2 insertions(+), 87 deletions(-)
Deleted branch feature/ReadMeFeature (was 141142c).

Summary of actions:
- The feature branch 'feature/ReadMeFeature' was merged into 'develop'
- Feature branch 'feature/ReadMeFeature' has been locally deleted
- You are now on branch 'develop'

PS C:\LabsGit\Lab3\lab-3-yura>
```

- 5) Запуская релиз Release1, изменяю там readme и так же коммичу это. Завершаю работу с релизом, создаю соответствующий тег

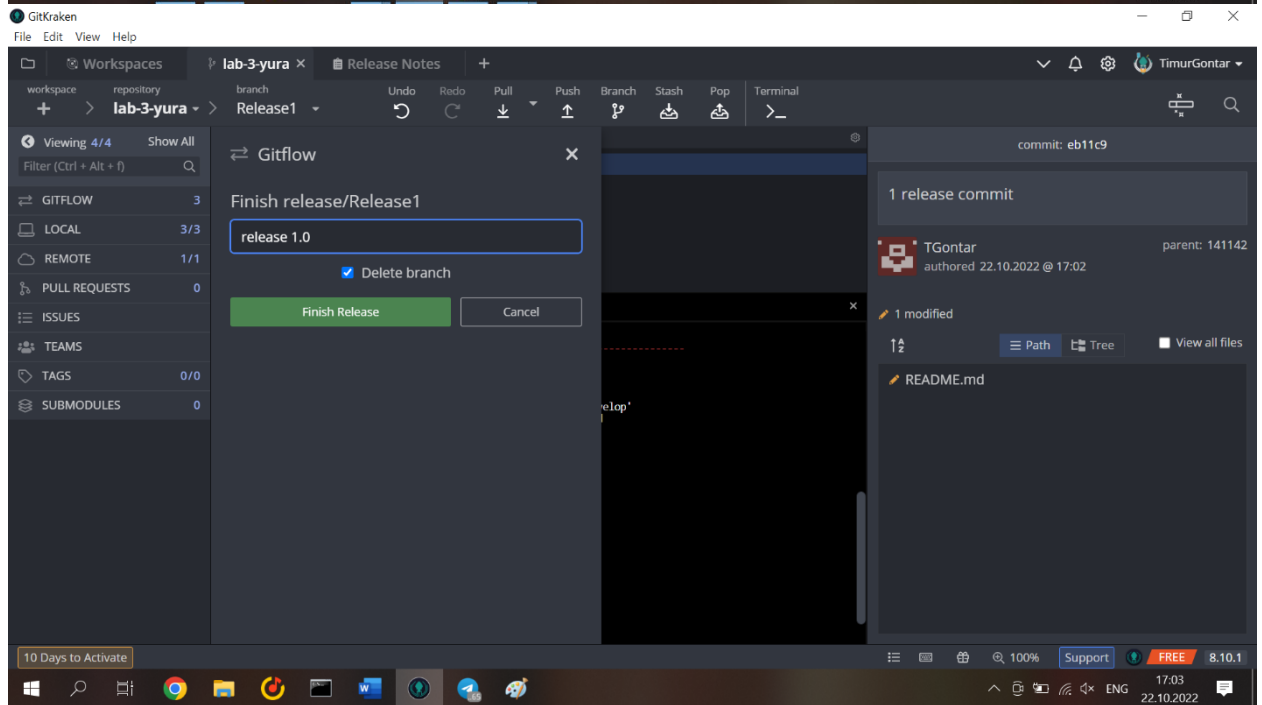
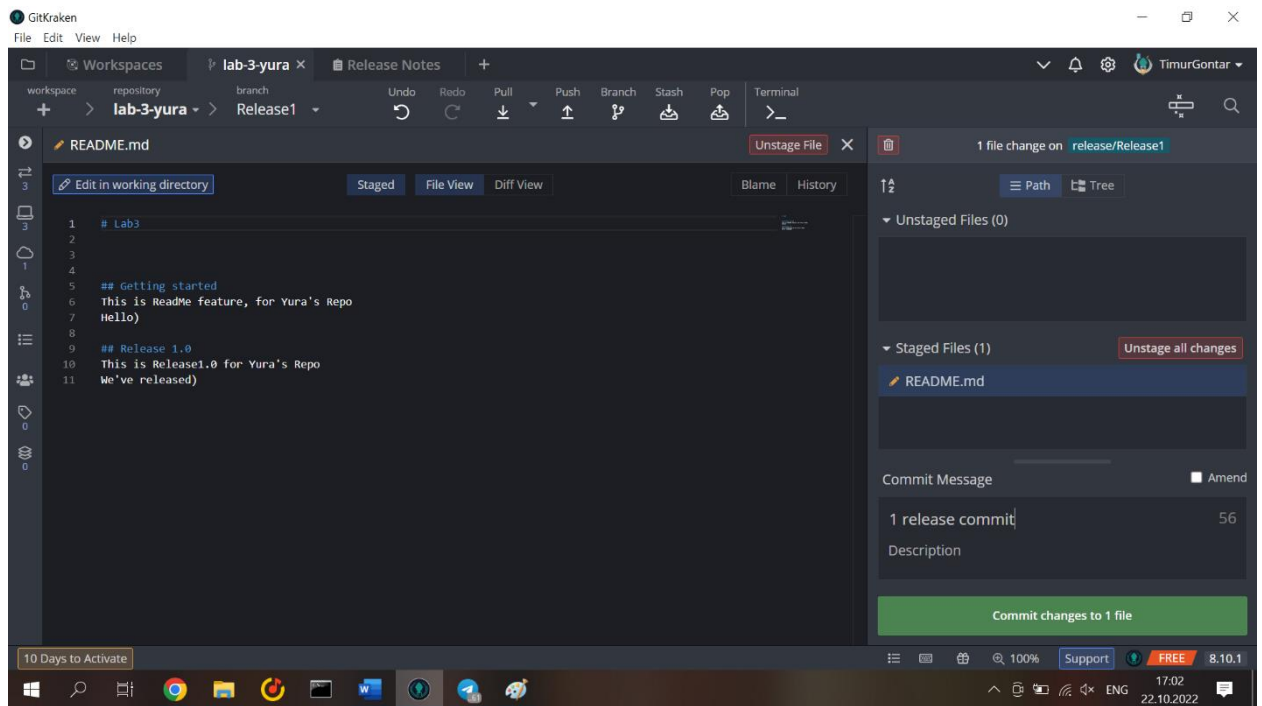
```
PS C:\LabsGit\Lab3\lab-3-yura> git flow release start Release1
Switched to a new branch 'release/Release1'

Summary of actions:
- A new branch 'release/Release1' was created, based on 'develop'
- You are now on branch 'release/Release1'

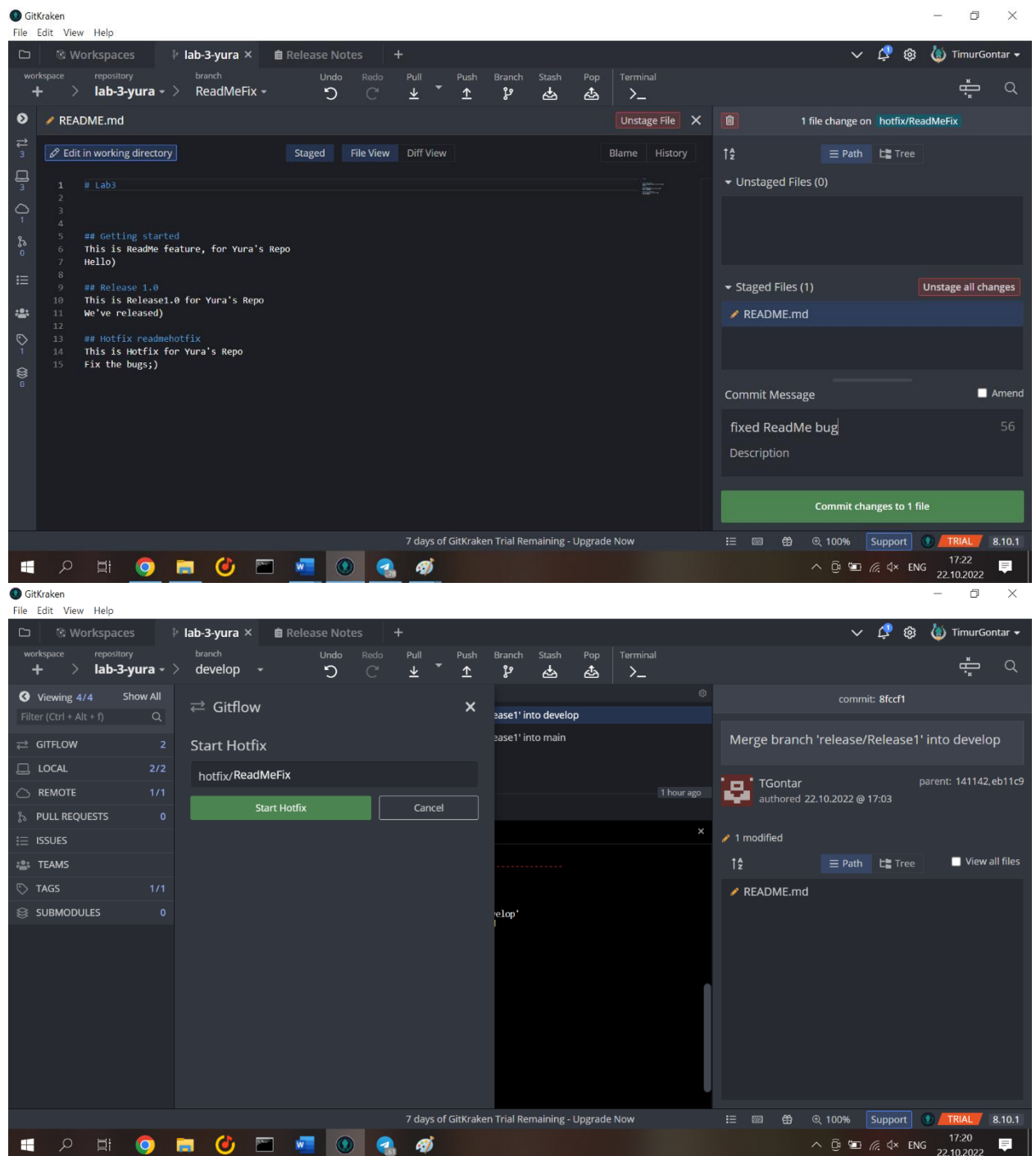
Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

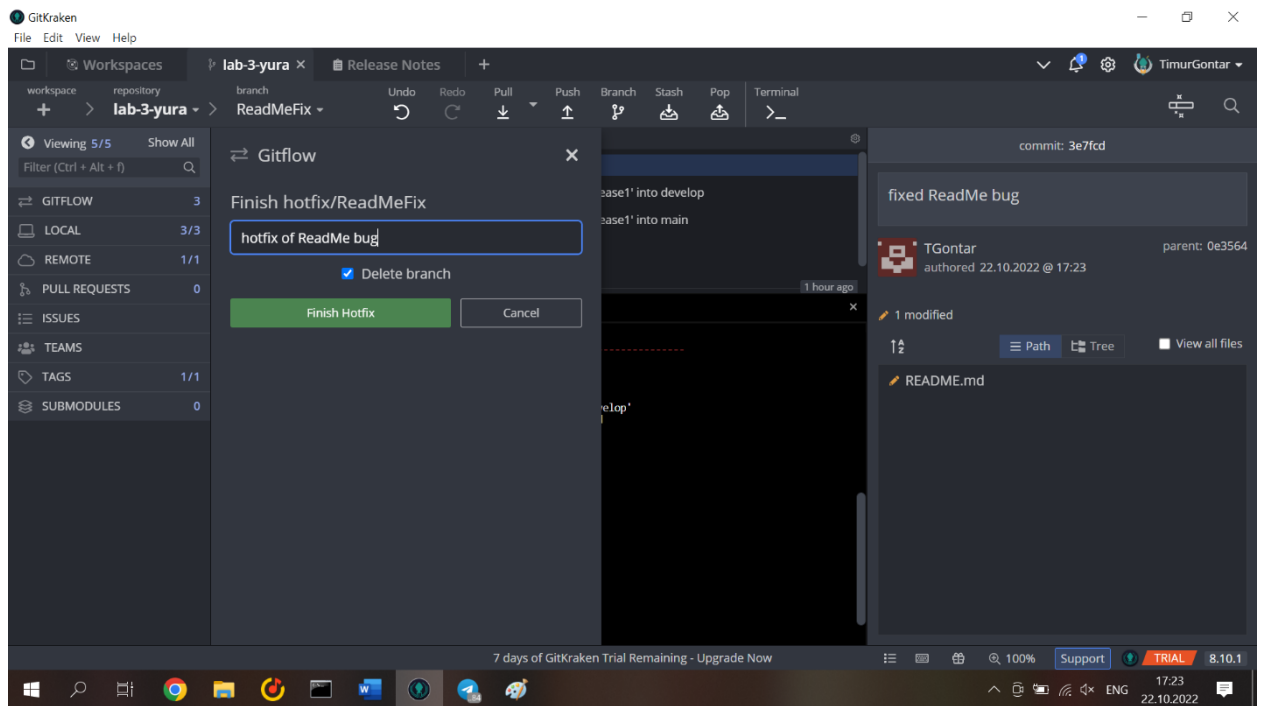
    git flow release finish 'Release1'

PS C:\LabsGit\Lab3\lab-3-yura>
```

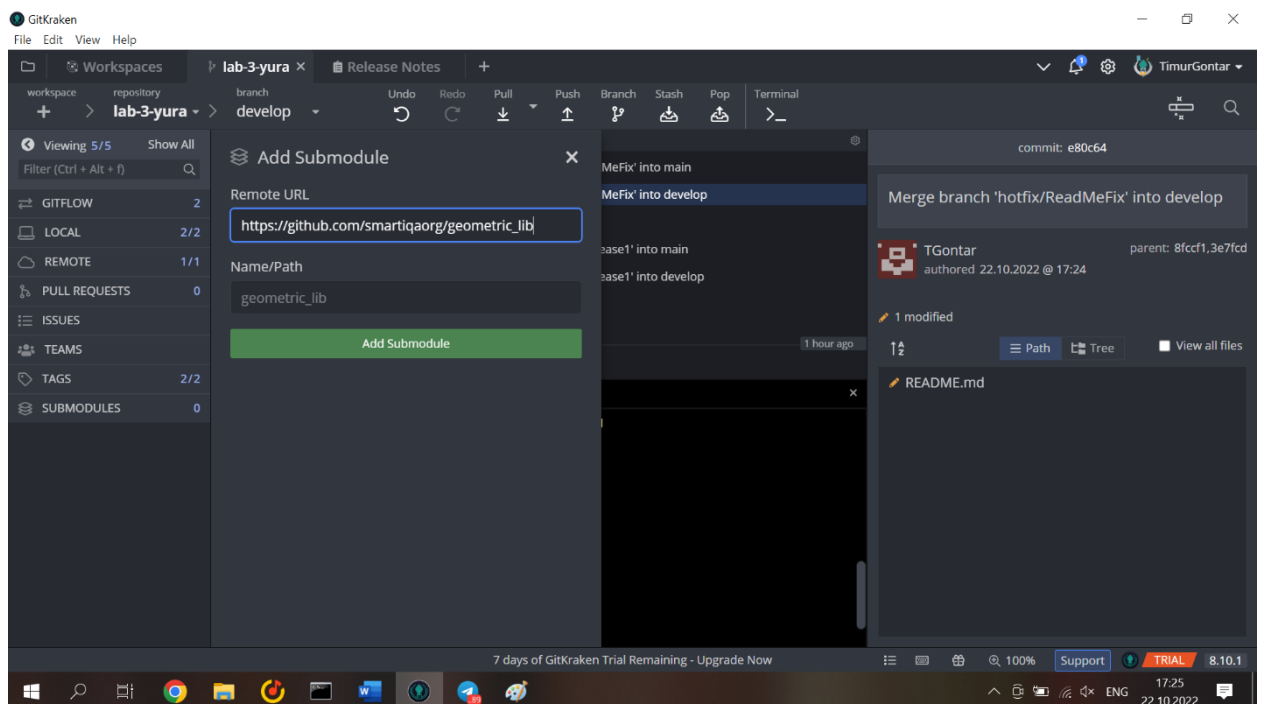


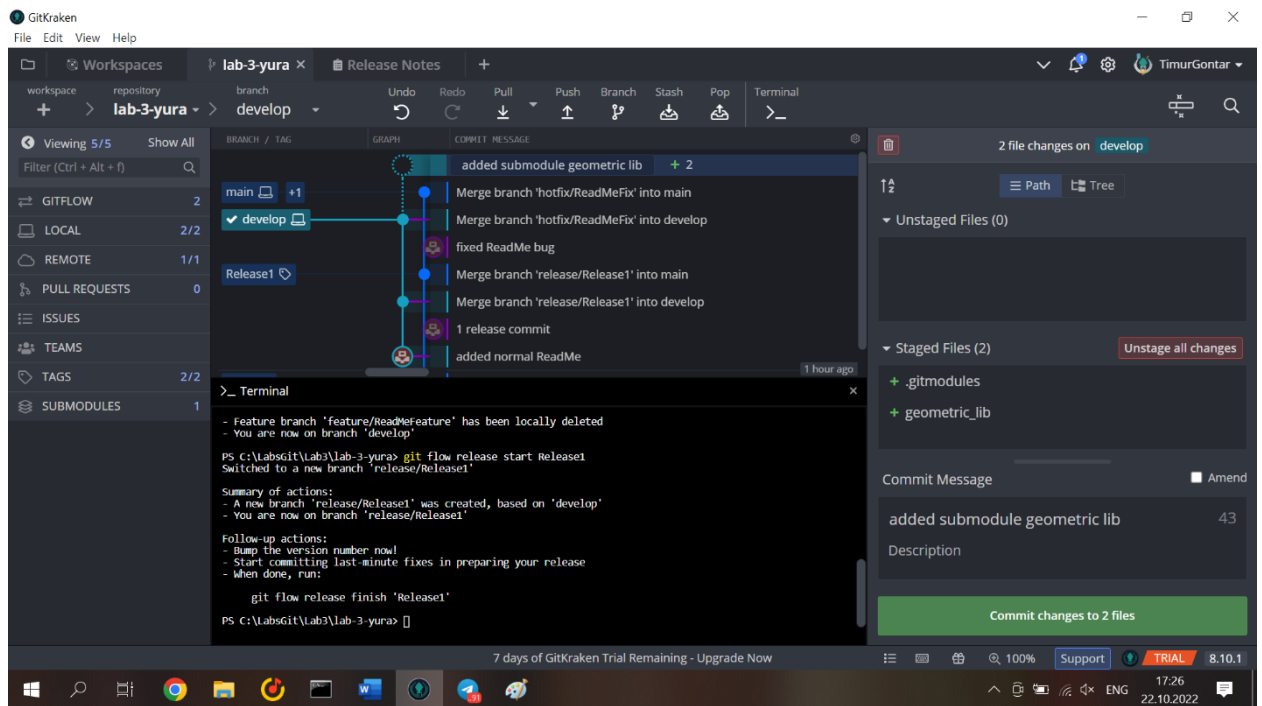
- 6) Инициализирую hotfix и так же изменяю там readme. Завершаю хотфикс и создаю соответствующий тег.



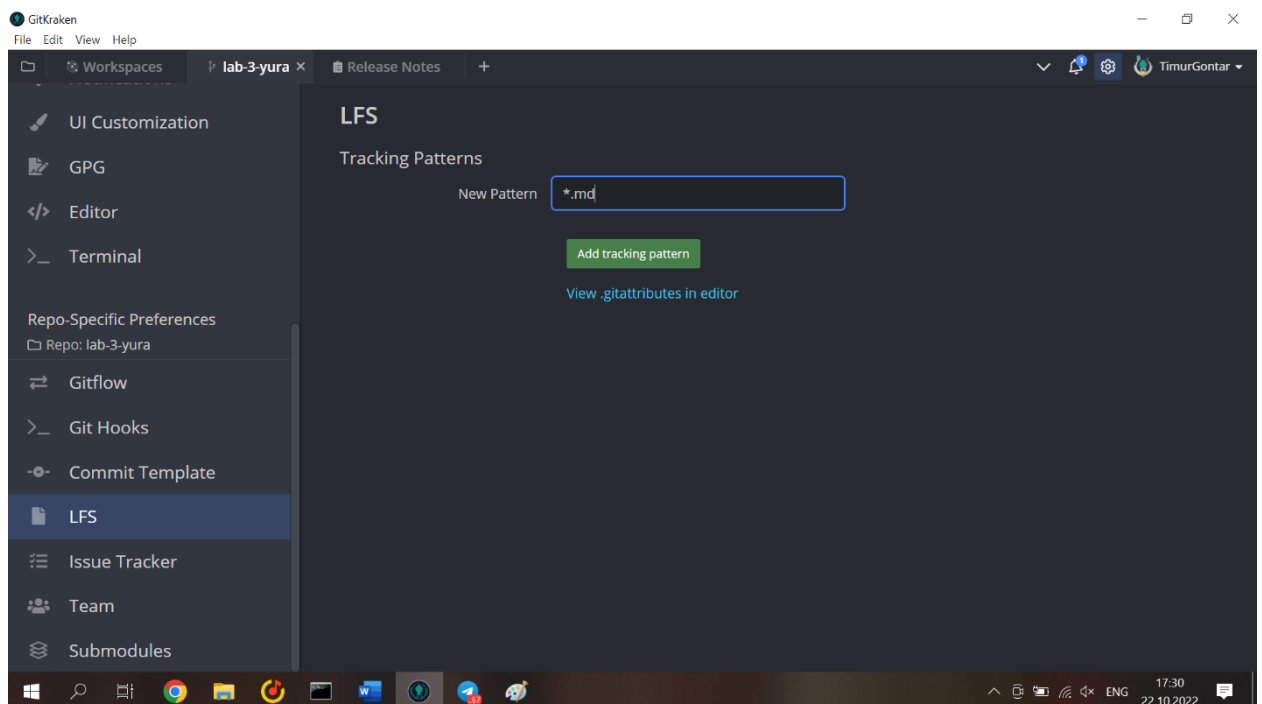


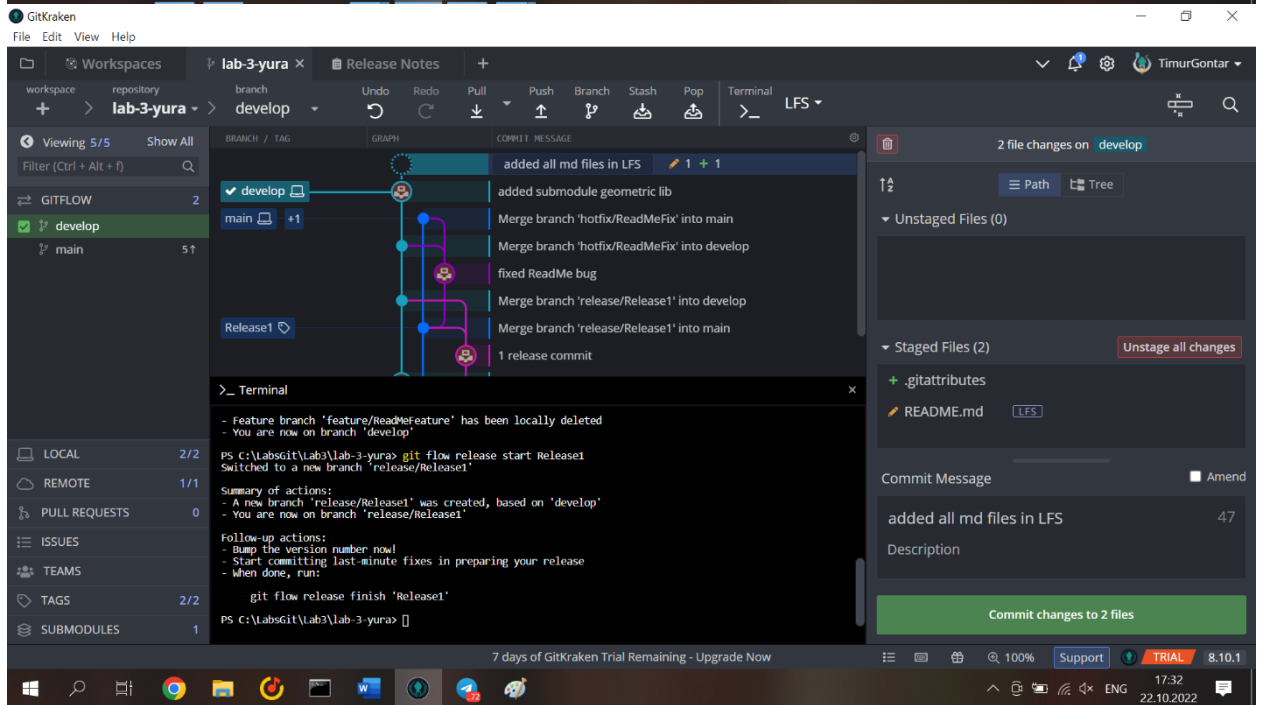
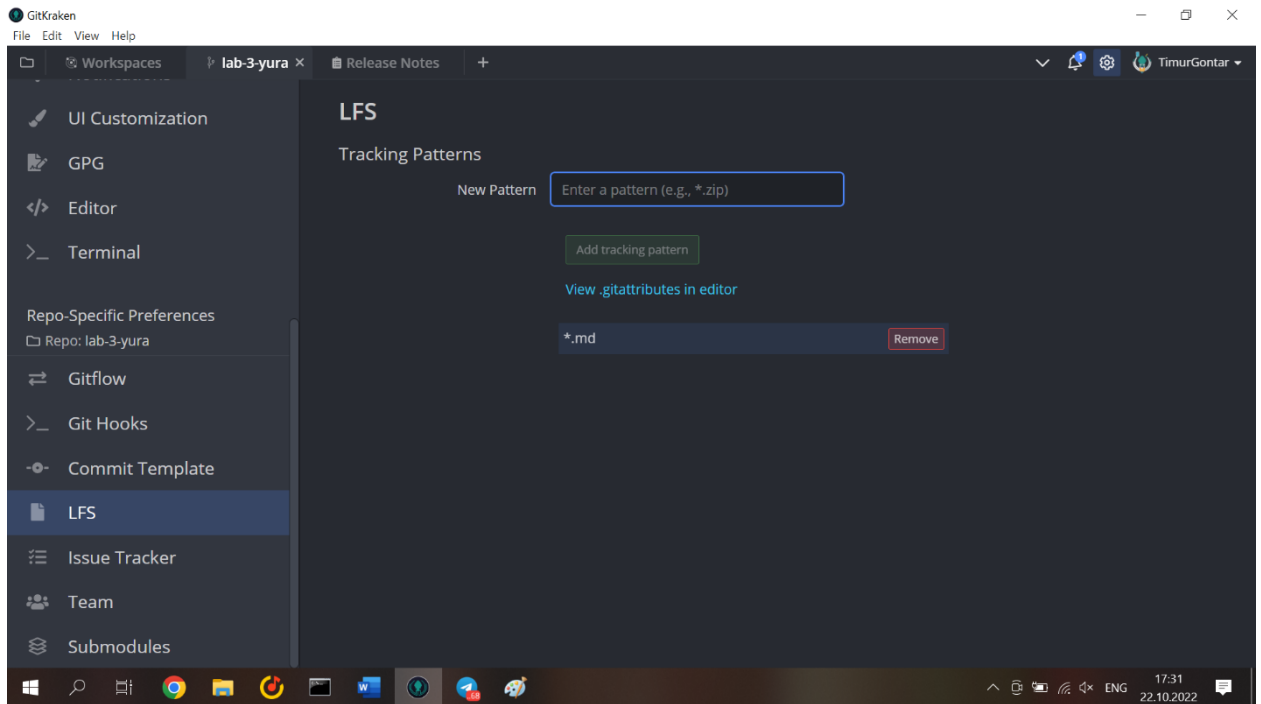
7) Добавляю сабмодуль, коммичу это.



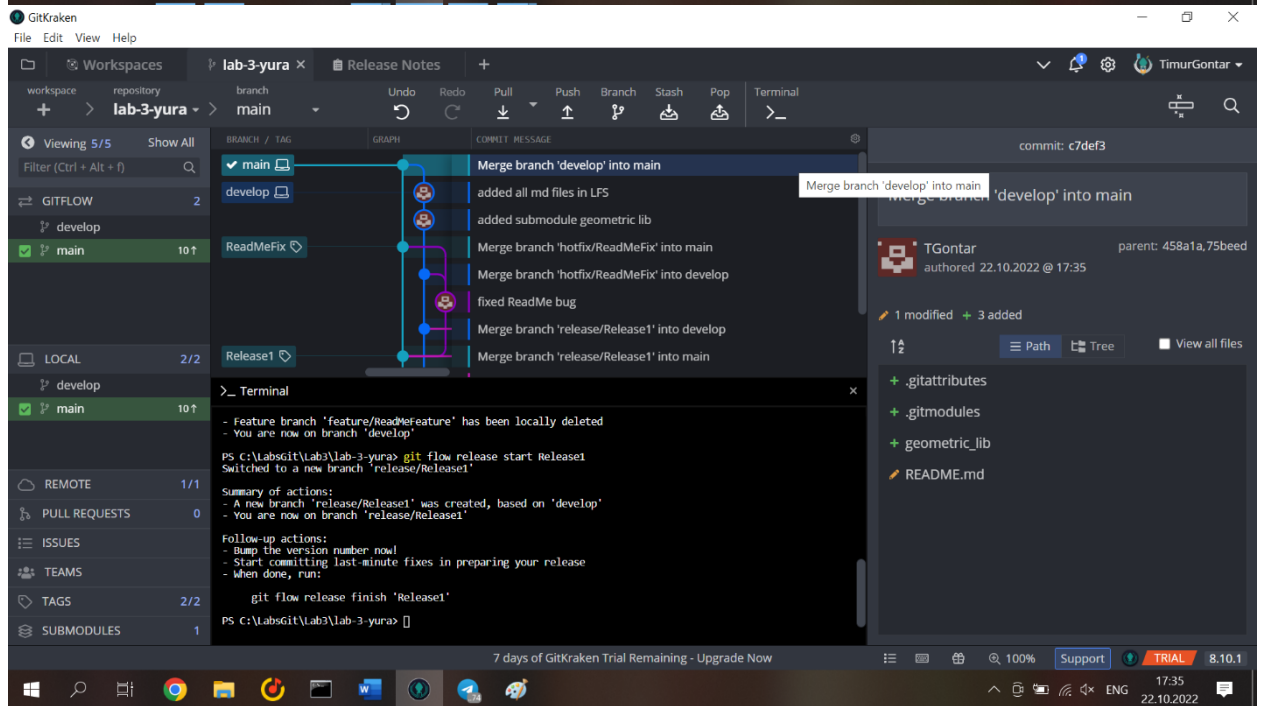
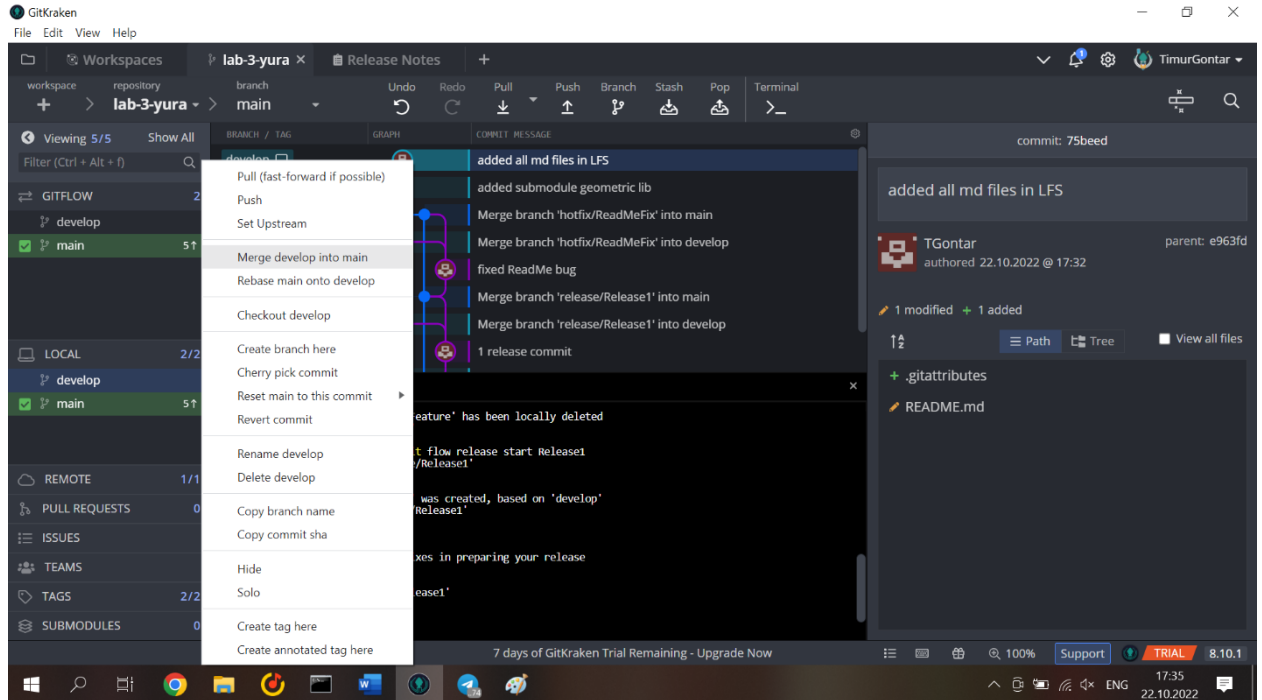


8) Инициализирую LFS в репозитории, добавляю туда все файлы md, коммичу это.





9) Вручную сливаю ветку develop в main, пушу всё это и теги тоже.



```
PS C:\LabsGit\Lab3\lab-3-yura> git push --tags
Enumerating objects: 2, done.
Counting objects: 100% (2/2), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 296 bytes | 98.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0)
To https://gitlab.com/TGontarLab/lab-3-yura.git
 * [new tag]      ReadMeFix -> ReadMeFix
 * [new tag]      Release1 -> Release1
PS C:\LabsGit\Lab3\lab-3-yura>
```

10) Кидаю пулл реквест напарнику и он его принимает

The image shows a workflow for creating and accepting a pull request. The top half displays the GitKraken application, and the bottom half displays the GitLab web interface.

GitKraken Interface:

- Workspace:** lab-3-yura
- Repository:** lab-3-yura
- Branch:** main
- Create Pull Request Dialog:**
 - From Repo:** TGontarLab/lab-3-yura
 - To Repo:** gafurov2/Lab3
 - Branch:** main
 - Title:** Pull request title
 - Description:** Pull request description

GitLab Interface:

- Repository:** Lab3
- Project ID:** 40439662
- Commits:** 14
- Branches:** 1
- Tags:** 0
- Project Storage:** 143 KB
- Recent Commits:**
 - geometric_lib @ d678c8d9: added submodule geometric lib (2 hours ago)
 - .gitattributes: returned lfs for md files (5 minutes ago)
 - .gitmodules: added submodule geometric lib (2 hours ago)
 - README.md: added all md files in LFS (2 hours ago)

The bottom half of the image shows the GitLab web interface, displaying the repository details and a list of recent commits. The repository is named "Lab3" and has 14 commits, 1 branch, and 0 tags. The project storage is 143 KB. The recent commits list shows the following changes:

Name	Last commit	Last update
geometric_lib @ d678c8d9	added submodule geometric lib	2 hours ago
.gitattributes	returned lfs for md files	5 minutes ago
.gitmodules	added submodule geometric lib	2 hours ago
README.md	added all md files in LFS	2 hours ago