МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ

Курсова Робота

По предмету: “Інженерія програмного забезпечення”

Виконав:

студент групи АМ-182

Борщов М. I.

Перевірила:

Шапоріна О.Л.

Одеса 2022

**Перелік завдань до курсової роботи**

Виконати наступні завдання:

1. Системы контроля версий. Введение в Git
2. Основи роботи в Git. Розгалуження
3. Основи командної работы в Visual Studio при помощи Git.
4. Формирование функциональных требований к ПО. Построение диаграмм вариантов использования
5. Построение диаграмм деятельности
6. Построение диаграмм классов анализа
7. Построение диаграмм классов
8. Построение диаграмм классов анализа
9. Побудова діаграм послідовності

**Хід роботи**

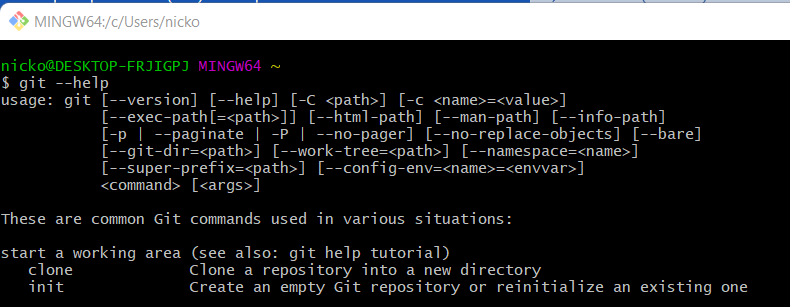
**Завдання 1**

1. Вивчити теоретичний матеріал.

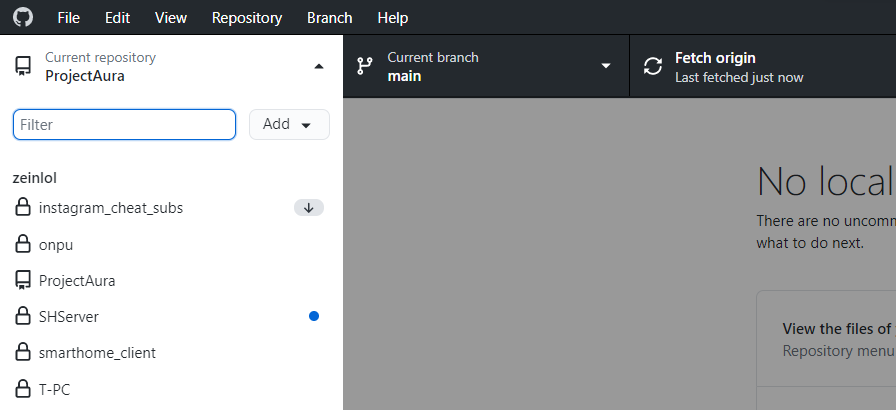
2. Встановити Git під Windows і виконати його первинне налаштування.

1. Первинне налаштування Git Bash та GitHub Desktop

Git Bash та GitHub Desktop в мене вже були встановленнi.

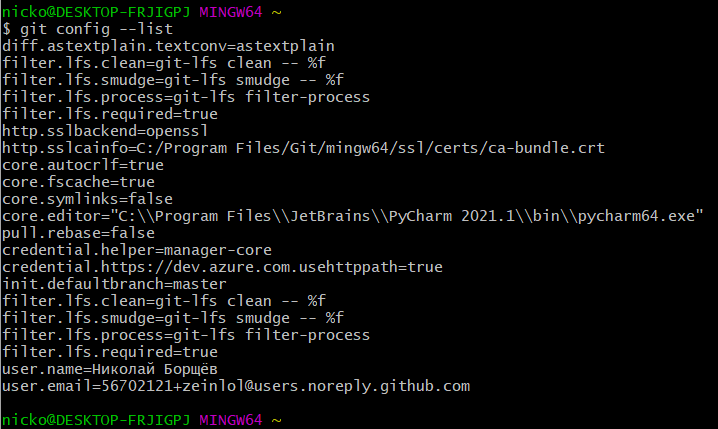


Команда допомоги в оболонцi Git



Лист репозиторіїв у GitHub Desktop

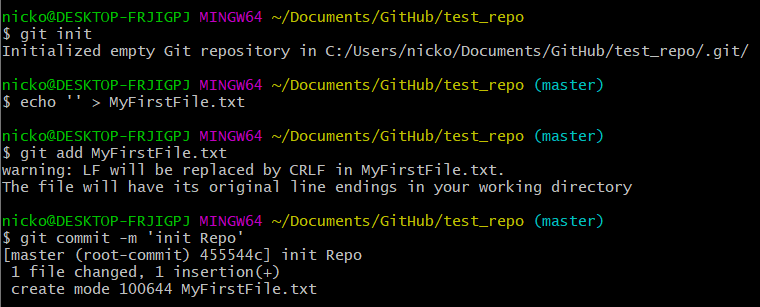
1. Первісна настройка Git (Вивід встановлених раніше параметрів)



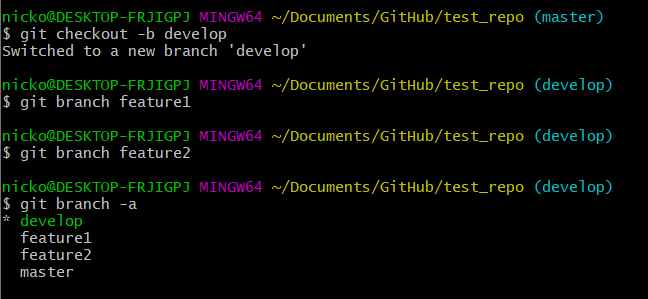
**Завдання 2**

Використовуючи GitBash і GitGUI, виконати наступні завдання:

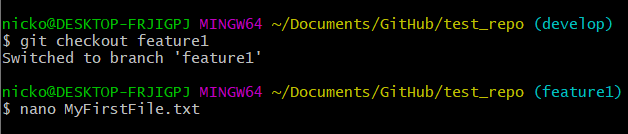
1. Створити порожній репозиторій. Додати в нього файл "MyFirstFile.txt". Зафіксувати зміни.

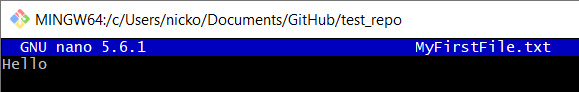


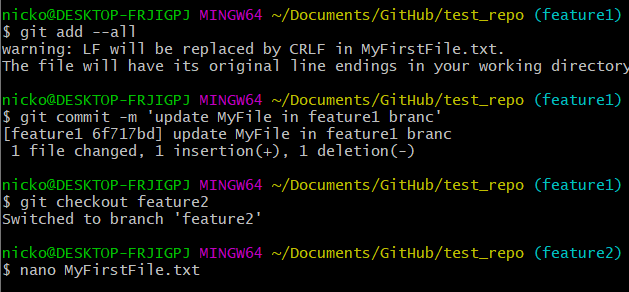
1. Створити гілку "develop" від гілки "master". Створити гілку "feature1" від гілки "develop". Створити гілку "feature2" від гілки "develop".

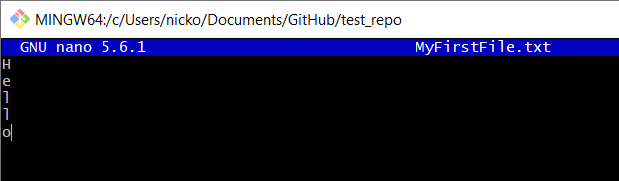


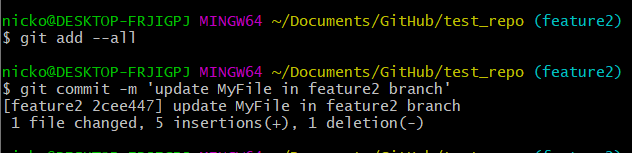
1. Переключитись на гілку "feature1". Додати рядок "Hello" в файл "MyFirstFile.txt". Зафіксувати зміни. Переключиться на гілку "feature2". Додати рядки "H", "e", "l", "l", "o" в файл "MyFirstFile.txt". Зафіксувати зміни.



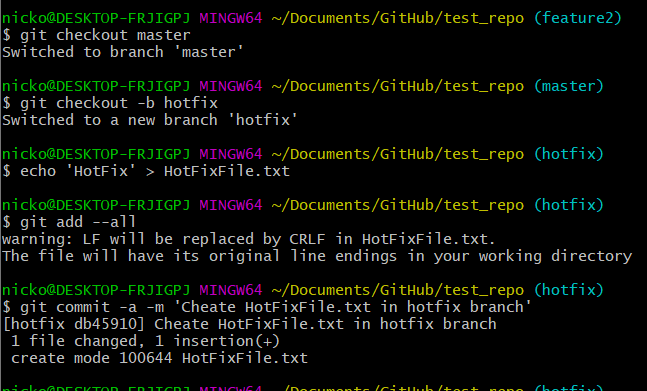




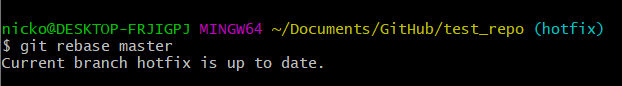




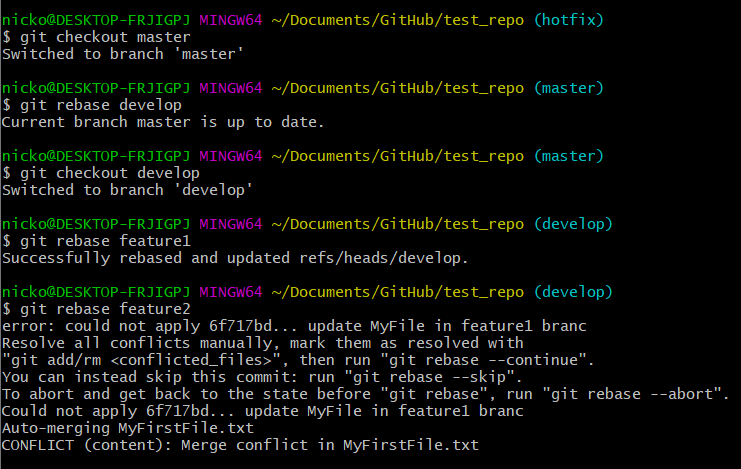
1. Створити з гілки "master" гілку "hotfix" і переключиться на неї. Створити файл "HotFixFile" з рядком "HotFix. Зафіксувати зміни. Додати в файл" HotFixFile.txt "новий рядок" HotFix2 ". Зафіксувати зміни.



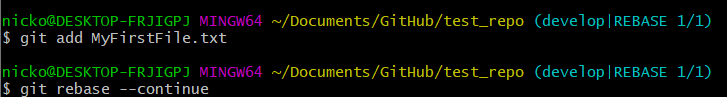
1. Зробити переміщення (rebase) гілки "master" на гілку "hotfix".



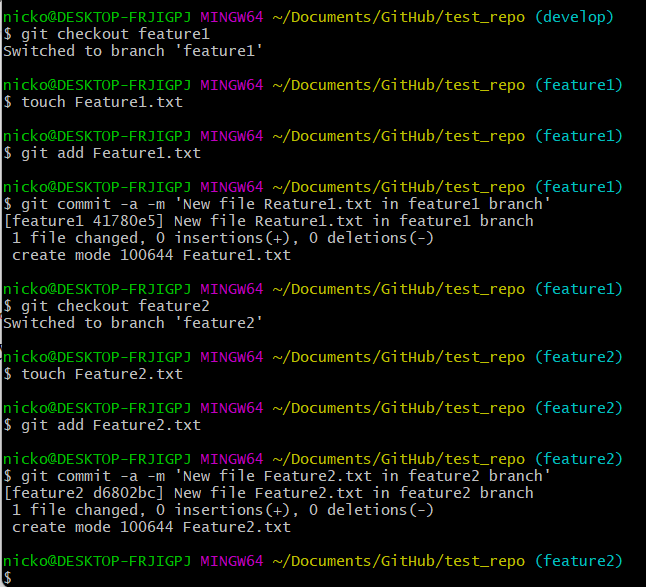
1. Зробити переміщення (rebase) гілки "develop" на гілку "master". Зробити переміщення (rebase) гілки "feature1" на гілку "develop". Зробити переміщення (rebase) гілки "feature2" на гілку "develop".



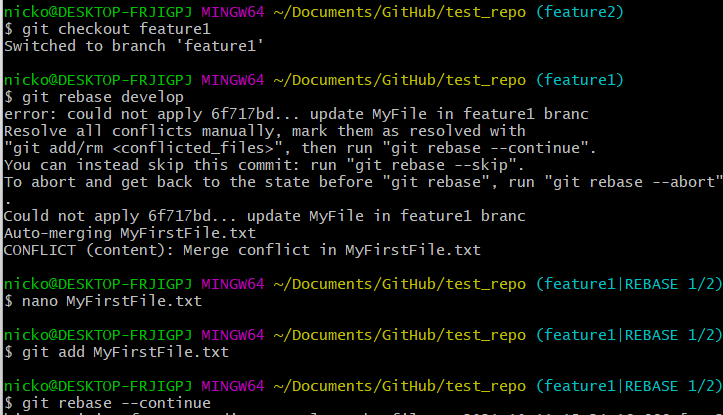
Під час переміщеня гілок виник конфлікт файлів. Це сталося бо зміст файлу MyFirstFile.txt різний у гілках feature1 та feature2.



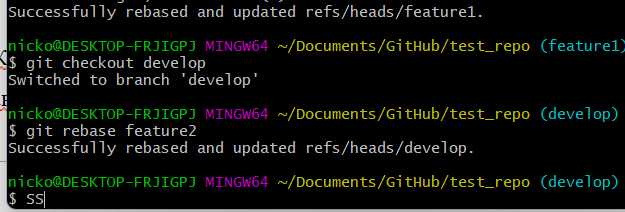
1. Переключитися на гілку "feature1". Додати файл Feature1.txt в репозиторій. Зафіксувати зміни. Переключитися на гілку "feature2". Додати файл Feature2.txt в репозиторій. Зафіксувати зміни.



1. Розробка функції "feature1" закінчена. Необхідно зробити переміщення (rebase) всіх змін з гілки "develop" на гілку "feature1".

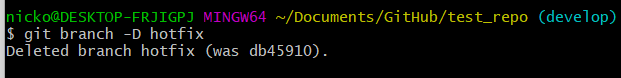


1. Зробити переміщення (rebase) гілки "feature2" на гілку "develop". Вирішити конфлікт, що виник у результаті об'єднання.

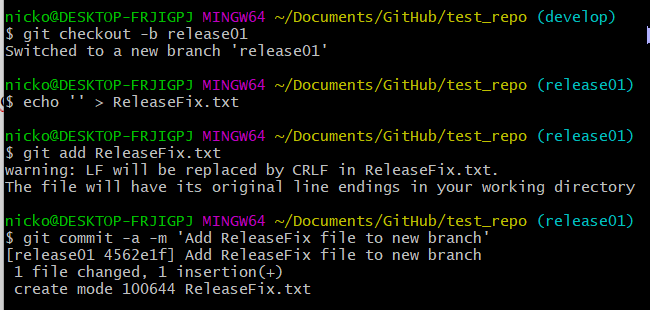


 Конфлікт переміщення був вирішений в попередніх завданнях

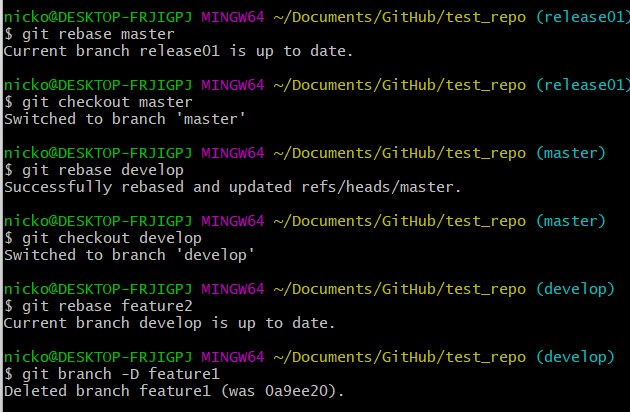
1. Видалити гілку "hotfix".



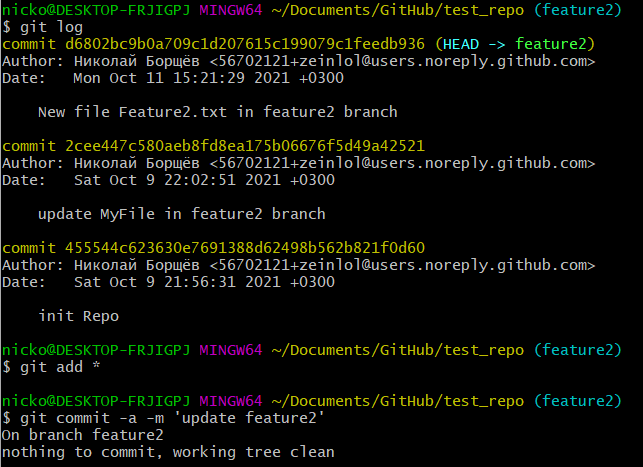
1. Створити гілку "release01" від гілки "develop". У ній буде готується реліз. Переключиться на гілку "release01". Створити файл "ReleaseFix.txt" і помістити його в репозиторій. Зафіксувати зміни.



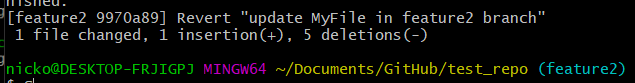
1. Припустимо, що з гілки "release01" був проведений реліз. Зробити переміщення (rebase) гілки "master" на гілку "release01". Зробити переміщення (rebase) гілки "develop" на гілку "master". Зробити переміщення (rebase) гілки "feature2" на гілку "develop". Видалити гілку "feature1".

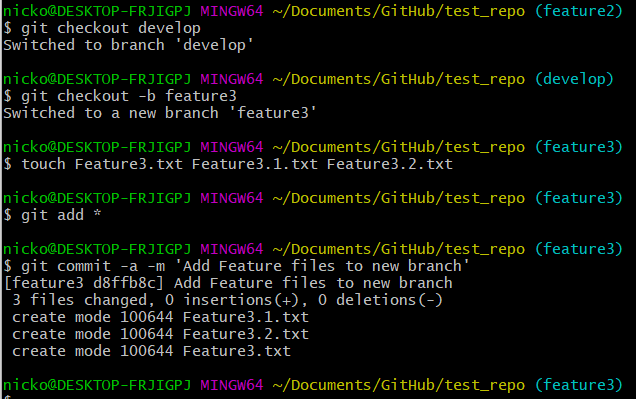


1. Припустимо, що останні зміни в гілці "feature2" не потрібні. Необхідно відкинути їх назад зробивши revert commit. Створити гілку "feature3" з гілки "develop". Додати в репозиторій 3 файлу: "Feature3.txt", "Feature3.1.txt" і "Feature3.2.txt". Зафіксувати зміни.

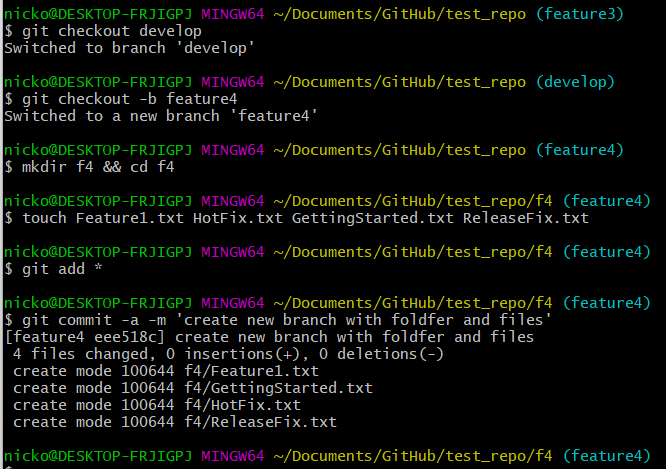




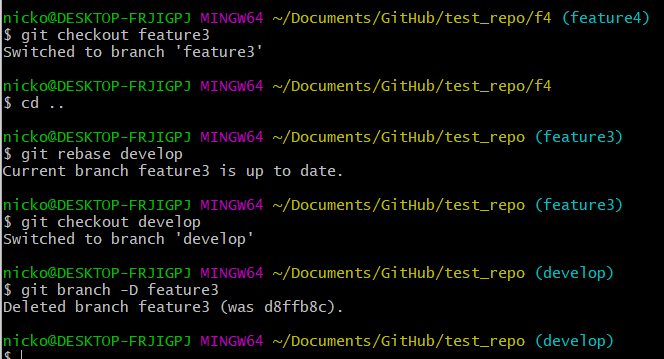




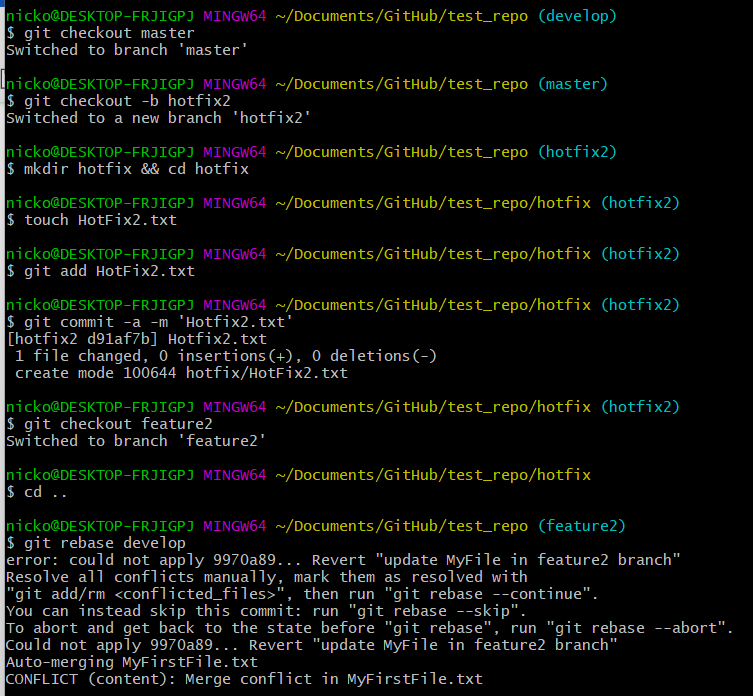
1. Створити гілку "feature4" від гілки "develop" і переключиться на неї. Додати в репозиторій папку "f4" з чотирма файлами всередині: "Feature1.txt", "HotFix.txt", "GettingStarted.txt", "ReleaseFix.txt". Зафіксувати зміни.

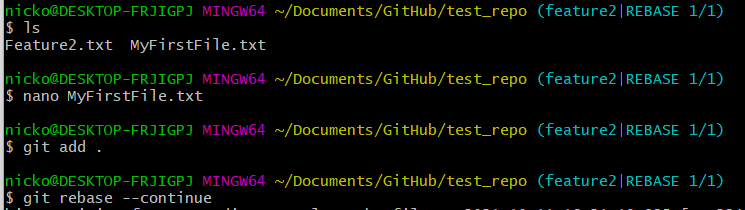


1. Зробити переміщення (rebase) гілки "develop" на гілку "feature3". Видалити гілку "feature3".



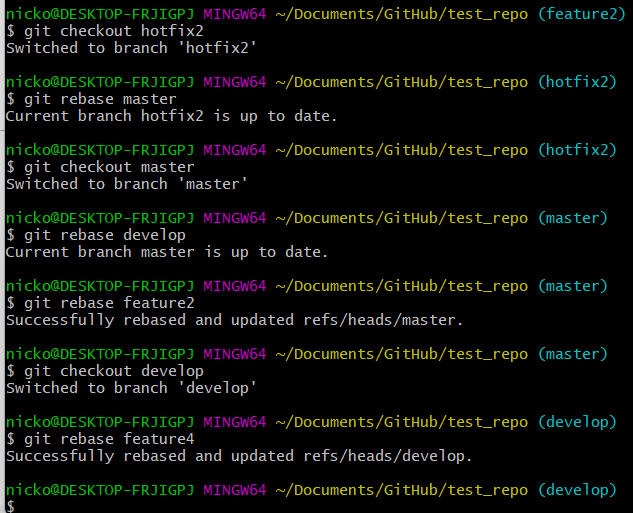
1. Створити гілку "hotfix2" від гілки "master" і переключиться на неї. Додати в репозиторій папку "hotfix" з файлом "HotFix2.txt" всередині. Зафіксувати зміни. Перемістити (rebase) гілку "develop" на гілку "feature2".



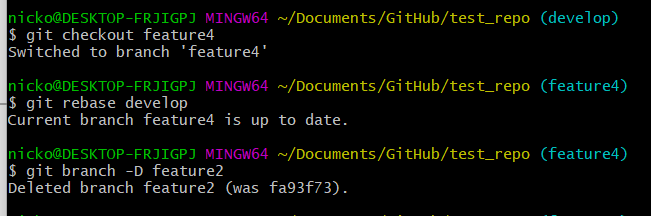


Вирішення merge  кофлікту

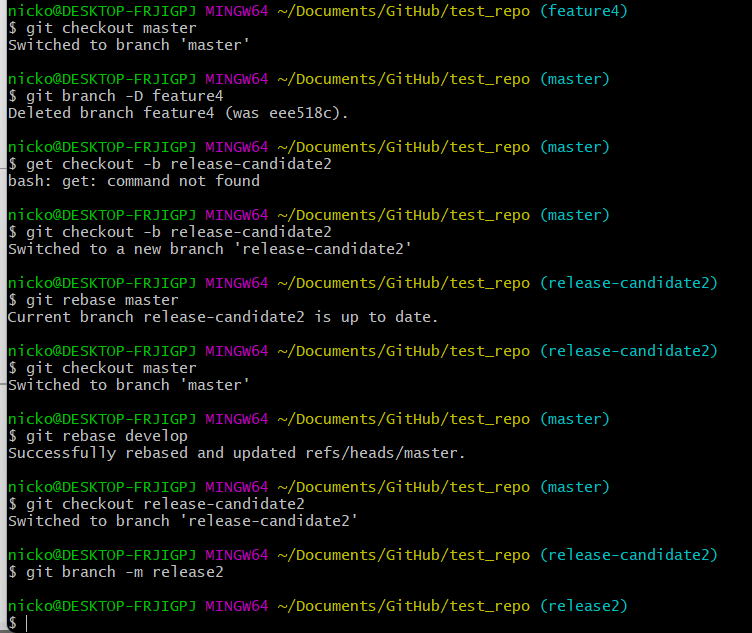
1. Зробити переміщення (rebase) гілки "master" на гілку "hotfix2". Зробити переміщення (rebase) гілки "develop" на гілку "master". Зробити переміщення (rebase) гілки "feature2" на гілку "master". Зробити переміщення (rebase) гілки "feature4" на гілку "develop".



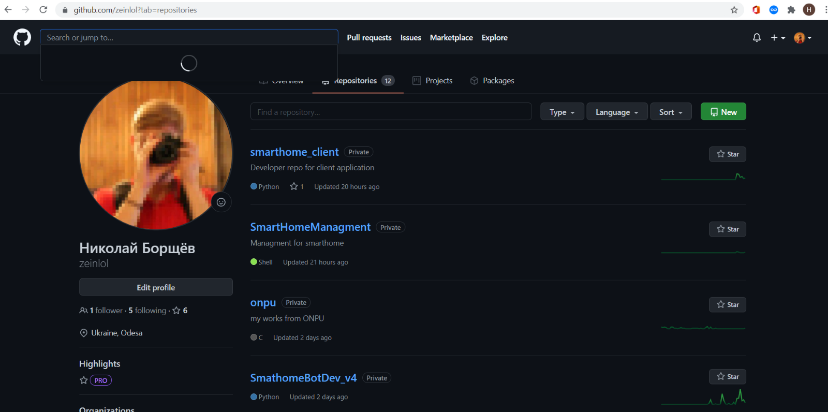
1. Зробити переміщення (rebase) гілки "develop" на гілку "feature4". Видалити гілку "feature2".

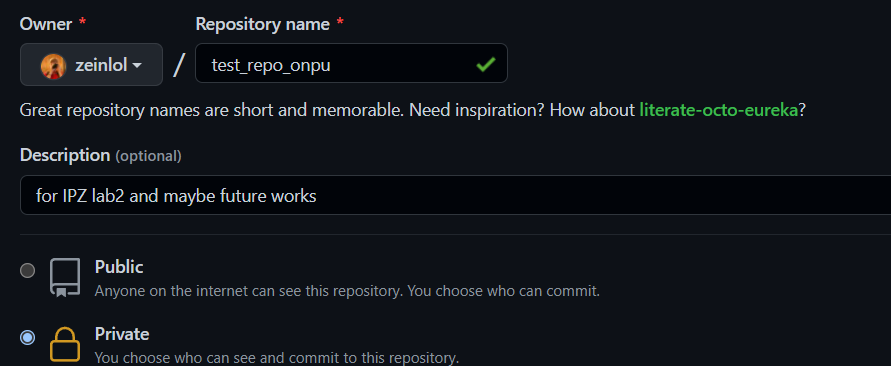


1. Видалити гілку "feature4". Зробити переміщення (rebase) гілки "master" на гілку "release-candidate2". Зробити переміщення (rebase) гілки "develop" на гілку "master". Перейменувати гілку "release-candidate2" в "release2".

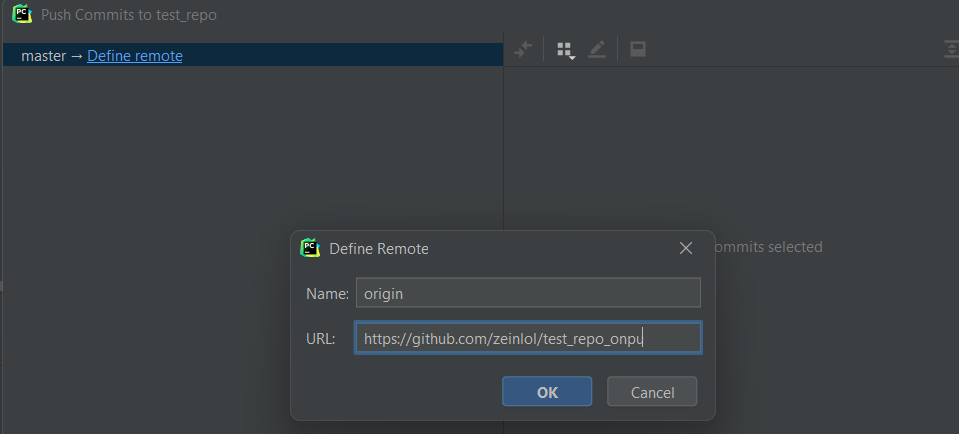


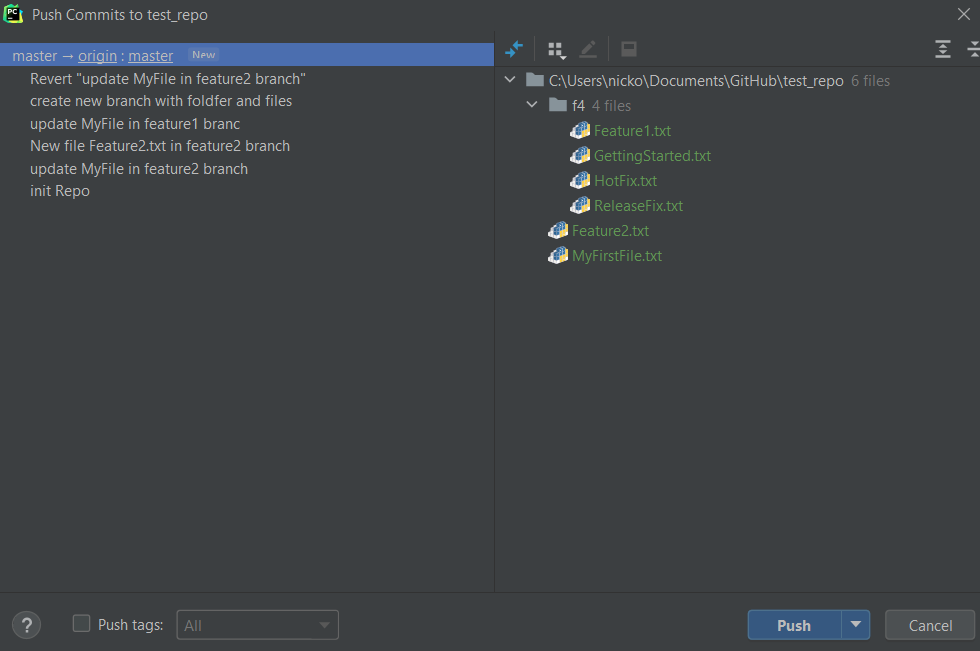
Далі на github.com зроблено репозиторій, на який було завантажено усі файли та історію змін.



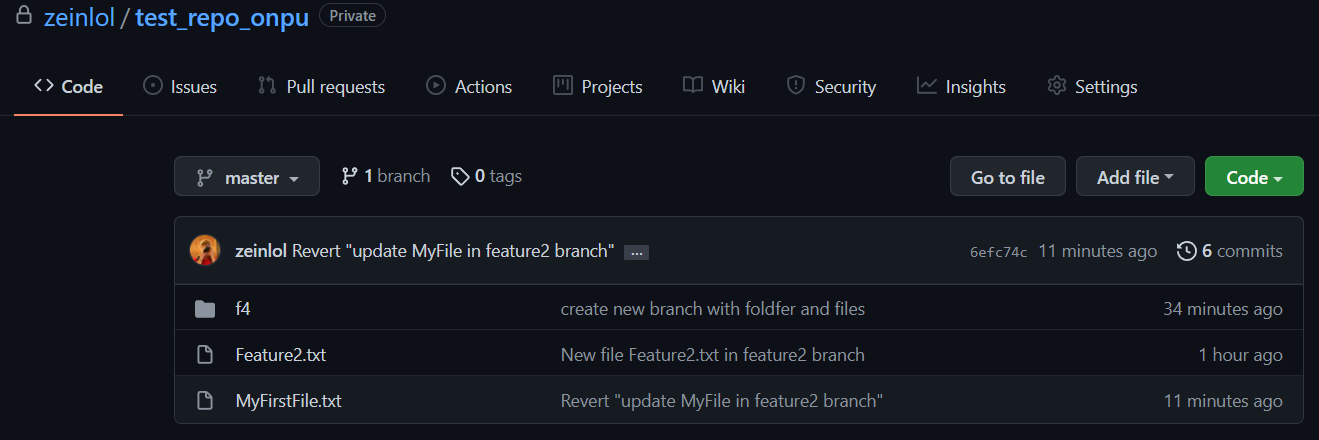


Для завантаження та історії змін було використано IDE PyCharm

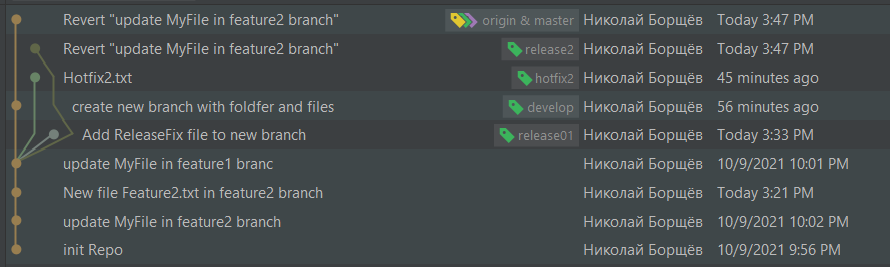




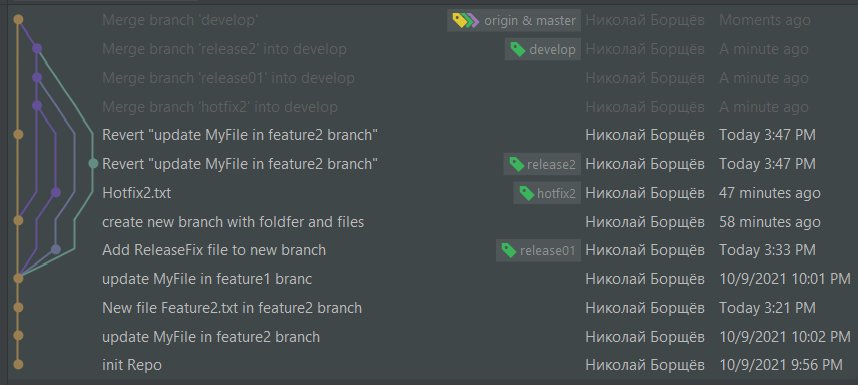
Результат



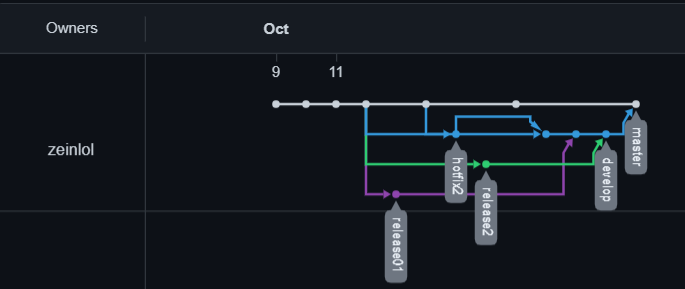
Історія змін



Розгалуження гілок не є ідеальним, бо не були закоммітчені рішення megre-кофліктів та частині гілок з конфліктами була видалена. У IDE PyCharm було виконано злиття усіх “робочих” гілок з гілкою develop, а потім вона була приєднана до гілки master.



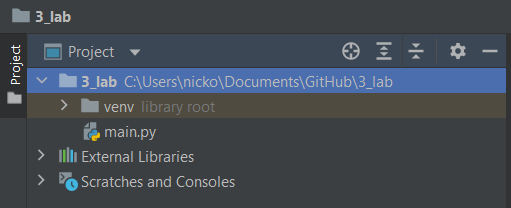
Історія гілок на сайті. Гілки можуть бути локальними або завандаженими у мережу, для того щоб їх могли використовувати інші розробники

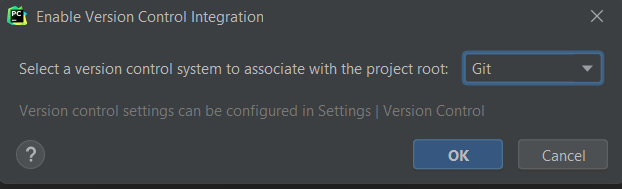


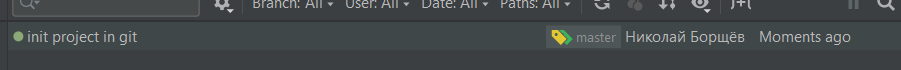
**Завдання 3**

Используя PyCharm, выполнить следующие задачи:

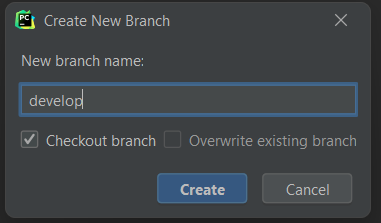
1. Створити проект та підключити Git:



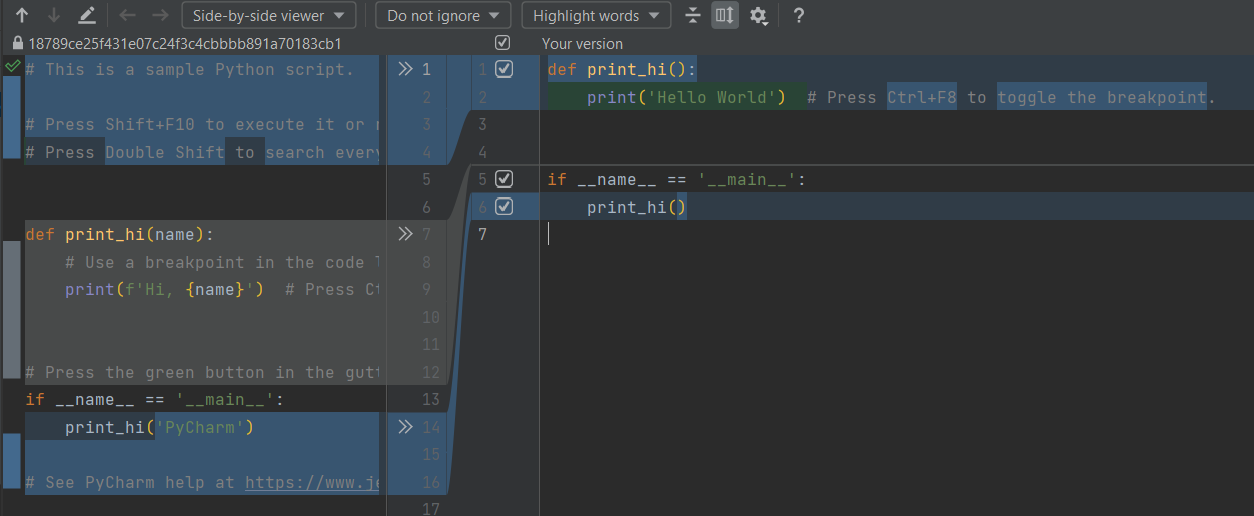


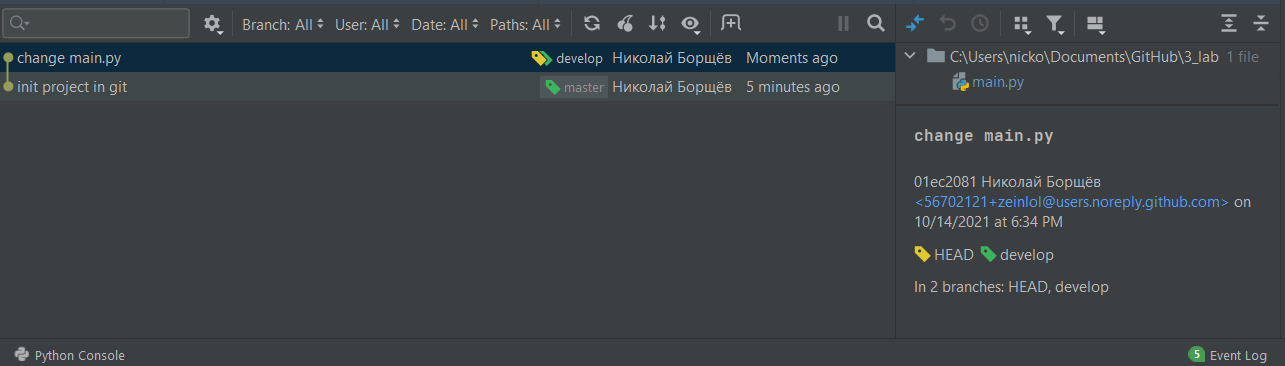


2. Створити нову гілку та додати файли та змінити їх

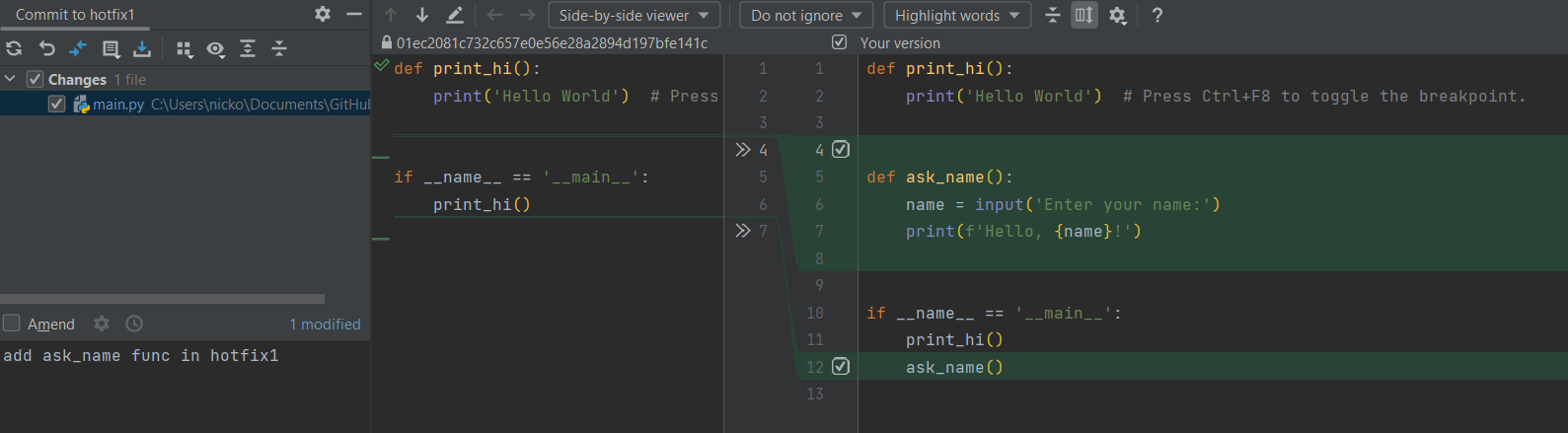


Змінення файлу main.py:

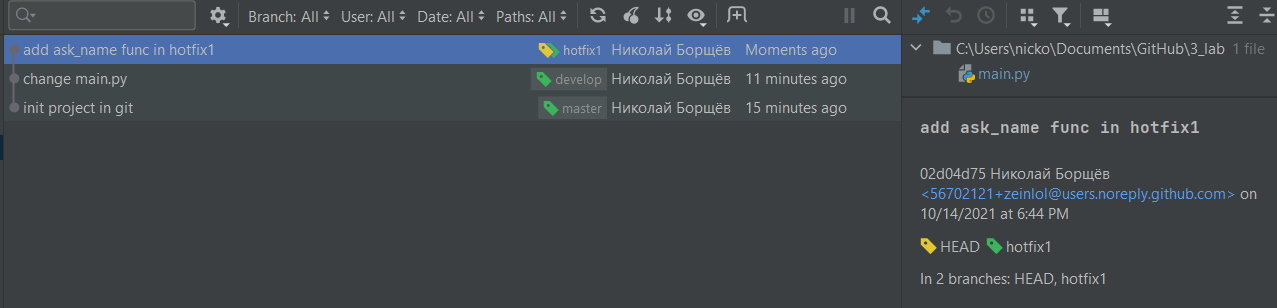




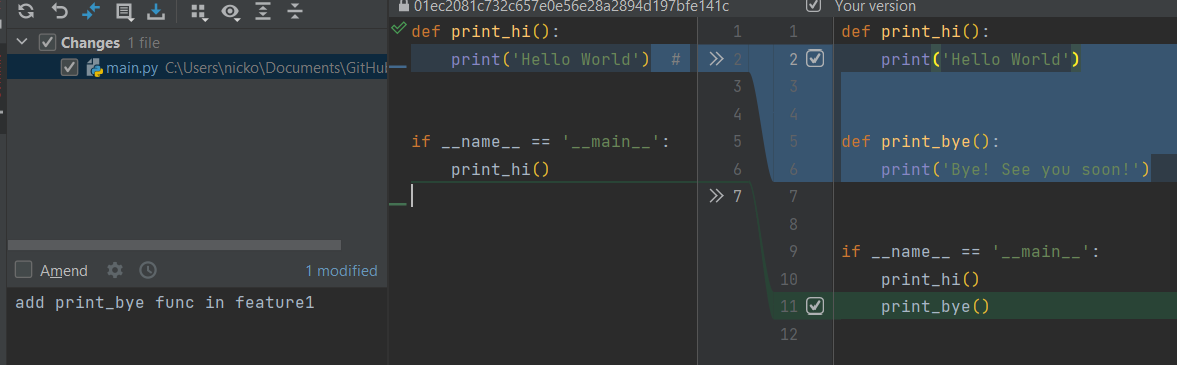
3. Створення нової гілки та внесення змін у файлі:

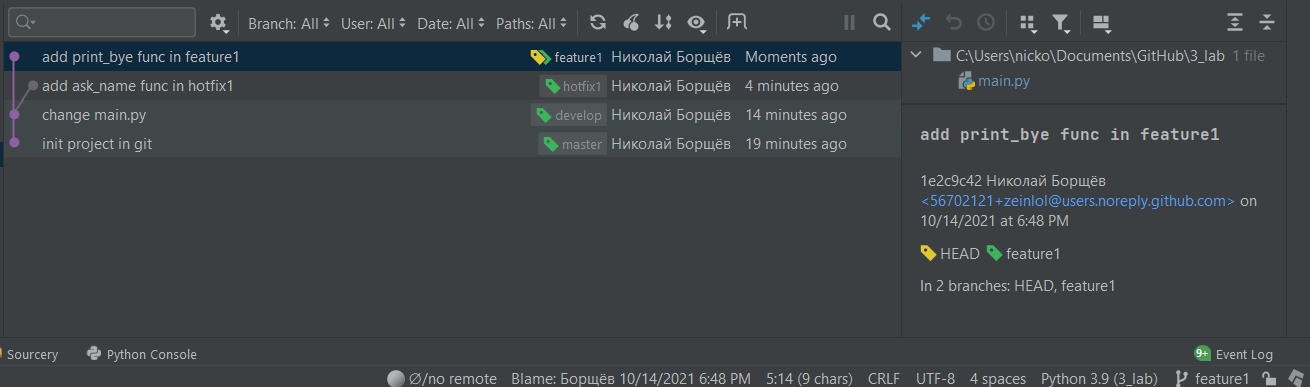


У проекті я використовую мову Python, тож у цій роботі я використовував аналогічні С# функції.

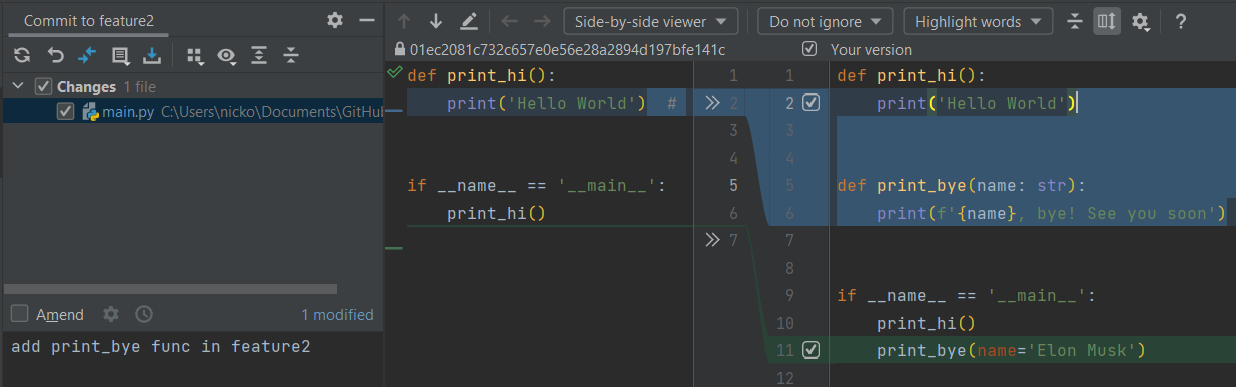


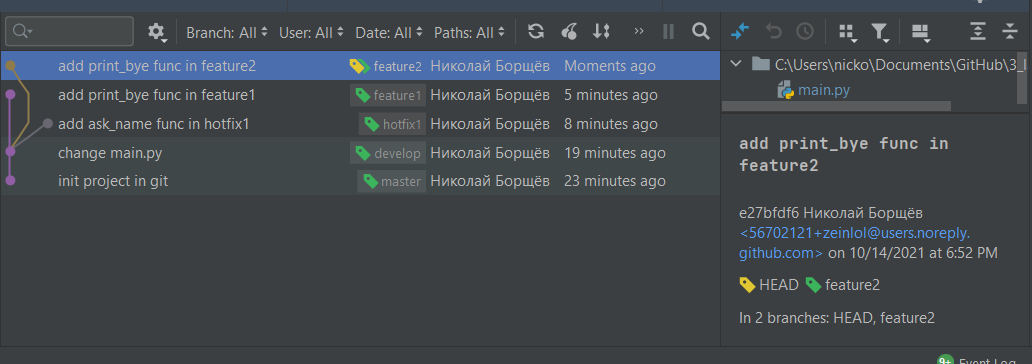
4. Створити гілку feature1 та внести зміни:



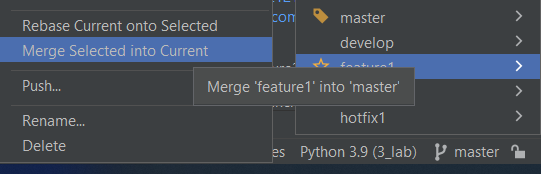


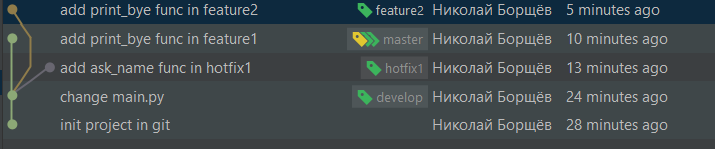
5. Створити гілку feature2 та внести подібні зміни:



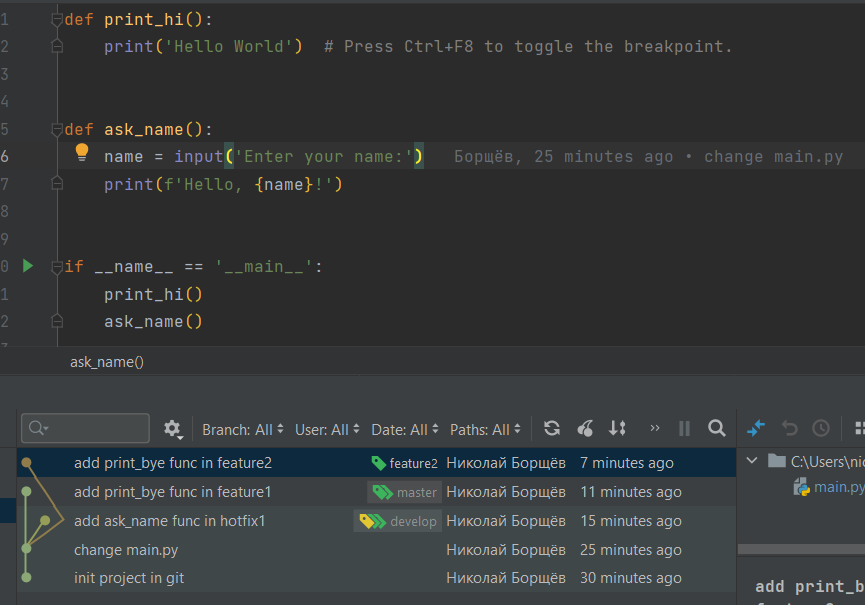


6. Перенести зміни з feature1 у master

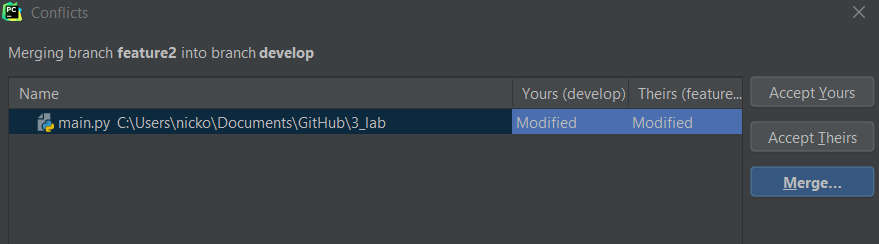




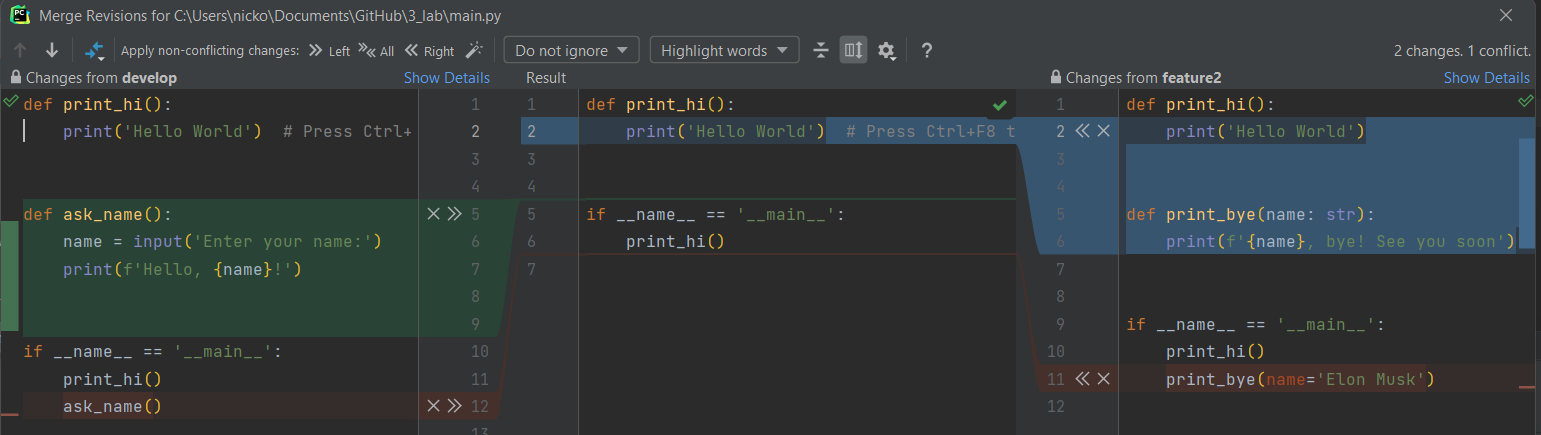
7. Перенесення змін з hotfix1 у develop



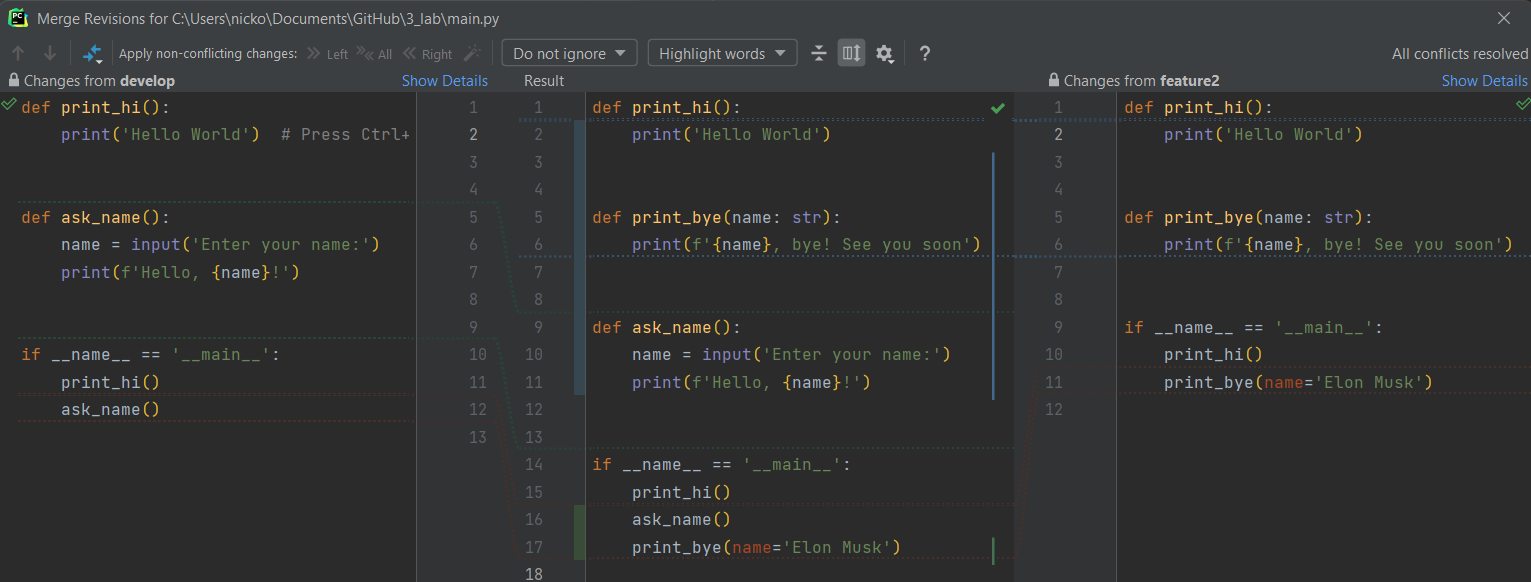
8. Перенесення змін з feature2 у develop

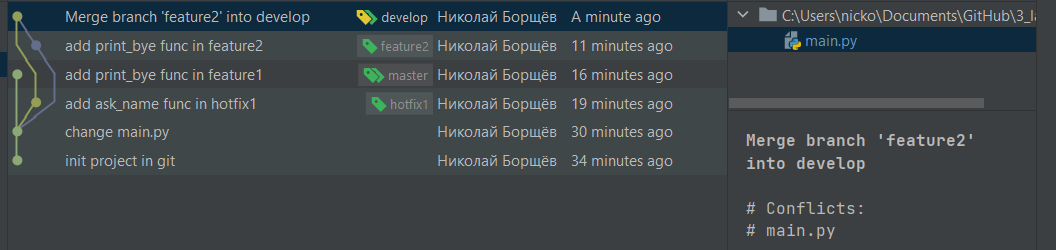


Megre конфлікт:

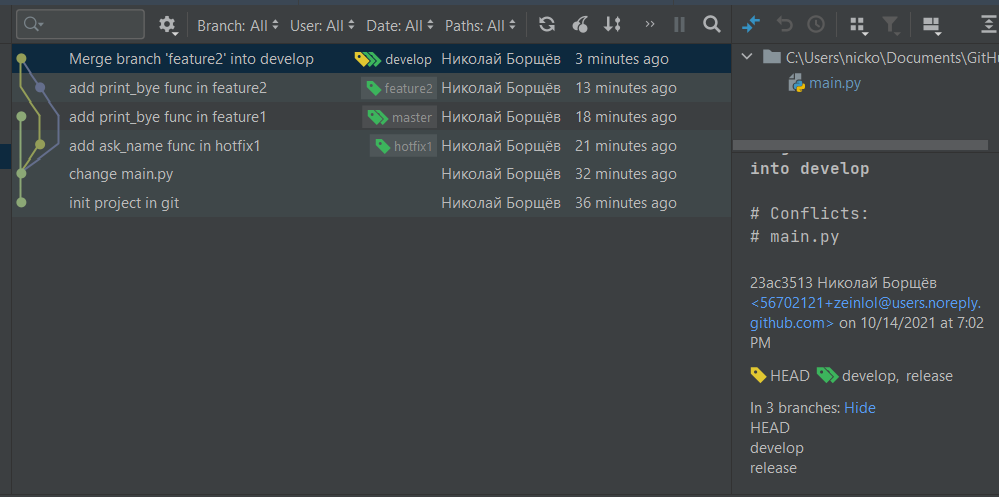


Вирішення кофлікту:

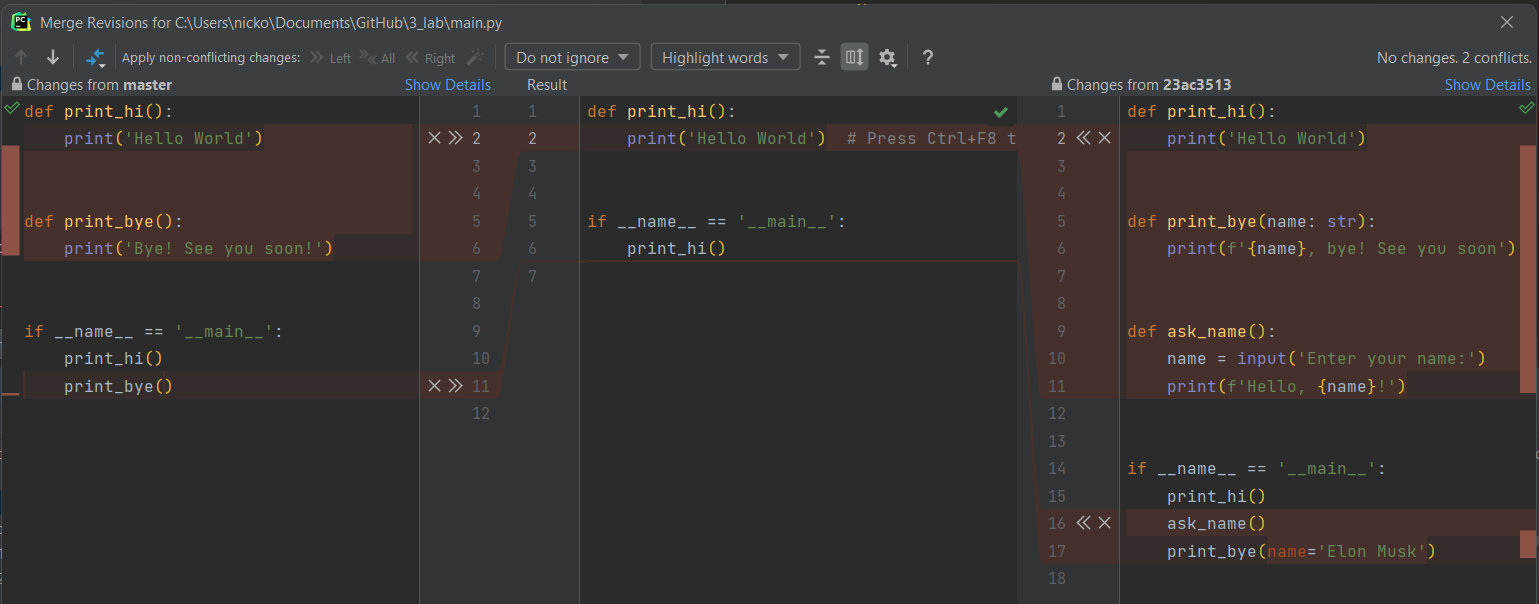


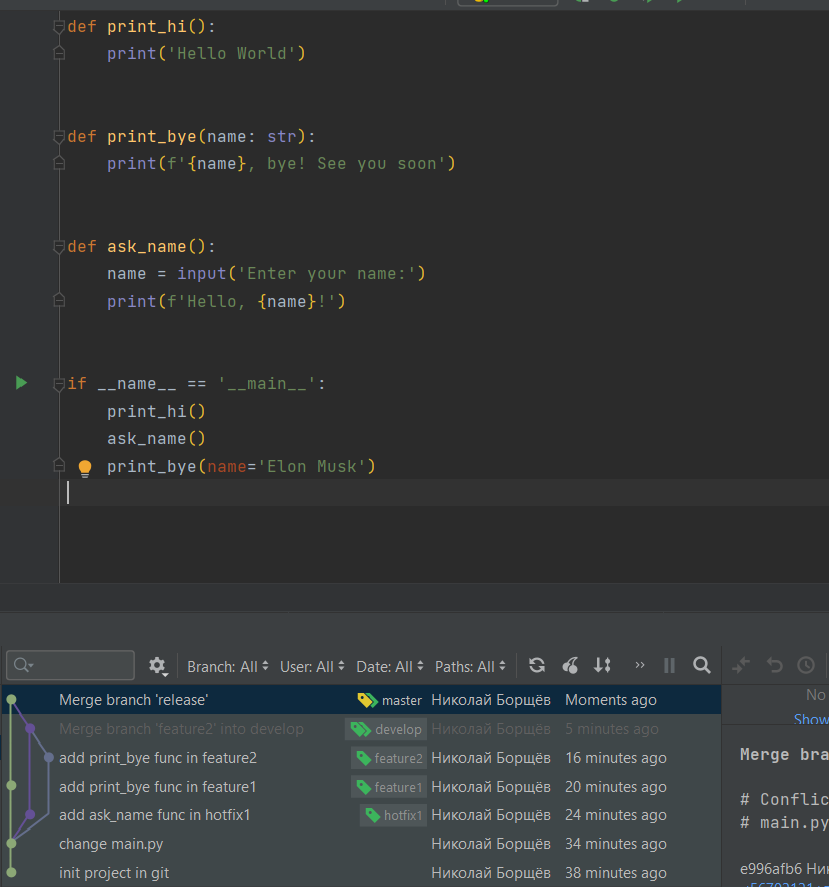


9. Створення нової гілки, зміна даних у гілках

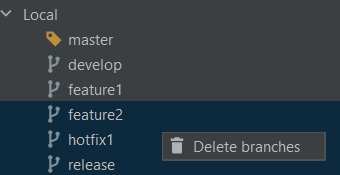


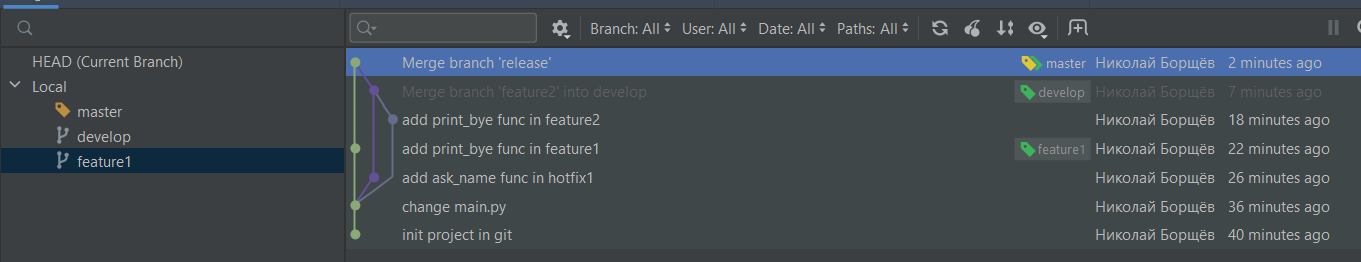
Вирішення конфлікту:





10. Видалення старих гілок





**Завдання 4**

Предметна область - Проектирование приложения для личного кабинета клиента банка.

1. Подготовить описание предметной области

Клиенту банка необходимо выполнять определённые действия над своими финансами в банке с помощью различных функций.

Основные элементы:

1. Клиент - выполняет операции, просматривает Баланс
2. Операции - определённые финансовые действия над Балансом
3. Баланс - финансы, которыми Клиент может распоряжаться
4. Кнопки - визуальные элементы, с помощью которых Клиент выполняет Операции над Балансом

Особенности и ограничения:

1. Клиент не может взаимодействовать с балансом других клиентов
2. У баланса есть лимиты - нельзя что бы он был ниже нуля, или есть кредитный лимит
3. Операции не могут быть выше или ниже определённой суммы

2. Сформулировать цель создания системы

Клиенту не придётся тратить время на поход в отделение банка, получение нужной информации от сотрудников, возможность лично выполнять операции без участия других людей.

Назначение данной системы -соответствие современным потребностям клиентов и предоставление необходимого комфортного для использования функционала.

Существующие аналоги:

С современным функционалом:

* ПриватБанк
* Монобанк

С устаревшим функционалом:

* АльфаБанк
* УкрСибБанк

Целевая аудитория - молодёжь (студенты), остальные люди, которые пользуются смартфонами / компьютерами

Ожидаемый уровень использования системы - данная система должна быть дополнительно изучена экспертами в экономике, финансовой сфере для дополнения и создания конкурентноспособной системы

3.Осуществить детализацию функций системы

Потребности клиента:

* Перевод средств другим клиентам
* Пополнение баланса телефона / других сервисов
* Просмотр истории операций

Категории пользователей:

* Неактивные - зарегистрировались, но не пользуются системой
* Малоактивные - выполнят небольшое количество операций в единицу времени
* Стандартный - выполняют среднее количество операций в единицу времени
* Активный - выполняют большое кол-во операций в единицу времени

4.Построить диаграммы вариантов использования

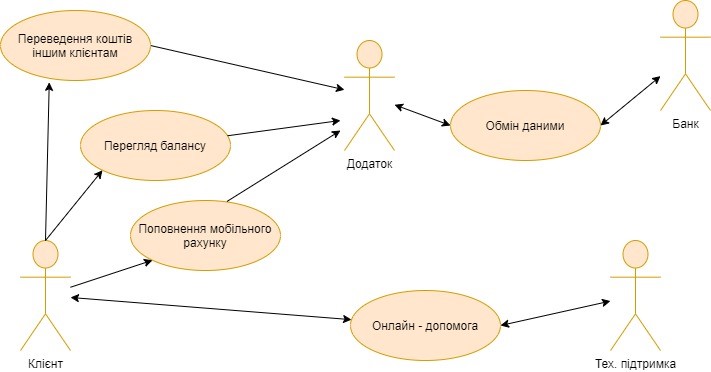


Рис. 1. Основные элементы их взаимодействие

Клиент должен войти в систему для её использования.

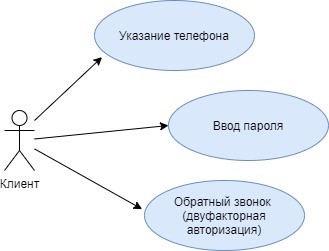


Рис. 2. Сценарий авторизации

Клиент переводит средства на счёт другого клиента

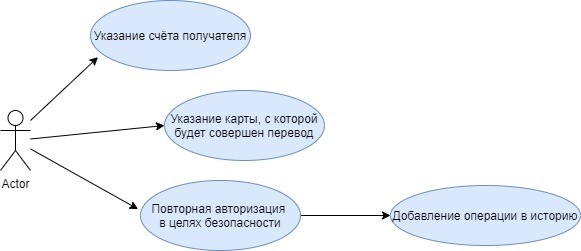


Рис. 3. Сценарий перевода средств

Клиент может просмотреть заготовленный список вопросов и получить ответ на него, или задать свой вопрос оператору через почту, онлайн-чат, по звонку.

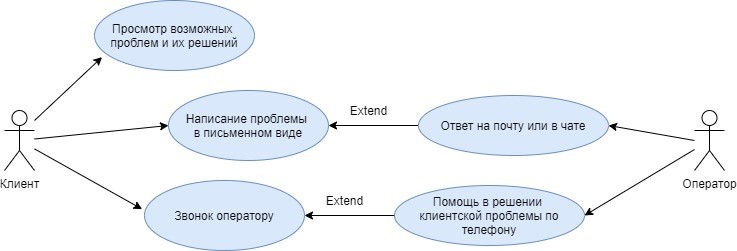


Рис. 4. Сценарий обращения в тех. Поддержку

Клиент может сделать запрос на пополнение своего счёта на определённую сумму у другого клиента.

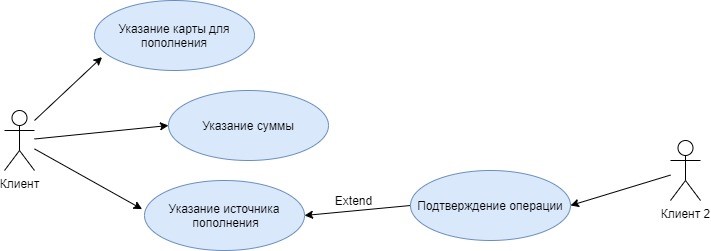


Рис. 5. Сценарий пополнения счёта

**Завдання 5**

**Завдання 6**

**Завдання 7**

**Завдання 8**

**Завдання 9**

**Висновок:**

Система Git дозволяє зручно вести контроль версії файлів та проектів на різних пристроях. Було розроблено власний тестовий репозиторій де використовувались механізми розгалужень, злиття, переміщення гілок, вирішенню merge-конфліктів та створенню файлів. Потім завантажив його та історію змін на сайт github.com.

Створив простий проект у середі розробки PyCharm, зробив багато гілок з різними змінами, потім потренирувався вирішувати megre-конфлікти при переміщені гілок.

Розробив діаграми сценаріїв використання системи роботи банківського додатка. Створив мінімальні базові елементи та діаграми їх взаємодії, описав декілька сценаріїв використання. Наступним кроком було створення діаграм використання елементів системи роботи банківського додатка. Створив діаграми базових операцій.

Також необхудно було додати діаграми класів аналізу та діаграм послідовностей для банківського додатку. Створив необхідні та описав усі елементи.