МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

Кафедра Програмної інженерії та інформаційних технологій управління

Звіт

З лабораторної роботи № 1

Виконав

студент групи КН-37г

Мирко П.О.

Перевірили

Двухглавов Д.Е.

ХАРКІВ 2020

**Тема:** Документація вимог та проектування архітектури системи за допомогою мови UML. Робота з гілками в системі GIT.

**Хід роботи:**

1. **Підготовка до виконання роботи**

1. У робочому каталозі (наприклад, D:\GIT\_PRACTICE) створити підкаталоги, в яких буде виконуватися вся подальша робота (наприклад, D:\GIT\_PRACTICE\ requirements і models).

2. Завантажити та встановити Visual Paradigm Community Edition, який буде використовуватися в даній роботі для створення UML-діаграм.

1. **Документація вимог**

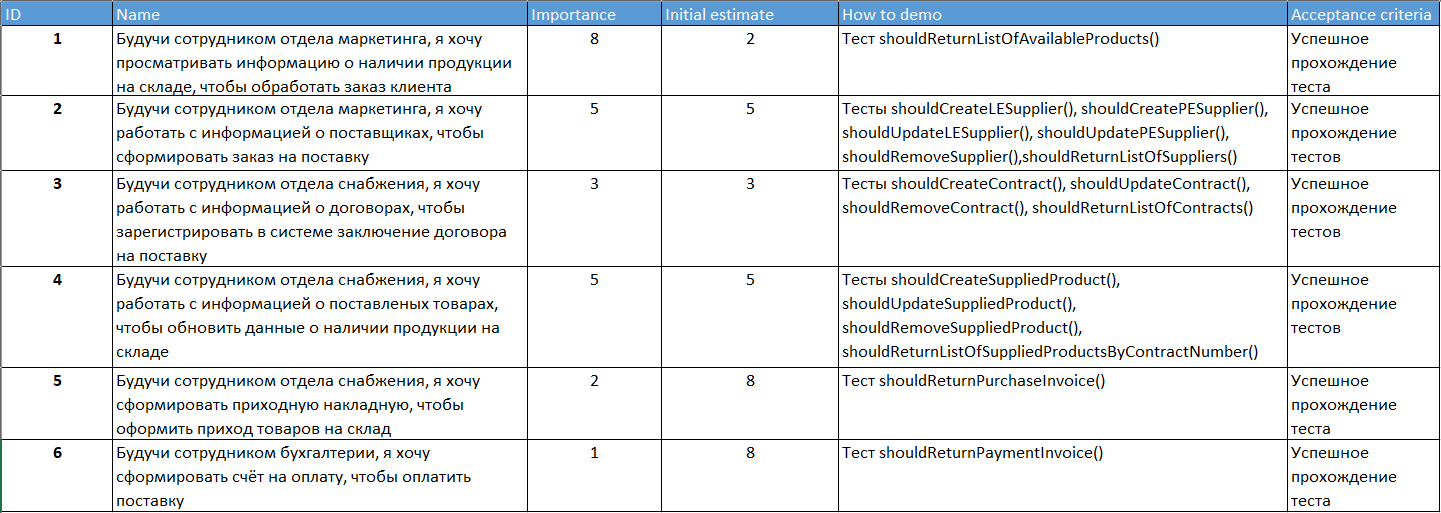
1. В робочому каталозі створити Excel файл (наприклад, dlvr\_user\_stories.xlsx), котрий буде містити користувацькі історії.

Рисунок 1 – користувацькі історії

2. Створити діаграми варіантів використання в Visual Paradigm CE та зберегти проект з ними в каталозі D:GIT\_PRACTICE\analysis.

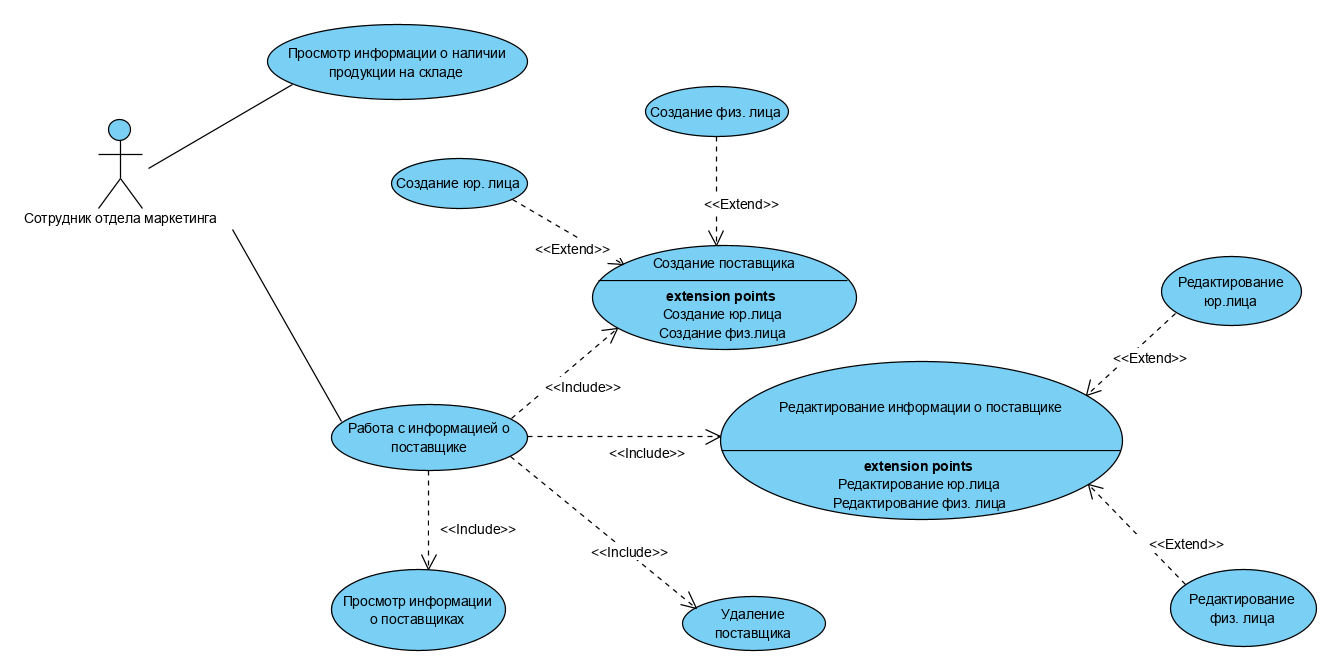


Рисунок 2 – взаємодія системи з співробітником відділу маркетингу

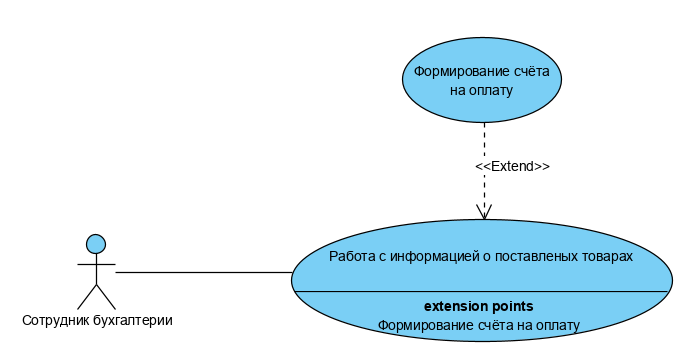


Рисунок 3 – взаємодія системи з співробітником відділу бухгалтерії

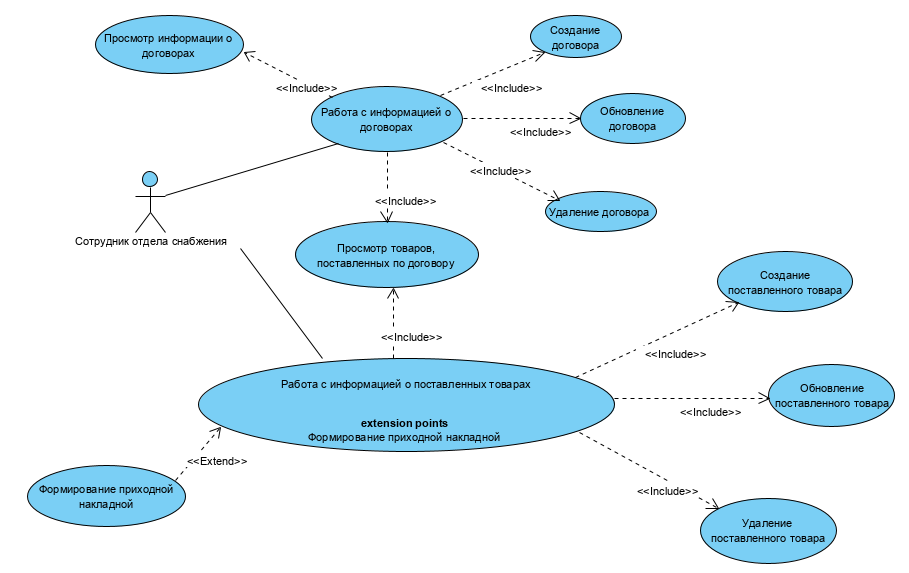


Рисунок 4 – взаємодія системи з співробітником відділу постачання

3. З С-вимог отримати D-вимоги шляхом аналізу варіантів використання та створення на основі них різноманітних діаграм.

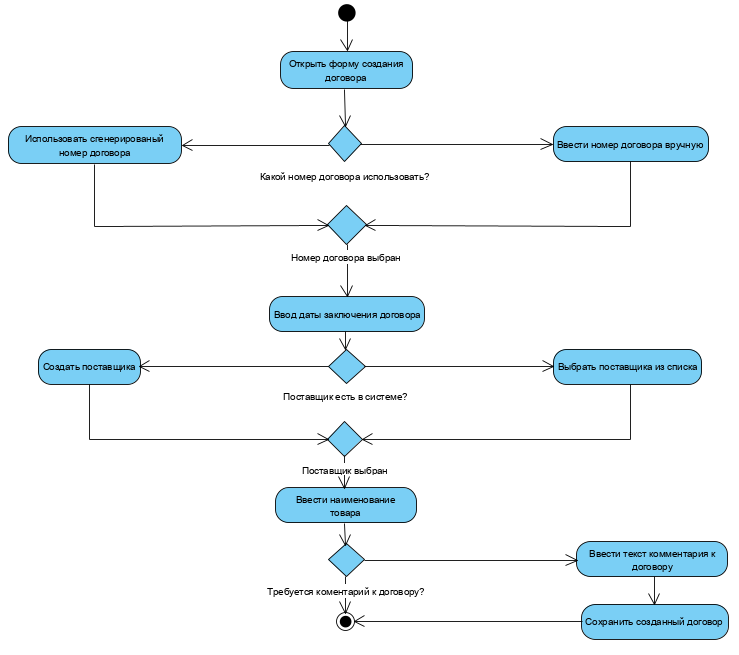


Рисунок 5 – діаграма діяльності для створення договору

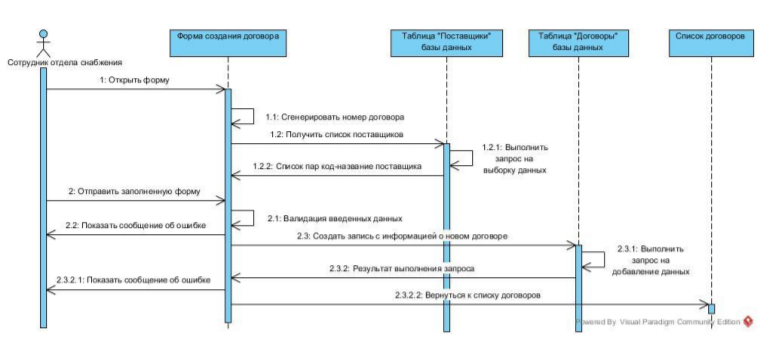


Рисунок 6 – діаграма послідовності дій для створення договору

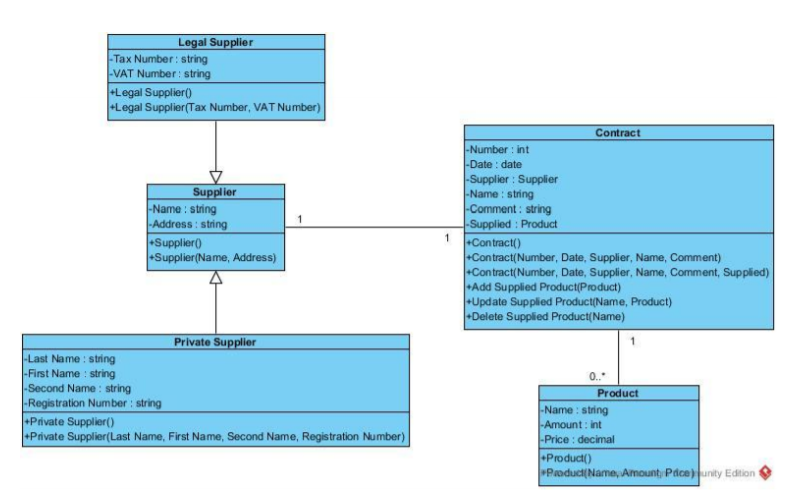


Рисунок 7 –діаграма класів

4. Проектування архітектури системи

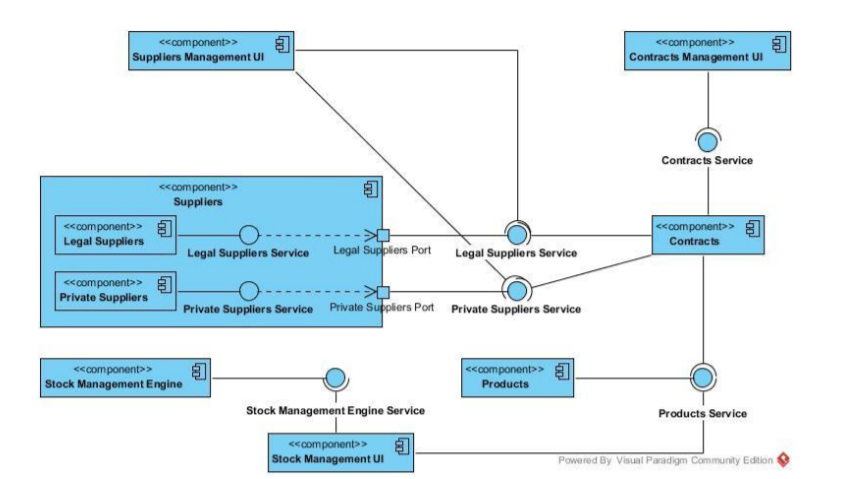


Рисунок 8 –діаграма компонентів

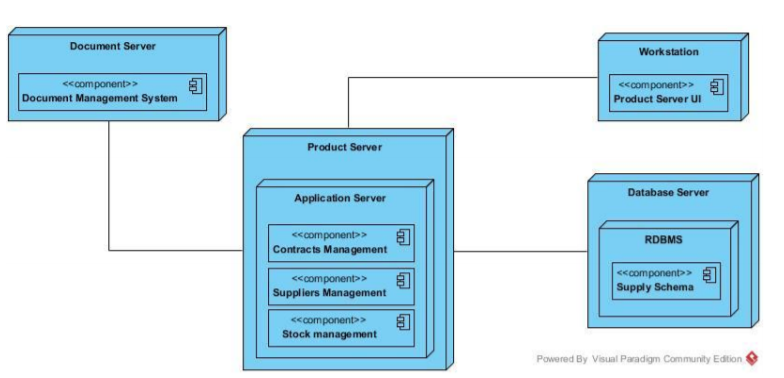


Рисунок 9 – діаграма розгортання

5. Розгалуження і злиття

Для створення гілки та переходу на неї використовується команда git checkout з ключем -b:



Рисунок 10 – створення нової гілки

Після зміни файлу index.html можна зафіксувати зміни в цій гілці:

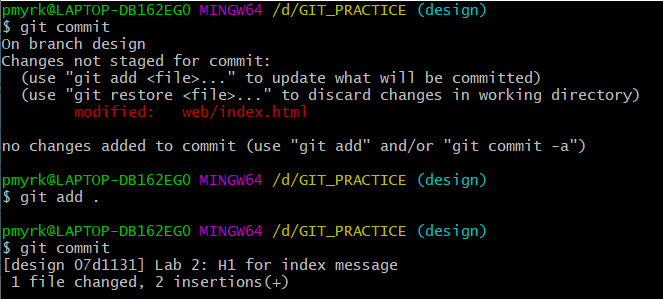


Рисунок 11 – фіксації змін в новій гілці

Перехід на між гілками здійснюється за допомогою команди git checkout <назва гілки>. Припустимо потрібно повернутися на гілку master та створити ще одну гілку hotfix. Це можна виконати за допомогою наступних команд:

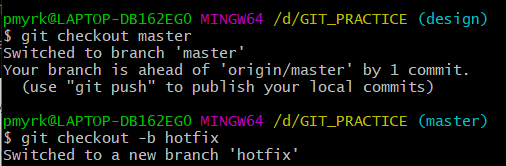


Рисунок 12 – перехід між гілками та створення нової

На цій гілці зафіксуємо зміни файлу index.html, після чтого виконаємо злиття гілки master та hotfix:

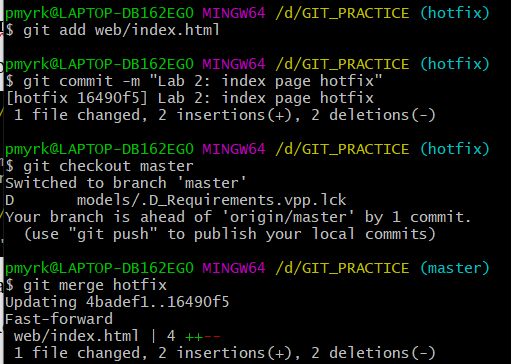


Рисунок 13 – злиття гілок

Тепер можна спробувати злиття гілок design та master. Це спочатку не вийде, оскільки зміни в файлі index.html не зафіксовані, про це повідомить сам GIT. Тож необхідно зафіксувати зміни та виконати злиття цих гілок:

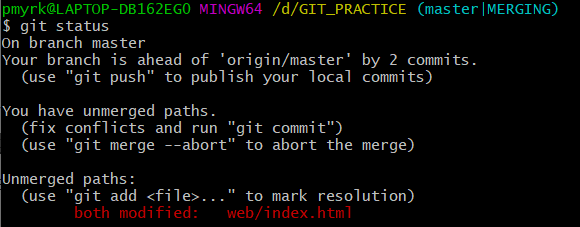


Рисунок 14 – перевірка статусу та повідомлення про незафіксовані зміни

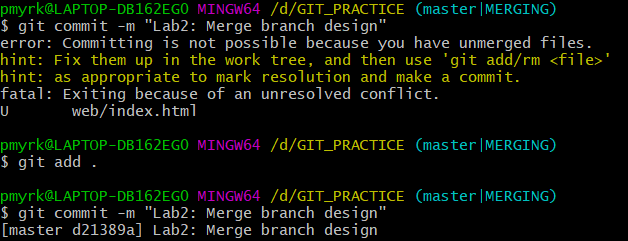


Рисунок 15 – фіксація змін та злиття гілок

**Висновок:** виконання лабораторної роботи розширило теоретичні знання та практичні навички роботи з GIT. Використання механізму розгалуження та злиття має величезну практичну цінність для контролю версіями.