

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
„КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. СІКОРСЬКОГО”
НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС
„ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ”

Лабораторна робота №3

з курсу «ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ»

Тема: «Розробка життєвого циклу проекту та системи багтрекінгу. Розробка
поетапного плану проекту.»

Виконала
Студентка 4 курсу
групи ДА-71
Опрісник Роксолана

Київ – 2020

Частина 1. Розробка життєвого циклу проекту та системи багтрекінгу.

Мета роботи: Вивчити типові життєві цикли розробки програмного забезпечення. Усвідомити методику роботи з багтрекінгом, навчитися створювати питання, задавати їх статус, а також навчитися оперувати статистикою багтрекінга.

Задача: Вивчити зміст прикладу інформаційної системи та створити опис життєвого циклу розробки відповідного програмного забезпечення.

Використовуючи систему багтрекінга MantisBT або подібну необхідно ознайомитися веденням журналу для повного циклу розробки та тестування, відтворити статуси для питання (ticket) та ролі у процесі розробки програмного забезпечення.

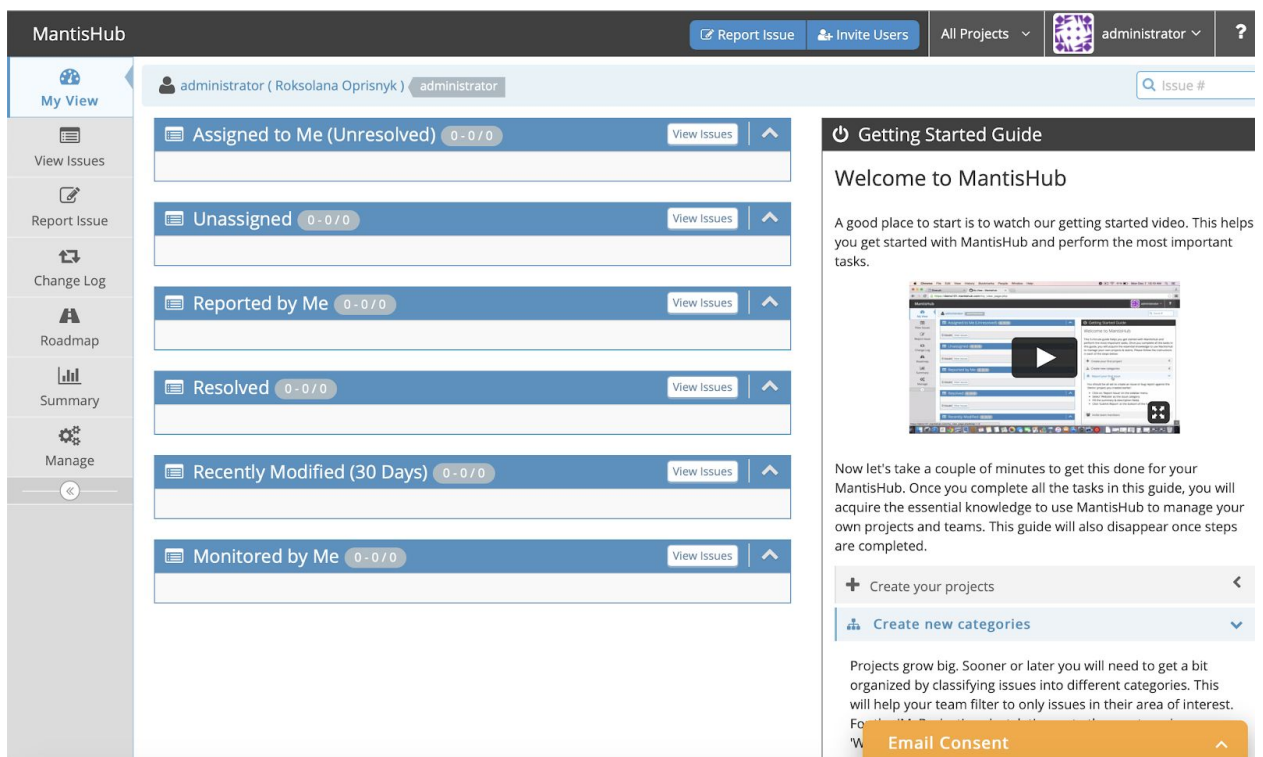


Рис.1 Знаходимось на головній сторінці.

Enter Issue Details

* Category: [All Projects] General

Severity: minor

Priority: normal

Assign To: [dropdown]

* Summary: [text area]

* Description: [text area]

Attach Tags: (Separate by ",") [text input] Existing tags [dropdown]

Upload Files: Maximum size: 250,000 KB. Attach files by dragging & dropping, selecting or pasting them.

View Status: ☒ public ☐ private

Report Stay: ☐ check to report more issues

Submit Issue

Рис.2 Створюємо задачу

View Issue Details

Send a Reminder | Jump to Notes | Jump to History

ID	Project	Category	View Status	Date Submitted	Last Update
0000003	MyProject	General	public	2020-11-26 22:56	2020-11-26 22:56

Reporter	administrator	Assigned To	administrator
Priority	normal	Severity	minor
Status	assigned	Resolution	open

Summary: 0000003: Зробити інтерфейс

Description: ...

Tags: No tags attached.

Attach Tags: (Separate by ",") [text input] Existing tags [dropdown] Attach

Edit | Assign To: [dropdown] | Change Status To: new | Monitor | Stick | Clone | Close | Move | Delete

Рис.3 Щойно створена задача

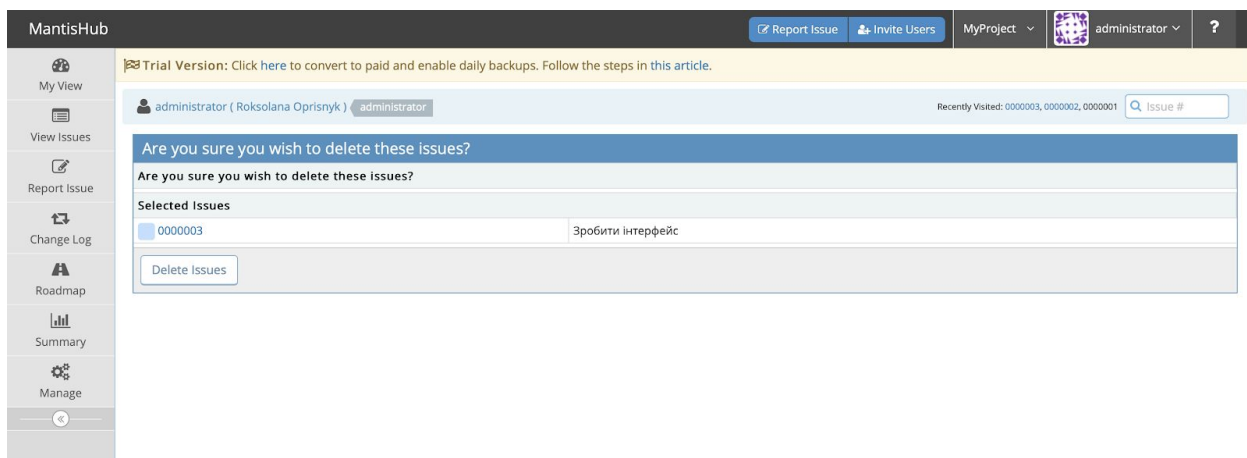


Рис. 4 Видаляємо щойно створену задачу.

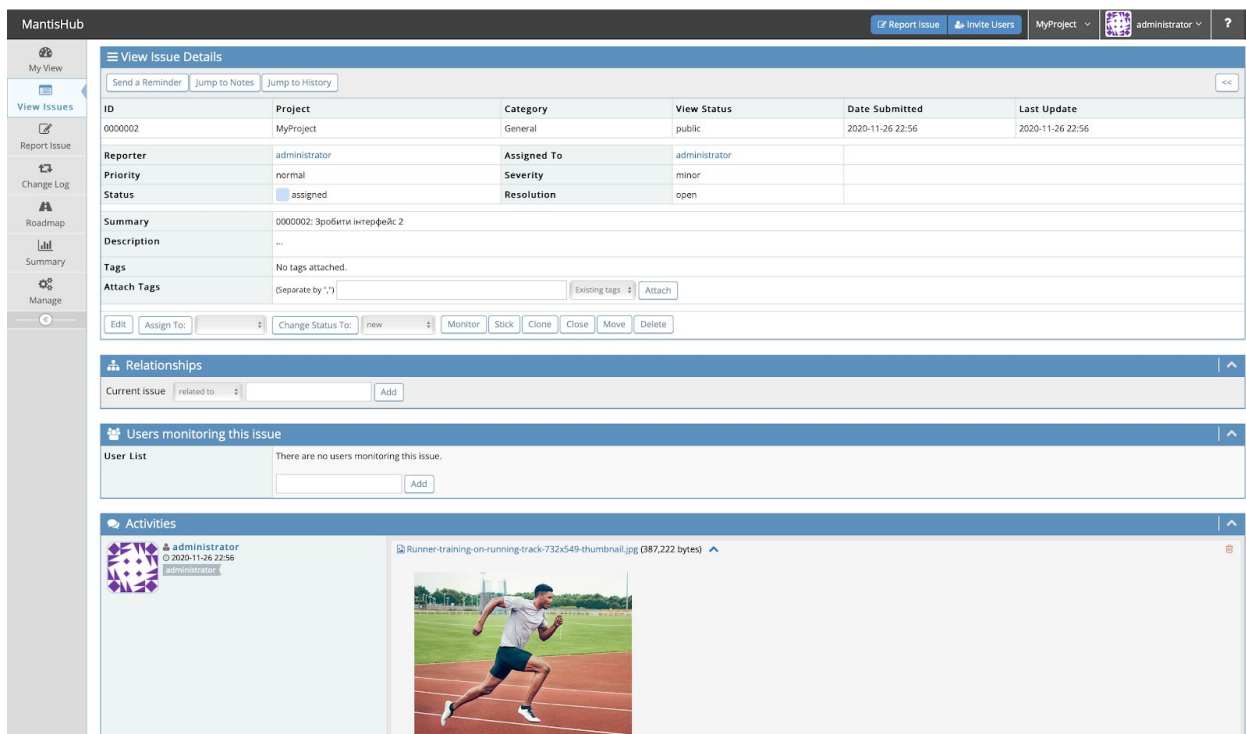


Рис.5 Створена заново задача з прикріпленим файлом та коментарем

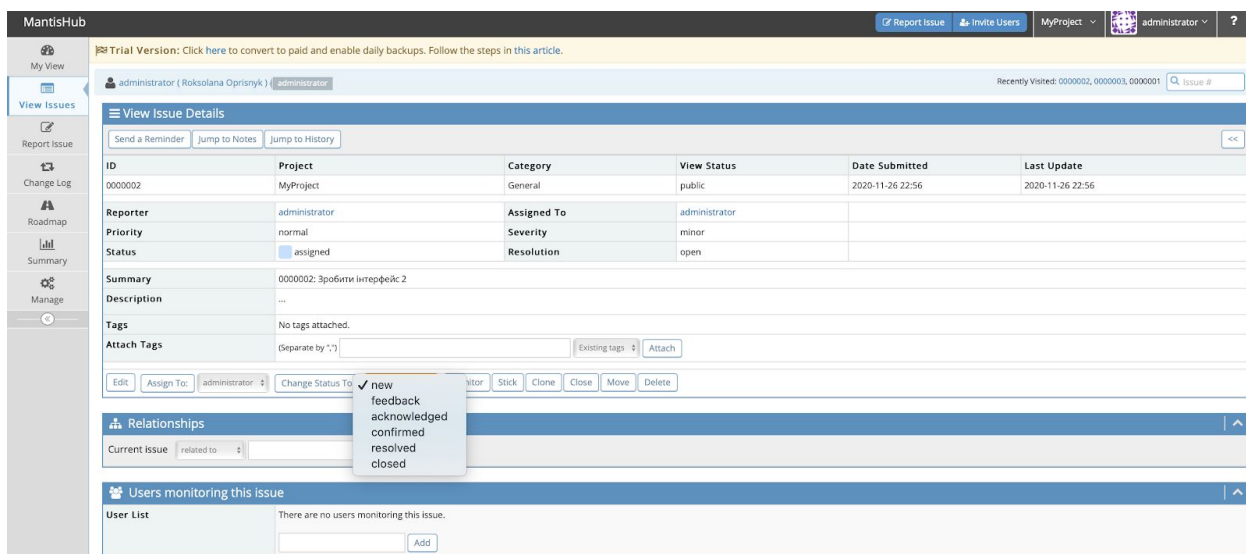


Рис. 6 Змінюємо статус задачі

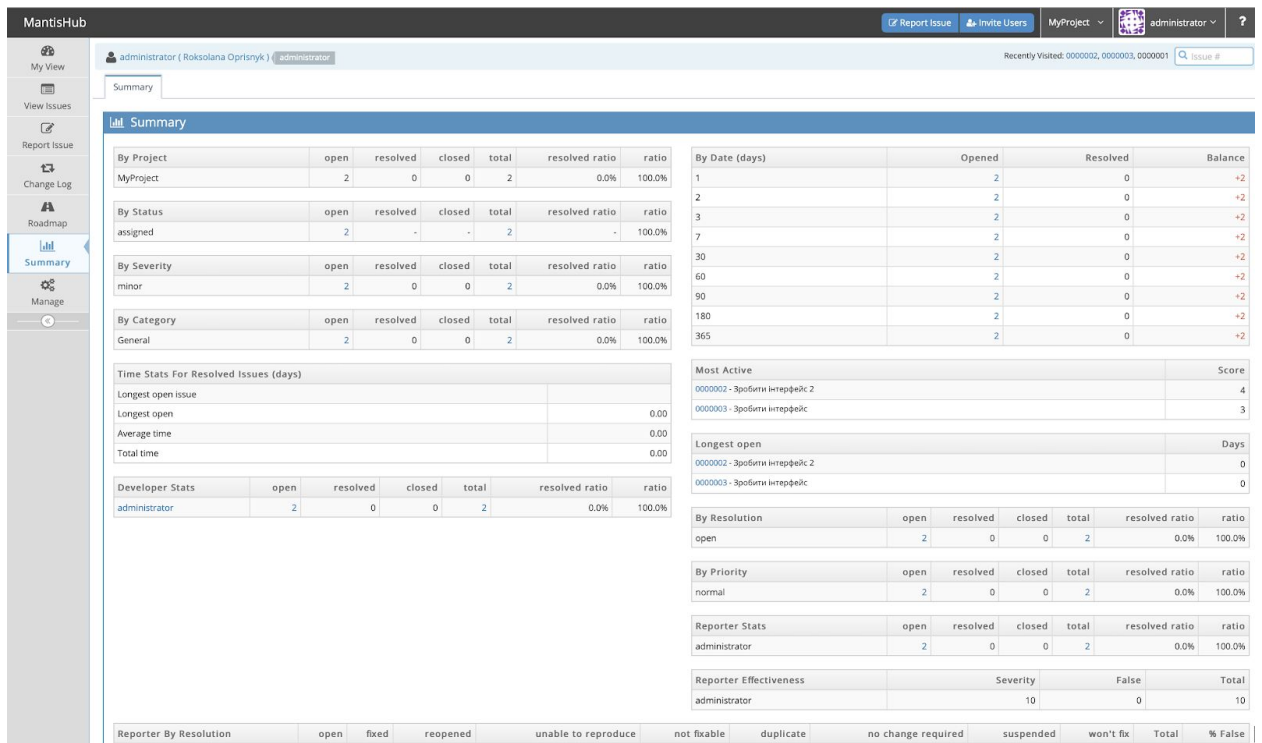


Рис. 7 Вікно статистики

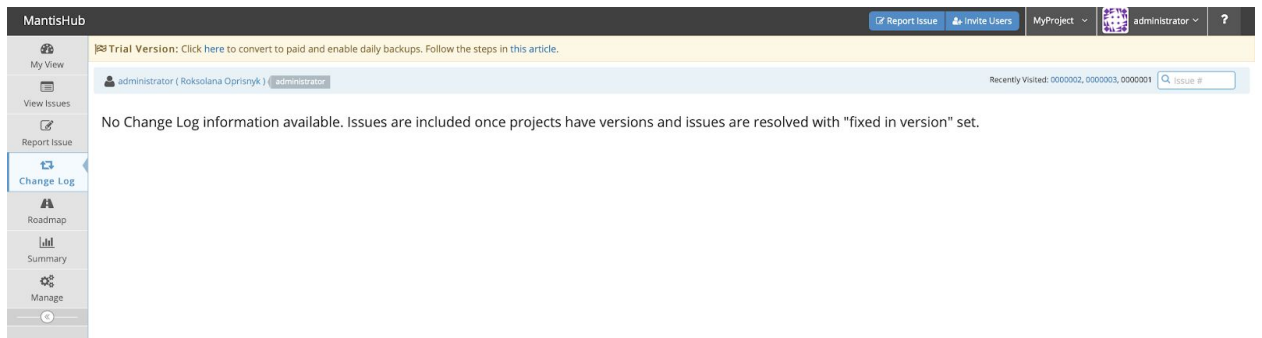


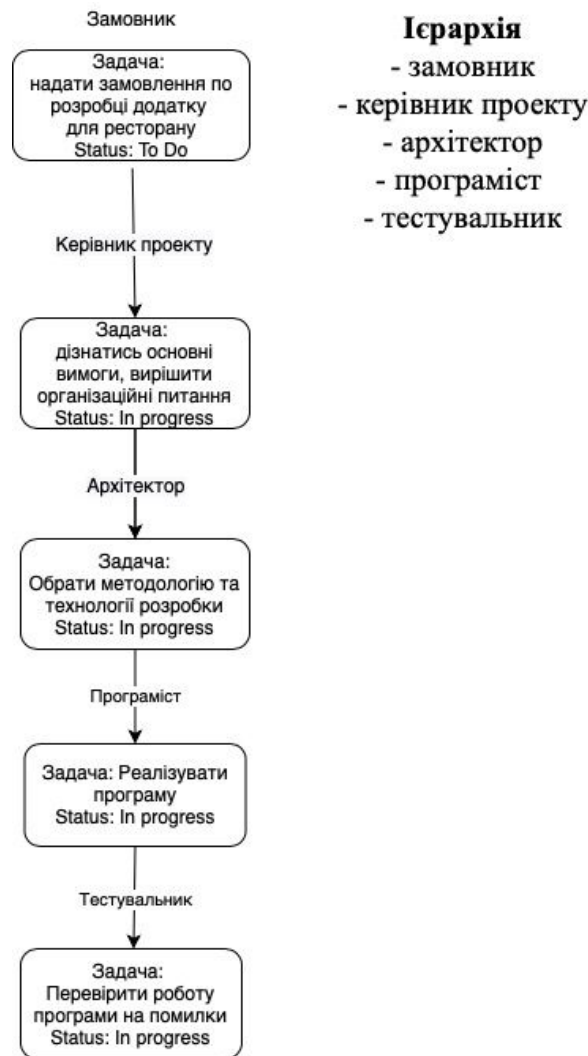
Рис. 8 Вікно журнал

Життєвий цикл питання (ticket):

1. Створення задачі з усіма подробицями(лаконічна назва, опис питання, якщо необхідні, то додаткові файли, ступінь важливості, і людину що буде виконувати цю задачу)
2. Далі програміст відмічає, що він бере задачу на виконання, або сповіщує, що вона заблокована та причину цьому
3. Далі програміст, після того, як зробить, відмічає завдання, як готове до тестування
4. Тестувальник перевіряє цю задачу на правильність і далі є два шляхи:
 - Якщо все добре, то задача є виконаною, якщо ні, то тестувальник змінює статус задачі з «готової до тестування» на «чекає на вирішення». І тоді все повертається до 2 і далі, поки задача не буде повністю вирішена.



Рис. Життєвий цикл дефекту



Висновки: Вивчила типові життєві цикли розробки програмного забезпечення. Усвідомила методику роботи з багтрекінгом, навчилася створювати питання, задавати їх статус, а також навчилася оперувати статистикою багтрекінгу.

Частина 2. Розробка поетапного плану проекту

Мета роботи: Скласти і описати поетапний план проекту.

Задача: розробити поетапний план проекту, створити діаграму Ганта, розбити проект на етапи, визначити обсяги постачання готового продукту в кожному проекті, розробити відповідну документацію. Для кожного етапу розрахувати ризики та розробити список анти-ризикових заходів.

Працюємо у Jira.

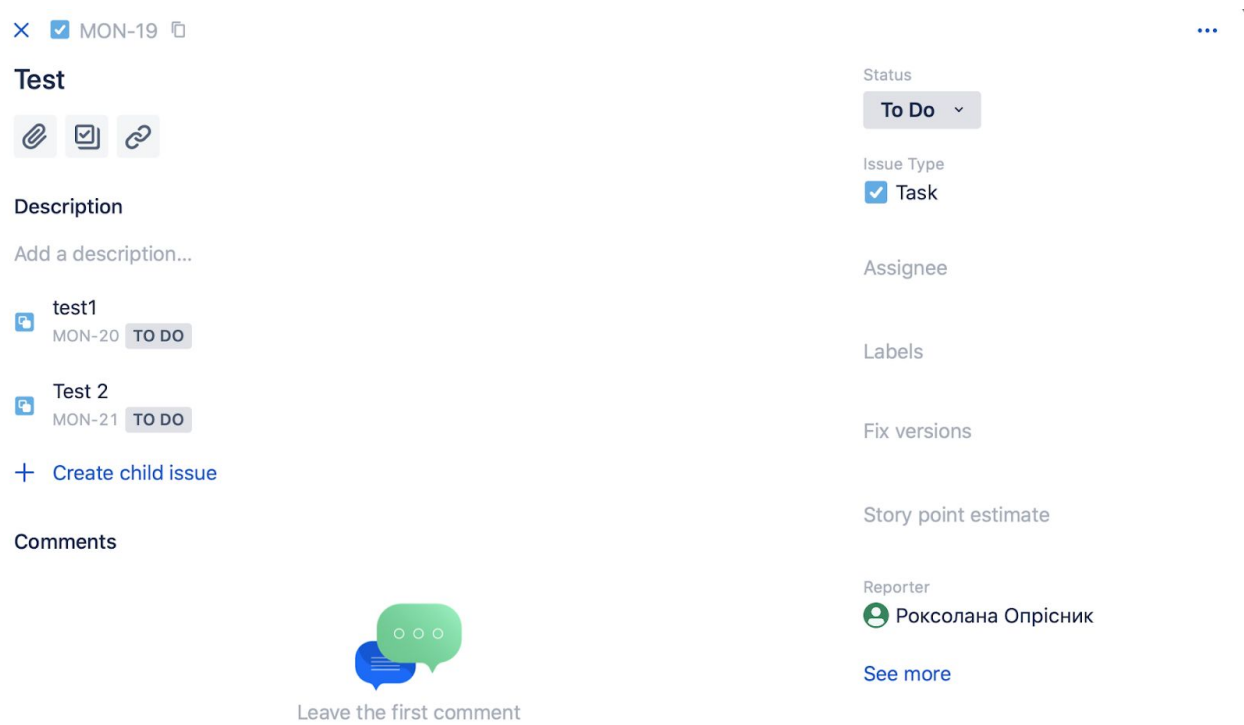


Рис. 10 Приклад однієї задачі з під-завданнями.

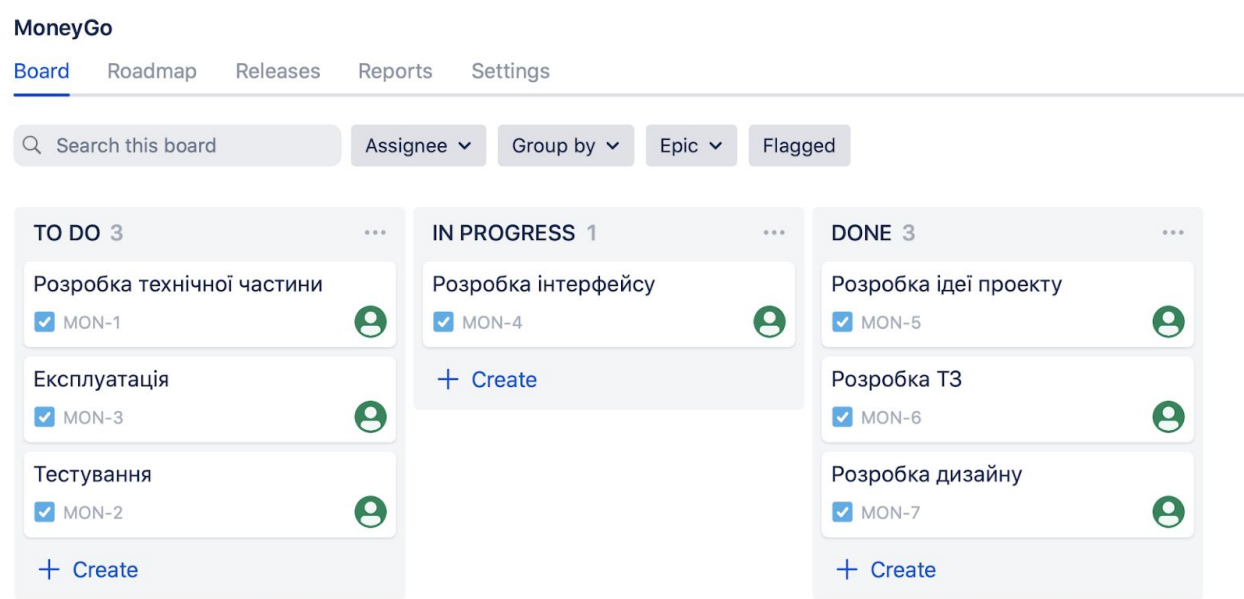


Рис. 11 Приклад всіх завдань з під-завданнями розташовані за статусом розробки.

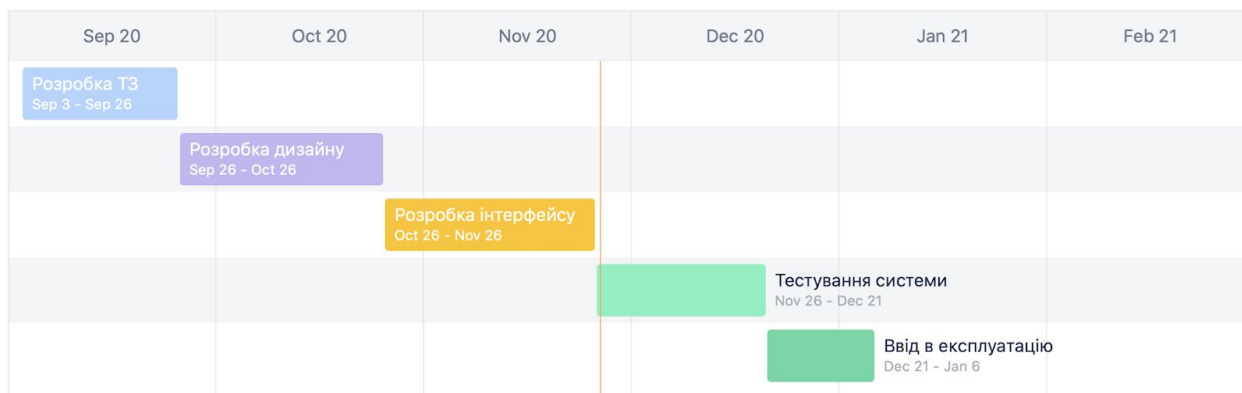


Рис. 13 Власне сама діаграма Ганта.

Ризик	Вплив	Ймовірність	Анти-ризикові заходи
Пошкодження місця зберігання проекту	Повна втрата розробленої програми	10%	Використовувати систему контролю версій з віддаленим репозиторієм
Ризики планування	Нераціональна втрата часу та затримки в отриманні результату	20%	Найняти менеджера
Ризик появи нових вимог	Укладнюється процес розробки	70%	Правильне проектування архітектури інформаційної системи
Ризики використання нестабільних технологій	Втрата працездатності частини програми	50%	Використання перевірених технологій

Ризики порушення закону про авторське право	Можливе притягнення до відповідальності та повне закриття проекту	35%	Перевірка використовуваних матеріалів на авторські права
Ризики появи багів	Можлива поява некоректної роботи програми	30%	Сформулювати зрозуміле технічне завдання. Прагнути виявити помилки до того, як на них наткнуться користувачі і не боятися їх виправити. Планувати не тільки розробку, але і тестування, реліз і виправлення помилок.

Висновок: Набула навичок складання і опису поетапного плану проекту, аналізу появи різноманітних ризиків при створенні інформаційної системи та знаходження їх рішення.