

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

ЛАБОРАТОРНА РОБОТА №3

з курсу: *«Проектування інформаційних систем»*
на тему: **„Розробка життєвого циклу проекту та системи
багтрекінгу. Розробка поетапного плану проекту.”**

виконав: студент IV курсу
групи ДА-71
Кузнєцов О.А1.

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.

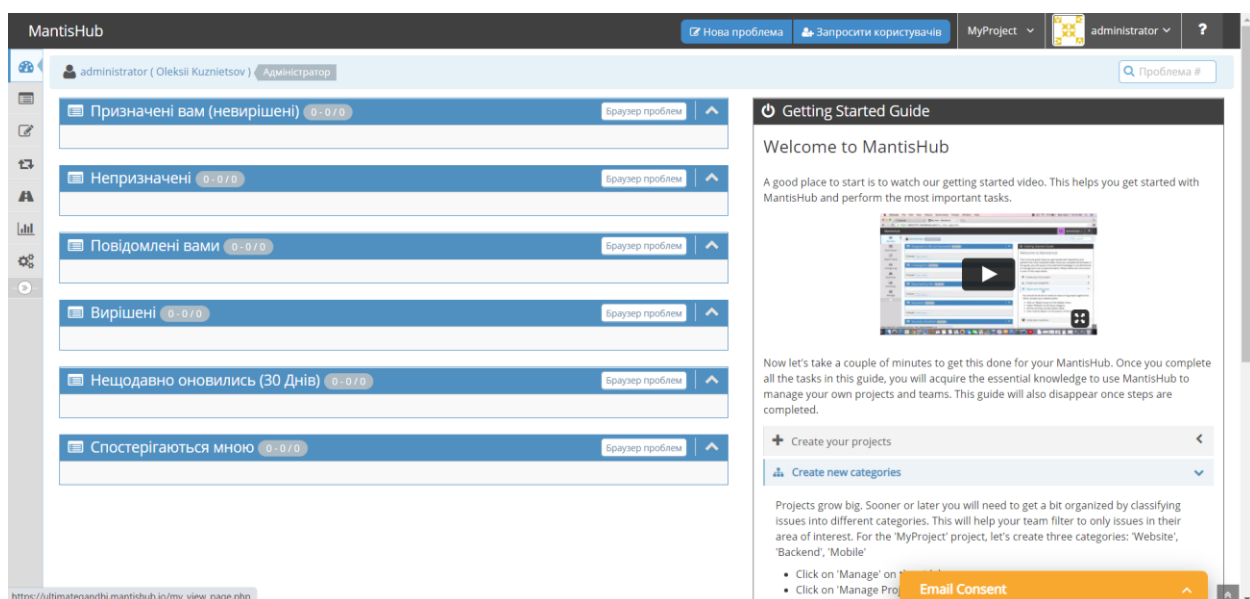
Завдання:

Використовуючи систему багтрекінга MantisBT або подібну необхідно виконати наступні дії:

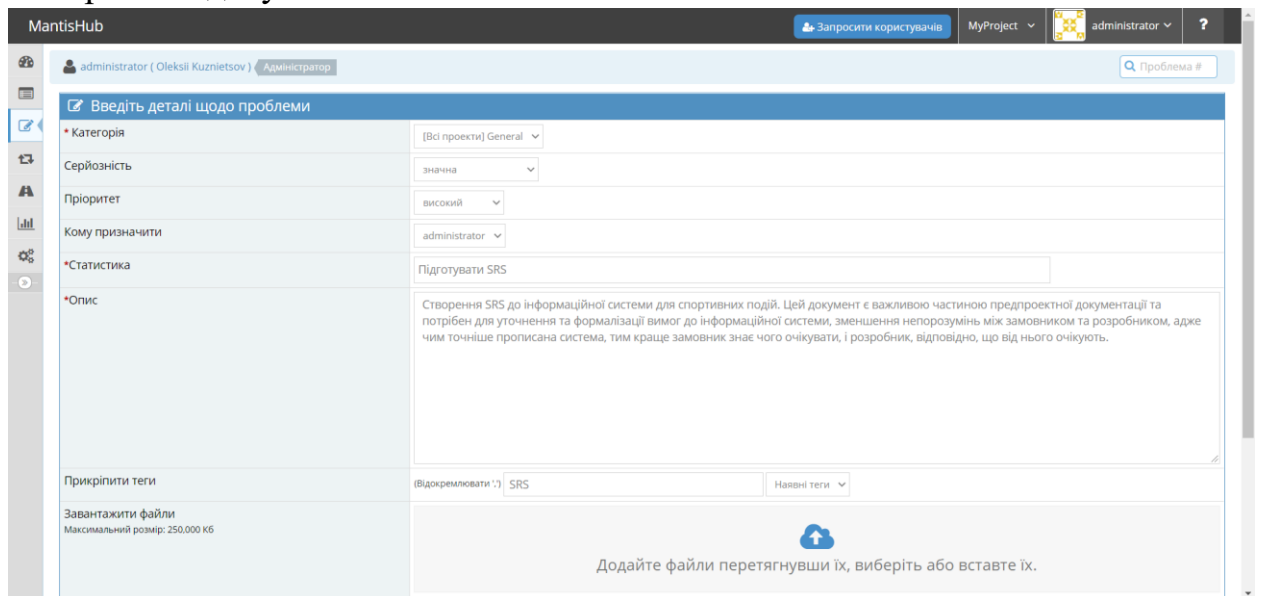
- створити питання (ticket)
- видалити питання
- змінити статус
- додати коментар
- прикріпити файл
- ознайомитися з журналом змін
- ознайомитися з виведенням статистики.

Хід роботи:

1. Розпочнемо роботу с MantisBT з головної сторінки:



2. Створимо задачу



The screenshot shows the 'Введіть деталі щодо проблеми' (Add Problem) form in MantisHub. The form includes fields for Category, Severity, Priority, Assignee, and Statistics. The description field contains text about creating an SRS for a sports event information system. The form also has a section for tags and a file upload area.

Введіть деталі щодо проблеми

* Категорія: [Всі проекти] General

Серйозність: значна

Пріоритет: високий

Кому призначити: administrator

*Статистика: Підготувати SRS

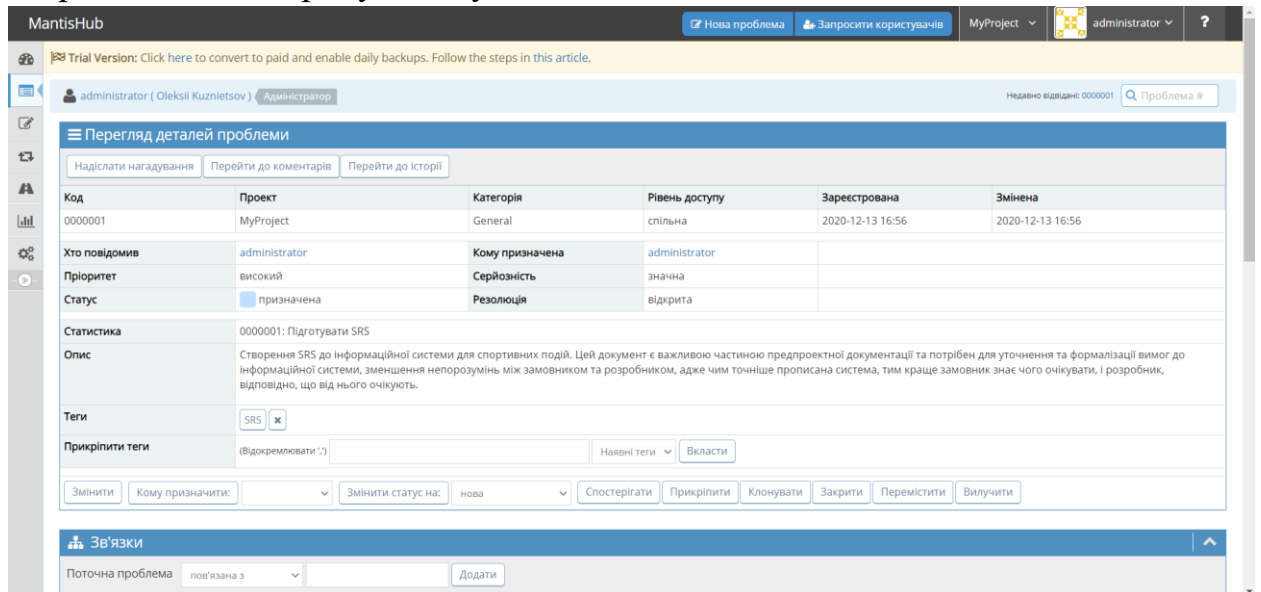
*Опис: Створення SRS до інформаційної системи для спортивних подій. Цей документ є важливою частиною предпроектної документації та потрібен для уточнення та формалізації вимог до інформаційної системи, зменшення непорозумінь між замовником та розробником, адже чим точніше прописана система, тим краще замовник знає чого очікувати, і розробник, відповідно, що від нього очікують.

Прикріпити теги: (Відокремлювати :') SRS Наявні теги

Завантажити файли: Максимальний розмір: 250.000 KB

Додайте файли перетягнувши їх, виберіть або вставте їх.

3. Переглянемо створену задачу



The screenshot shows the 'Перегляд деталей проблеми' (View Problem Details) page in MantisHub. It displays a table with problem details, including code, project, category, access level, registration date, and change date. Below the table, there are sections for statistics, description, tags, and actions.

Перегляд деталей проблеми

Надіслати нагадування | Перейти до коментарів | Перейти до історії

Код	Проект	Категорія	Рівень доступу	Зареєстрована	Змінена
0000001	MyProject	General	спільна	2020-12-13 16:56	2020-12-13 16:56

Хто повідомив: administrator | Кому призначена: administrator

Пріоритет: високий | Серйозність: значна

Статус: призначена | Резолюція: відкрита

Статистика: 0000001: Підготувати SRS

Опис: Створення SRS до інформаційної системи для спортивних подій. Цей документ є важливою частиною предпроектної документації та потрібен для уточнення та формалізації вимог до інформаційної системи, зменшення непорозумінь між замовником та розробником, адже чим точніше прописана система, тим краще замовник знає чого очікувати, і розробник, відповідно, що від нього очікують.

Теги: SRS ж

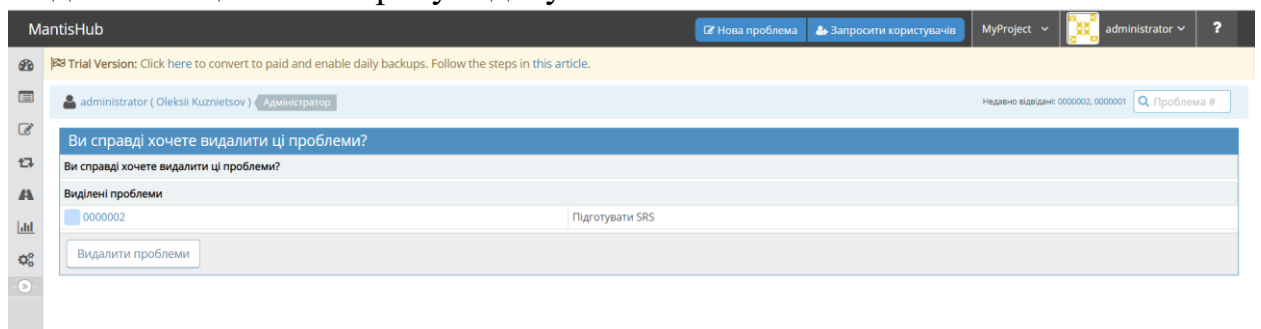
Прикріпити теги: (Відокремлювати :') Наявні теги: Вкласти

Змінити | Кому призначити: | Змінити статус на: нова | Спостерігати | Прикріпити | Клонувати | Закрити | Перемістити | Вилучити

Зв'язки

Поточна проблема: пов'язана з | Додати

4. Видаляємо щойно створену задачу



The screenshot shows the 'Ви справді хочете видалити ці проблеми?' (Are you really going to delete these problems?) confirmation page in MantisHub. It displays a table with problem details, including code, project, category, access level, registration date, and change date. Below the table, there is a button to delete the problem.

Ви справді хочете видалити ці проблеми?

Ви справді хочете видалити ці проблеми?

Виділені проблеми	
0000002	Підготувати SRS

Видалити проблеми

5. Додаємо до задачі коментар та файл:



6. Змінюємо статус задачі:

7. Переглянемо журнал змін:

8. Переглянемо статистику:

Життєвий цикл тікету:

1. Створення задачі з усіма подробицями: лаконічна назва, опис питання, якщо необхідні, то додаткові файли, ступінь важливості, і відповідального за 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 пункту 2 поки задача не буде повністю вирішена

Висновки: У цій роботі ми дослідили типові життєві цикли розробки програмного забезпечення. Вивчили методику роботи з багтрекінгом на прикладі системи MantisHub: навчитися створювати задачі, задавати їх статус, а також навчитися оперувати статистикою багтрекінга.

Частина 2. Розробка поетапного плану проекту.

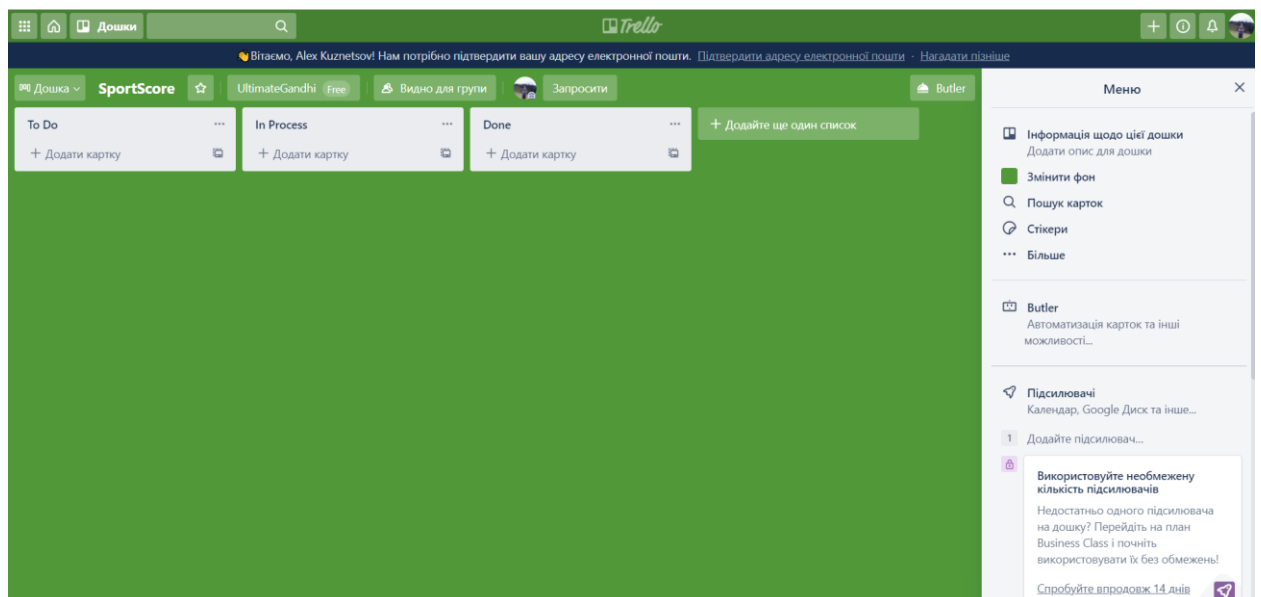
Мета роботи: Скласти і описати поетапний план проекту

Завдання:

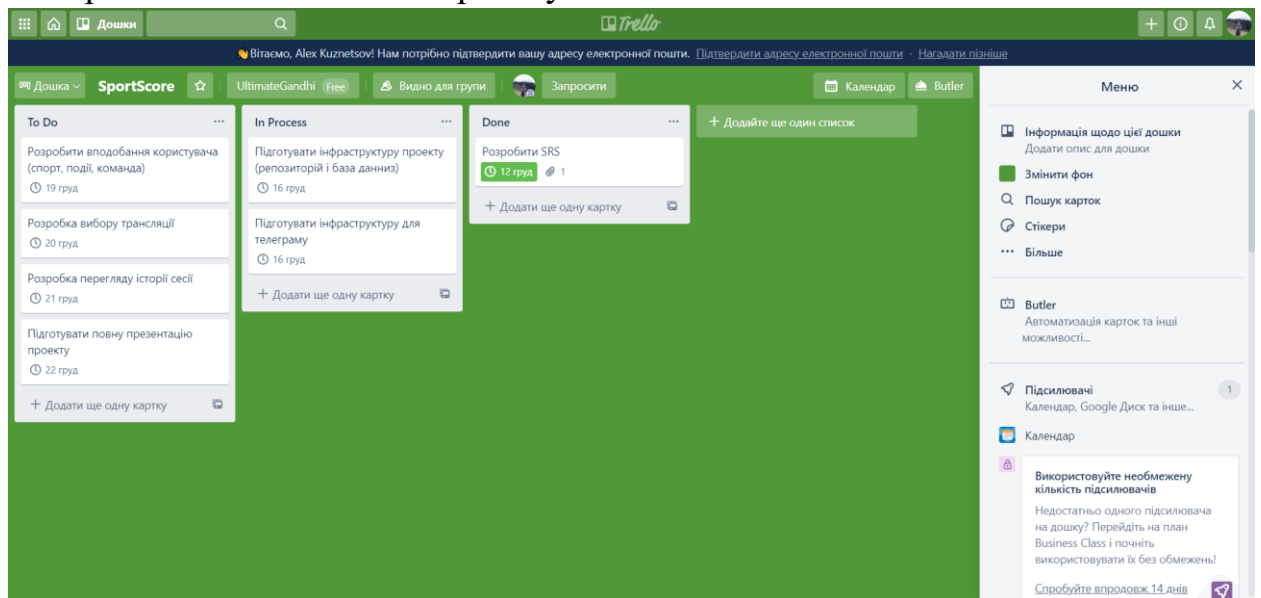
1. Опис об'єкту взяти з SRS в лабораторній роботі № 2.
2. Використовувати методологію створення програмного забезпечення: upper/lower design methods.
3. Створити поетапний план для об'єкта проектування за допомогою OpenProj або аналогічного програмного забезпечення.
4. Оформити діаграму Ганта для етапів проекту.
5. Оформити для кожного етапу проекту список ризиків, їх вплив на проект та ймовірність виникнення.
6. Для кожного ризику розробити список анти-ризикових заходів.

Хід роботи:

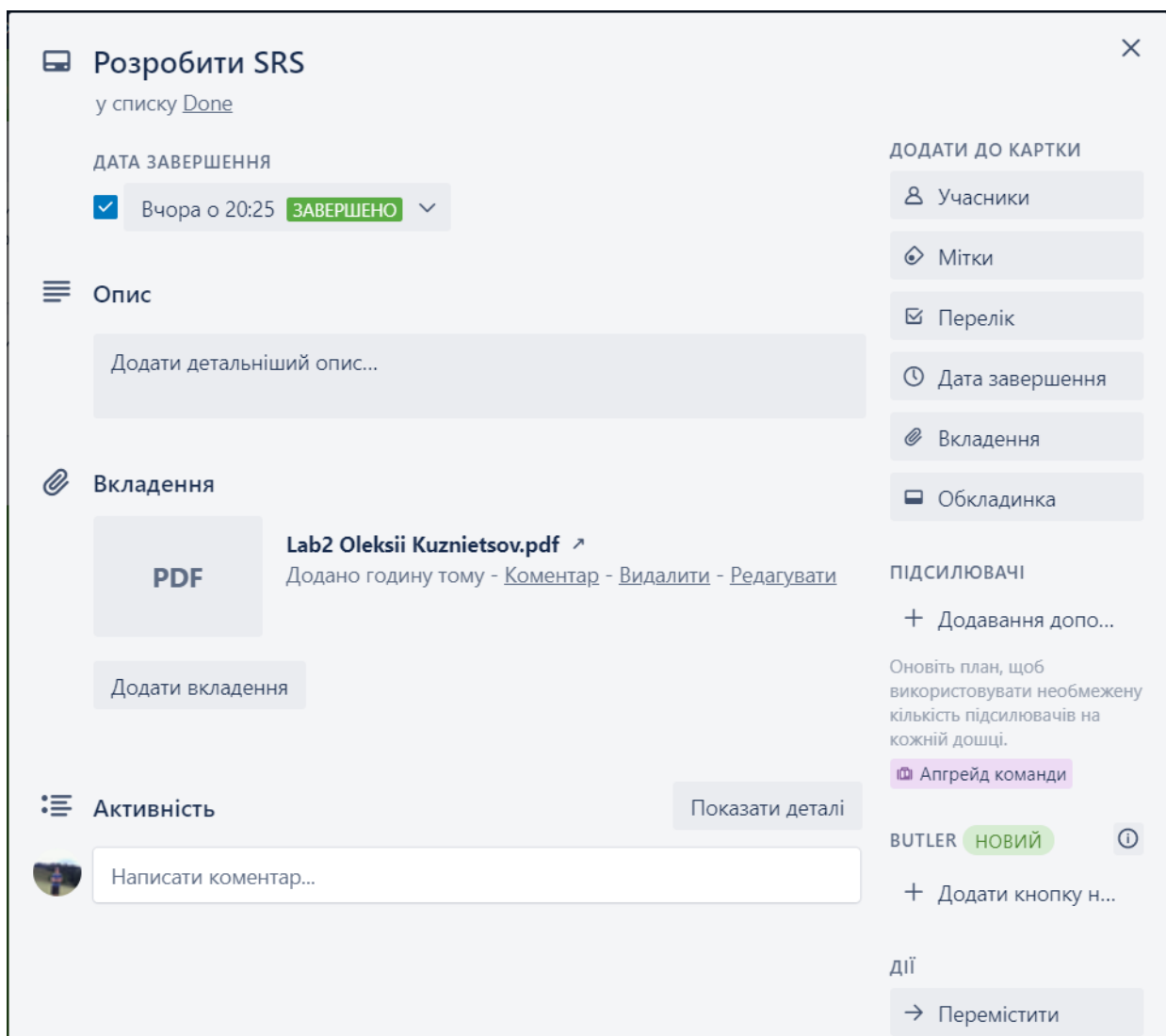
1. Розпочнемо роботу с Trello з нової дошки:



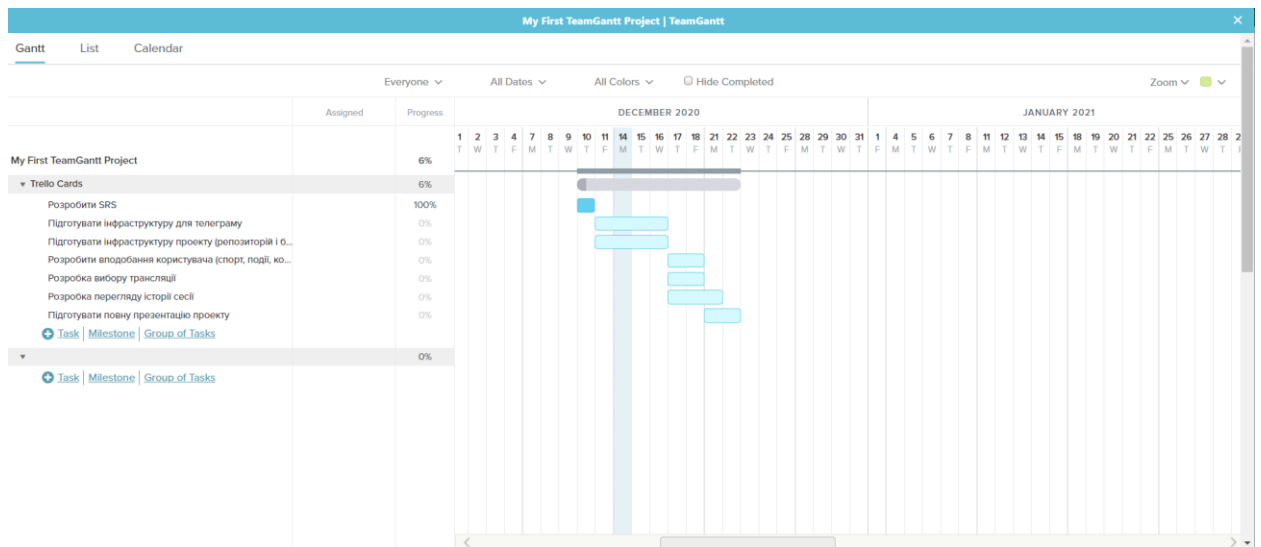
2. Створимо поетапний план проекту:



3. Приклад створення картки:

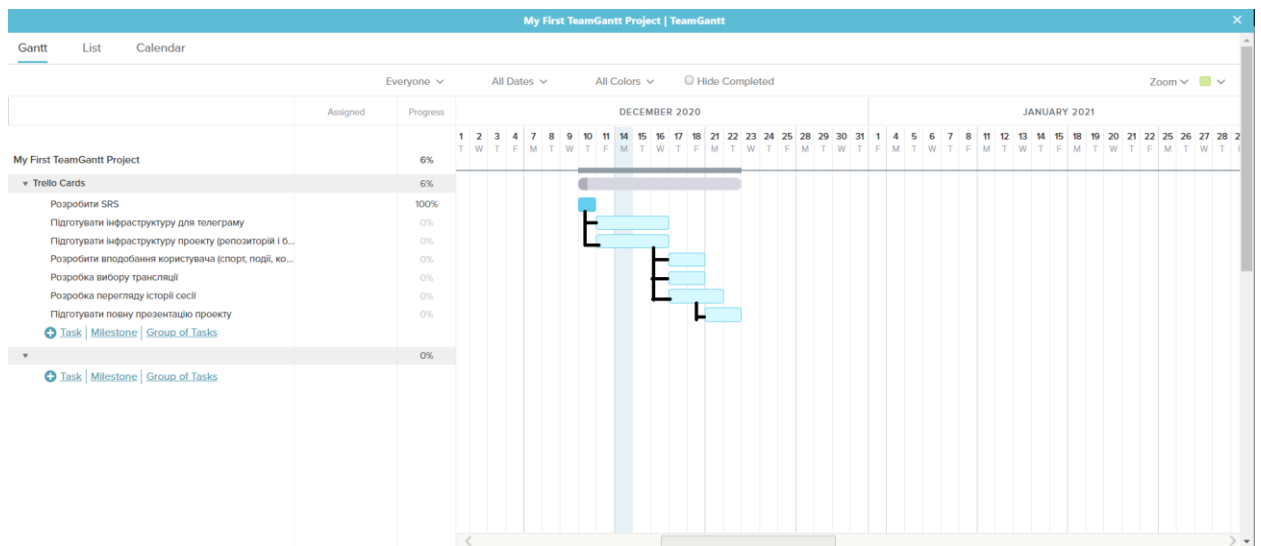


4. За допомогою інтеграції Trello з TeamGantt переглянемо діаграму Ганта проекту:



5. На базі діаграми Ганта побудуємо критичний шлях проекту.

Критичний шлях - це одна або кілька завдань, що впливають на термін завершення проекту. Це найкоротший шлях виконання проекту, тобто неможливо виконати проект раніше, ніж будуть завершені роботи по ланцюжку завдань проекту, що лежать на критичному шляху.



6. Оформимо для кожного етапу проекту таблицю ризиків, їх вплив на проект та ймовірність виникнення.

Ризик	Вплив	Ймовірність	Анти-ризикові заходи
Пошкодження місця зберігання проекту	Повна втрата розробленої програми	1%	Використовувати систему контролю версій з віддаленим репозиторієм або регулярно робити бекапи локальної системи
Ризики планування	Нераціональна втрата часу та затримки в отриманні результату	25%	Виконати перевірку ризиків проекту
Ризик появи нових вимог	Ускладняється процес розробки, частина проекту може стати неактуальною	75%	Правильне проектування архітектури інформаційної системи
Ризики порушення закону про авторське право	Можливе притягнення до відповідальності та повне закриття проекту	30%	Перевірка використовуваних матеріалів на авторські права

Висновки: У цій роботі ми розробили поетапний план проекту за допомогою хмарного веб-сервісу для управління проектами невеликих груп Trello. Також за допомогою вбудованих засобів візуалізували цей план у вигляді діаграми Гранта. Після цього виконали аналіз ризиків проекту та розробили список заходів, що мінімізують їх ймовірність.