**Міністерство освіти І науки України**

**національний університет “Львівська політехніка”**

**Кафедра СКС**

****

**Звіт про виконання практики**

На тему: «Аналітичний огляд джерел інформації за темою БКР»

Виконав: ст. гр. КІ-48

Папіш О. Р.

Прийняв: професор

Дунець Р.Б.

**Львів – 2022**

**Зміст**

[**АНАЛІЗ 3**](#_Toc103291028)

[**1.1 Аналіз попередніх проектних рішень 3**](#_Toc103291029)

[**1.1.1 Trello 3**](#_Toc103291030)

[**1.1.2 Jira 7**](#_Toc103291031)

[**1.1.3 ClickUp 10**](#_Toc103291032)

[**1.2.4 Todoist 15**](#_Toc103291033)

[**1.2 Аналіз технічного завдання 18**](#_Toc103291034)

[**1.3 Аналіз випадків використання 20**](#_Toc103291035)

[**1.4 Аналіз алгоритму роботи користувача 22**](#_Toc103291036)

[**1.5 Аналіз архітектурного рішення 24**](#_Toc103291037)

[**ВИБІР ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ 25**](#_Toc103291038)

[**2.1 Перелік технологій та пояснення вибору 25**](#_Toc103291039)

[**2.2 Короткий опис використаних технологій 26**](#_Toc103291040)

[**2.2.1 TypeScript 26**](#_Toc103291041)

[**1.2.2 Angular 26**](#_Toc103291042)

[**1.2.3 HTML 27**](#_Toc103291043)

[**1.2.4 SCSS 27**](#_Toc103291044)

[**1.2.5 C# 27**](#_Toc103291045)

[**1.2.5 ASP.NET Core 28**](#_Toc103291046)

[**1.2.6 EntityFramework Core 29**](#_Toc103291047)

[**1.2.7 SQLite 29**](#_Toc103291048)

[**Висновок 30**](#_Toc103291049)

[**Список літературних джерел 31**](#_Toc103291050)

# **АНАЛІЗ**

## **1.1 Аналіз попередніх проектних рішень**

### **1.1.1 Trello**

Trello [1] – це неймовірний веб-інструмент для керування проектами та спільної роботи, який допоможе вам планувати свої проекти на одній платформі. Він неймовірно візуальний і дуже адаптивний, що полегшує ваше життя, коли над вашим проектом працює кілька людей, особливо коли вони розкидані по різних місцях.

Співзасновником платформи є засновник Fog Creek Software Джоел Спольскі та випущена на заході TechCrunch у 2011 році. Іншим засновиником є Майкл Прайор. Джоел вже знайоме ім‘я у світі підриємництва та стартапів, а його блог Joel on Software дійсно дуже популярний. Він ділиться досвідом із запуску нових продуктів у своєму блозі, а тепер допомагає керувати проектами Trello.

Легкість роботи з Trello пояснюється його трирівневою інформаційною системою дошки, списки та картки. Проети та інформація про ці проекти будуть організовані в дошки, які міститимуть списки.

Списками буде присвоєно назви та вони будуть мати власні картки. Самі ці картки становлять основиний блок дошки Trello. Користувачі можуть перетягувати різні списки на певну дошку, виконувати завдання в цих списках, створювати скільки завгодно дощок, зберігати їх у хмарі або у вашому профілі, залежно від того що вам більше подобається.

**Способів використання Trello може бути багато. Ось деякі з них:**

1. Поділитися файлами з членами вашої команди
2. Прокоментувати карту, щоб оновити думки працівників
3. Слідкувати за списками справ
4. Встановити кольорові етикетки відповідно до приорітету завдань
5. Виконувати пакетну обробку карток у списку
6. Вставляти нові карти (їх може бути необмежена кількість)

**Переваги Trello**

1. Менш складна структура ціноутворення

У порівнянні з іншими інструментами управління проектами, Trello має менш складну структуру цін. У вас є версія, де ви можете запрошувати необмежену кількість учасників, створювати дошки, карти та списки. Але версія бізнес-класу коштує 25 доларів США на місяць, але вона надає безліч функцій, таких як інтеграція Google Apps, легкий масовий кспорт і можливість адміністратора отримувати доступ до всіх дошок і керувати ними(включаючи приватні), обмежує видимість дошки...

1. Миттєве розуміння того, коли наближається кінцевий термін виконання завдань

З Trello Board ви ніколи не пропустите дедлайн. Коли ви створюєте картки, ви можете додати до них терміни виконання. Коли дата наближається, картка стане жовтою, а коли ви пройдете дату, стане червоною. Ви можете позначити дату виконання як «повну», щоб уникнути червого кольору на дошці.

1. Мобільний

Trello працює на кожній платформі. На комп‘ютері, планшеті, чи телефоні інструмент може переформатувати себе до будь-якого розміру екрана.

1. Trello дотримується системи Канбан

Система Канбан допомагає вам планувати завдання на менші компонети за допомогою системи дошок і карток. Ця система, заснована Toyota в 1950-х роках, відіграє важливу роль у виявленні вузьких місць у процесі розробки.

1. На вашому столі більше немає нотаток

Уявіть, що ви маєте простий стіл без жодних листівок, які прикрашають стіни та краї столів і стільці. Trello – це ваш новий Post-it, і він знаходиться в Інтернеті, до якого ви можете отримати доступ до нього з будь-якої точки світу. Насправді Trello відомий як «Канбан майбутнього». Більше ніяких зім‘ятих і викинутих листівок Post-it у смітник.

1. Миттєві сповіщення

Ніколи не пропускайте завдання, оскільки у вас безперебійні сповіщення, коли завдання оновлюються, коментуються або видалються. І це надійде до вас електронною поштою.

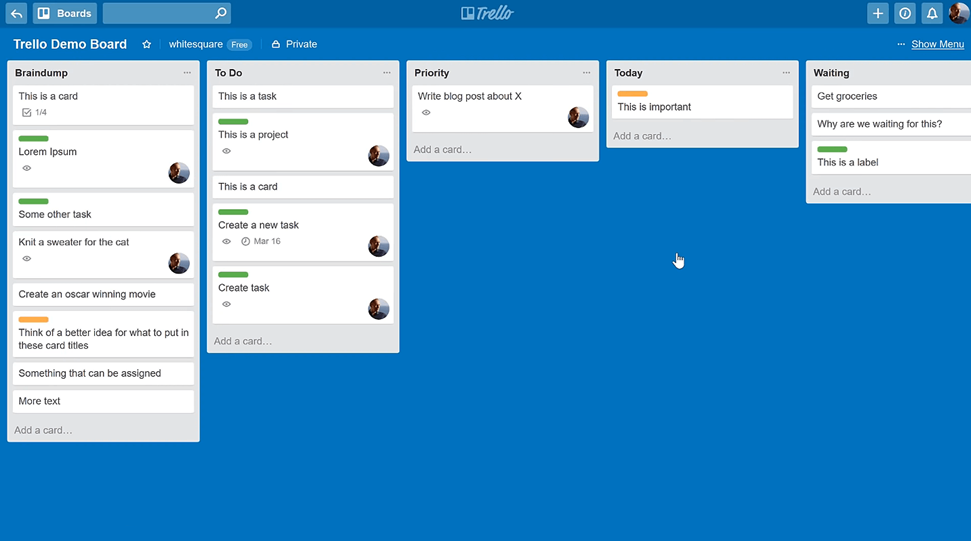


Рис. 1.1. Дошка Trello

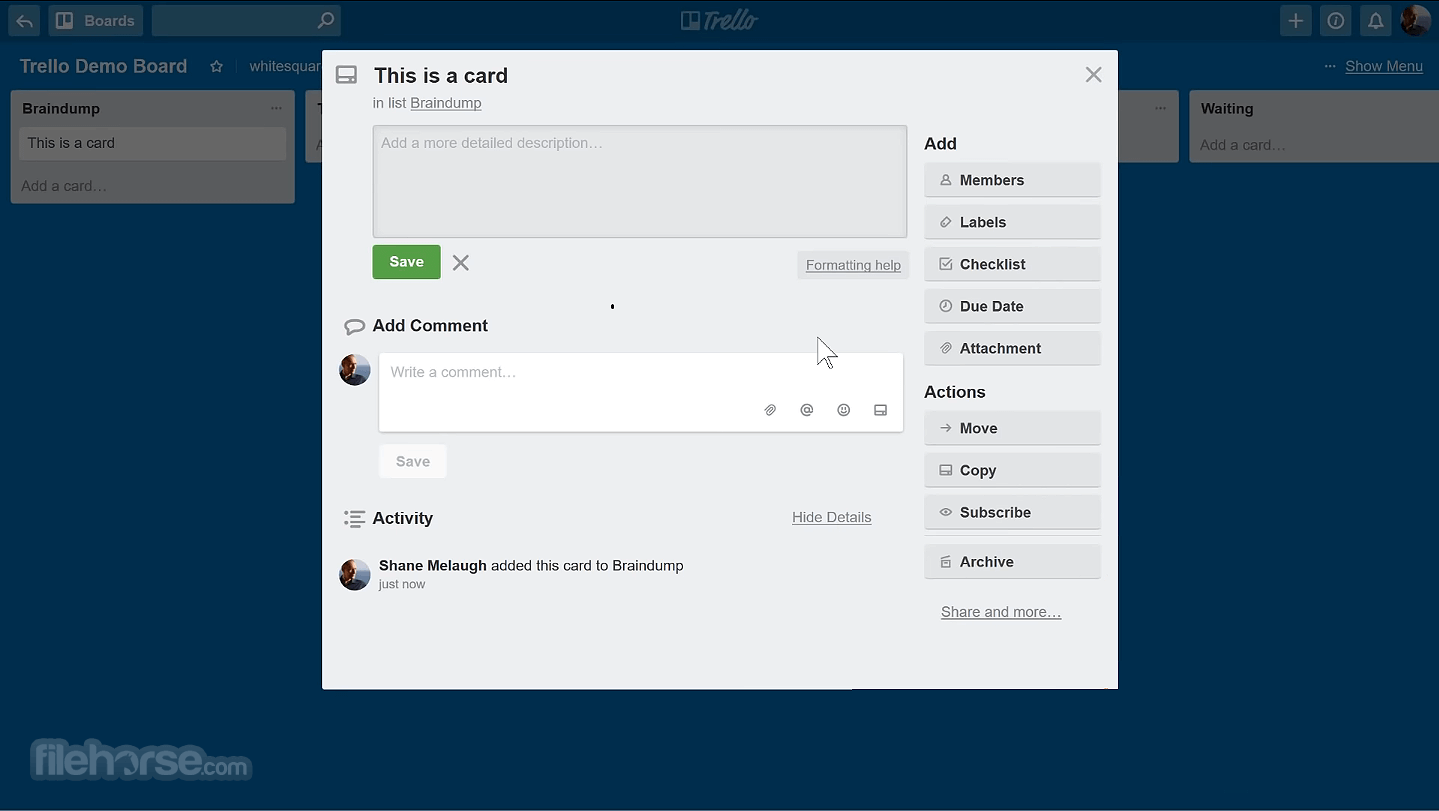


Рис. 1.2. Деталі картки

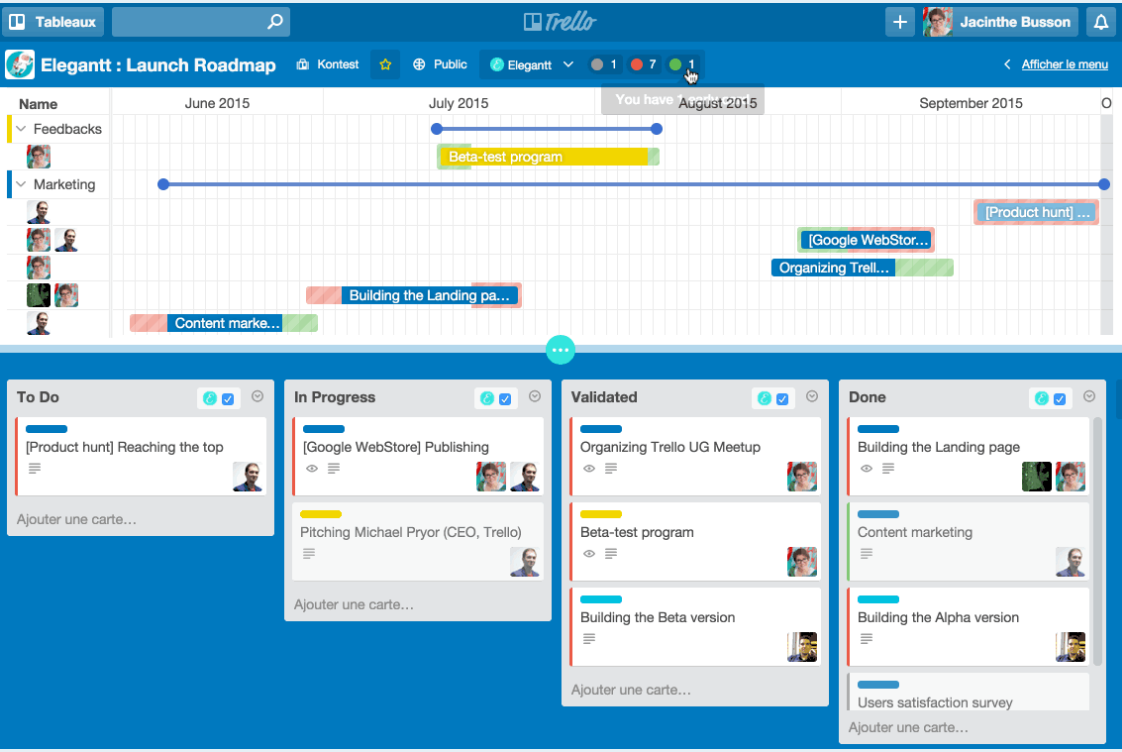


Рис. 1.3. Графік

### **1.1.2 Jira**

Jira [2] – це інструмент відстеження помилок, який, в основному, був розроблений австралійською компанією Atlassian у 2002 році. По суті, це інсрумент відстеження помилок. Пізніше в 2012 році, він почав свої послуги, такі як керування проектами. Цей інструмент в основному використовується для відстеження помилок програмного забезпечення та мобільних додатків.

Jira в основному використовується Agile-розробниками для налаштування робочого процесу та для командної співпраці. Цим інструментом користувалися 75000 користувачів у 122 країнах. Jira містить 4 пакети. Вони згадуються нижче:

1. Jira core
2. Jira Software
3. Jira Ops
4. Jira service desk

**Пакети Jira**

1. Jira core

Це рішення для управніння проектами та завданнями. В організації кожна ділова людина використовує план і звіт про роботу. Новий бізнес-шаблон цього ядра – дуже ефективний для підтримки кількох команд, проектів і робочих процесів.

1. Jira Software

Програмне забезпечення Jira розроблено для відстеження помилок програмного забезпечення та мобільних додатків.

1. Jira Ops

Це місце для реагування на інциденти, систем зв‘язку та створення послідовного процесу. На програмному забезпеченні Atlassian Jira побудовано Jira Ops. Його мета – автоматизувати й упорядкувати реагування на інциденти, а також забезпечити видимість поточних інцидентів, а також надання оновлення про чат і статус.

1. Jira service desk

Jira service desk – сучасне програмне забезпечення для служби обслуговування. Він забезпечує хорошу видимість і забезпечує хорошу структуру для роботи, щоб забезпечити кращий сервіс.

**Переваги Jira**

1. В Jira доступна інтеграція. Це означає, що програмне забезпечення для відстеження проблем і проектів інтегроване з кількома програмами інших розробників.
2. Jira підходить для різних типів користувачів, таких як розробники, менеджери проектів, інженери, нетехнічні спеціалісти тощо.
3. Підтримує вимогу дорожньої карти.
4. Jira дозволяє створювати користувачам будь які проблеми.

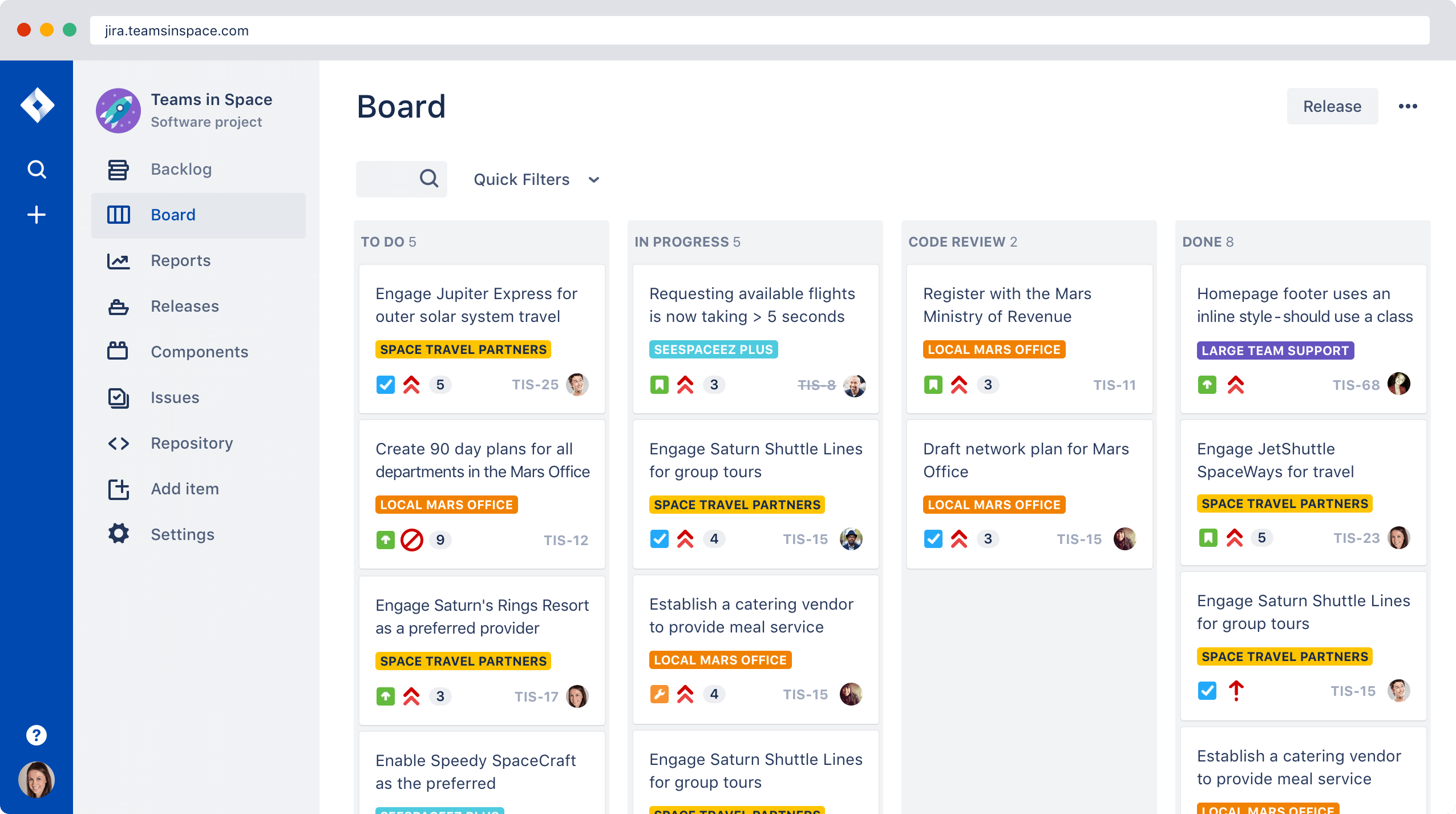


Рис. 1.4. Дошка Jira

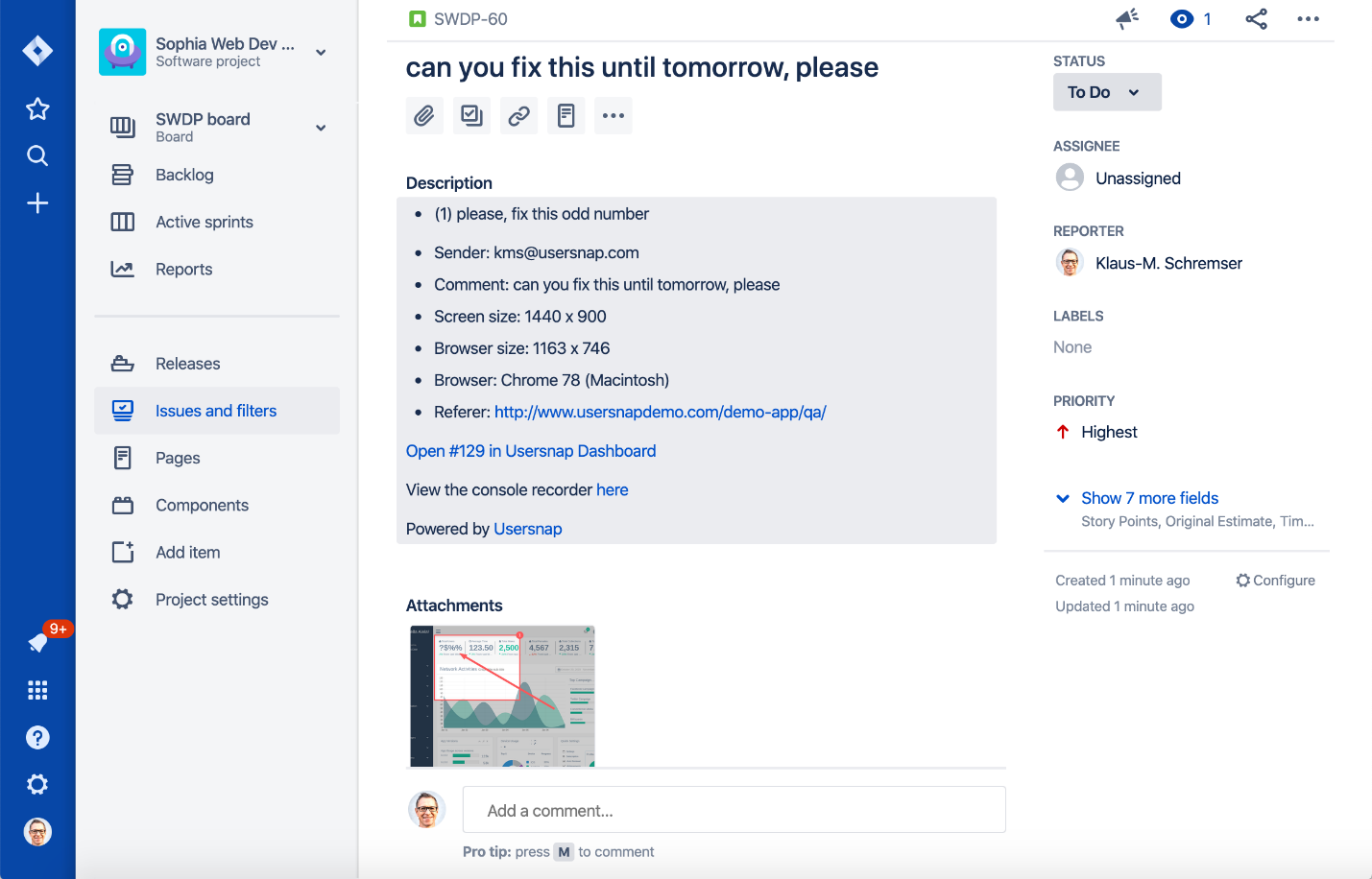


Рис. 1.5. Деталі картки

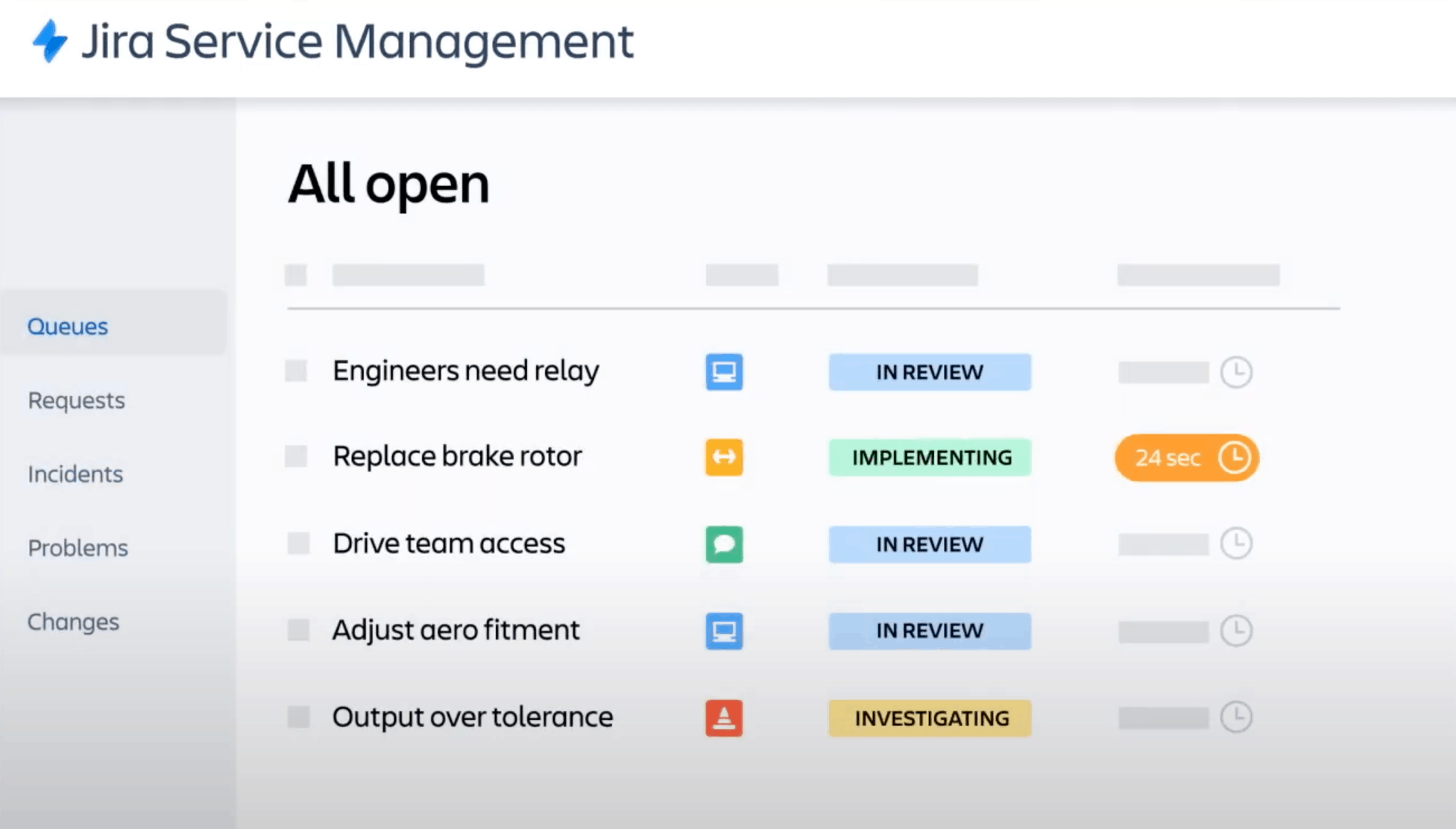


Рис. 1.6. Відкриті черги

### **1.1.3 ClickUp**

ClickUp [3] – це хмарна робоча платформа для всіх типів і розмірів команд і компаній. Він поєднує важливі бізнес-додатки та централізує інформацію про копанію в єдине онлайн-рішення. Призначайте завдання членам команди, керуйте проектами для клієнтів і співпарцюйте з колегами на документами. ClickUp надає всі інструменти та функції для ефективного, видимого та доступного виконання роботи.

Крім того, додаток для підвищення продуктивності роботи дозволяє переглядати робочі елементи та дані в кількох представленнях для кращого розуміння та швидкого відстеження. Виберіть подання списку завдань, дошку для робочих процесів, вікно для інформаційних панелей або подання Ганта для розкладів проекту. ClickUp також надає вам перегляд календаря, подання діяльності, карти розуму, подання робочого навантаження, подання таблиці та карти, серед іншого. Програмне забезпечення налаштовується, тому ви можете вносити зміни до свого робочого простору, опису робочого стану, кольорів і тем, а також вибору функцій.

ClickUp дозволяє командам і компаніям вирішувати бізнес-проблеми залежно від того, якою сферою вони хочуть керувати:

1. Управління процесами – оптимізовуйте проекти та робочі процеси за допомогою правильних кроків.
2. Керування завданнями – організуйте завдання, відстежуйте зміни та керуйте командою.
3. Керування часом – створюйте розклади, упорядковуйте календарі, керуйте потужністю та відстежуйте час.
4. Співпраця та звітність – спілкуйтесь безпосередньо та в контексті, створюйте готові до використання або натсроювані звіти та безпечно діліться інформацією.

**Функції ClickUp**

Компанії будь-якого розміру використовують програмне забезпечення для управління проектами ClickUp, щоб допомогти своїм командам досягти бізнес-цілей. Ось випадки використання, коли ClickUp надає потрібні функції для роботи:

1. Управління проектом

ClickUp має функції, які допомагають проектним командам планувати, а також візуалізувати свій план з допомогою кількох переглядів. Використовуйте перегляд списку, щоб керувати пріоритетами, перегляд календаря, щоб перевірити часові шкали, і перегляд Ганта, щоб відстежувати процес. Команди можуть легко співпрацювати, додаючи коментарі до будь-якого завдання чи документа та отримуючи сповіщення про будь-які зміни. Вони можуть використовувати інформаційні панелі, щоб швидко отримувати інформацію, а також підтримувати графіки та бюджети відповідно до відстеження часу.

1. Гнучка розробка

Функції ClickUp можна налаштувати, щоб команди з розробки програмного забезпечення могли використовувати платформу для відстеження помилок, керування спринтами або запуску продуктів. Вони можуть створювати інформаційні панелі Scrum і переглядати прогрес за допомогою різних гнучких діаграм. Розробники можуть співпрацювати над кодом, автоматизувати систему точок спринту, виділяти синтаксик та інтегрувати Git.

1. Віддалена робота

Команди можуть продовжувати свою роботу, виконуючи завдання та виокнувати проекти, працюючи віддалени. ClickUp дозволяє командам призначати завдання та співпрацювати над ними, ділитися документами для встановлення чітких процесів, створювати нагадування та переглядати, як виконуються завдання, де б вони не були.

**Переваги ClickUp**

1. Підходить для команд та одиночних користувачів

ClickUp надає командам потужні інструменти. Але на відміну від інших доступних рішень для управління проектами, ClickUp добре працює і для окреми користувачів. Його набір функцій ідеально підходить для керування щоденними справами. Інтерфейс користувача простий для розуміння та використання, а інтеграції допомагають заощадити один дорогоцінний час для команди під час роботи з кількома додатками.

1. ClickUp надає чудові візуальні зображення та інформацію зі своїх інформаційних панелей

Перемикання преглядів показує різні точки зору, що допомогає побачити цінну інформацію. Ця функція ClickUp відрізняє програмне забезпечення від його конкурентів.

1. Повнофункціональна безкоштовна версія

Більшість безкоштовних версій програмного забезпечення містять основні та обмежені функції. На щастя, ClickUp пропонує значну кількість функцій, які зазвичай містяться в платних версіях іншого програмного забезпечення. Наприклад, він включає в себе різні подання, панелі інструментів, перевірку, спільну роботу в реальному часі, залежності завдань, інтеграцію, відстеження часу, підтримку 24/7 та 2FA.

1. Орієнтований на клієнта

ClickUp постійно надає нові функції, які потрібні клієнтам.

1. Дружній макет

ClickUp дозволяти легко побачити та зрозуміти інформацію. Мекет дозволяє командам організовувати декілька проектів.

1. Економічно ефективний

Командам і організаціям доводиться використовувати для своїх проектів два або більше додатків, і це може швидко накопичити витрати. ClickUp може замінити кілька програм, необхідних для управління різними типами проектів. Це мінімізує витрати, надаючи єдину платформу знайому всім користувачам, і не вимагає використання іншого інструменту керування завданнями.

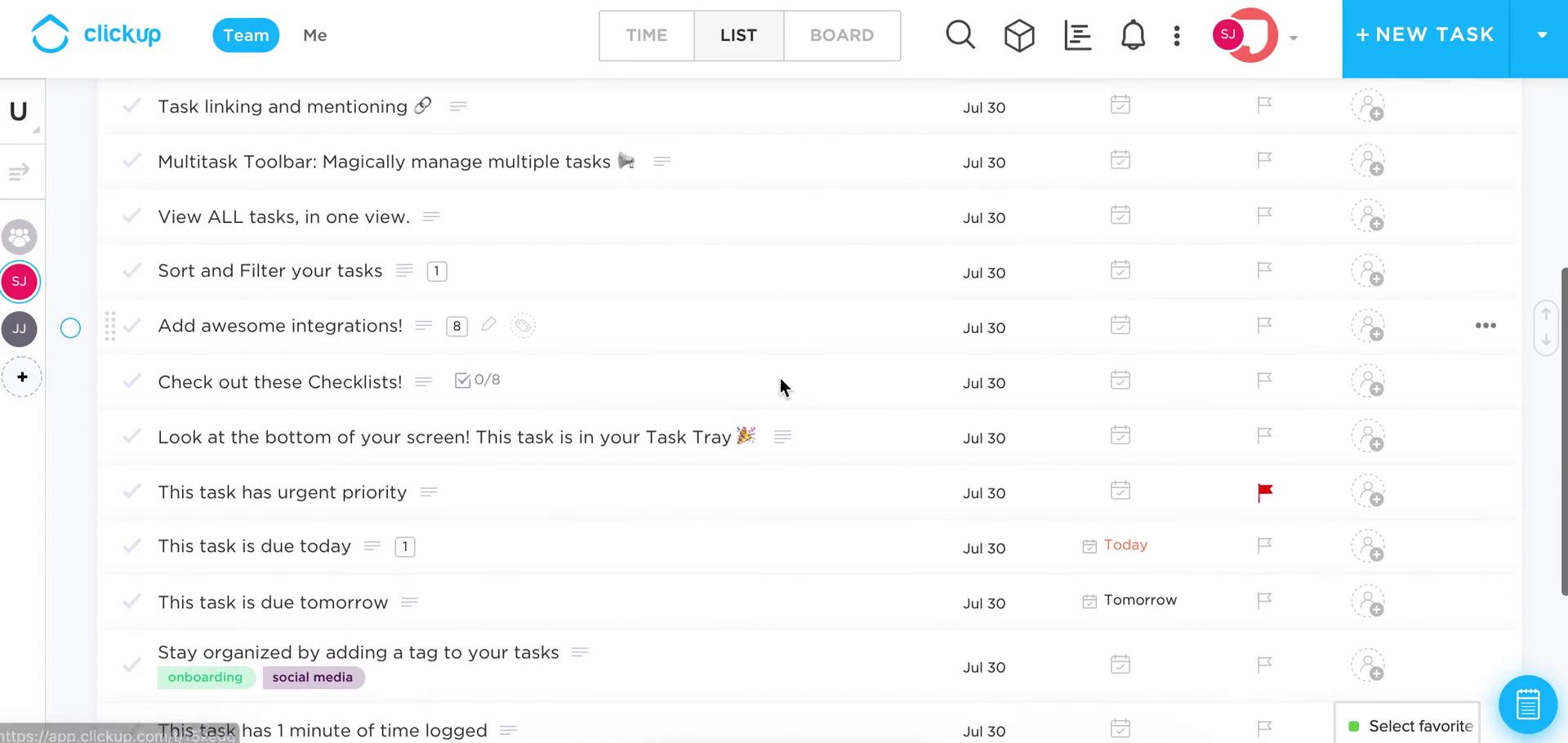


Рис. 1.7. Дошка ClickUp

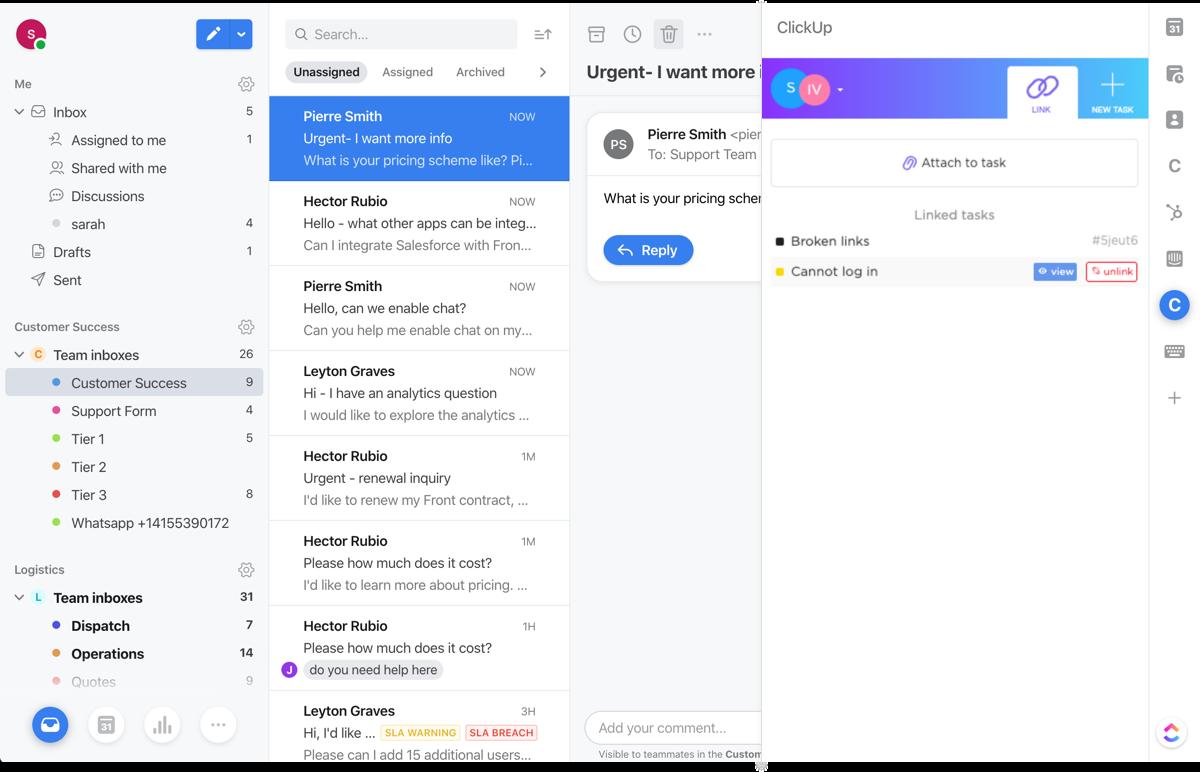


Рис. 1.8. Комунікація між командою

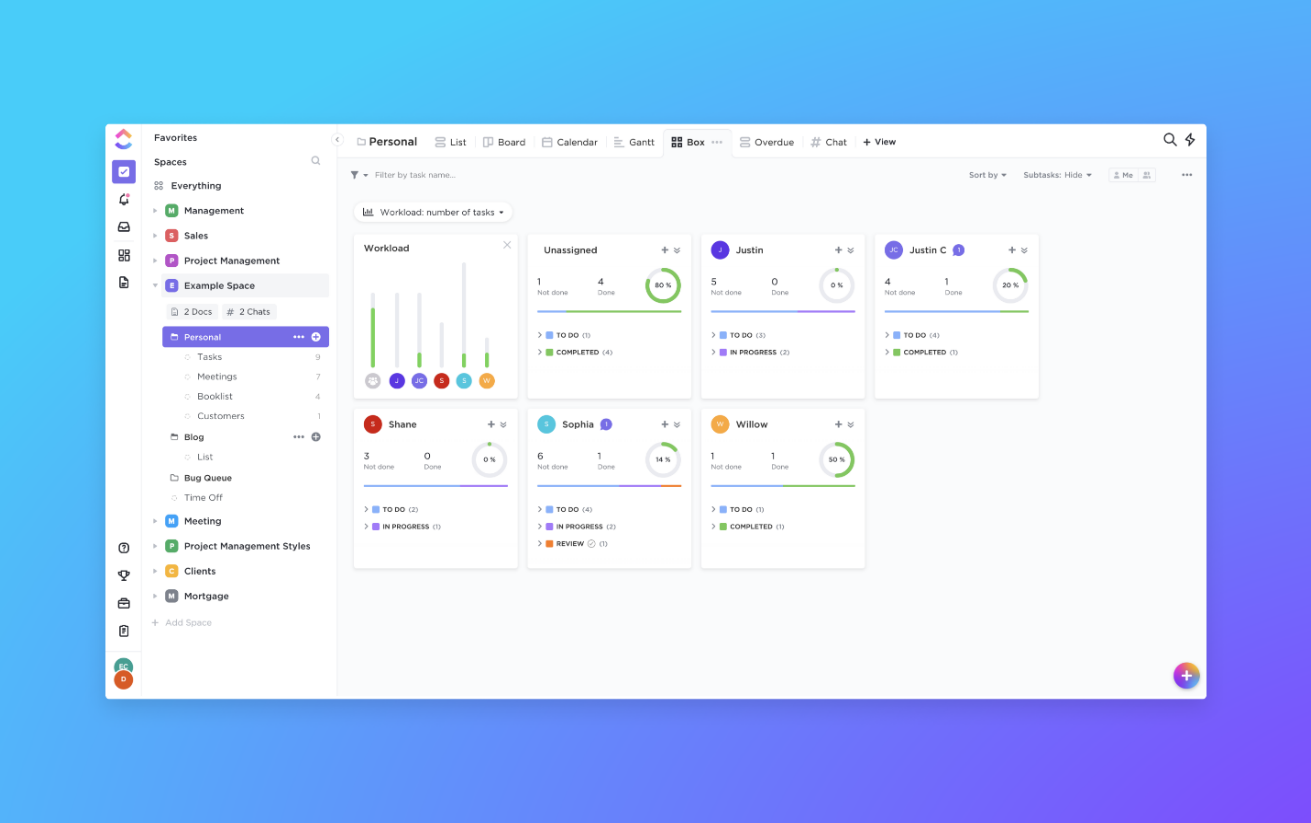


Рис. 1.9. Графіки

### **1.2.4 Todoist**

Todoist [4] є одним із найуживаніших програм на сучаному ринку. Він наповнений чудовими функціями, які підвищують продуктивність, організовують завдання та встановлюють графіки, а також терміни. Система відрізняється від інших подібних продуктів на ринку, що робить її настільки популярною, надаючи користувачам безліч плагінів і програм для барузерів, електронної пошти, мобільних пристроїв, настільних комп‘ютерів і ноутбуків. Todoist пропонується в пакеті freemium, однак найкращі функції та інструменти, які надає система, можна отримати через обліковий запис Premium. Його можна отримати за 29 доларів США на рік.

Функції Todoist:

1. Кілька приорітетів
2. Сповіщення
3. Централізована командна оплата
4. Підпроекти
5. Синхронізація даних в режимі реального часу
6. Завдання
7. Легко додавати та видаляти користувачів
8. Коментарі
9. Підзавдання

10) Шаблони проектів

1. Інтуїтивні фільтри
2. Проекти
3. Етикетки
4. Нагадування
5. Завантаження файлів
6. Візуалізація продуктивності
7. Пошук в архіві
8. Автоматичне резервне копіювання

**Переваги Todoist**

Основними перевагами Todoist є його міжплатформенність, зручний інтерфейс та різноманітні інструменти класифікації, підтримка спільної роботи та його здатність працювати в автоматичному режимі. Ось більше деталей:

1. Кросплатформенність

Todoist – це програма для виконання завдань, яка добре працює на різних платформах і пристроях, таких як Mac, Apple iOS, Android та Windows. Його також можна використовувати в увсіх провідних веб-браузерах.

1. Інтуїтивно зрозумілий інтерфейс користувача та інструменти класифікації

Система може похвалитися інтуїтивно зрозумілим інтерфейсом користувача, який оснащений інтрегрованими діаграмами продуктивності, доступними ддля преміалних облікових записів.

1. Посилена співпраця

Ще однією сильною стороною Todoist є розширена співпраця, яку він забезпечує користувачам. Вони можуть легко спілкуватись один з одним під час виконання свої завдань. Це збільшує їх загальну продуктивність.

1. Офлайн підтримка

Користувачам також не протрібно турбуватися про роботу над своїми завданнями, поки вони не підключені до мережі. Це тому, що Todoist має автономну функціональність, якою можна скористатися в таких ситуаціях.

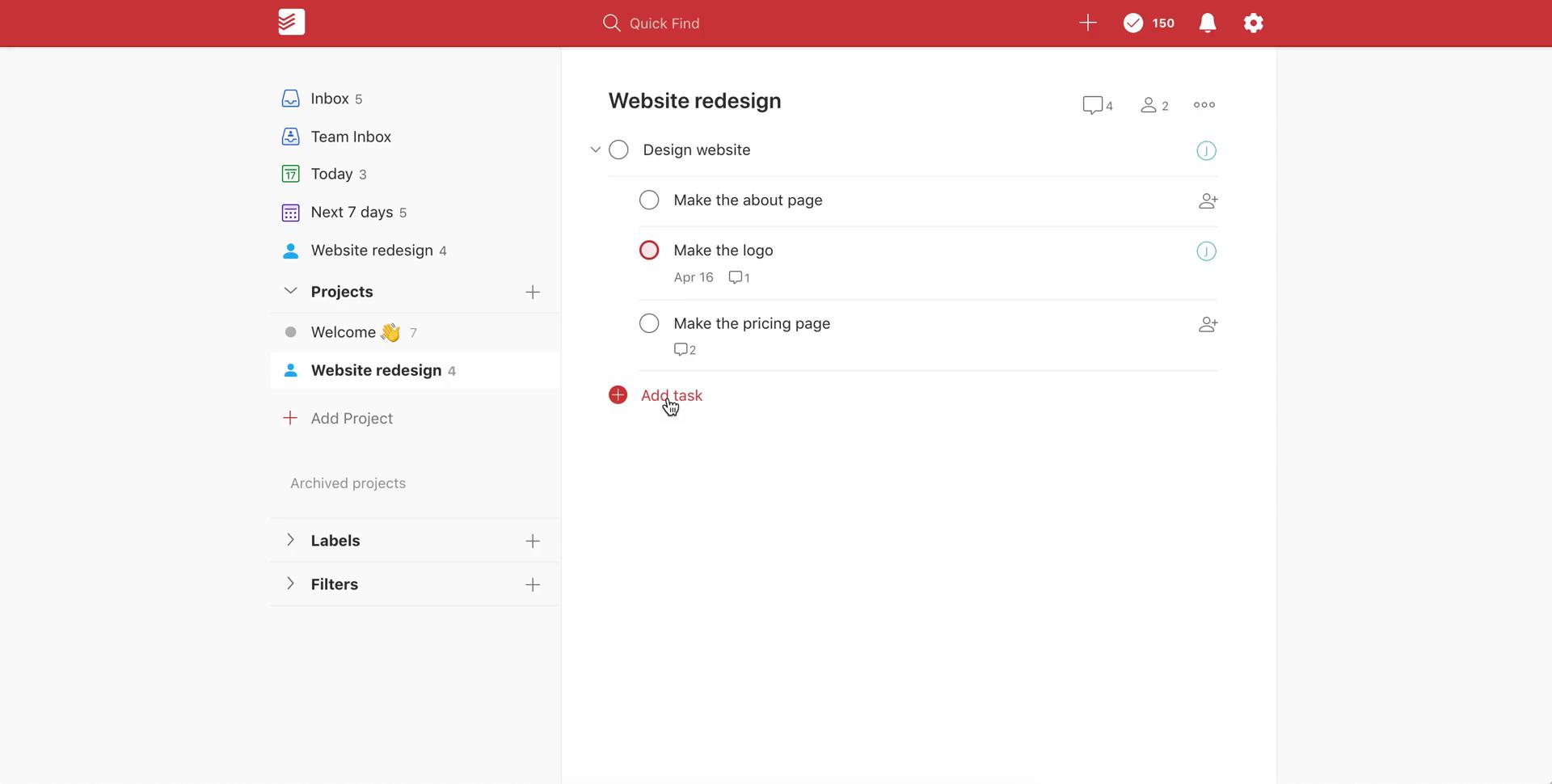


Рис. 1.10. Дошка Todoist

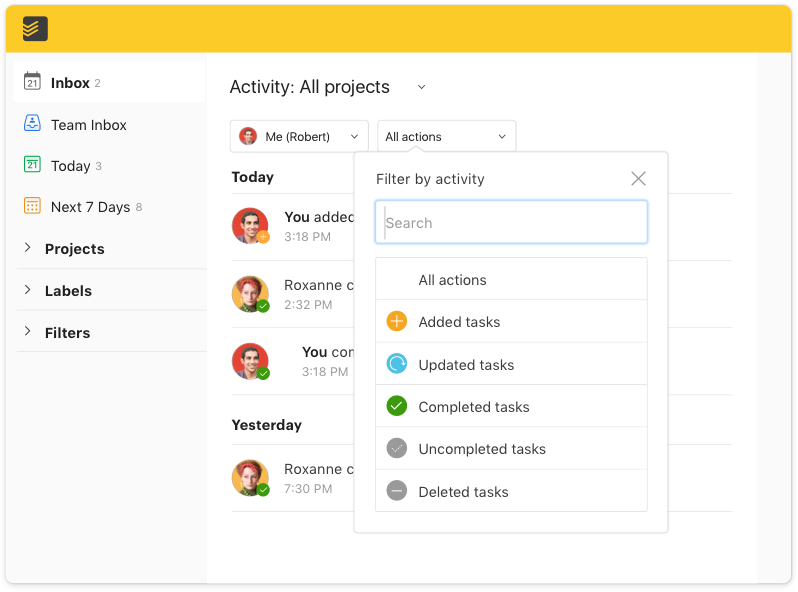


Рис. 1.11. Активності

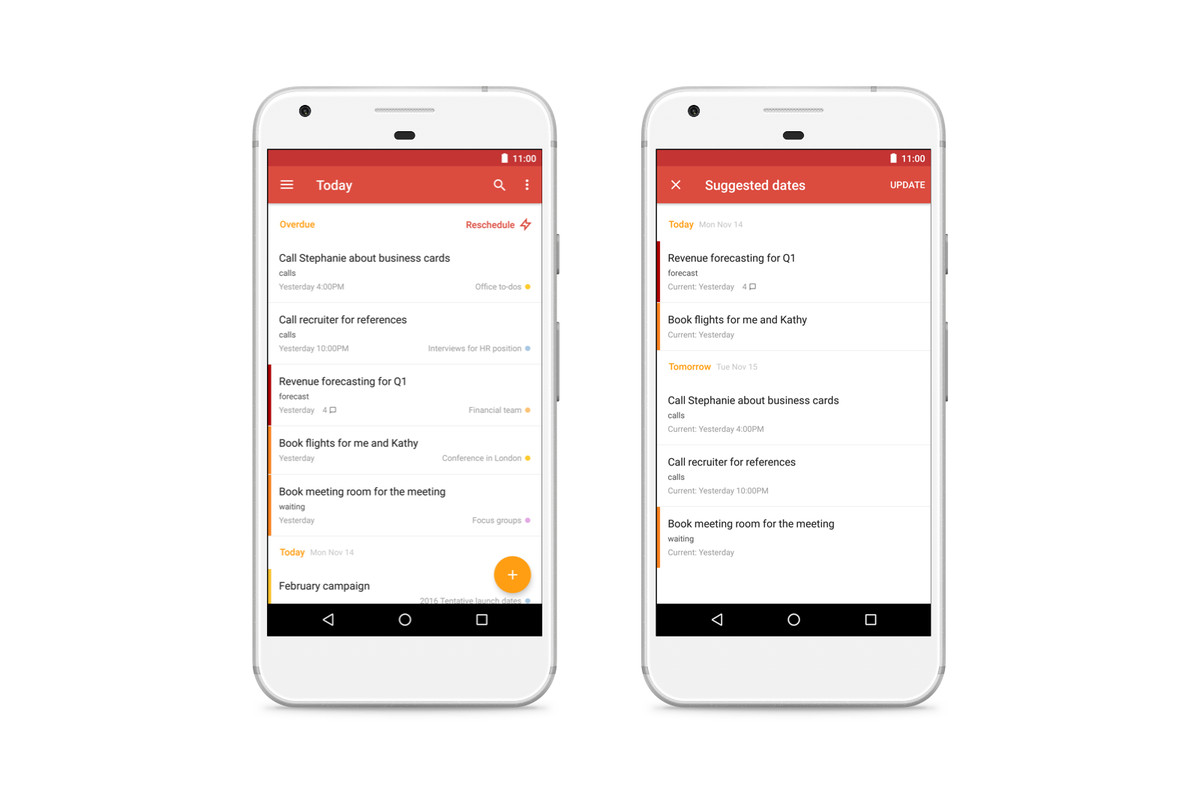


Рис. 1.12. Мобільна версія

## **1.2 Аналіз технічного завдання**

Спеціалізована програма розподілу навантаження між викладачами кафедри є унікальною. Ні один з вищевказаних подібних програмних рішень не підтримує потрібні функції. В програмному рішенні обов‘язоково повинні бути імплементовані наступні функції:

1) Форми для ручного заповнення

Форми для ручного заповнення даних потрібні для корегування або заповнення даних при не можливості імпорту з іншого джерела.

2) Кросплатформенність

Кросплатформенність потрібна для того щоб використовувати розроблене програмне рішення під керуванням будь якої операційної системи.

3) Зручний інтерфейс

Зручний інтерфейс потрібен для зручності використання продукту. Чим зручніший інтерфес, тим менше потрібно потрати часу на виконання завдання.

4) Валідація даних

Валідація даних потрібна для уникнення помилок повязаних з людським фактором.

1. Динамічна рекалькуляція розподілу навантаження між викладачами

Динамічна рекалькуляція розподулу навантаження між викладачами потрібна для зменшення часу виконання завдання а також для зменшення помилки в розрахунках. Попередньо всі дані перераховувалися в ручну.

1. Можливість експорту даних в csv, xlsx

Можливість експорту даних потрібна для подальшого їх використання деканатом, або іншими керуючими апаратами.

1. Проста СУБД

Проста СУБД потрібна для спрощення переносу даних.

## **1.3 Аналіз випадків використання**

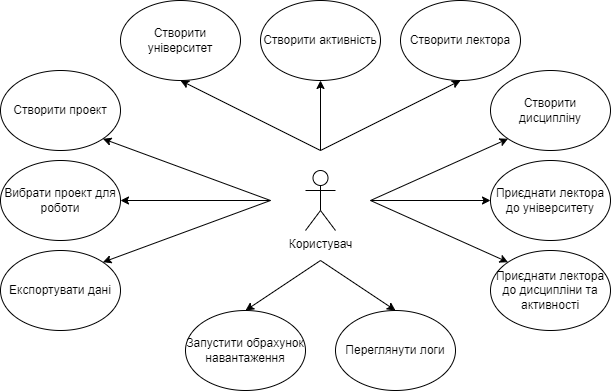
****

Рис. 1.13. Випадки використання

В користувача є 11 випадків використання програмного продукту:

1. Створити проект

Проект це абстактна сутність, до якої можуть прив‘язуватися інші сутності. Він слугує одиницею розмежування роботи. Різні проекти можуть містити свої унікальні дисципліни, викладачів тощо.

1. Вибрати проект для роботи

Після вибору проекту користувач може створювати інші сутності. Вони будуть використовуватись для збереження і обрахунку даних.

1. Створити університет

Сутність універститету допомагає досягнути універсальності. В майбутньому можна використовувати аплікацію для різних кафедр та університетів.

1. Створити активність

Під активністю мається на увазі «Лаборатна», «Курсова», «Димпломна» робота тощо. Теж допомагає досягти універсальності.

1. Створити лектора

Сутність лектора потрібна для співставленням з реальним лектором.

1. Створити дисципліну

Сутність дисципліни потрібна для співставлянням з реальною дисципліною. Вона повинна містити кількість студентів, груп тощо.

1. Приєднати лектора до університету

Сутність зв‘язку Лектор-Університет уникає дублювання записів в базі даних. Один лектор може викладати в декількох університетах або на різних кафедрах.

1. Приєднати лектора до активності і дисципліни.

Сутність зв‘язку Лектор-Активність-Дисципліна теж уникає дублювання записів в базі даних. Один лектор може проводити як лабораторні так і лекції в декількох дисциплінах.

1. Запустити обрахунок даних

Запуск обрахунку даних розподіляє навантаження між викладачами опираючись на попередньо додані дані.

10) Експортувати дані

Експорт даних це міграція даних з бази даних до xlsx або csv для подальшого використання деканатом або іншими керуючими органами.

1. Переглянути логи

Перегляд логів це системна функція. В логах буде зберігатися інформація про роботу програми або помилки, які виникли під час роботи. Опираючись на логи буде простіше пояснити розробнику про проблеми, які могли виникнути під час експлуатації програмного продукту. Таким чином розробник витратить менше часу на виправлення помилок в коді програми.

## **1.4 Аналіз алгоритму роботи користувача**



Рис. 1.14. Алгоритм роботи користувача

Всі блоки алгоритму були детально пояснені в попередньому підрозділі.

## **1.5 Аналіз архітектурного рішення**

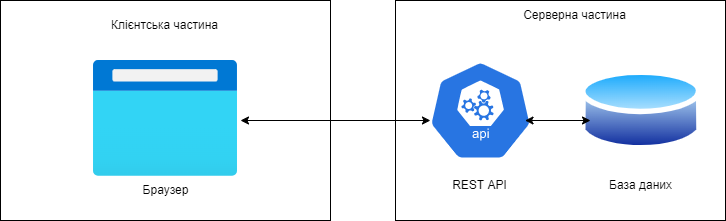


Рис. 1.15. Архітектура аплікації

Покладаючись на висунуті технічні вимоги я побудував артіхектуру, яка складається з двох частин. Перша – клієнтська частина, друга – серверна частина. В майбутньому серверну частину можна розгорнути на будь-якому сервері або в хмарі. Внаслідок цього користувач може працювати з аплікацією з будь-якого пристрою, знаходячись в будь-якій точці світу де є доступ до Інтернету. Клієнтська частина(Браузер) обмінюється даними з REST API. Який в свою чергу оперує даними з бази даних. В базі даних зберігаються потрібні користувачу дані.

# **ВИБІР ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ**

## **2.1 Перелік технологій та пояснення вибору**

Всі використані технолої були вибрані опираючись на досвід використання в попередніх комерційних проектах. Технології показали максимальну швидкодію, мінімальний час на розробку а також легкість використання.

Короткий перелік технологій та місця їх використання:

1. Клієнтська частина

* TypeScript – об‘єктно-орієнтована мова програмування, яка компілюється в JavaScript. Виконується в браузері та допомагає анімувати веб-сторінку.
* Angular – фреймворк, який спрощує роботу з TypeScript і зменшує час, який потрібен на розробку програмного рішення.
* HTML – декларативна мова для опису веб-сторінок. Відповідає за наповнення web-сторінки.
* SCSS – декларативна мова, яка компілюється в CSS. Відповідає за відображення web-сторінки.

1. Серверна частина

* C# - флагманська мова програмування компанії Microsoft. Відповідає за обробку даних та взаємодію з базою даних.
* ASP.NET Core – фреймворк, який спрощує роботу з C#. В ньому імплементований функціонал для роботи з HTTP.
* EntityFramework Core – фреймворк який спрощує взаємодію з базою даних. API між базою даних і C# кодом.
* SQLite – компактна СУБД. Використвоується для збереження даних.

## **2.2 Короткий опис використаних технологій**

### **2.2.1 TypeScript**

TypeScript [5] - це мова програмування, в якій виправлено багато недоліків JavaScript. Код на TypeScript виглядає майже так само, як і код на JS, і якщо у вас є досвід frontend-розробки, вивчити TypeScript досить просто. Особливо з огляду на те, що ви можете писати JS-код прямо в TS-скриптах.

Код на TypeScript компілюється в JS і підходить для розробки будь-яких проектів під будь-які браузери - тим більше, що можна вибрати версію JS, в яку компілюватиметься код.

TypeScript - проект з відкритим вихідним кодом, тому він швидко розвивається. Багато що з'являється в TS пізніше переходить і в JavaScript: наприклад, let і const, стрілочні функції і так далі.

**Переваги TypeScript:**

1. Строга типізація

Багато проблем у JavaScript виникають через динамічну типізацію і в цілому дивну поведінку типів даних. У TypeScript типізація статична, що позбавляє багатьох проблем. Є числовий тип, рядковий, логічний та інші.

1. Поліпшене ООП

І в JS, і в TS є підтримка об'єктно-орієнтованого програмування: класи, об'єкти, спадкування. Однак TypeScript зробив крок далі і використовує більше можливостей ООП. У тому числі, наприклад, інтерфейси. Інший великий плюс – модифікатори доступу. Їх у TypeScript три: public, private та protected.

### **1.2.2 Angular**

Angular [6] – це платформа розробки, побудована на TypeScript. Як платформа, Angular включає:

1. Компонентний фреймворк для створення масштабованих веб-додатків.
2. Колекція добре інтегрованих бібліотек, які охоплюють широкий спектр функцій, включаючи маршрутизацію, керуування формами, зв‘язок клієнт-сервер тощо.
3. Набір інструментів для розробників, які допоможуть вам розробляти, створювати, тестувати та оновлювати код.

З Angular розробник користується перевагами платформи, яка може масштабуватися від проектів окремого розробника до програм корпоративного рівня.

### **1.2.3 HTML**

HTML (Hypertext Markup Language) [7] – це код, який використовується для структурування та відображення веб-сторінки та її вмісту. Наприклад, вміст може бути структурованим всередині множини параграфів, маркованих списків або з використанням зображень та таблиць даних.

### **1.2.4 SCSS**

SCSS (Syntacically Awesome Style Sheet) [8] – надмножина CSS. SCSS є більш просунутою версією CSS. SCSS розроблений Хемптоном Кетліном, Крісом Еппштейном і Наталі Вайзенбаум.

Переваги SCSS над CSS:

1. SCSS містить усі функції CSS. А також ті, яких немає в CSS, що робить його хорошим вибором для розробників.
2. SCSS наповнений розшириними функціями.
3. SCSS пропонує змінні. Це велика перевага над звичайним CSS.
4. SCSS дозволяє використовувати вкладений синтаксис.

### **1.2.5 C#**

C# [9] - це строго типізована об‘єктно-орієнтована мова програмування.

C# - це мова програмування, розроблена та запущена компанією Microsoft у 2001 році. C# - проста, сучасна об‘єктно орієнтована мова програмування, яка надає сучасним розробникам гнучкість та можливості створення програмного забезпечення, яке не тільки працюватими сьогодні, але й буде застосовуватися протягом багатьох років.

**Основні переваги C#:**

1. C# сучасний та простий

C# — це проста, сучасна й об'єктно-орієнтована мова програмування. Метою C# була розробка мови програмування, яку не тільки легко вивчити, але й підтримувати сучасну функціональність для всіх видів розробки програмного забезпечення

1. C# є кросплатформним

C# — це кросплатформна мова програмування. Можна створювати програми .NET, які можна розгорнути на платформах Windows, Linux і Mac. Програми C# також можна розгортати в хмарі та в контейнерах.

### **1.2.5 ASP.NET Core**

ASP.NET Core [10] — це нова версія веб-фреймворку ASP.NET, яка в основному призначена для роботи на платформі .NET Core.

ASP.NET Core — безкоштовний, з відкритим вихідним кодом та кросплатформна платформа для створення хмарних додатків, таких як веб-програми, програми Інтернету речей і мобільні серверні програми. Він призначений для роботи як у хмарі, так і локально.

Так само, як і .NET Core, він був розроблений модульно з мінімальними накладними витратами, а потім інші більш розширені функції можуть бути додані як пакети NuGet відповідно до вимог програми. Це призводить до високої продуктивності, потребує менше пам’яті, меншого розміру розгортання та легкого обслуговування.

### **1.2.6 EntityFramework Core**

Entity Framework Core [11] — це нова версія Entity Framework після EF 6.x з відкритим вихідним кодом, легка, розширювана і кросплатформна версія технології доступу до даних Entity Framework.

Entity Framework — це структура об’єктного/реляційного відображення (O/RM). Це вдосконалення ADO.NET, яке надає розробникам автоматизований механізм доступу та зберігання даних у базі даних.

EF Core призначений для використання з додатками .NET Core. Однак його також можна використовувати зі стандартними додатками на основі .NET 4.5+.

### **1.2.7 SQLite**

SQLite [12] — це бібліотека, яка реалізує автономний, безсерверний, транзакційний механізм баз даних SQL з нульовою конфігурацією. Код для SQLite є загальнодоступним і, таким чином, безкоштовний для будь-яких цілей, комерційних чи приватних. SQLite — це найпоширеніша база даних у світі з більшою кількістю додатків, ніж ми можемо розрахувати, включаючи кілька високопрофільних проектів.

# **Висновок**

Під час виконання роботи було зроблено аналітичний огляд джерел інформації. Проаналізовано існуючі програмні рішення, технічне завдання, випадки використання, алгоритм роботи користувача а також архіректуру майбутньої аплікації. Вибрано технології для проектування програмного рішення а також наведено переваги цих технологій.

# **Список літературних джерел**

1. <https://www.software-developer-india.com/advantages-and-disadvantages-of-trello/>
2. <https://smartifysol.com/blog/testing/what-is-jira-advantages-and-disadvantages-of-jira-blog/>
3. <https://project-management.com/pros-and-cons-of-using-clickup/>
4. <https://comparecamp.com/todoist-review-pricing-pros-cons-features/>
5. <https://skillbox.ru/media/code/typescript_kak_s_nim_rabotat_i_chem_on_otlichaetsya_ot_javascript/>
6. <https://angular.io/guide/what-is-angular>
7. <https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/HTML_basics>
8. <https://www.geeksforgeeks.org/what-is-the-difference-between-css-and-scss/>
9. <https://www.c-sharpcorner.com/article/what-is-c-sharp/>

10) https://www.tutorialsteacher.com/core/aspnet-core-introduction

11) <https://www.entityframeworktutorial.net/efcore/entity-framework-core.aspx>

12) <https://www.sqlite.org/about.html>