

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Навчально-науковий інститут новітніх освітніх технологій
Кафедра інформаційно-обчислювальних систем і управління

ЛИМАР Вікторія Романівна

**Інформаційна панель згадок для відстеження та керування
комунікаціями програмного середовища Jira / Mentions dashboard
for tracking and managing Jira software communication**

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Кваліфікаційна робота

Виконала студентка групи КН-42
В. Р. Лимар

Науковий керівник
к.т.н., доцент І.В. Турченко

Кваліфікаційну роботу допущено до
захисту

«___» _____ 20___ р.

В.о. завідувача кафедри
_____ Н.М. Васильків

Тернопіль – 2025

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність: 122 «Комп'ютерні науки»
освітньо-професійна програма – «Комп'ютерні науки»

«ЗАТВЕРДЖУЮ»
В.о. завідувача кафедри
_____ Н.М. Васильків
«_____» _____ 2024р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ
ЛИМАР Вікторія Романівна
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Інформаційна панель згадок для відстеження та керування комунікаціями програмного середовища Jira / Mentions dashboard for tracking and managing Jira software communication

керівник Турченко І. В., к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 20 грудня 2024 р. №938.

2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:

– проаналізувати специфіку комунікацій у програмному середовищі Jira, зокрема особливості згадок і коментарів; виявити проблеми у відстеженні важливих згадок у великій кількості проєктної інформації;

– сформулювати концепцію інформаційної панелі, яка агрегує згадки з різних задач, коментарів і подій, групуючи їх за контекстом, авторством або датою;

– розробити архітектуру системи, яка забезпечує інтеграцію з Jira API, обробку отриманих даних, їх візуалізацію та взаємодію з користувачем через інтерактивний інтерфейс;

– реалізувати інформаційну панель на основі Atlassian Forge, що забезпечує контекстне виведення згадок безпосередньо в інтерфейсі Jira.

5. Перелік графічного матеріалу у роботі:

– структура функціональних модулів інформаційної панелі для управління згадуваннями;

– діаграма послідовностей;

– схема алгоритму роботи інформаційної панелі для моніторингу згадувань;

– алгоритм збору та фільтрації даних;

– макет інтерфейсу інформаційної панелі згадок у середовищі Jira.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 20 грудня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи	до 01.01. 2025 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02. 2025 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 01.04.2025 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2025 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2025 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2025 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2025 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2025 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06.2025 р.	

Студент _____ Лимар В.Р.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Турченко І.В.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Інформаційна панель згадок для відстеження та керування комунікаціями програмного середовища Jira» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 58 сторінок і містить 22 ілюстрацій, 1 таблицю, 3 додатки та 28 використаних джерел.

Метою кваліфікаційної роботи є створення інформаційної панелі згадок для відстеження та керування комунікаціями програмного середовища Jira.

Об'єктом дослідження є процеси організації та оптимізації комунікації в середовищі командної розробки, зокрема механізми згадок користувачів, взаємодії із задачами, коментарями та подіями у системі Jira.

Розроблено інформаційну панель згадок у середовищі Jira, що реалізована на платформі Atlassian Forge з використанням React та Atlassian Design System.

Розроблена панель дозволяє відображати отримані та відправлені згадування користувачів, інтегрується з коментарями та додатком контрольних списків, підтримує персоналізовані фільтри та механізми налаштування, забезпечує оперативний моніторинг взаємодій між учасниками проєкту, дозволяючи швидко реагувати на важливі коментарі та завдання. Вона підвищує ефективність комунікацій і сприяє кращому управлінню командною співпрацею.

Ключові слова: ІНФОРМАЦІЙНА ПАНЕЛЬ, Jira, ЗГАДУВАННЯ, КОНТРОЛЬНІ СПИСКИ, КОМУНІКАЦІЇ.

ANNOTATION

Qualification work on the topic «Mentions dashboard for tracking and managing Jira software communication» for obtaining a bachelor's degree in the specialty 122 «Computer Science» of the educational program «Computer Science» is written on 58 pages and it contains 22 figures, 1 table, 3 annexes and 28 references.

The purpose of the qualification work is to create a mentions dashboard for tracking and managing communication in the Jira software environment.

The object of research is the processes of organizing and optimizing communication in a team development environment, particularly the mechanisms of user mentions, interaction with tasks, comments, and events within the Jira system.

A mentions dashboard has been developed in the Jira environment, implemented on the Atlassian Forge platform using React and the Atlassian Design System.

The dashboard enables the display of received and sent mentions, integrates with comments and checklist functionality, supports personalized filters and configuration mechanisms, and ensures real-time monitoring of user interactions. This allows teams to promptly respond to important comments and tasks, improving communication efficiency and supporting better management of team collaboration.

Keywords: MENTIONS DASHBOARD, Jira, MENTIONS, CHECKLISTS, COMMUNICATIONS.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної області та постановка задачі проектування.....	10
1.1 Опис структури та керування комунікаціями програмного середовища Jira.....	10
1.2 Огляд і аналіз літературних джерел та існуючих рішень.....	13
1.3 Постановка задачі проектування.....	20
2 Структура та алгоритмічне забезпечення інформаційної панелі згадок для відстеження та керування комунікаціями програмного середовища Jira.....	22
2.1 Загальна структура проєктованої інформаційної панелі згадок.....	22
2.2 Алгоритмічне забезпечення.....	26
2.3 Партнерська інтеграція.....	31
3 Реалізація інформаційної панелі згадок для відстеження та керування комунікаціями програмного середовища Jira.....	33
3.1 Програмна реалізація.....	33
3.2 Інтерфейс користувача.....	37
3.3 Тестування розробленої програми.....	42
Висновки.....	44
Список використаних джерел.....	46
Додаток А Програмний код основного функціоналу інформаційної панелі.....	49
Додаток Б Довідка про використання.....	52
Додаток В Копія опублікованих результатів.....	53

ВСТУП

У сучасному бізнес-середовищі, де компанії стикаються з постійним зростанням обсягів роботи та необхідністю ефективного управління процесами, команди шукають оптимальні інструменти для організації завдань, відстеження виконання робіт та налагодження ефективної комунікації [1].

Сучасні програмні середовища управління проектами стали невід'ємною частиною ефективної командної взаємодії в галузі інформаційних технологій. Ці платформи не лише підтримують процеси планування, відстеження та контролю виконання завдань, але й формують основу для комунікацій між учасниками проектів [2]. З огляду на зростання обсягів інформації, масштабування проектів і збільшення кількості залучених осіб, особливої ваги набуває потреба у систематизованому підході до управління комунікаціями між користувачами, командою та організаціями [2].

Зважаючи на ці виклики, багато організацій обирають Jira – це потужна платформа для управління проектами та командною взаємодією, створена компанією Atlassian. Завдяки гнучкості у налаштуванні та масштабуванні, Jira стала одним із найпопулярніших рішень для організації командної роботи в сфері інформаційних технологій, зокрема у розробці програмного забезпечення, а також у сферах маркетингу, аналітики, управління ризиками тощо.

Однією з переваг Jira є можливість згадування інших користувачів за допомогою символу "@", що дозволяє оперативно привертати увагу до важливої інформації або запитань. Втім, на практиці така функціональність, попри її корисність, має обмеження: згадування розпорошуються серед численних задач і коментарів, що ускладнює їх відстеження, особливо в умовах великих організацій.

Вбудовані механізми Jira не надають можливості централізованого моніторингу згадувань, а існуючі сторонні рішення, як було виявлено в процесі аналізу, не повністю задовольняють потреби користувачів – вони не підтримують фільтрацію, швидкий пошук або контекстне реагування на згадування [3].

Відповідно, постає задача реалізації інформаційної панелі згадок як інструмента для відстеження, обробки та керування згадуваннями користувачів у середовищі Jira. Така панель повинна вирішити низку проблем: забезпечити централізований перегляд згадувань, дозволити фільтрацію за авторами, надати механізм швидкої відповіді без перемикання між задачами, а також інтегруватися у стандартний інтерфейс Jira без порушення звичного робочого середовища.

Метою кваліфікаційної роботи є створення інформаційної панелі згадок для відстеження та керування комунікаціями програмного середовища Jira.

Для досягнення мети необхідно виконати наступні завдання:

- провести аналіз предметної області;
- проаналізувати існуючі аналоги;
- зробити постановку задачі проектування;
- розробити структуру інформаційної панелі згадок для відстеження та керування комунікаціями;
- розробити алгоритмічне забезпечення інформаційної панелі згадок для відстеження та керування комунікаціями;
- дослідити технічні особливості роботи з користувацькими полями та спроектувати партнерську інтеграцію для розширення функціональності збору згадувань;
- розробити користувацький інтерфейс інформаційної панелі;
- реалізувати та протестувати інформаційної панелі згадок для відстеження та керування комунікаціями.

Робота складається з трьох основних розділів. У першому розділі проведено аналіз особливостей комунікаційної структури Jira, існуючих рішень для управління згадуваннями та обґрунтовано потребу у створенні нового інструмента. У другому – подано алгоритмічне забезпечення: архітектура системи, діаграми послідовності, алгоритми збору, фільтрації даних, швидкої відповіді, налаштування панелі. Третій розділ присвячено реалізації програмного забезпечення: представлено структуру застосунку, користувацький інтерфейс та результати тестування.

Результати виконаної роботи матимуть практичну цінність, оскільки розроблювальний інструмент може підвищити ефективність командної взаємодії, дозволяє швидко виявляти та реагувати на важливі комунікації в Jira, зменшуючи ризик пропущених повідомлень. Це буде корисно керівникам проєктів, розробникам, тестувальникам, аналітикам, а також будь-яким учасникам команд, які працюють із Jira.

Сфери застосування:

- управління проєктами допомагає планувати завдання, відстежувати терміни виконання, розподіл ресурсів та контроль прогресу реалізації проєктів різної складності;
- розробка програмного забезпечення веде технічні завдання, відстеження помилок, управління випусками версій та координація роботи команд розробників;
- IT-підтримка та DevOps (development і operations), включає обробку запитів користувачів, управління інцидентами, автоматизацію процесів розгортання та моніторинг системної інфраструктури;
- бізнес-аналітика охоплює збір вимог до продукту, аналіз бізнес-процесів, створення звітності та відстеження ключових показників ефективності організації.

Результати дослідження опубліковано в матеріалах науково-практичної конференції «Інтелектуальні інформаційні технології в прикладних дослідженнях» (ІТАР – 2025), м. Тернопіль, 27–29 травня 2025 р. (додаток В).

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ПРОЕКТУВАННЯ

1.1 Опис структури та керування комунікаціями програмного середовища Jira

Як було вище згадано, Jira – це головна система для всього життєвого циклу розробки, що надає автономним командам контекст для швидкого просування, залишаючись при цьому націленими на досягнення більшої бізнес-цілі [4].

Jira як система управління проектами складається з кількох основних структурних компонентів, які разом утворюють цілісну екосистему для ефективної взаємодії команд. Використання Jira неймовірно широке, але ось деякі з її основних застосувань [5]:

- гнучкі методології - ідеально підходить для команд, які практикують Scrum або Kanban, Jira надає вбудовані шаблони проектів та дошки для покращення робочих процесів та ефективності;

- управління проектами - менеджери проектів можуть відстежувати роботу за допомогою завдань Jira, кожне з яких містить ключові деталі і виконується за визначеним робочим циклом, який представляє ваш процес;

- відстеження помилок та управління завданнями.

Перш за все, робота з Jira починається зі створення проектів - контейнерів для організації роботи над певним продуктом, сервісом чи ініціативою. Найчастіше проекти можна розглянути як набір завдань, які необхідно виконати для досягнення певного результату [6].

Структура проектів складається з низки елементів, нижче наведено основні:

- ключ проекту або унікальний ідентифікатор (наприклад, "PROJ"), який використовується для створення ідентифікаторів задач;

- тип проекту, що визначає шаблон проекту, наприклад, розробка програмного забезпечення, бізнес-проект, служба підтримки тощо;

- схема робочого процесу, що дозволяє визначати стадії, через які проходять задачі, наприклад, "Зробити", "Взято в роботу", "Виконано" тощо;

— дозволи встановлюють, які користувачі мають доступ до проекту та які дії вони можуть виконувати.

Наступною основною складовою роботи з Jira є дошки – це візуальне представлення задач, що дозволяє команді бачити поточний стан роботи, слідкувати за прогресом та ефективно управляти робочим процесом. Дошки складаються з певних колонок (рисунок 1.1), які відображають статуси задач у робочому процесі, а також карток задач, тобто візуальне представлення окремих задач з ключовою інформацією.

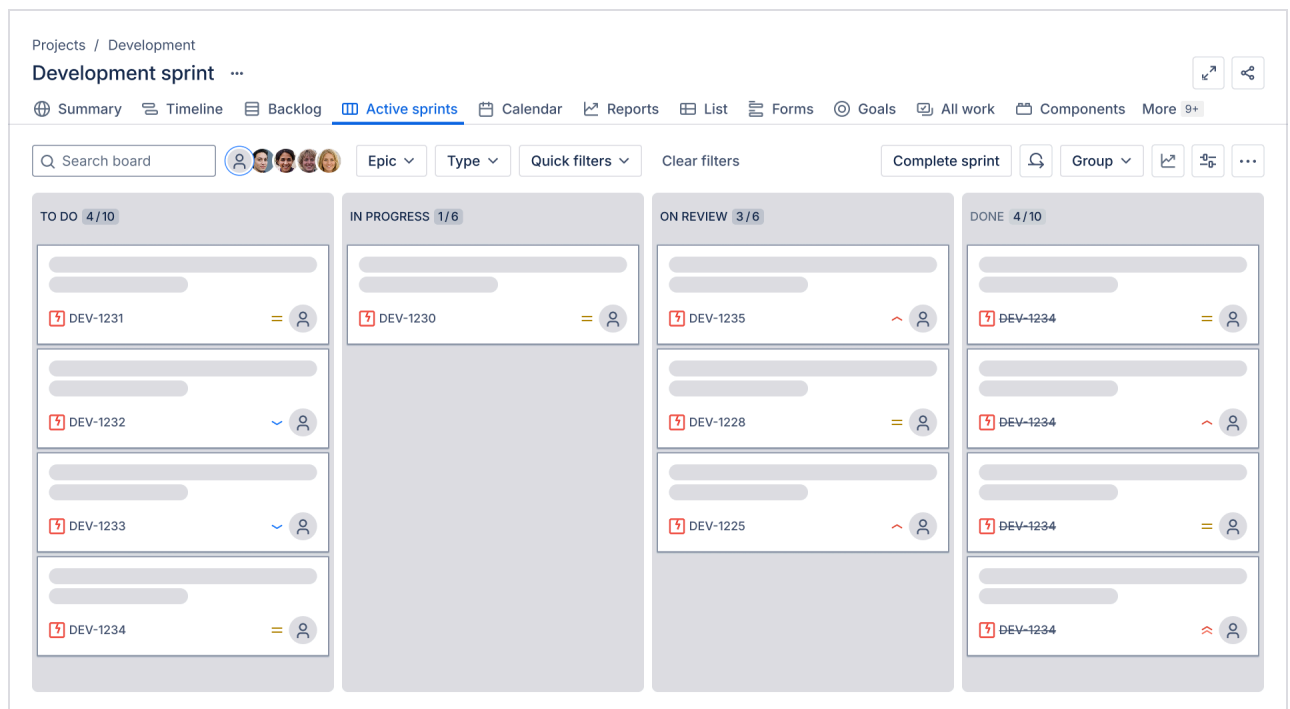


Рисунок 1.1 – Інтерфейс дошки в Jira

Jira пропонує дві дошки з унікальними функціями, які дозволяють командам легко почати практикувати гнучкі методології та відстежувати свою роботу [7]. Ці дошки попередньо налаштовані і готові до використання з шаблонами Jira [7].

Для легшого відслідковування процесів, керування задачами, моніторингу активності команд тощо, в Jira існують програми або додатки, які можуть бути розроблені різними організаціями та безпосередньо Jira. Ці розширення збільшують функціональність базової платформи. Вони доступні через Atlassian Marketplace і можуть значно розширити можливості системи. Окрім програм

також є інформаційні панелі (dashboards) – це панелі, які відображають важливу інформацію про проєкти, задачі та ключові показники ефективності в зручному для сприйняття форматі.

В Jira інформаційна панель – це набір "віджетів" або попередньо встановлених дисплеїв даних, які показують різні типи інформації в реальному часі, щоб допомогти відстежувати і контролювати роботу над проєктами. Інформаційні панелі можуть бути приватними або доступними для інших користувачів. Вони також можуть містити інформацію з одного або кількох проєктів, що дає доступ до відповідної інформації з усього екземпляра [8].

Найважливішою одиницею роботи в Jira є задачі, або робочі елементи (рисунок 1.2). Вони представляють конкретні завдання, які потрібно виконати в рамках проєкту. Робочі елементи можна оновлювати, тобто показати прогрес роботи або надати відповідну інформацію, наприклад, хто призначений для даного завдання, детальний опис, вкладення тощо.

Jira має різні типи робочих елементів, які можуть допомогти визначити, класифікувати та розрізняти різні типи робіт [9].

The image shows a Jira task card. At the top, it has a breadcrumb trail: 'Projects / Development / DEV-1238'. The title of the task is 'Feature deployment causing unexpected errors'. Below the title is a 'Description' section with the text: 'A newly implemented feature has caused unexpected errors, disrupting core functionality. Conduct an analysis, apply fixes, and ensure stable deployment.' There is a 'Comments' section with two entries. The first comment is from Anna Jonson, dated January 4, 2025 at 10:53 AM, mentioning '@Ronald Richards, fix has been applied and preliminary testing shows improvements.' The second comment is from Ronald Richards, dated December 28, 2024 at 9:13 PM, mentioning '@Anna Jonson, proceeding with extensive validation to confirm full resolution before deployment.' On the right side of the card, there is a 'To Do' dropdown menu and a 'Details' section. The 'Details' section lists: Assignee (Ronald Richards), Reporter (Anna Jonson), Priority (High), Sprint (DEV Sprint 13), and Due date (May 30, 2025).

Projects / Development / DEV-1238

Feature deployment causing unexpected errors

Description

A newly implemented feature has caused unexpected errors, disrupting core functionality. Conduct an analysis, apply fixes, and ensure stable deployment.

Comments

Anna Jonson January 4, 2025 at 10:53 AM
@Ronald Richards, fix has been applied and preliminary testing shows improvements.

Ronald Richards December 28, 2024 at 9:13 PM
@Anna Jonson, proceeding with extensive validation to confirm full resolution before deployment.

To Do ▾

Details

Assignee Ronald Richards

Reporter Anna Jonson

Priority High

Sprint DEV Sprint 13

Due date May 30, 2025

Рисунок 1.2 – Інтерфейс робочого елементу

Одним із важливих аспектів комунікації в Jira є можливість згадувати (тегати) інших користувачів в задачах, використовуючи символ “@” з подальшим вибором імені користувача зі списку. Ця функція значно покращує взаємодію всередині команди та забезпечує своєчасне інформування зацікавлених осіб про зміни чи питання, що потребують їхньої уваги. Згаданий користувач може отримати як повідомлення через сповіщення в інтерфейсі Jira, так і листа на електронну пошту.

Однак, у реальних умовах використання Jira виникає проблема зручного перегляду списку згадувань і вона не є локальною або поодинокую. Це системне обмеження, на яке регулярно звертають увагу користувачі на Atlassian Community [10]. Пошук за ключовими словами “mentions list” повертає десятки обговорень, де користувачі скаржаться на відсутність централізованого перегляду, обмежену фільтрацію та неможливість зручно відстежувати, коли і де їх згадали. Ці звернення свідчать про наявність реального запиту на покращення механізмів моніторингу згадувань, що додатково обґрунтовує доцільність розробки спеціалізованої інформаційної панелі, особливо це стосується великих організацій з багатьма проектами та задачами.

1.2 Огляд і аналіз літературних джерел та існуючих рішень

На початковому етапі було проведено аналіз наукових статей з баз даних Scopus, Web of Science та інших джерел, щоб глибше дослідити проблему відстеження та управління комунікаціями у сфері інформаційних технологій. Для прикладу, автори [11] висвітлюють практичний підхід до використання інструментів Jira, детально аналізуючи їхню роль у командній співпраці, зокрема функції комунікації між учасниками, відстеження подій та автоматизацію процесів.

Крім того, у роботі [12] розглядаються особливості комунікації в розробницьких командах, зокрема використання згадок, тегів, коментарів та методи оптимізації інформаційного навантаження. Значну увагу також було

приділено питанням побудови зручних інформаційних панелей, які забезпечують ефективне представлення великих обсягів даних та відповідають ключовим принципам UX/UI-дизайну для аналітичних інтерфейсів [13].

Було проаналізовано вплив сучасних комунікаційних інструментів, таких як Slack та Jira, на загальну ефективність командної взаємодії в контексті Agile-розробки [14]. Дослідження демонструє значення реорганізації Scrum-дошок та інтеграції Jira з платформою Slack для покращення комунікаційних процесів і підвищення продуктивності співпраці.

Проаналізувавши всі вищезгадані статті, було здійснено дослідження безпосередньо самої системи сповіщень Jira, яка в свою чергу є важливою складовою, оскільки забезпечує інформування користувачів про різноманітні події та зміни в проєктах та робочих елементах [15]. Основною метою цієї системи є підтримка комунікації між членами команди та забезпечення своєчасного реагування на зміни, що відбуваються в робочому процесі [16].

Базовий функціонал системи сповіщень Jira включає панель сповіщень – розташована у верхній частині інтерфейсу Jira, представлена іконкою дзвіночка. При натисканні на неї відкривається випадаючий список з останніми сповіщеннями (рисунком 1.3). Паралельно з внутрішніми сповіщеннями, Jira також надсилає повідомлення на електронну пошту користувачів відповідно до їхніх налаштувань.

Аналіз існуючої системи сповіщень Jira, зокрема в контексті роботи зі згадуваннями користувачів, виявляє ряд переваг та суттєвих недоліків, які впливають на ефективність комунікації в проєктних командах. Перевагами даної системи є те, що сповіщення вбудовані в основний інтерфейс Jira, що забезпечує єдину точку доступу до всіх функцій платформи [17].

Крім того, дублювання сповіщень через електронну пошту дозволяє користувачам отримувати інформацію навіть за відсутності доступу до веб-інтерфейсу Jira. Є можливість швидкого переходу безпосередньо до відповідної задачі або коментаря з інтерфейсу сповіщень.

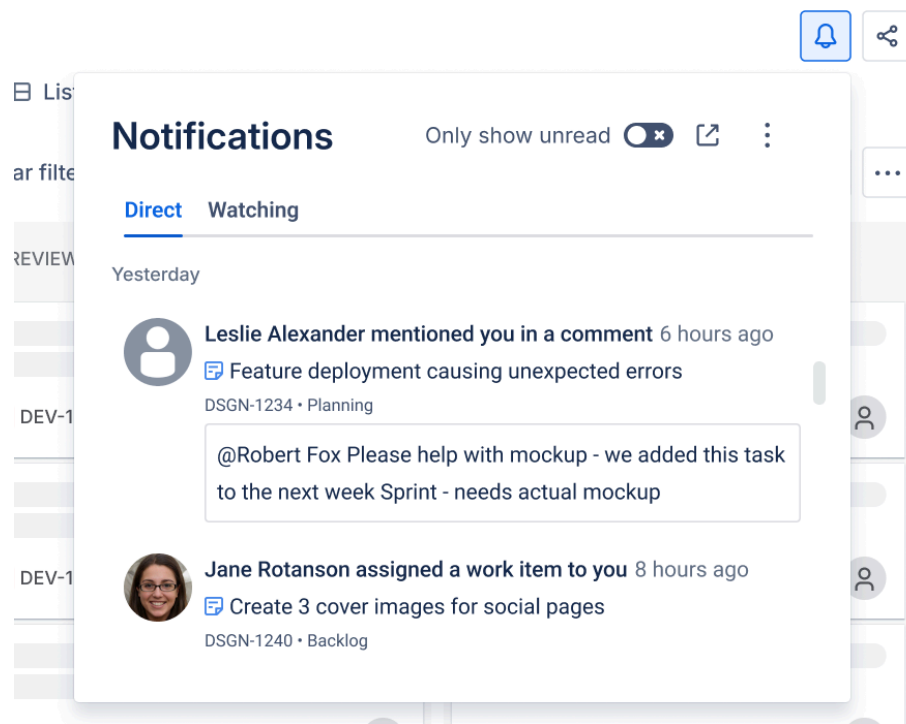


Рисунок 1.3 – Інтерфейс випадаючого вікна сповіщень Jira

Незважаючи на ряд переваг, особливої уваги заслуговує проблема структурованості інтерфейсу сповіщень в Jira. Окрім згадувань, інтерфейс містить усі типи оновлень, пов'язаних з роботою в системі, що призводить до ряду негативних наслідків:

- користувачі Jira скаржаться на неможливість точної фільтрації згадувань. Багато хто вдавався до обхідних шляхів, таких як використання введення запитів JQL (Jira Query Language) - це розширений пошук, який дозволяє створювати структуровані запити для пошуку робочих елементів у проектах і між ними [18], для фільтрації завдань, де вони були згадані хоча б один раз, але цей метод не підтримує фільтрацію коментарів за згадуваннями безпосередньо. У результаті користувачі повинні просіювати нерелевантні завдання, щоб знайти згадки;

- більшість покладаються на сповіщення електронною поштою для відстеження своїх згадувань. Однак скринька пошти може бути надто перевантаженою, а важливі коментарі часто губляться серед потоку менш важливих сповіщень. У великих організаціях це може призвести до пропущених

дедлайнів, затримки відповідей та відсутності відповідальності за важливі завдання;

– на даний момент у Jira немає спеціалізованої інформаційної панелі або інтерфейсу для перегляду всіх згадувань в одному місці. Користувачі повинні вручну прокручувати завдання і сповіщення, часто пропускаючи важливі оновлення в процесі. Без централізованого способу відстежувати згадки в проєктах, користувачам часто стає викликом підтримувати структуровані комунікації.

На ринку представлено декілька рішень, які намагаються вирішити проблему управління згадуваннями в Jira. Нижче розглянуто основні з них, проаналізувавши їхні переваги та недоліки.

User Mentions Gadget for Jira є одним з основних прямих конкурентів у сфері управління згадуваннями (рисунок 1.4).

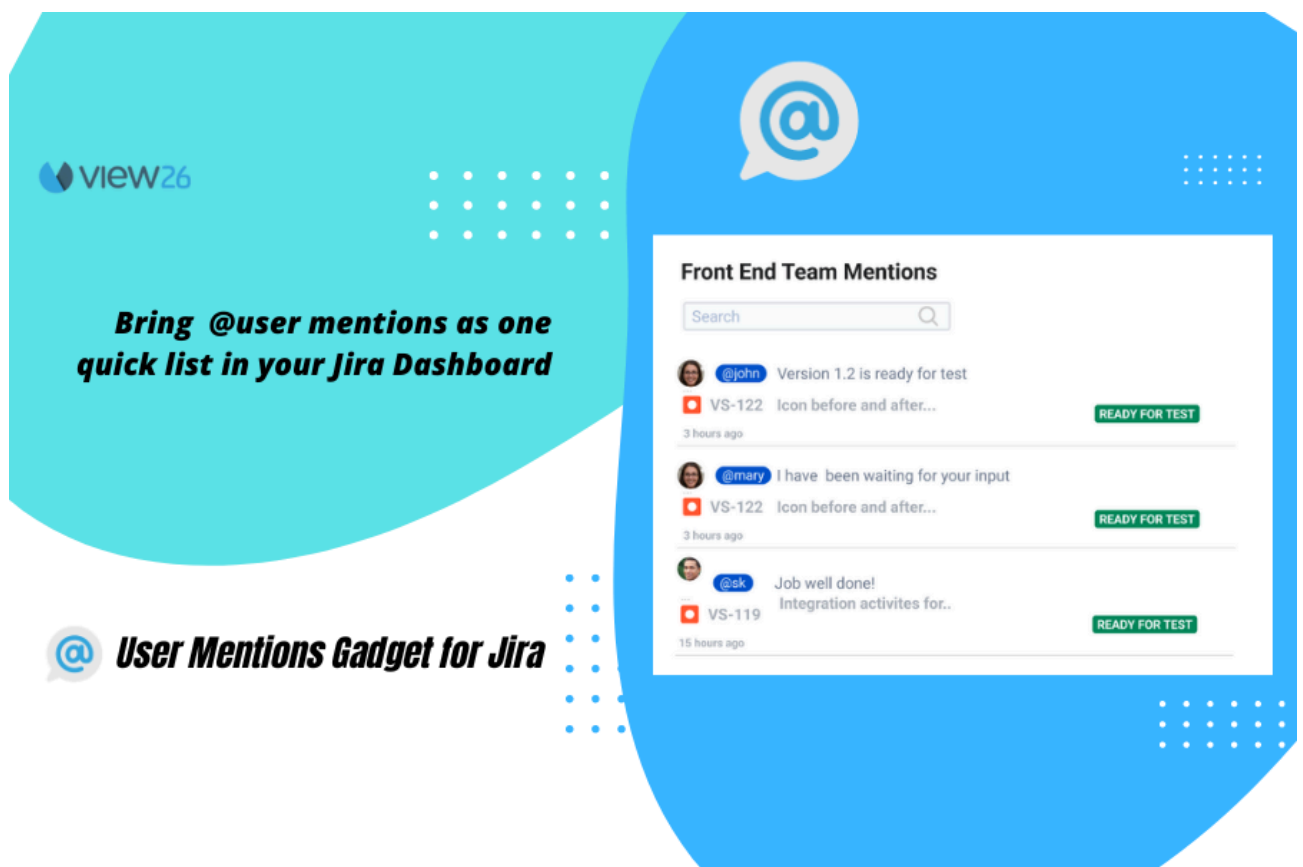


Рисунок 1.4 – Інтерфейс додатку “User Mentions Gadget for Jira” [19]

Цей продукт представляє собою панель Jira, яка спеціалізується на відстеженні згадувань і тегів користувачів в Jira, як надісланих, так і отриманих. Він об'єднує згадки з різних проєктів і завдань в єдиний список, дозволяючи користувачам відстежувати свої згадування і залишатися в курсі певних повідомлень.

Функціональними можливостями цього продукту є:

- відображення згадувань поточного користувача в системі Jira;
- можливість переглядати згадування обраного користувача;
- відображення контексту згадування (задача, коментар);
- підтримка переходу до оригінального коментаря.

Перевагами цього рішення може слугувати забезпечення легкої навігації по сторінках з великою кількістю згадувань, вбудований прапорець для вибору для вибору користувача та пошукове поле, яке розташоване на панелі.

Недоліками рішення є відсутність можливості перегляду надісланих згадувань, тобто користувач не може відстежувати власні згадування інших учасників, неможливість відповіді на коментарі з панелі, обмежена інформація про автора згадування та контекст, а також обмежені можливості фільтрації та сортування.

Наступним існуючим рішенням є Threaded Comments for Jira (рисунк 1.5), що належить до категорії непрямих конкурентів, оскільки його основний фокус – покращення структури комунікацій через організацію коментарів у ланцюжки обговорень, а не безпосереднє управління згадуваннями.

Перевагами даного рішення є чітка візуалізація зв'язків між коментарями, можливість згортання неактуальних гілок обговорення, інтуїтивно зрозумілий інтерфейс та збереження історії обговорень [20].

Однак тут недостатньо певного функціоналу, наприклад, для перегляду ланцюжків коментарів потрібно щоразу активувати додаток. Цей продукт тримає фокус на структуризації коментарів, а не на управлінні згадуваннями, що не вирішує початкову проблему користувачів про відстеження згадувань.



Рисунок 1.5 – Інтерфейс додатку “Threaded Comments for Jira” [21]

Для систематизації результатів аналізу існуючих рішень доцільно порівняти їх за ключовими параметрами, що визначають ефективність управління згадуваннями в Jira (таблиця 1.1).

Таблиця 1.1 – Порівняльна таблиця існуючих рішень

Параметр	Сповіщення Jira	User Mentions Gadget for Jira	Threaded Comments for Jira
Інтеграція з Jira	Вбудовані в основний інтерфейс, єдина точка доступу	Панель Jira, спеціалізована на згадуваннях	Додаток для структуризації коментарів
Фільтрація згадувань	Неможлива точна фільтрація згадувань через JQL	Обмежені можливості фільтрації та сортування	Не фокусується на управлінні згадуваннями
Відображення згадувань	Змішано з усіма типами оновлень	Відображення згадувань поточний/обраний користувач	Не спеціалізується на згадуваннях

Продовження таблиці 1.1

Параметр	Сповіщення Jira	User Mentions Gadget for Jira	Threaded Comments for Jira
Навігація	Швидкий перехід до задачі і/або коментаря	Легка навігація по сторінках, підтримка перехід до коментаря	Чітка візуалізація зв'язків між коментарями
Інтерфейс користувача	Неструктурований і перевантажений	Вбудований контрольний елемент для вибору користувача	Інтуїтивно зрозумілий інтерфейс
Додаткова функціональність	-	Обмежена інформація про автора та контекст	Згортання неактуальних гілок обговорення
Основні переваги	Єдина точка доступу, швидкий перехід, email-дублювання	Спеціалізація на згадуваннях, зручна навігація	Структуризація коментарів, візуалізація зв'язків
Основні недоліки	Відсутність централізованого перегляду, перевантаженість email, неможливість точної фільтрації	Неможливість відповіді з панелі, відсутність перегляду надісланих згадувань	Потребує активації для перегляду, не вирішує проблему відстеження згадувань

Аналіз порівняльної таблиці демонструє, що жодне з існуючих рішень не забезпечує оптимального балансу між спеціалізацією на управлінні згадуваннями, функціональною повнотою та зручністю використання.

Тому існує чітка потреба в розробці інструменту для управління згадуваннями в Jira, який би забезпечував відстеження згадувань, фільтрацію та пріоритезацію згадок, дозволяючи командам швидко знаходити важливу інформацію, уникати інформаційного шуму та оптимізувати робочі процеси. А

також пропонував зручний інтерфейс для швидкої відповіді на згадування та забезпечував якісну інтеграцію з існуючими робочими процесами в Jira.

1.3 Постановка задачі проєктування

На основі проведеного аналізу структури та функціональності системи Jira, а також специфіки її використання в різноманітних організаціях, було виявлено ряд суттєвих проблем, пов'язаних із моніторингом згадувань користувачів. Ці проблеми набувають особливої актуальності в умовах зростання масштабів проєктів, збільшення кількості користувачів системи та інтенсифікації комунікаційних процесів у межах робочих груп.

Аналіз існуючих підходів показав відсутність комплексного рішення, яке б ефективно адресувало всі виявлені проблеми моніторингу згадувань у Jira. Жоден з них не забезпечує одночасно централізацію, гнучку фільтрацію, інтеграцію з робочими процесами та масштабованість рішення. Враховуючи виявлені проблеми та обмеження інших продуктів, основним завданням дослідження є розробка спеціалізованого рішення в Jira.

Mentions Dashboard for Jira представляє собою інформаційну панель, інтегровану в стандартний інтерфейс Jira, що надає можливість централізовано відображати, сортувати та керувати згадуваннями користувачів, забезпечуючи зручну навігацію та ефективне управління інформацією. Рішення реалізується як набір взаємопов'язаних гаджетів, що можуть бути розміщені на персоналізованій або спільній інформаційних панелях користувача або команд, відповідно.

Формат у вигляді інформаційної панелі має ряд переваг порівняно з реалізацією окремої програми або додатку, тому що створення панелей дозволяє забезпечити оптимальну комбінацію різних представлень даних на одному екрані. Наприклад, керівник хоче мати можливість контролювати всі згадування, що містять посилання на конкретні команди. Для цього система організовує дані у вигляді окремих панелей: одна відображає згадування команди маркетологів, інша

– тестувальників, ще одна – розробників тощо. Це забезпечує зручну навігацію та ефективне управління інформацією.

Крім того, користувачі також можуть одночасно бачити не лише список останніх згадувань по командах, а й додатково, наприклад, діаграми, що показують тенденцію виконаних задач або перелік налаштованих інших фільтрацій або додатків. Така консолідація інформації значно підвищує ефективність роботи порівняно з переходами між різними екранами окремої програми.

Завдяки панелі візуалізації згадувань дозволить об'єднати всі отримані та відправлені згадування в єдиному просторі, забезпечуючи зручний моніторинг та фільтрацію за командою або департаментом. Це значно спрощує процес моніторингу та реагування на згадування, зменшує когнітивне навантаження на користувачів та підвищує загальну ефективність комунікації в команді. Реалізацію цього продукту дозволить заощадити значний час, який раніше витрачався на ручний пошук, та гарантуватиме, що жодне важливе згадування не буде пропущене через обмеження стандартних механізмів фільтрації.

Отже, основна ідея полягає у створенні єдиного центру моніторингу всіх комунікацій, пов'язаних зі згадуваннями, безпосередньо в середовищі Jira, без необхідності переходу до сторонніх систем або використання електронної пошти. Це досягається шляхом агрегації даних про згадування з різних проєктів та задач, їх структурування та представлення у зручному для аналізу та взаємодії форматі.

Проведений аналіз підтверджує актуальність розробки рішення для управління згадуваннями в Jira, яке враховуватиме обмеження існуючих продуктів та забезпечить користувачів необхідним функціоналом для ефективної комунікації в межах платформи.

2 СТРУКТУРА ТА АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ ПАНЕЛІ ЗГАДОК ДЛЯ ВІДСТЕЖЕННЯ ТА КЕРУВАННЯ КОМУНІКАЦІЯМИ ПРОГРАМНОГО СЕРЕДОВИЩА JIRA

2.1 Загальна структура проєктованої інформаційної панелі згадок

Інформаційна панель для відслідковування та керування комунікаціями програмного середовища Jira “Mentions Dashboard for Jira” призначена для розширення функціональних можливостей системи Jira в контексті відстеження та управління користувацькими згадуваннями. Головним завданням панелі є забезпечення централізованого доступу до всіх згадок користувачів, що дозволяє значно підвищити ефективність командної комунікації та уникнути втрати важливої інформації.

Для реалізації обрано об'єктно-орієнтований підхід, який передбачає представлення системи як сукупності взаємопов'язаних класів та об'єктів. Такий підхід найкраще відповідає архітектурі Jira та дозволяє створити гнучку, масштабовану систему з можливістю подальшого розширення функціональності.

Інформаційна панель складається з декількох функціональних модулів, що забезпечують ефективне управління згадуваннями (рисунк 2.1):

– модуль налаштування:

- 1) вибір джерела згадувань (коментарі Jira, контрольні списки);
- 2) налаштування фільтрації за автором згадування;
- 3) вибір користувачів, чий згадування будуть відображатися;
- 4) збереження налаштованих параметрів;

– модуль обробки даних:

- 1) збір даних про згадування з Jira;
- 2) фільтрація згадувань відповідно до налаштувань користувача;
- 3) обробка метаданих (автор, час створення, контекст);
- 4) формування структурованого набору даних для відображення;

– модуль візуалізації:

- 1) відображення загальної інформації про налаштування (згадані користувачі, джерело згадувань);
- 2) представлення списку згадувань з усією необхідною інформацією;
- 3) швидка відповідь на коментарі безпосередньо з інтерфейсу панелі;
- 4) пагінація для навігації по великих обсягах даних;
- 5) індикація часу останнього оновлення панелі.



Рисунок 2.1 – Структура функціональних модулів інформаційної панелі для управління згадуваннями

Ця структура забезпечує цілісне рішення для моніторингу згадувань, ефективне управління інформацією та інтеграцію в робочі процеси.

Для забезпечення наочності та структурованості процесів, що реалізовані в межах програмного забезпечення, було використано діаграми послідовності (UML sequence diagrams). Цей тип діаграм дає змогу моделювати послідовність взаємодії між основними компонентами системи та зовнішніми користувачами в межах окремих сценаріїв використання.

У межах розробленого рішення обрано два сценарії, що ілюструють функціональні аспекти роботи інформаційної панелі:

- налаштування панелі користувачем зображено на рисунку 2.2, що демонструє параметри відображення даних згадувань;

– перегляд згадувань користувача зображено на рисунку 2.3, що означає динамічне завантаження та візуалізація даних, отриманих із системи Jira.

Діаграми відображають взаємодію таких компонентів, як:

- користувач – ініціатор дій;
- інтерфейс – клієнтська частина додатку;
- система – логіка обробки запитів;
- API Jira – зовнішній інтерфейс для отримання даних про згадування.

Кожна діаграма демонструє часову послідовність обміну повідомленнями, викликів методів, обробки відповідей і оновлення інтерфейсу.

Перший сценарій демонструє повну послідовність дій, які виконує користувач для налаштування параметрів панелі згадувань відповідно до власних потреб. Спочатку користувач ініціює дію, відкриваючи розділ налаштувань панелі у призначеному для користувача інтерфейсі. Після цього надсилається запит до системи для отримання актуального стану параметрів, а система, у свою чергу, обробляє запит, звертається до збережених налаштувань і повертає їх інтерфейсу.

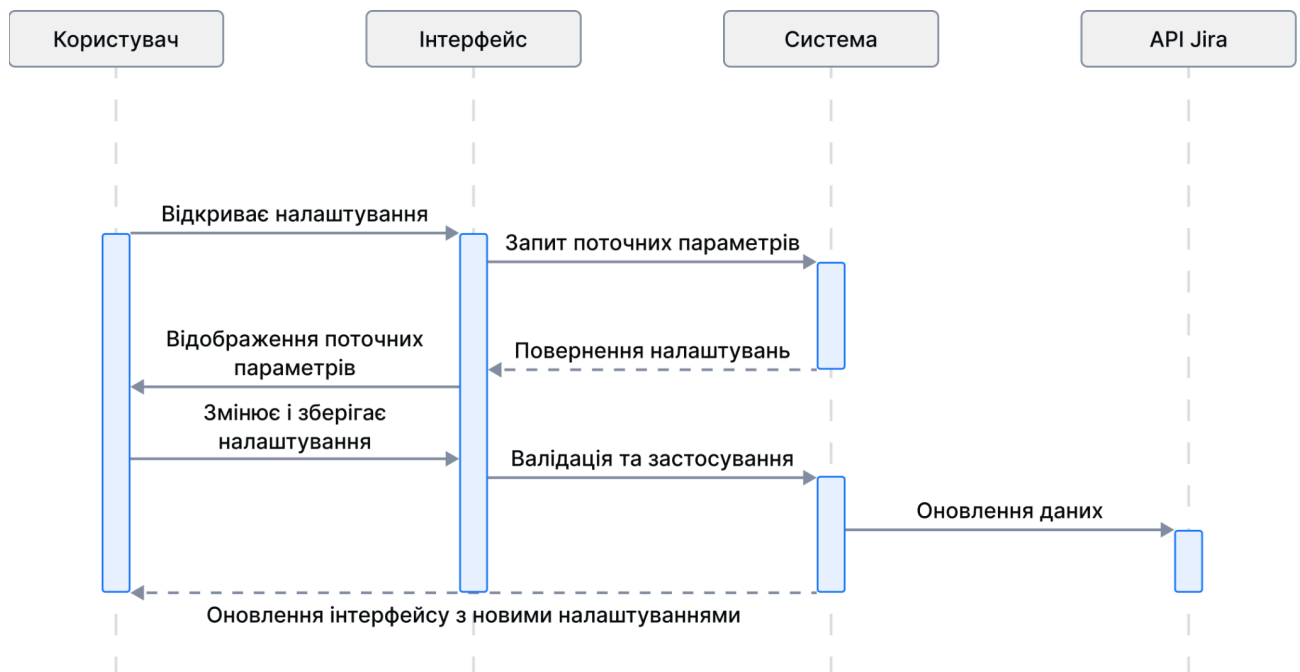


Рисунок 2.2 – Діаграма послідовності для сценарію налаштування панелі

На наступному етапі інтерфейс відображає поточні налаштування у формі, доступній для редагування, і користувач вносить корективи. Він може змінювати тип згадувань, які слід відображати, наприклад, вибираючи між коментарями, контрольними списками або обома варіантами. Після внесення змін оновлені параметри передаються назад до системи, яка перевіряє коректність введених даних, щоб уникнути можливих конфліктів.

Якщо перевірка проходить успішно, система зберігає нові параметри та, за необхідності, звертається до API Jira для оновлення пов'язаних атрибутів. У результаті оновлені налаштування застосовуються в інтерфейсі, і користувач одразу бачить зміни у відображенні згадувань.

Другий сценарій демонструє типову поведінку системи при запиті користувачем згадувань, у яких його було зазначено. Інформація динамічно завантажується з Jira та відображається в інтерфейсі. Спочатку користувач відкриває інформаційну панель або переходить до відповідного розділу в додатку, ініціюючи запит. Після цього інтерфейс надсилає запит до серверної частини для отримання згадувань, пов'язаних із його обліковим записом.

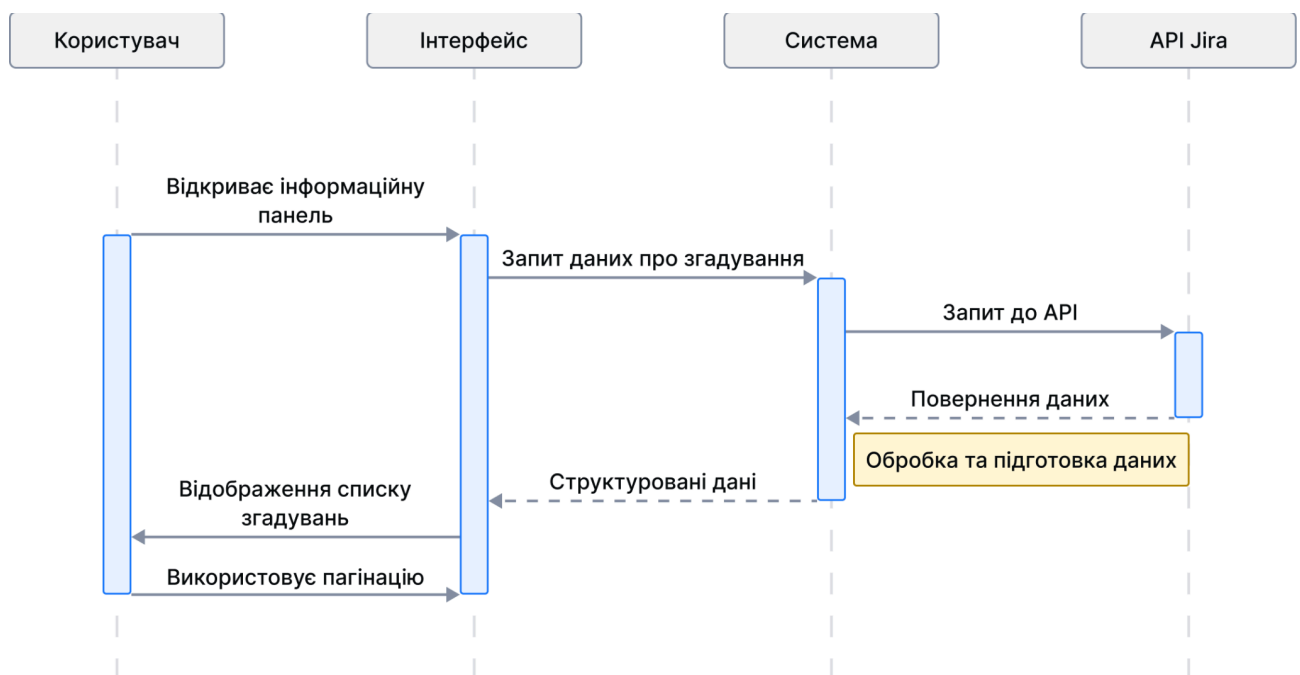


Рисунок 2.3 – Діаграма послідовності для сценарію перегляд згадувань користувача

Серверна частина, у свою чергу, формує відповідні запити до API Jira, наприклад, з використанням JQL, щоб отримати згадування з коментарів або контрольних списків. Отримавши відповідь, API Jira повертає результати – задачі, де користувача було згадано. Після отримання даних система проводить їхню попередню обробку: структурує, фільтрує та групує згадування, а також формує додаткову інформацію, включаючи тип джерела, час та проєкт.

На наступному етапі інтерфейс отримує оброблені дані та виводить їх у вигляді списку згадувань. Якщо згадувань багато, користувач може ініціювати додаткові запити через нижню навігацію. У результаті нові згадування додаються до наявного списку без перезавантаження сторінки, що забезпечує безперервний користувацький досвід. Такий механізм дозволяє користувачеві легко переглядати згадування, зберігаючи інтерактивність і зручність взаємодії з системою.

2.2 Алгоритмічне забезпечення

Алгоритмічне забезпечення інформаційної панелі для моніторингу згадувань у системі Jira визначає логіку функціонування системи та способи формування результатів. Нижче розглянуто основні алгоритми, що забезпечують роботу системи.

Важливою складовою розробленої системи “Mentions Dashboard for Jira” є інтуїтивно зрозумілий процес налаштування та використання гаджетів для моніторингу згадувань. Загальний алгоритм роботи з системою представлений на рисунку 2.4 і складається з послідовності чітко визначених кроків:

- 1) початок алгоритму;
- 2) додавання гаджета на інформаційну панель – на цьому етапі користувач обирає серед доступних гаджетів “Mentions Dashboard for Jira” та розміщує його на своїй персоналізованій інформаційній панелі в середовищі Jira;

3) за замовчуванням система пропонує опцію "Jira коментарі" як основне джерело для відслідковування згадок, що відповідає найбільш типовому сценарію використання;

4) за замовчуванням система пропонує опцію "Будь-хто", яка в подальшому дозволяє відстежувати згадування від усіх користувачів системи;

5) вибір змінити тип джерела:

5.1) якщо користувач не змінює за замовчуванням опцію, то система зберігає стандартне налаштування "Jira коментарі" і переходить до кроку "6.1";

5.2) в іншому випадку, система пропонує альтернативне джерело "Контрольні списки" для відстеження згадувань саме в них;

6) вибір фільтра за авторством згадувань:

6.1) якщо користувач не змінює за замовчуванням опцію, то система зберігає систему зберігає фільтр "Будь-хто", який в подальшому дозволяє відстежувати згадування від усіх користувачів системи;

6.2) в іншому випадку, система пропонує фільтр "Поточний користувач", що обмежує моніторинг лише згадуваннями від поточного користувача системи;

7) ввід користувачів, які були згадані безпосередньо в тих активностях, які користувач налаштував попередні фільтри;

8) підтвердження налаштування;

9) відображення списку згаданих користувачів, враховуючи налаштовані фільтри;

10) кінець алгоритму.

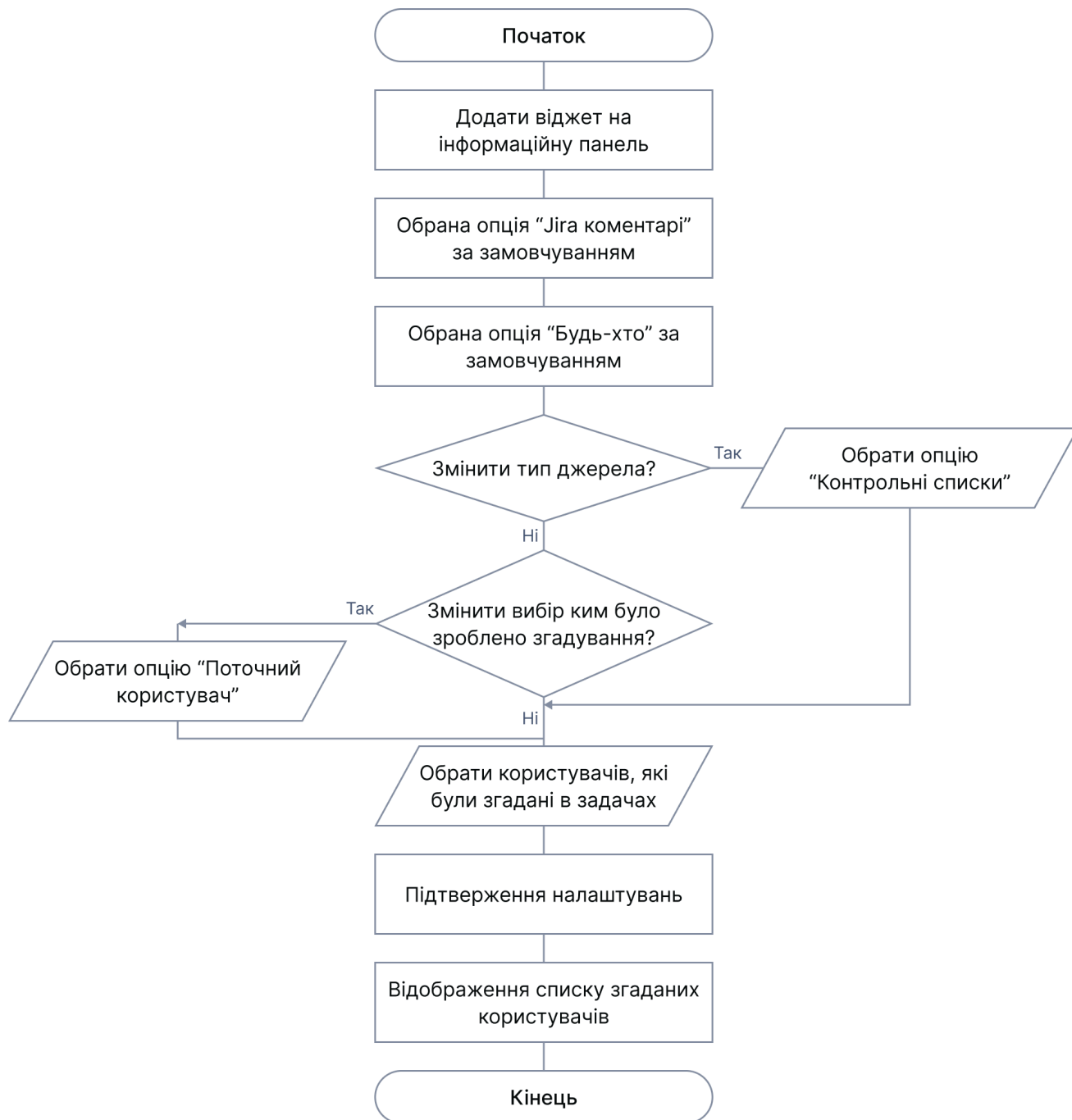


Рисунок 2.4 – Схема алгоритму роботи інформаційної панелі для моніторингу згадувань

На рисунку 2.5 представлено алгоритм збору та фільтрації даних, що забезпечує точність та цінність інформації, що відображається на панелі. Робота алгоритму починається з отримання параметрів, таких як джерело згадувань, автори та згадані користувачі. На основі цих параметрів формується JQL-запит для пошуку коментарів, що містять потенційні згадки, після чого виконується запит до API Jira для отримання базового набору даних.

Наступним етапом є аналіз текстового вмісту коментарів. Система виявляє згадування шляхом пошуку символу "@" з подальшим ідентифікатором користувача. Отримані результати проходять додаткову фільтрацію відповідно до заданих параметрів: перевіряється автор коментаря, наявність згаданих користувачів у списку для відображення, а також джерело згадування (коментарі Jira або контрольні списки).



Рисунок 2.5 – Схема алгоритму збору та фільтрації даних

Збагачення даних забезпечує отримання додаткової інформації про задачу, у якій розміщено коментар, повні дані про автора та згаданих користувачів, а також формування посилань для швидкого переходу до контексту згадки. Після цього результати сортуються за часом, щоб найновіші згадування відображалися першими. Для зручності навігації дані розподіляються на сторінки згідно з параметрами пагінації, а потім повертаються у вигляді структурованого набору для відображення.

Точність роботи збору та фільтрації забезпечується завдяки використанню регулярних виразів, які коректно виявляють тексти коментарів. Взаємодія з API Jira гарантує актуальність отриманої інформації, що сприяє високій достовірності даних.

Особливість швидкої відповіді на згадування надає одну з основних функцій системи – можливість відповідати на коментарі безпосередньо з інтерфейсу інформаційної панелі без необхідності переходу до задачі.

Це включає отримання вхідних даних, таких як, текст відповіді, ідентифікатор задачі, де розміщено коментар та ідентифікатор коментаря, на який надається відповідь. Також присутнє формування запиту до API Jira для створення нового коментаря, тобто йде підготовка структури даних для відправки, а також включення посилання на оригінальний коментар.

Налаштування інформаційної панелі забезпечує персоналізацію відображення згадувань відповідно до потреб користувача.

Форма параметрів відображається з поточними налаштуваннями, що дозволяє користувачеві налаштувати параметри згадувань відповідно до своїх потреб. Під час обробки введених даних користувач може вибрати джерело згадувань, наприклад коментарі Jira або контрольні списки. Також доступний вибір автора згадувань – це може бути поточний користувач або будь-який інший. Крім того, користувач має можливість визначити, яких саме згаданих користувачів слід враховувати: конкретних осіб або всіх.

Перед збереженням налаштувань система виконує валідацію параметрів. Вона перевіряє сумісність вибраних параметрів, а також наявність необхідних дозволів для доступу до даних. Після успішної перевірки параметри можуть бути збережені: тимчасово для поточної сесії або постійно для подальшого використання.

Після внесення змін система застосовує нові параметри. Заголовок панелі оновлюється, відображаючи вибрані налаштування, а також ініціюється запит на отримання відповідних даних. У результаті змінений інтерфейс відображає актуальні згадування відповідно до нових параметрів.

Якщо під час налаштування інформаційної панелі виникають помилки, користувач отримує зрозумілі повідомлення про проблеми. Система пропонує підказки щодо коректних налаштувань, а у разі – серйозних помилок дає змогу повернутися до попередньої. Це забезпечує стабільність та зручність користування інтерфейсом.

2.3 Партнерська інтеграція

У рамках стратегічного розвитку продукту було ініційовано партнерську інтеграцію з компанією HeroCoders [22], розробником популярного додатку "Checklists for Jira (Pro)" [23]. Ця співпраця представляє собою взаємовигідне партнерство, спрямоване на створення комплексного користувацького досвіду через інтеграцію функціональності контрольних списків та системи згадувань користувачів.

Перед реалізацією інтеграції було проведено детальне дослідження технічних можливостей партнерського продукту. Аналіз документації HeroCoders виявив декілька основних аспектів для успішної інтеграції контрольних списків.

Дослідження API партнера показало, що витяг переліків здійснюється через спеціалізовані точки доступу, які дозволяють отримувати як повний вміст списків, так і їх текстове представлення у режимі "лише для читання". Особливу увагу було приділено вивченню загальних налаштувань контрольних списків, оскільки від них залежить формат та структура даних, які передаватимуться через інтеграцію.

Дослідження також виявило, що для успішного підключення та перегляду переліків необхідно пройти низку специфічних налаштувань на стороні партнерського застосунку. Ці налаштування включають налаштування прав доступу, встановлення відповідних дозволів API та налаштування параметрів синхронізації між системами.

Хоча партнерський додаток має власний віджет для відображення згадувань у контрольних списках, інтеграція дозволить елементам цих списків з'являтися в

інформаційній панелі нашого продукту, створюючи комплексний користувацький досвід. Ця інтеграція означає, що користувачі зможуть отримувати уніфіковане представлення всіх згадувань та задач в одному місці, незалежно від того, чи походять вони з традиційних коментарів Jira, чи з інтерактивних контрольних списків.

Такий підхід значно покращує ефективність роботи команд, оскільки користувачам не потрібно переключатися між різними інтерфейсами для відстеження всіх релевантних активностей. Реалізація партнерської інтеграції демонструє здатність продукту адаптуватися та інтегруватися з іншими інструментами, що є важливим для корпоративних клієнтів, які використовують технологічні комплекси.

Довгострокова перспектива цього партнерства включає можливість створення платформи інтеграцій, де різні додатки можуть взаємодіяти через наш продукт як центральний вузол для управління згадуваннями та сповіщеннями в екосистемі Atlassian. Це позиціонує продукт не просто як окремий додаток, а як інтеграційну платформу для покращення спільної роботи команд.

3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ПАНЕЛІ ЗГАДОК ДЛЯ ВІДСТЕЖЕННЯ ТА КЕРУВАННЯ КОМУНІКАЦІЯМИ ПРОГРАМНОГО СЕРЕДОВИЩА JIRA

3.1 Програмна реалізація

Інформаційна панель згадок для відстеження та керування комунікаціями програмного середовища Jira реалізована як застосунок на платформі Atlassian Forge, який інтегрується безпосередньо в екосистему Jira Cloud.

У Додатку А представлено програмний код, що використовується для реалізації описаних алгоритмів та методів. Він містить структуровані фрагменти коду, які забезпечують функціональність розглянутої системи.

Архітектура системи базується на модульному підході, що забезпечує гнучкість розширення функціональності та легкість супроводження коду. Структурна схема програмної системи складається з чотирьох основних компонентів, які взаємодіють між собою через добре визначені інтерфейси.

Презентаційний шар відповідає за відображення користувацького інтерфейсу та обробку взаємодії з користувачем. Цей шар включає компоненти для налаштування параметрів пошуку, відображення результатів згадувань та управління станом додатку.

Логічний шар містить бізнес-логіку обробки даних, включаючи алгоритми пошуку згадувань у коментарях та контрольних списках, фільтрацію результатів за критеріями користувача та форматування даних для відображення. Шар інтеграції забезпечує взаємодію з API Jira (рисунок 3.1) та додатковими сервісами.

```
const getResponse = async (route, options) : Promise<...> => { Show usages
  try {
    const response : Response = await requestJira(route, options);
    return await response.json();
  } catch (e) {
    console.log(e);
    return null;
  }
}
```

Рисунок 3.1 – Лістинг запиту до Jira API для обробки відповіді

Він включає модулі для роботи з REST API Jira, обробки автентифікації та авторизації, а також інтеграції з додатком контрольних списків HeroCoders (рисунок 3.2). Шар даних відповідає за тимчасове зберігання стану застосунку та кешування отриманих з API даних для покращення продуктивності.

```
export const isChecklistAppInstalled = async () : Promise<...> => { Show usages
  try {
    const response :... = await getResponse( route: '/rest/api/3/field', options: {
      headers: { 'Accept': 'application/json' }
    });

    if (!response || !Array.isArray(response)) {
      return false;
    }

    const checklistField = response.find(field => field.name === 'Checklist Text');
    return checklistField !== undefined;
  } catch (error) {
    console.error("Error checking if checklist app is installed:", error);
    return false;
  }
};

export const getIssuesWithChecklists = async (startAt = 0, maxResults = 100) : Promise<any> => { Show usages
  const jql = `"Checklist Text[Paragraph]" IS NOT EMPTY ORDER BY updated DESC`;
  const fields = ["key", "summary", "issuetype", "status", "Checklist Text"];
  return await getIssues(jql, startAt, maxResults, fields);
};
```

Рисунок 3.2 – Лістинг перевірки встановлення додатку “Checklists” та витяг задач з контрольними списками

Клас CombinedList виступає центральним координатором, який управляє процесом завантаження та об'єднання даних з різних джерел (рисунок 3.3). Він взаємодіє з API Jira для отримання коментарів та з додатком контрольних списків для завантаження відповідних даних, після чого об'єднує результати в єдиний потік для відображення користувачу.

Вибір програмних засобів реалізації обумовлений специфікою платформи Atlassian Forge та вимогами до інтеграції з екосистемою Jira. Forge – це платформа для розробки хмарних застосунків від Atlassian, яка дозволяє розробникам створювати безпечні та інтегровані рішення [24].

```

return (
  <div style={{
    display: 'flex',
    flexDirection: 'column',
    height: '480px',
    overflowY: 'auto',
    paddingRight: token( path: 'space.050'),
    gap: token( path: 'space.100')}}>
    {combinedItems.map((item, index :number) => (
      item.type === 'mention' ? (
        <Mention key={`mention-${index}`} mention={item} selected={selected} siteUrl={context?.siteUrl} isUserMode={isUserMode}/>
      ) : (
        <Checklist key={`checklist-${index}`} issueKey={item.key} summary={item.summary} issuetype={item.issuetype} status={item.status}
          siteUrl={context?.siteUrl}
          lastUpdate={item.lastUpdate}
          filteredItems={item.filteredItems}
        />
      )
    ))}
  </div>
);

```

Рисунок 3.3 – Лістинг компонента CombinedList для рендерингу списку
згадувань та контрольних списків

Робота з платформою Forge виявилася значно складнішою, ніж очікувалося на початковому етапі проєкту. Специфіка цієї платформи потребувала глибокого вивчення її архітектурних особливостей та обмежень, що значно відрізняються від традиційних підходів до веб-розробки. Платформа має власну екосистему з унікальними правилами безпеки, особливостями розгортання та специфічними програмними інтерфейсами.

Додаткове дослідження документації Forge стало необхідним через відсутність чітких прикладів для складних сценаріїв інтеграції, особливо при роботі з специфічними полями сторонніх додатків. Документація платформи містить базові приклади, але для реалізації партнерської інтеграції з контрольними списками потрібно було досліджувати поглиблені аспекти роботи з програмними інтерфейсами Jira через Forge. Це включало вивчення особливостей обробки помилок, управління правами доступу та оптимізації продуктивності в умовах хмарного середовища.

Особливу складність становило розуміння життєвого циклу застосунків Forge та їх взаємодії з основною платформою Jira. На відміну від традиційних веб-застосунків, Forge накладає певні обмеження на використання зовнішніх бібліотек та програмних інтерфейсів, що потребувало переосмислення архітектурних рішень та адаптації існуючих підходів до розробки.

React Framework використовується як основна бібліотека для побудови користувацького інтерфейсу завдяки його природній інтеграції з платформою Forge та можливості створення компонентних архітектур. React забезпечує ефективне управління станом додатку через хуки `useState` та `useEffect`, що особливо важливо для додатків, які активно взаємодіють з зовнішніми API.

Atlassian Design System (ADS) вибрано як UI-бібліотеку для забезпечення консистентності з дизайном Jira та інших продуктів Atlassian. Використання компонентів ADS гарантує, що додаток природно інтегрується в інтерфейс Jira та надає користувачам звичне середовище взаємодії.

Платформа Forge надає вбудовані механізми аутентифікації та авторизації, що дозволяє додатку безпечно взаємодіяти з API Jira без необхідності самостійної реалізації цих аспектів безпеки. Модульна архітектура коду забезпечує можливість незалежного тестування окремих компонентів та спрощує процес розширення функціональності в майбутньому (рисунок 3.4) [25].

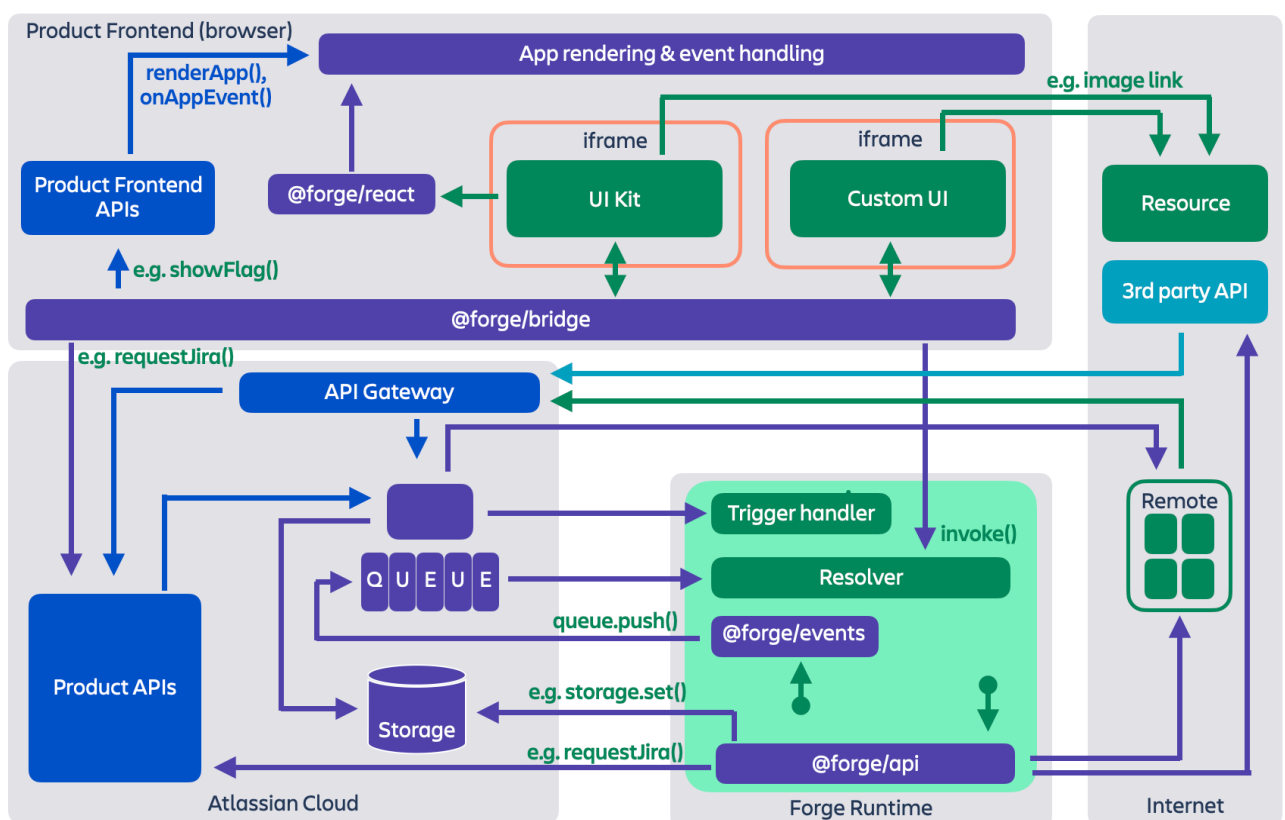


Рисунок 3.4 – Діаграма, що демонструє архітектуру платформи Forge [25]

3.2 Інтерфейс користувача

Користувацький інтерфейс системи відстеження згадувань розроблено з урахуванням принципів зручності використання та інтеграції в робочий процес користувачів Jira. Основним завданням інтерфейсу є надання швидкого та інтуїтивно зрозумілого способу відстеження згадувань користувачів у коментарях та контрольних списках без відволікання від основної роботи в Jira.

Головна сторінка додатку містить панель налаштувань, де користувач може визначити параметри пошуку згадувань. Інтерфейс налаштування організовано у вигляді вертикальної форми з логічним групуванням елементів керування (рисунок 3.5). Перший блок дозволяє вибрати джерела пошуку згадувань, включаючи коментарі Jira та контрольні списки від партнера HeroCoders. Використання випадаючого меню з можливістю множинного вибору забезпечує гнучкість налаштувань інформаційної панелі без перевантаження інтерфейсу.

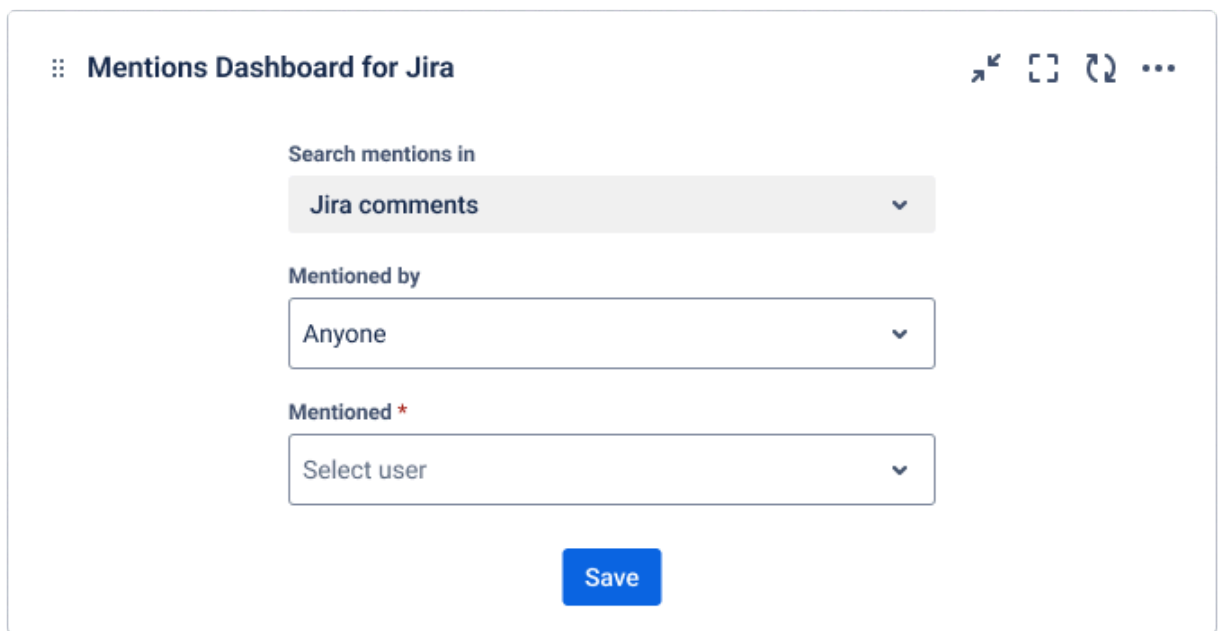


Рисунок 3.5 – Інтерфейс сторінки налаштувань

Другий блок налаштувань дозволяє обрати режим пошуку за автором згадування. Користувач може вибрати між режимом "Поточний користувач" (рисунок 3.6) та режимом "Будь-хто", коли показуються всі згадування незалежно

від автора. Такий підхід забезпечує персоналізацію досвіду використання додатку відповідно до ролі та потреб конкретного користувача.

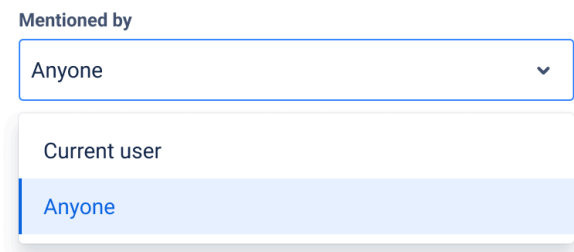


Рисунок 3.6 – Блок налаштувань за автором згадування

Третій блок містить компонент вибору користувачів для відстеження. Інтерфейс реалізовано у вигляді поля з автозаповненням та підтримкою множинного вибору (рисунок 3.7), що дозволяє швидко знайти та додати потрібних користувачів. Компонент відображає аватари користувачів та їхні імена, що покращує візуальне розрізнення та спрощує процес вибору. Обмеження на максимальну кількість вибраних користувачів запобігає перевантаженню системи та забезпечує оптимальну продуктивність.

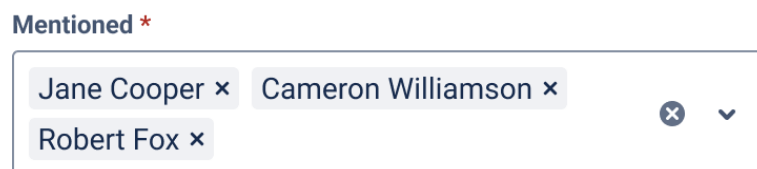


Рисунок 3.7 – Блок налаштувань за вибором користувачів для відстеження

Основна сторінка перегляду результатів організована у вигляді хронологічного списку згадувань з можливістю прокручування (рисунок 3.8). Кожен елемент списку представлено у вигляді картки, що містить ключову інформацію про згадування. Візуальна ієрархія інформації побудована таким чином, що найважливіші деталі, такі як ім'я автора згадування та час створення, розміщені на найпомітніших позиціях.

Кожна картка згадування включає аватар автора, що забезпечує швидку візуальну ідентифікацію, інформацію про час створення згадування та контекст, в

якому воно з'явилося. Для згадувань у коментарях відображається фрагмент коментаря з виділенням згаданого користувача, а для контрольних списків показується відповідний пункт списку зі статусом виконання.

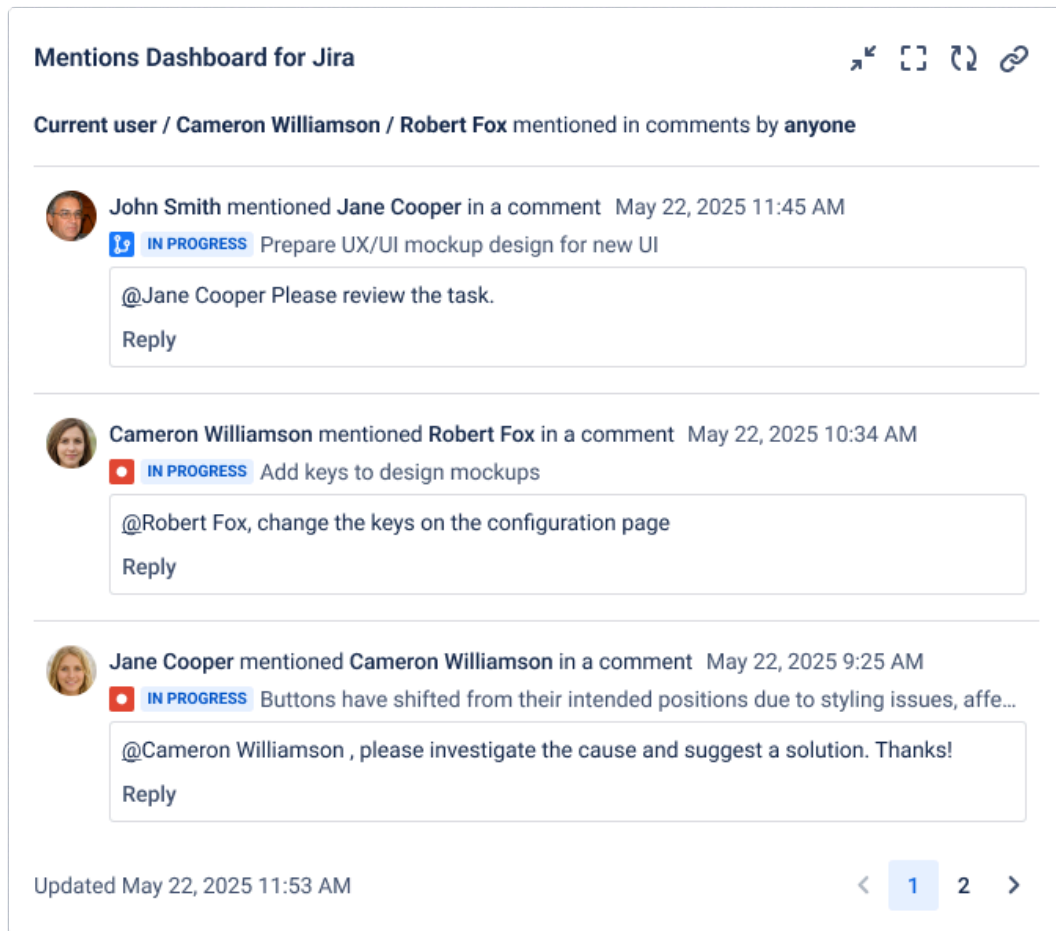


Рисунок 3.8 – Інтерфейс основної сторінки з відображенням списку згадувань у вигляді карток

Механізм навігації реалізовано через систему посилань, що дозволяють швидко переходити від згадування до відповідної задачі в Jira. Клік на номер задачі або її назву відкриває задачу в новій вкладці, зберігаючи контекст роботи з додатком. Додаткові дії, такі як швидка відповідь на коментар, доступні через інлайн-елементи керування безпосередньо в картці згадування.

Адаптивність інтерфейсу забезпечує коректне відображення на різних розмірах екранів та пристроях. Компоненти автоматично адаптуються до доступного простору, зберігаючи читабельність та зручність взаємодії. Система

нижньої навігації дозволяє ефективно працювати з великими обсягами даних без перезавантаження інтерфейсу.

Більше того, додаток повністю підтримує темну тему інтерфейсу (рисунок 3.9), автоматично адаптуючись до налаштувань користувача в Jira. Ця функціональність реалізована через використання компонентів Atlassian Design System, які динамічно змінюють кольорову схему відповідно до обраної теми.

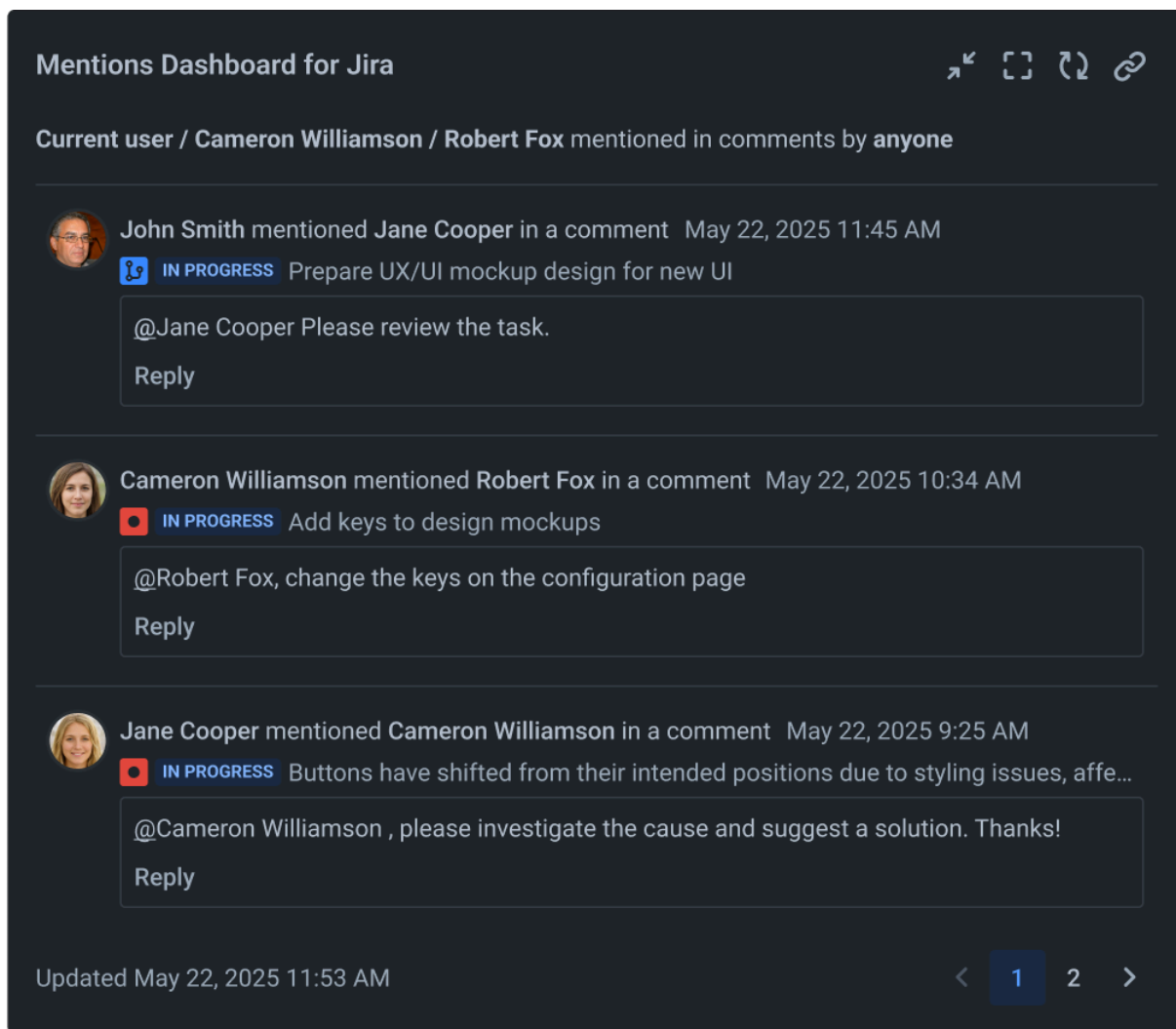


Рисунок 3.9 – Інтерфейс головної сторінки у темному оформленні

З точки зору користувацького досвіду, підтримка темної теми значно покращує комфорт роботи, особливо під час тривалих сесій або роботи в умовах слабого освітлення. Темна тема зменшує навантаження на зір, знижує втому очей та покращує читабельність тексту в темних приміщеннях [26].

Обробка помилок та виняткових ситуацій реалізована через систему інформативних повідомлень, що з'являються в контексті відповідних дій зображені на рисунках 3.10 і 3.11.

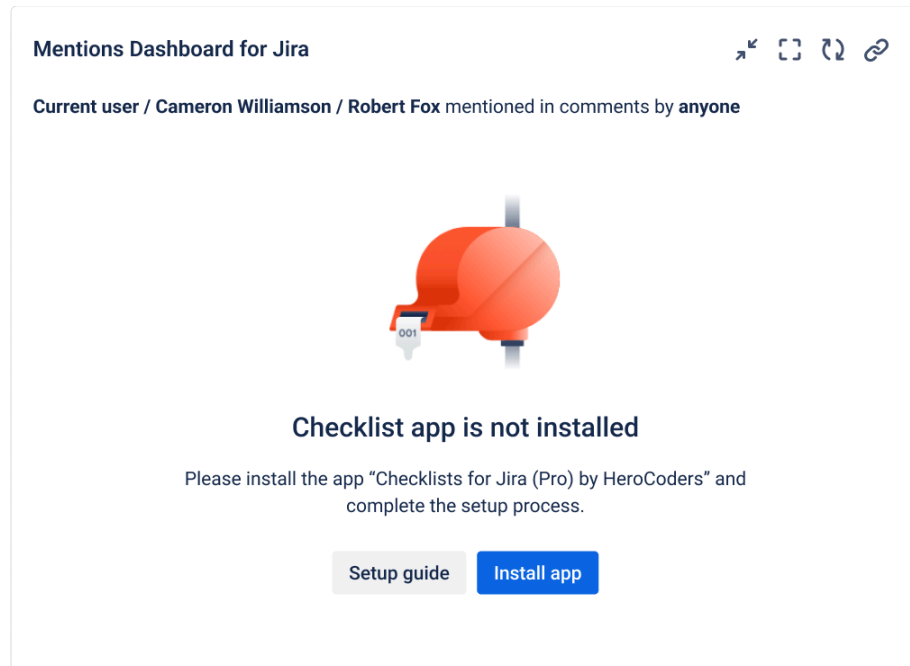


Рисунок 3.10 – Інформативне повідомлення про те, що додаток “Checklists” не встановлено

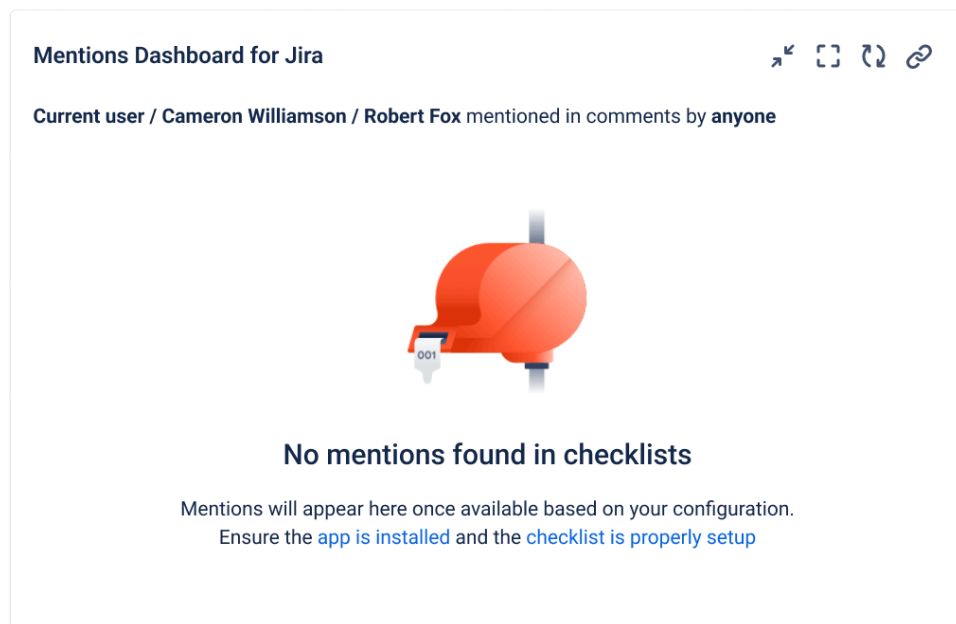


Рисунок 3.11 – Інформативне повідомлення про те, що не знайдено згадок в контрольних списках

Повідомлення містять зрозумілі пояснення проблеми та рекомендації щодо її вирішення. Для ситуацій, коли необхідно встановити додаткові компоненти або налаштувати інтеграції, система надає прямі посилання на відповідні ресурси та інструкції.

3.3 Тестування розробленої програми

Тестування системи відстеження згадувань користувачів проводилося за методологією багатоетапного тестування, що включає модульне тестування окремих компонентів, інтеграційне тестування взаємодії між модулями та системне тестування повного робочого циклу додатку.

Інтеграційне тестування включало перевірку взаємодії з API Jira та додатком контрольних списків HeroCoders. Однією з технічних особливостей виявилася специфіка роботи з нестандартними полями Jira. Оскільки "Checklists for Jira (Pro)" є додатковою програмою, її дані зберігаються в нестандартних полях, що потребувало розробки спеціального алгоритму їх ідентифікації. Початкові спроби отримання доступу до переліків через фіксовані ідентифікатори полів виявилися неефективними.

Подальше дослідження показало деяке спостереження, що ідентифікатори нестандартних полів різняться між різними примірниками Jira. Це спостереження призвело до необхідності розробки більш гнучкого підходу, який базується на пошуку полів за їх повною назвою замість використання статичних ідентифікаторів. Такий підхід забезпечив універсальність інтеграції та її функціонування в різних корпоративних середовищах.

Загалом тестування проводилося в контрольованому середовищі з використанням тестових даних, що імітують реальні робочі сценарії. У процесі тестування було виявлено критичну помилку, суть якої полягала була в тому, що не відображалися контрольні списки та з'явилося "вічне" завантаження. У ході роботи цю проблему було усунуто. Причиною виявилось, що спочатку контрольні

списки витягувалися по всіх задачах, але пізніше було змінено на JQL-запит (рисунок 3.12), що спростив роботу додатку в рази.

```
export const getIssuesWithChecklists = async (startAt = 0, maxResults = 100) : Promise<any> => {  
  const jql = `"Checklist Text[Paragraph]" IS NOT EMPTY ORDER BY updated DESC`;  
  const fields = ["key", "summary", "issuetype", "status", "Checklist Text"];  
  return await getIssues(jql, startAt, maxResults, fields);  
};
```

Рисунок 3.12 – Лістинг витягу контрольних списків через JQL-запит

Під час системного тестування було перевірено функціональність додатку, виконуючи стандартні робочі сценарії. Було налаштовано параметри пошуку, відстежено згадування користувачів та переміщення між результатами пошуку. Тестування виявило декілька технічних особливостей, які потребували уваги:

- до проведення відповідних налаштувань згідно з інструкцією, віджет відображав сторінку з некоректним повідомленням про помилку і ввів через неправильне посилання. Для вирішення цієї проблеми було додано пряме посилання на інструкцію з налаштуванням та перенаправленням користувачів на сторінку глобальних параметрів системи;

- було виявлено, що віджет змінює свої розміри при відображенні певного вмісту, не зберігаючи початкову висоту. Ця особливість поведінки інтерфейсу потребувала додаткового дослідження для забезпечення стабільності відображення елементів.

На основі проведеного тестування всі виявлені проблеми були успішно усунуті. У результаті функціональність працює коректно та відповідає очікуваним вимогам.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи:

1. Проведено аналіз структури Jira, її функціональних можливостей, системи сповіщень та згадувань, який показав, що існуючі інструменти платформи не забезпечують зручного, гнучкого і масштабованого механізму для моніторингу згадувань.

2. Огляд рішень виявив їхню неповну функціональність: відсутність можливості фільтрації, обмежений доступ до контексту згадування, незручна навігація та низька інтеграція з основними сценаріями роботи в Jira. Це стало обґрунтуванням доцільності створення власного рішення і зробити постановку задачі проектування.

3. Розроблено структуру програмного забезпечення, яка включає такі компоненти, як модуль збору даних із використанням JQL, фільтрації, візуалізації та відповіді на згадування. Реалізовано підтримку багатоджерельності (коментарі Jira, контрольні списки HeroCoders) та налаштування за автором згадування і списком користувачів.

4. Розроблено алгоритм збору, фільтрації та відображення згадувань користувачів, який забезпечує точну ідентифікацію згадок через регулярні вирази, інтеграцію з API Jira для отримання актуальних даних та персоналізоване налаштування параметрів відображення. Алгоритм включає функціональність швидкої відповіді на коментарі безпосередньо з інтерфейсу панелі, що значно підвищує ефективність взаємодії користувачів.

5. Реалізовано партнерську інтеграцію з додатком "Checklists for Jira (Pro)", яка дозволяє об'єднати згадування з контрольних списків та традиційних коментарів в єдиному інтерфейсі. Інтеграція передбачає динамічний пошук нестандартних полів за назвою для забезпечення сумісності з різними зразками Jira та створює основу для розвитку екосистеми партнерських рішень у майбутньому.

6. Розроблено повноцінний програмний засіб, що дозволяє централізовано відстежувати згадування користувачів у задачах Jira, ефективно керувати інформаційними потоками та покращити якість внутрішньої комунікації.

7. Розроблено інтерфейс користувача, відповідно до стандартів Atlassian Design System. Інтерфейс дозволяє легко здійснювати налаштування, переглядати згадування у вигляді карток з контекстною інформацією, переходити до відповідної задачі, а також швидко відповідати на коментарі. Така функціональність значно спрощує взаємодію з системою та мінімізує втрати інформації в процесі командної роботи.

8. Тестування програмного продукту засвідчило його надійність, зручність та відповідність вимогам користувачів. Зокрема, було виявлено та усунуто важливу помилку з обробкою контрольних списків, а результати тестів підтвердили правильність реалізації алгоритмів пошуку, фільтрації та відображення згадувань.

9. Запропоноване рішення має потенціал до подальшого розвитку як інтеграційна платформа для взаємодії з іншими інструментами Atlassian та сторонніми сервісами. Його можна використовувати як базу для розширення екосистеми віджетів Jira, створення індивідуально налаштованих панелей керування згадуваннями для різних команд та департаментів.

10. Загалом, реалізована інформаційна панель вирішує важливу проблему сучасного проєктного менеджменту – прозорість і керованість згадувань, що є досить важливими для забезпечення оперативної взаємодії та відповідальності в командах, які працюють у динамічному середовищі.

11. Кваліфікаційна робота має практичну значимість, а результати її виконання плануються до використання (додаток Б). Інформаційна панель згадок у Jira може використовуватись менеджерами проєктів, розробниками та аналітиками для контролю залученості в обговорення, відстеження важливих звернень і швидкого реагування на коментарі. Вона допомагає ефективно керувати завданнями та уникати втрати важливої інформації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Muszyńska K. Measuring communication management performance in virtual project teams. *Procedia Computer Science*. 2022. Vol.207. P.3507–3515.
2. Vaskiv R., Veretennikova N. Information and communication tools for effective functioning of distributed project teams. *Scientific Journal of Information Systems and Networks*. 2024. Vol.15. P.357–369.
3. Aglibar K.D., Alegre G.C., Del Mundo G., Duro K.F., Rodelas N. Ticketing System: A Descriptive Research on the Use of Ticketing System for Project Management and Issue Tracking in IT Companies. *National Research Conference in Computer Engineering and Technology*. 2022.
4. Get Started with Jira - Comprehensive Beginner's Guide. Atlassian. Product guide. Getting started. Introduction to Jira. What is Jira? URL: <https://www.atlassian.com/software/jira/guides/getting-started/introduction#what-is-jira-software>.
5. Jira vs. Jira Align: Which One is Right for Your Agile Strategy? Idalko. URL: <https://idalko.com/jira-vs-jira-align-which-one-is-right-for-your-agile-strategy/>.
6. Master Jira Projects: Key Elements and Essentials. Atlassian. Product guide. Projects. Overview. What is a Jira project? URL: <https://www.atlassian.com/software/jira/guides/projects/overview#what-is-a-jira-project>
7. A Brief Introduction to Jira Boards. Atlassian. Product guide. Boards. Overview. What types of boards are available in Jira? URL: <https://www.atlassian.com/software/jira/guides/boards/overview#board-vs-project>.
8. Learn About Jira Software Reports & Dashboards. Atlassian. Product guide. Reports and Dashboards. Overview. What are dashboards? URL: <https://www.atlassian.com/software/jira/guides/reports-dashboards/overview#what-are-dashboards>.
9. Introduction to Jira Software Work Items. Atlassian. Product guide. Work items/issues. Overview. What are work types? URL:

<https://www.atlassian.com/software/jira/guides/issues/overview#what-are-work%20item-types>.

10. Mentions list – Atlassian Community. URL: https://community.atlassian.com/forums/forums/searchpage/tab/message?advanced=false&allow_punctuation=false&q=mentions+list.

11. O’Connell P. Jira Software Essentials: Plan and manage your projects efficiently using the all-new Jira Software Cloud. *Packt Publishing*.2020. P.354.

12. Бушуєв С.Д. Управління проектами у розвитку суспільства. *Тези доповідей*. Київ: КНУБА, 2024. С. 270.

13. Few S. Information Dashboard Design: Displaying Data for At-a-Glance Monitoring. 2nd ed. Burlingame: Analytics Press, 2013. P.246.

14. Calefato, F., Giove, A., Losavio, M., & Lanubile, F. A case study on tool support for collaboration in agile development. *15th IEEE/ACM International Conference on Global Software Engineering*. 2020.

15. Mastering Notifications in Jira. Atlassian Community. URL: <https://community.atlassian.com/forums/App-Central-articles/Mastering-Notifications-in-Jira/ba-p/2667471>.

16. Jira Notifications Hub EAP: What's changed? Atlassian Community. URL: <https://community.atlassian.com/forums/Jira-articles/Jira-Notifications-Hub-EAP-What-s-changed/ba-p/2543661>.

17. How to Set Up the Perfect Jira Notification Scheme Idalko. URL: <https://idalko.com/jira-notification-scheme/>.

18. Unlock Advanced Search with Jira Query Language (JQL). Atlassian. Product guide. JQL. Overview. What is a JQL? URL: <https://www.atlassian.com/software/jira/guides/jql/overview#what-is-jql>.

19. User Mentions Gadget for Jira. Atlassian Marketplace. URL: <https://marketplace.atlassian.com/apps/1229393/user-mentions-gadget-for-jira?tab=overview&hosting=cloud>.

20. Threaded Comments for Jira Cloud. URL: <https://docs.votazz.co/threaded-comments-for-jira-cloud/>.

21. Threaded Comments for Jira. Atlassian Marketplace. URL: <https://marketplace.atlassian.com/apps/1212145/threaded-comments-for-jira?hosting=cloud&tab=overview>.
22. HeroCoders. URL: <https://www.herocoders.com/>.
23. Checklists for Jira (Pro) by HeroCoders. Atlassian Marketplace. URL: <https://marketplace.atlassian.com/apps/1213231/checklists-for-jira-pro-by-herocoders?hosting=cloud&tab=overview>.
24. Taking the Ecosystem Forward: An Update on the Future of Connect - Work Life by Atlassian. URL: <https://www.atlassian.com/blog/developer/taking-the-ecosystem-forward-an-update-on-the-future-of-connect>.
25. The Forge platform. URL: <https://developer.atlassian.com/platform/forge/introduction/the-forge-platform/>.
26. Sengsoon P., Intaruk R. Immediate Effects of Light Mode and Dark Mode Features on Visual Fatigue in Tablet Users. *International Journal of Environmental Research and Public Health*. 2025. Vol.22, No.4. Article 609.
27. Лимар В.Р., Турченко І.В. Інформаційна панель згадок для відстеження та керування комунікаціями програмного середовища Jira. *Інтелектуальні інформаційні технології в прикладних дослідженнях (ІІТАР-2025)*: збірник тез доповідей студентської науково-практичної конференції (Тернопіль, 27–29 травня 2025 р.). Тернопіль: ЗУНУ, 2025. С. 342-345.
28. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Ліп'яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

Додаток А

Програмний код основного функціоналу інформаційної панелі

```
const getResponse = async (route, options) : Promise<...> => { Show usages
  try {
    const response : Response = await requestJira(route, options);
    return await response.json();
  } catch (e) {
    console.log(e);
    return null;
  }
}
```

Рисунок А.1 – Лістинг функції отримання відповіді з API Jira

```
export const getAllUsers = async() : Promise<any> => { Show usages
  const result = [];
  let startAt = 0;
  let responseLength;
  do {
    const response : ... = await getResponse( route: `/rest/api/3/users/search?maxResults=1000&startAt=${startAt}`,
      headers: {
        'Accept': 'application/json'
      }
    );
    if (!response || !Array.isArray(response)) {
      break;
    }
    responseLength = response.length;
    startAt += responseLength;
    result.push(...response);
  } while (responseLength !== 1000);
  return result;
}
```

Рисунок А.2 – Лістинг функції отримання всіх користувачів системи Jira

```
export const getUser = async(accountId) : Promise<any> => { Show usages
  const response : ... = await getResponse( route: `/rest/api/3/user?accountId=${accountId}`, options: {
    headers: {
      'Accept': 'application/json'
    }
  });
  return response;
}
```

Рисунок А.3 – Лістинг функції отримання даних конкретного користувача

```

export const getIssues = async (jql, startAt, maxResults, fields) : Promise<any> => {
  const bodyData : any | string = JSON.stringify( value: {
    fields,
    jql,
    startAt,
    maxResults
  })
  const response : ... = await getResponse( route: `/rest/api/3/search`, options: {
    method: 'POST',
    headers: {
      'Accept': 'application/json',
      'Content-Type': 'application/json'
    },
    body: bodyData
  });
  return response;
}

```

Рисунок А.4 – Лістинг функції отримання задач за JQL-запитом

```

export const addComment = async (issueKey, bodyData) : Promise<any> =>{ Show usages
  const response : ... = await getResponse( route: `/rest/api/3/issue/${issueKey}/comment`, options: {
    method: 'POST',
    headers: {
      'Accept': 'application/json',
      'Content-Type': 'application/json'
    },
    body: bodyData
  });
  return response;
}

```

Рисунок А.5 – Лістинг функції додавання коментаря до задачі

```

export const getIssuesWithChecklists = async (startAt = 0, maxResults = 100) : Promise<any> => {
  const jql = `"Checklist Text[Paragraph]" IS NOT EMPTY ORDER BY updated DESC`;
  const fields = ["key", "summary", "issuetype", "status", "Checklist Text"];
  return await getIssues(jql, startAt, maxResults, fields);
};

```

Рисунок А.6 – Лістинг функції отримання задач із контрольними списками

```

export const isChecklistAppInstalled = async () : Promise<...> => { Show usages
  try {
    const response :... = await getResponse( route: '/rest/api/3/field', options: {
      headers: { 'Accept': 'application/json' }
    });

    if (!response || !Array.isArray(response)) {
      return false;
    }

    const checklistField = response.find(field => field.name === 'Checklist Text');
    return checklistField !== undefined;
  } catch (error) {
    console.error("Error checking if checklist app is installed:", error);
    return false;
  }
};

```

Рисунок А.7 – Лістинг функції перевірки на наявність інстальованої програми контрольних списків

```

export const getIssueProperty = async (issueIdOrKey, fieldKey) : Promise<...> => { Show usages
  return await getResponse( route: `/rest/api/3/issue/${issueIdOrKey}?fields=${fieldKey}`, options: {
    headers: { 'Accept': 'application/json' }
  });
};

```

Рисунок А.8 – Лістинг функції отримання значення певного поля задачі за її ідентифікатором

```

export const getIssueChangelog = async (issueIdOrKey) : Promise<...> => { Show usages
  const fieldId = await getChecklistFieldId();
  if (!fieldId) {
    return null;
  }
  const response = await getResponse( route: `/rest/api/3/issue/${issueIdOrKey}/changelog`, options: {
    headers: { 'Accept': 'application/json' }
  });
  if (response && response.values) {
    const checklistChanges = response.values
      .filter(change => change.items && change.items.some(item => item.fieldId === fieldId))
      .sort((a, b) => new Date(b.created) - new Date(a.created));

    return checklistChanges.length > 0 ? checklistChanges[0] : null;
  }
  return null;
};

```

Рисунок А.9 – Лістинг функції отримання історії змін задачі з фільтрацією за полем контрольного списку

Додаток Б

Додаток В
Копія опублікованих результатів

Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління



ЗБІРНИК ТЕЗ ДОПОВІДЕЙ

Студентської науково-практичної конференції
ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ПРИКЛАДНИХ
ДОСЛІДЖЕННЯХ
(ПТАР-2025)

27-29 травня 2025 року

Тернопіль
2025

Свірський Максим, Кочан Володимир	315
АРХІТЕКТУРА СИСТЕМИ РОЗУМНОЇ БІБЛІОТЕКИ НА БАЗІ ІoT І SDN	315
Трач Мар'ян, Осолінський Олександр	319
КОНЦЕПЦІЯ ПРОГРАМНО КЕРОВАНОЇ МОДУЛЬНОЇ АРХІТЕКТУРИ ІoT	319
Цанько Андрій, Осолінський Олександр	322
ІНТЕЛЕКТУАЛЬНА СИСТЕМА МОНІТОРИНГУ ПАРАМЕТРІВ МІКРОКЛІМАТУ	
ТЕПЛИЦІ НА БАЗІ ІoT	322
Циб Андрій	325
ІНТЕЛЕКТУАЛЬНА СИСТЕМА ВИЯВЛЕННЯ АТАК У СЕРЕДОВИЩІ ІНТЕРНЕТУ	
РЕЧЕЙ НА ОСНОВІ ТУМАННИХ ОБЧИСЛЕНЬ	325
СЕКЦІЯ 4. ІІІ В УПРАВЛІННІ ПРОЄКТАМИ	327
Бубнюк Надія, Черній Іван	327
ОГЛЯД ІНТЕЛЕКТУАЛЬНИХ МЕТОДІВ В УПРАВЛІННІ ПРОЄКТАМИ	327
Вередюк Тетяна, Дорош Віталій	330
АВТОМАТИЗАЦІЯ ПРОЦЕСІВ ПЛАНУВАННЯ ТА МОНІТОРИНГУ ПРОЄКТІВ З	
ВИКОРИСТАННЯМ ІШТУЧНОГО ІНТЕЛЕКТУ	330
Грицюк Іванна, Гладій Григорій	333
ІНТЕЛЕКТУАЛЬНИЙ МОДУЛЬ УПРАВЛІННЯ ПРОЦЕСАМИ ГУМАНІТАРНИХ	
МІСІЙ НА ОСНОВІ РСА ТА КЛАСТЕРИЗАЦІЇ	333
Демчук Максим, Лендюк Тарас	336
МОДУЛЬ КЛАСИФІКАЦІЇ КОМЕНТАРІВ REDDIT З ВИКОРИСТАННЯМ МОДЕЛІ	
BERT	336
Кузьмич Богдан, Биковий Павло	339
ПРОГРАМНИЙ МОДУЛЬ ПРОГНОЗУВАННЯ ПЕРЕШКОД У ЛАБІРИНТІ НА	
ОСНОВІ МАШИННОГО НАВЧАННЯ	339
Лимар Вікторія, Турченко Ірина	342
ІНФОРМАЦІЙНА ПАНЕЛЬ ЗГАДОК ДЛЯ ВІДСТЕЖЕННЯ ТА КЕРУВАННЯ	
КОМУНІКАЦІЯМИ ПРОГРАМНОГО СЕРЕДОВИЩА LRA	342
Лось Марія, Ліп'яніна-Гончаренко Христина	346
ВИКОРИСТАННЯ МОДЕЛІ CodeBERT ДЛЯ АНАЛІЗУ ТА ГЕНЕРАЦІЇ КОДУ В	
СУЧАСНОМУ ПРОГРАМУВАННІ	346
Магильницький Дмитро, Коваль Василь	349
МЕТОД ПОБУДОВИ ОПТИМАЛЬНИХ АРХІТЕКТУР НЕЙРОННИХ МЕРЕЖ	
ПРЯМОГО ПОШИРЕННЯ	349

Лимар Вікторія
студентка групи КН-42
lymarvikar@gmail.com

Турченко Ірина

к.т.н., доцент

itu@wunu.edu.ua

Західноукраїнський національний університет
Тернопіль, Україна

ІНФОРМАЦІЙНА ПАНЕЛЬ ЗГАДОК ДЛЯ ВІДСТЕЖЕННЯ ТА КЕРУВАННЯ КОМУНІКАЦІЯМИ ПРОГРАМНОГО СЕРЕДОВИЩА JIRA

У сучасних умовах бізнесу, коли компанії змушені справлятися з постійним збільшенням робочих навантажень і необхідністю ефективного управління процесами, команди прагнуть знайти найкращі інструменти для організації завдань, контролю виконання роботи та забезпечення ефективної взаємодії [1].

Jira – система від Atlassian, призначена для управління проектами, зокрема для організації завдань, відстеження їхнього прогресу та комунікації в командах тощо, яка містить власні інструменти штучного інтелекту. Однією з її корисних функцій є можливість згадування користувачів в завданнях через символ "@", що дозволяє оперативно залучати учасників до обговорення. Однак, у реальних умовах використання Jira виникає проблема зручного перегляду списку згадок і вона не є локальною або поодиноким. Це системне обмеження, на яке регулярно звертають увагу користувачі на Atlassian Community [2]. Пошук за ключовими словами "mentions list" повертає десятки обговорень, де користувачі скаржаться на відсутність централізованого перегляду, обмежену фільтрацію та неможливість зручно відстежувати, коли і де їх згадали. Ці звернення свідчать про наявність реального запиту на покращення механізмів моніторингу згадувань.

Таким чином, актуальним є створення спеціалізованого інструмента для зручного моніторингу згадувань у Jira, який забезпечуватиме централізований перегляд, фільтрацію, пошук і швидку реакцію на комунікації.

Хоча Jira має базову систему сповіщень, яка включає інтерфейс сповіщень та email-інформування. Проте із вищезгаданих запитів від користувачів вона не дозволяє точно фільтрувати згадки, поєднує їх з усіма іншими типами оновлень і часто призводить до перевантаження користувача не корисною для нього інформацією.

Окрім базової системи сповіщень у Jira існують і сторонні додатки, які намагаються частково вирішити проблему з управлінням згадуваннями. Зокрема, User Mentions Gadget for Jira [3] дозволяє переглядати згадування користувача у форматі окремої панелі, підтримує фільтрацію за обраною особою та перехід до

оригінального коментаря. Втім, його функціональність є обмеженою через неможливість побачити згадування, створені самим користувачем, а також бракує можливості відповіді на коментарі безпосередньо з панелі.

Ще одним прикладом є Threaded Comments for Jira [4], який структурує коментарі у вигляді ланцюжків, покращуючи читабельність комунікацій. Проте цей додаток не фокусується саме на згадуваннях, і не дозволяє їх централізовано переглядати чи фільтрувати. Загалом, жодне з доступних рішень не пропонує повноцінного централізованого механізму для зручного моніторингу, фільтрації та взаємодії зі згадуваннями – що й стало поштовхом для розробки нового інструмента.

Для вирішення вказаних проблем було розроблено інформаційну панель Mentions Dashboard for Jira, що є вбудованим додатком до хмарної Jira та реалізована на платформі Atlassian Forge з використанням React та Atlassian Design System.

Основні функціональні модулі панелі:

а) налаштування вибору джерел згадувань (коментарі, контрольні списки), фільтрація за автором та списком згаданих користувачів;

б) обробка даних, інтеграція з API Jira, пошук згадувань за допомогою регулярних виразів та JQL, фільтрація і сортування результатів;

в) візуалізація виводу у вигляді карток з контекстом, зручна нижня навігація, інтерактивна відповідь на згадування без переходу до задачі.

Основні етапи роботи з інформаційною панеллю згадок можна сформулювати наступним чином:

а) користувач додає віджети на інформаційну панель та задає необхідні фільтри, зокрема вибір джерела згадування (коментарі чи контрольні списки), визначення автора згадувань (поточний користувач або будь-хто), а також вибір користувачів, які були згадані у відповідних активностях згідно з заданими параметрами;

б) для оптимізації пошуку система автоматично формує відповідні JQL-запити на основі заданих фільтрів – це дозволяє ефективно витягувати лише ті задачі, в яких згадування дійсно присутні, та уникати надмірного навантаження;

в) після отримання початкових даних система аналізує текстовий вміст кожного коментаря чи елемента контрольного списку, визначаючи наявність згадувань за допомогою пошуку по символу “@” та ідентифікатору користувача;

г) виявлені згадування проходять додаткову фільтрацію відповідно до заданих параметрів;

д) відображення згадувань здійснюється у вигляді карток зі зручним інтерфейсом, де користувач може ознайомитися з контекстом згадки, побачити автора, час створення, уривок коментаря або пункту контрольного списку.

Нижче на рисунку 1.1 зображено загальну схему алгоритму роботи користувача із панеллю.

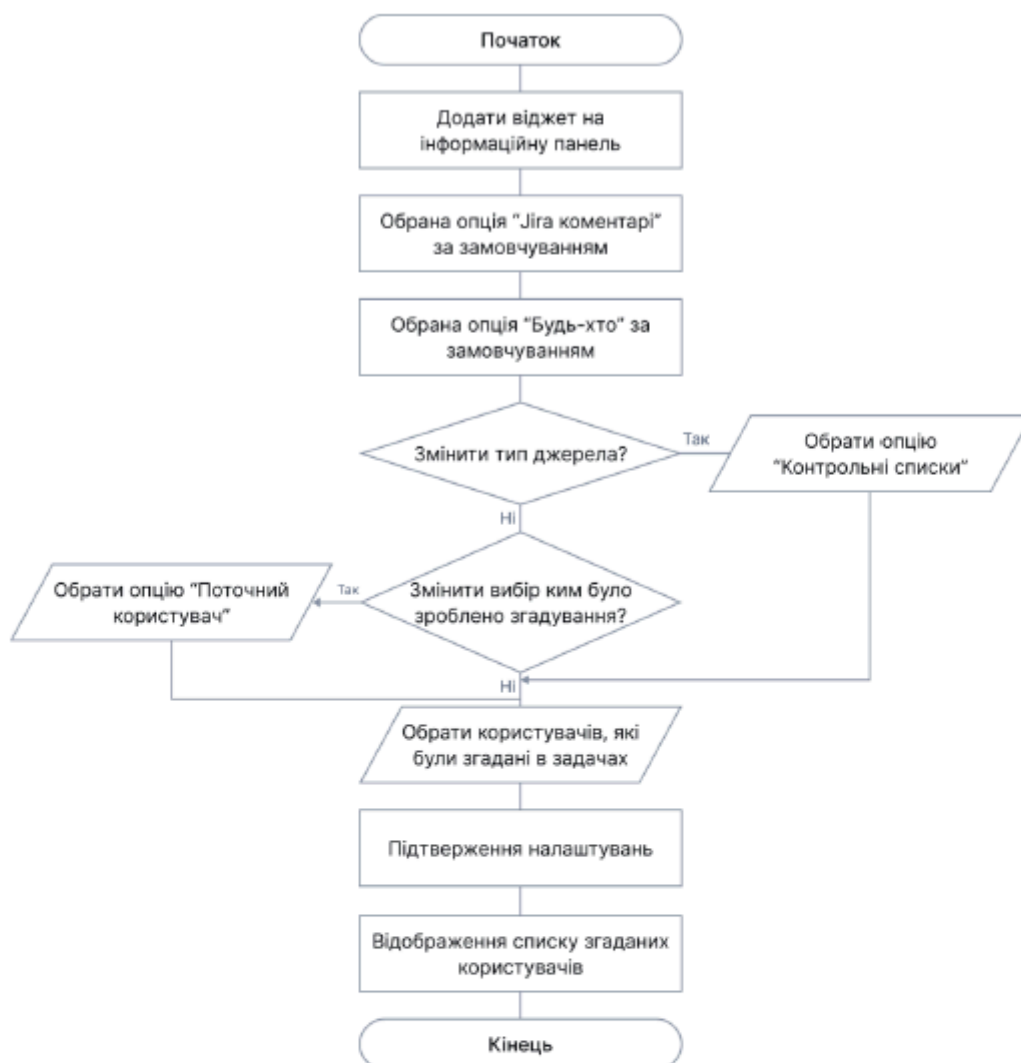


Рисунок 1 – Схема алгоритму роботи інформаційної панелі для моніторингу згадувань

Ще однією особливістю реалізації є можливість миттєвої відповіді без переходу до оригінальної задачі прямо з панелі. Це реалізовано через вбудовану форму відповіді, яка автоматично додає коментар у відповідне завдання через API Jira, забезпечуючи безперервний робочий процес без необхідності перемикання між вікнами.

Також приділено особливу увагу інтеграції з партнерським додатком – Checklists for Jira (Pro) by HeroCoders [5], згадування з якого також включаються до панелі, що дозволяє відображати повну картину комунікацій у проєктах. Такий підхід забезпечує уніфіковане середовище відстеження згадувань незалежно від їхнього джерела.

Розроблена інформаційна панель “Mentions Dashboard for Jira” вирішує нагальну проблему сучасного проєктного менеджменту – відсутність прозорого та зручного моніторингу згадувань. Завдяки інтеграції з екосистемою Jira, підтримці кількох джерел згадувань, можливості налаштування фільтрів і швидкого реагування, система значно підвищує ефективність командної взаємодії.

Практична цінність цього рішення проявляється в таких аспектах:

- а) зменшення ризику пропущених повідомлень і кінцевих термінів;
- б) підвищення персональної відповідальності та залученості членів команди;
- в) спрощення пошуку та навігації у великій кількості задач;
- г) можливість масштабування та розширення інтеграцій із іншими додатками.

Розроблена інформаційна панель згадок може бути ефективно використана менеджерами проєктів, аналітиками, розробниками, тестувальниками, які щоденно працюють із Jira. Воно також має потенціал розвитку як інтеграційна платформа для взаємодії зі сторонніми сервісами, що відкриває нові можливості для покращення корпоративної комунікації.

Список використаних джерел

1. Muszyńska K. Measuring communication management performance in virtual project teams // *Procedia Computer Science*. – 2022. – Vol. 207. – P. 3507–3515. – DOI: 10.1016/j.procs.2022.09.409.
2. Mentions list – Atlassian Community [Електронний ресурс]. – URL: https://community.atlassian.com/forums/forums/searchpage/tab/message?advanced=false&allow_punctuation=false&q=mentions+list.
3. User Mentions Gadget for Jira | Atlassian Marketplace. URL: <https://marketplace.atlassian.com/apps/1229393/user-mentions-gadget-for-jira?tab=overview&hosting=cloud>.
4. Threaded Comments for Jira Cloud. URL: <https://docs.votazz.co/threaded-comments-for-jira-cloud/>.
5. Checklists for Jira (Pro) by HeroCoders | Atlassian Marketplace. URL: <https://marketplace.atlassian.com/apps/1213231/checklists-for-jira-pro-by-herocoders?hosting=cloud&tab=overview>.