**Лабораторна робота №1**

**Мета:**Навчитися збирати, аналізувати, валідувати, документувати та погоджувати вимоги до ПЗ; створювати SRS; сформувати Product Backlog і базовий процес виконання (Kanban/Scrum).

**Завдання 1. Discovery Phase (збір вимог)**

1. Сформулювати бачення продукту: цілі бізнесу, межі (Scope/Out of Scope), припущення.
2. Визначити всіх **stakeholders**
3. Обрати та обґрунтувати методи елесітації (інтерв’ю, воркшоп, опитування, спостереження, аналіз документів, прототипування, JAD тощо).
4. Скласти список первинних вимог (мін 20 вимог)

**Завдання 2. Аналіз і декомпозиція вимог**

1. Класифікувати вимоги
2. Скласти каталог вимог з атрибутами
3. Декомпозувати вимоги **Epic → Feature → User Story**.
   * User Story у форматі: As a [persona] I want [goal] so that [value].
4. Нормалізувати вимоги: усунення дублів, атомарність, трасованість, узгодженість термінів (за потреби)
5. Для функціональних вимог створіть Use Case Diagram

**Завдання 3. Валідація та верифікація (V&V)**

1. Перевірити вимоги за критеріями: повнота, коректність, узгодженість, однозначність, здійсненність, необхідність, тестованість, трасованість.
2. Для кожної User Story сформувати Acceptance Criteria

**Завдання 4. Approval і документація**

1. Модель SCRUM-команди + RACI-matrix

* Для визначених вимог створіть команду для розробки ПП.

1. **Product Backlog**

* Створити **Product Backlog** на основі вимог, які були визначенні в попередніх пунктах.
* Таблиця (рекомендовано): ID, Title, User Story, Priority (MoSCoW), Value (1–5), Effort (SP), AC і тд.

1. **Kanban-дошка** для **Product Backlog**

* Колонки (рекомендовано): Backlog → Ready → In Progress → Review → Testing → Done.

1. Планування спринтів і дошки

* Оберіть довжину спринту (напр., 2 тижні).
* Сплануйте мінімум 3 спринти: визначіть **Sprint Goal**, оберіть необхідні **User Story** з **Product Backlog**.
* Для кожного спринту створіть окрему **Kanban-дошку**
* Створити діаграму **Ganat**

1. Software Requirements Specification (SRS)

На основі всіх вище пройденних завдань створіть SRS за зразком)

**Хід роботи:**

Продокументувати розробку ПЗ а також SRS

**Варіант №19.** **Перевірка контенту: факти й плагіат - цитування, ризики, подібність**

**Завдання 1. Discovery Phase (збір вимог)**

1. **Бачення продукту:** Вся інформація має бути перевірена на достовірність та унікальність. Джерела інформації потрібно правильно цитувати, щоб не було плагіату та юридичних ризиків. Контент має відображати персональне бачення людини і функціональність продукту, а не копірайт.

**Цілі бізнесу:**

* Створити продукт який значно вирішить проблеми копірайтингу.
* Забезпечити оригінальність та унікальність контенту.
* Підвищити ефективність роботи копірайтерів, еконмолячи час на редагування контенту.
* Сформувати унікальне бачення продукту, яке буде відрізняти його від схожих проектів.
* Залучити людей, особливо тих, хто багато працює з текстом, воно облегшить їм життя.

**Характеристика кінцевих користувачів:**

* **Копірайтери:** професійні автори текстів (статті, блоги, рекламні матеріали), які перевіряють унікальність та якість тексту для економії часу та підвищення продуктивності.
* **Маркетологи:** спеціалісти з реклами та PR, які використовують систему для перевірки достовірності фактів і уникнення репутаційних ризиків у рекламному контенті.
* **Користувачі-початківці:** студенти, блогери-початківці, аматори, які прагнуть навчитися писати грамотно та користуються підсвіткою помилок і рекомендаціями.
* **Бізнес-власники:** підприємці та власники контент-платформ, які зацікавлені у створенні якісного, унікального контенту для розвитку бренду.
* **Адміністратори системи або розробники:** технічні спеціалісти, які відповідають за стабільність роботи, логування, продуктивність та масштабованість системи.

**Межі:**

**Scope:**

* Перевірка контенту на факти, плагіат а також оригінальність.
* Надання звітів та оцінки ризиків плагіату.
* Генерація рекомендацій для покращення унікальності інформації.
* Підтримка статей, рекламних постів.

**Out of Scope:**

* Робота з зображеннями, відео, аудіо.
* Створення повністю готового документу без участі користувача.
* Підтримка інших форматів, такі як .pdf, .pptx.
* Наукові статті мають бути перевірені на факти з гарантією на 100%.

**Припущення:**

Припустим, що користувачі будуть активно використовувати програму для перевірки унікальності та якості текстів. Тексти, що перевіряються, знаходяться у форматах, яка підтримує програма. Система не гарантує 100% точність перевірки фактів, але зменшує ризики плагіату.

1. **Stakeholders:**

* Копірайтери – це будуть наші основні користувачі, яким наш продукт буде допомагати створювати якісний контент.
* Маркетологи – будуть використовувати продукт для контролю контенту і зниженню ризиків плагіату.
* Користувачі-початківці – це ті, хто хоче швидше створювати тексти та перевіряти їх на оригінальність без досвіду в копірайтингу.
* Розробники – відповідатимуть за функціонал, стабільність а також безпеку продукту.
* Власники бізнесу – це люди які зацікавлені в прибутковості, в впізнаваності бренду. Впливають на розвиток продукту.

1. **Методи елесітації**

* Проводимо співбесіди з копірайтерами, маркетологами для глибшого розуміння їхніх потреб, проблем а також очікувань від продукту.
* Також проводимо опитування для того, щоб швидко зібрати думки великої кількості користувачів щодо оцінки унікальності контенту та функцій продукту.
* Спостерігаємо за робочими процесами копірайтерів, щоб визначити де можна підвищити ефективність роботи продукту.
* Аналізуємо вже існуючі матеріали, контент-плани, технічні завдання та політики щодо плагіату, щоб точно визначити вимоги та стандарти продукту.
* Організовуємо спільні сесії із ключовими stakeholders для узгодження вимог.
* Створюємо швидкі макети інтерфейсу та функціонал, щоб отримати реакцію користувачів до розробки продукту.

1. **Первинні вимоги**

* Аналіз достовірності фактів.
* Перевірка унікальності контенту та виявлення плагіату.
* Зручний інтерфейс для користувачів.
* Генерування рекомендацій щодо покращення структури контенту.
* Надання детальних звітів про перевірку контенту.
* Підтримка основних текстових форматів такі як: статті, рекламний контент…
* Конфіденційність інформації користувачів.
* Можливість збереження та експорт результатів.
* Підтримка багатомовності.
* Логування дій користувача для покращення продукту.
* Підсвітка помилок в контенті.
* Підтримка масштабного навантаження без зниження швидкості роботи продукту.
* Можливість відстеження змін у контенті.
* Вбудований словник термінів.
* Повідомлення про критичні помилки у контенті.
* Можливість порівняння декількох версій контенту між собою.
* Можливість налаштування рівня перевірки унікальності, тобто швидка або детальна.
* Фільтрація результатів перевірки за відповідними категоріями.
* Функція виявлення дублікатів у внутрішній базі текстів користувача
* Автоматична перевірка орфографії, граматики, пунктуації.

**Завдання 2. Аналіз і декомпозиція вимог**

|  |  |
| --- | --- |
| **Тип вимог** | **Опис** |
| **Функціональні** | Перевірка унікальності контенту та виявлення плагіату |
| Надання звітів про перевірку контенту |
| Генерування рекомендацій щодо покращення контенту |
| Автоматична перевірка орфографії, граматики, пунктуації |
| Відстеження змін у контенті |
| Підсвітка помилок у тексті |
| Аналіз достовірності фактів |
| Підтримка основних текстових форматів (статті, рекламний контент тощо) |
| Порівняння кількох версій контенту |
| Фільтрація результатів перевірки за категоріями |
| Виявлення дублікатів у внутрішній базі текстів користувача |
| Налаштування рівня перевірки унікальності (швидка/детальна) |
| Вбудований словник термінів |
| Повідомлення про критичні помилки у контенті |
| Можливість збереження та експорту результатів |
| Підтримка багатомовності |
| Логування дій користувача |
| **Нефункціональні** | Зручний інтерфейс для користувачів |
| Конфіденційність інформації користувачів |
| Підтримка масштабного навантаження без зниження швидкодії |

1. **Декомпозиція вимог**

**Epic 1: Перевірка контенту на унікальність та якість**

**Feature 1: Перевірка контенту на плагіат.**

**Story A:** *As a* ***copywriter****, I want to check my text for plagiarism so that I can ensure originality.*

**Story B:** *As a* ***marketer****, I want to detect plagiarism in ads so that I avoid reputational risks.*

**Feature 2: Автоматична перевірка орфографії, пунктуації та граматики.**

**Story A:** *As a* ***copywriter****, I want to check punctuation automatically so that I don’t waste time on manual corrections.*

**Story B:** *As a* ***novice user****, I want to see grammar and spelling mistakes highlighted so that I can quickly fix them.*

**Feature 3: Аналіз достовірності фактів**

**Story A:** *As a* ***marketer****, I want to verify facts in content so that I don’t publish false information.*

**Feature 4: Виявлення дублікатів у внутрішній базі користувачів**

**Story A:** *As a* ***user****, I want to detect duplicates in my text database so that I can avoid reusing the same content.*

**Epic 2: Робота з результатами перевірки**

**Feature 5: Надання звітів про перевірку контенту**

**Story A:** *As a* ***copywriter****, I want to get a detailed report about plagiarism so that I understand weak points in my text.*

**Story B:** *As a* ***marketer****, I want risk-level reports so that I can evaluate legal and reputational threats.*

**Feature 6: Генерація рекомендацій для покращення унікальності**

**Story A:** *As a* ***novice user****, I want to get recommendations for improving my content so that I can learn better writing.*

**Story B:** *As a* ***copywriter****, I want the system to suggest rephrasing sentences so that my text is more unique.*

**Feature 7: Порівняння версій контенту**

**Story A:** *As a* ***user****, I want to compare two text versions so that I can see what has improved.*

**Feature 8: Фільтрація результатів перевірки**

**Story A:** *As a* ***copywriter****, I want to filter plagiarism results by source type so that I can focus on relevant risks.*

**Epic 3: Інтерфейс та продуктивність системи**

**Feature 9: Зручний інтерфейс**

**Story A:** *As a* ***novice user****, I want a simple interface so that I can easily start using the product*.

**Story B:** *As a* ***copywriter****, I want a clean UI with highlighted mistakes so that I can quickly edit content.*

**Feature 10: Збереження та експорт результатів у файл**

**Story A:** *As a* ***marketer****, I want to export reports in .docx so that I can share them with the team.*

**Story B:** *As a* ***copywriter****, I want to save previous checks so that I can track progress over time.*

**Feature 11: Підтримка багатомовності**

**Story A:** *As a* ***user****, I want to use the system in English and Ukrainian so that I can work with different projects.*

**Feature 12: Масштабованість продукту**

**Story A:** *As a* ***business owner****, I want the system to handle high loads so that many users can work simultaneously.*

**Story B:** *As a* ***developer****, I want logs of user actions so that I can improve system performance.*

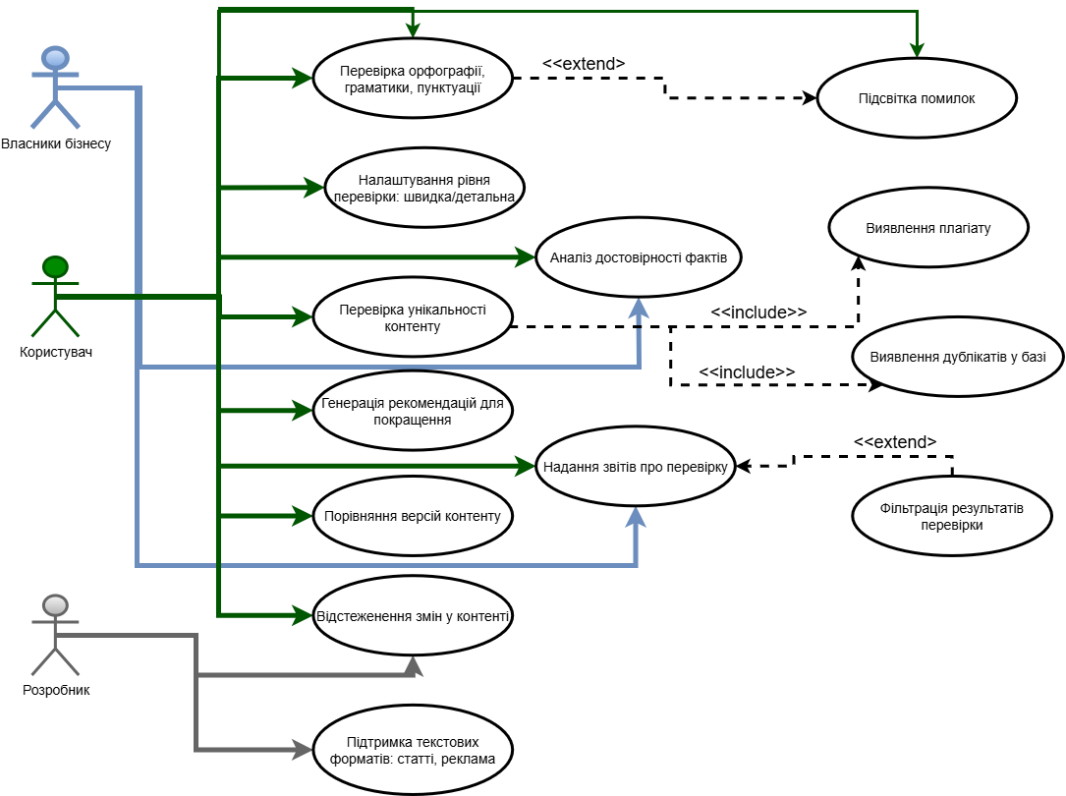
**Feature 13:****Історія змін та відкат змін**

**Story A:** *As a user, I want to restore previous text versions so that I can undo unwanted edits.*

1. **Каталог вимог з атрибутами**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epic | Feature | User Story | Priority | Effort (SP) | Value |
| Перевірка контенту на унікальність та якість | Перевірка контенту на плагіат | As a copywriter, I want to check my text for plagiarism so that I can ensure originality. | Must | 8 | High |
| Перевірка контенту на унікальність та якість | Перевірка контенту на плагіат | As a marketer, I want to detect plagiarism in ads so that I avoid reputational risks. | Must | 13 | High |
| Перевірка контенту на унікальність та якість | Автоматична перевірка орфографії, пунктуації та граматики | As a copywriter, I want to check punctuation automatically so that I don’t waste time on manual corrections. | Should | 5 | Medium |
| Перевірка контенту на унікальність та якість | Автоматична перевірка орфографії, пунктуації та граматики | As a novice user, I want to see grammar and spelling mistakes highlighted so that I can quickly fix them. | Should | 5 | Medium |
| Перевірка контенту на унікальність та якість | Аналіз достовірності фактів | As a marketer, I want to verify facts in content so that I don’t publish false information. | Should | 13 | Medium |
| Перевірка контенту на унікальність та якість | Виявлення дублікатів у внутрішній базі користувачів | As a user, I want to detect duplicates in my text database so that I can avoid reusing the same content. | Must | 8 | High |
| Робота з результатами перевірки | Надання звітів про перевірку контенту | As a copywriter, I want to get a detailed report about plagiarism so that I understand weak points in my text. | Must | 5 | High |
| Робота з результатами перевірки | Надання звітів про перевірку контенту | As a marketer, I want risk-level reports so that I can evaluate legal and reputational threats. | Should | 3 | Medium |
| Робота з результатами перевірки | Генерація рекомендацій для покращення унікальності | As a novice user, I want to get recommendations for improving my content so that I can learn better writing. | Could | 3 | Medium |
| Робота з результатами перевірки | Генерація рекомендацій для покращення унікальності | As a copywriter, I want the system to suggest rephrasing sentences so that my text is more unique. | Must | 5 | High |
| Робота з результатами перевірки | Порівняння версій контенту | As a user, I want to compare two text versions so that I can see what has improved. | Should | 4 | Medium |
| Робота з результатами перевірки | Фільтрація результатів перевірки | As a copywriter, I want to filter plagiarism results by source type so that I can focus on relevant risks. | Could | 3 | Medium |
| Інтерфейс та продуктивність системи | Зручний інтерфейс | As a novice user, I want a simple interface so that I can easily start using the product. | Must | 3 | High |
| Інтерфейс та продуктивність системи | Зручний інтерфейс | As a copywriter, I want a clean UI with highlighted mistakes so that I can quickly edit content. | Must | 5 | High |
| Інтерфейс та продуктивність системи | Збереження та експорт результатів у файл | As a marketer, I want to export reports in .docx so that I can share them with the team. | Should | 3 | Medium |
| Інтерфейс та продуктивність системи | Збереження та експорт результатів у файл | As a copywriter, I want to save previous checks so that I can track progress over time. | Must | 4 | High |
| Інтерфейс та продуктивність системи | Підтримка багатомовності | As a user, I want to use the system in English and Ukrainian so that I can work with different projects. | Could | 4 | Medium |
| Інтерфейс та продуктивність системи | Масштабованість продукту | As a business owner, I want the system to handle high loads so that many users can work simultaneously. | Must | 5 | High |
| Інтерфейс та продуктивність системи | Масштабованість продукту | As a developer, I want logs of user actions so that I can improve system performance. | Should | 4 | Medium |
| Інтерфейс та продуктивність системи | Історія змін та відкат змін | As a user, I want to restore previous text versions so that I can undo unwanted edits. | Must | 4 | High |

1. **Use of** **Case Diagram**

****

**Завдання 3. Валідація та верифікація (V&V)**

**Перевірка вимог за критеріями:**

1. **Повнота**

**Функціональні вимоги** охоплюють усі основні функції системи перевірки контенту такі як: унікальність, плагіат, рекомендації, перевірка орфографії і т.д.

У **нефункціональних вимогах** описані UI/UX, безпека користувачів, продуктивність, повідомлення про наявність помилок а також збереження або експорт контенту у відповідний файл.

**Технічні вимоги**, усі базові аспекти продукту враховані.

Отже, повнота повністю забезпечена.

1. **Коректність**

Усі вимоги повністю пов’язані з цілями нашого продукту тобто, забезпечення унікальності, ефективність роботи та допомога копірайтерам.

Зробимо висновок, коректність продукту на вищому рівні.

1. **Узгодженість**

У нас деякі терміни постійно повторюються такі як, “унікальність” та “оригінальність”.

Комбіновані вимоги були поділені, це були “перевірка орфографії/граматики/пунктуації” на окремі вимоги.

Виходячи з цього, можемо сказати що в нас все узгоджено.

1. **Однозначність**

Більшість вимог є чіткі та зрозумілі для нашого продукту, але деякі можуть потребувати уточнень. Це такі як: 1. “Можливість відстеження змін у контенті”, 2. “Фільтрація результатів за категоріями” 3. “Підтримка масштабного навантаження”.

Потрібно ще деталізувати дані вимоги, щоб все було чітко поставлено і виконано.

1. **Здійсненість**

В наших вимогах, функції реалізовані сучасними технологіями, окрім стовідсоткової точності перевірки фактів.

Отже, здійсненість в нас є, але нам варто враховувати ресурси.

1. **Необхідність**

Кожна з функцій спрямована на досягнення потреб користувачів а також цілей бізнесу, зайвих функцій немає.

Усі вимоги необхідні.

1. **Тестованість**

Більшість з вимог тестуються. Вимоги такі як: “Перевірка унікальності” “Генерація звітів” “Підсвітка помилок” тестуються.

Перевірка унікальності потребує автоматичних тестів на різний контент.

Генерація звітів потребує перевірки формату та наявності даних.

Підсвітка помилок потребує перевірки, що помилки виділятимуться коректно.

Також потрібно визначити метрики для тестування нефункціональних вимог (наприклад, продуктивність та масштабованість).

1. **Трасованість**

Вимоги можна пов’язати з бізнес-цілями та stakeholders.

* Забезпечення унікальності → перевірка контенту, фільтрація та виявлення плагіату.
* Підвищення ефективності копірайтерів → підсвітка помилок, рекомендації та перевірка орфографії/граматики.
* Залучення користувачів → зручний та зрозумілий інтерфейс, а також багатомовність.
* Стабільність та безпека користувачів → масштабованість, конфіденційність а також логування.

Можемо зробити висновок, що трасованість забезпечена, можна легко показати зв’язок вимог з бізнес-цілями та stakeholders.

**Визначаємо Acceptance Criteria для кожної User Story.**

**User Story:** “*As a* ***copywriter****, I want to check my text for plagiarism so that I can ensure originality.”*

**AC 1:** The system analyzes the entered text and returns a uniqueness value.

**AC 2:** Fragments of plagiarism will be highlighted.

**AC 3:** Sources of plagiarized fragments will be displayed in the result.

**User Story:** *“As a marketer, I want to detect plagiarism in ads so that I avoid reputational risks.”*

**AC 1:**The system will check advertising content for plagiarism.

**AC 2:** The results will be grouped by risk level, i.e. low, medium, high.

**AC 3:** You will be able to export a report for the team.

**User Story:** “*As a copywriter, I want to check punctuation automatically so that I don’t waste time on manual corrections.*”

**AC 1:** The system checks for all punctuation errors..

**AC 2:** The user will see highlighted errors.

**AC 3:** Automatic corrections will be offered.

**User Story:** “*As a novice user, I want to see grammar and spelling mistakes highlighted so that I can quickly fix them.*”

**AC 1:** All grammatical and spelling errors will be highlighted in a different color.

**AC 2:** The system offers a correction option.

**AC 3:** The highlight will update after editing the text..

**User Story:** “*As a marketer, I want to verify facts in content so that I don’t publish false information.*”

**AC 1:** The system checks key facts.

**AC 2:** False and questionable facts will be highlighted.

**AC 3:** The user will receive links to sources to verify the information.

**User Story:** *“As a user, I want to detect duplicates in my text database so that I can avoid reusing the same content.”*

**AC 1:** The system checks the new text for matches in the user's internal database..

**AC 2:** Duplicate fragments will be displayed with a link to the source from which it was taken..

**AC 3:** The user will be able to filter duplicates by time or another category.

**User Story:** “*As a copywriter, I want to get a detailed report about plagiarism so that I understand weak points in my text.”*

**AC 1:** The report can be exported in .docx or .pdf format.

**AC 2:** Sources and percentage of repetitions are indicated..

**AC 3:** Reports are generated with plagiarism highlights.

**User Story:** “*As a marketer, I want risk-level reports so that I can evaluate legal and reputational threats.*”

**AC 1:** Risk levels will be automatically assigned to any text.

**AC 2:** Reports will contain recommendations for risk reduction.

**AC 3:** You can filter content by risk level.

**User Story:** *“As a novice user, I want to get recommendations for improving my content so that I can learn better writing.”*

**AC 1:** The system analyzes the content and suggests improvements to the structure.

**AC 2:** Each recommendation is explained briefly, clearly and clearly.

**AC 3:** The user can, at their own discretion, apply recommendations to improve content, or may deviate from them..

**User Story: *“****As a copywriter, I want the system to suggest rephrasing sentences so that my text is more unique.****”***

**AC 1:** User can change the phrase with a single click.

**AC 2:** The system detects repeated phrases and suggests an alternative.

**AC 3:** Suggestions do not change the essence of the content.

**User Story:** “*As a user, I want to compare two text versions so that I can see what has improved*.”

**AC 1:** Any changes in content are highlighted.

**AC 2:** Two versions of the text will be displayed side by side.

**AC 3:** You can export the comparison in the report.

**User Story:** *“As a copywriter, I want to filter plagiarism results by source type so that I can focus on relevant risks.”*

**AC 1:**The system saves the last filter so that it can be reused..

**AC 2:** The user will be able to choose source categories (online, internal databases, libraries).

**AC 3:** The result will be updated depending on the selected filters..

**User Story:** “*As a novice user, I want a simple interface so that I can easily start using the product.”*

**AC 1:** No complicated settings.

**AC 2:** The user will receive a prompt for each action.

**AC 3:** Basic functions are available on the home screen.

**User Story:***“As a copywriter, I want a clean UI with highlighted mistakes so that I can quickly edit content.”*

**AC 1:** All errors will be highlighted in the appropriate color.

**AC 2:** The interface is not overloaded with unnecessary elements.

**AC 3:** Ability to edit text without going to other screens.

**User Story:** “*As a marketer, I want to export reports in .docx so that I can share them with the team.”*

**AC 1:** The report is generated in .docx..

**AC 2:** Formatting remains readable.

**AC 3:** You can choose which sections to include in the report.

**User Story:** “*As a copywriter, I want to save previous checks so that I can track progress over time.*”

**AC 1:** The user will be able to view the history of checks.

**AC 2:** You can rerun the check of past texts.

**AC 3:** Each check stores the date and time.

**User Story:** *“As a user, I want to use the system in English and Ukrainian so that I can work with different projects.”*

**AC 1:** You can choose the interface language.

**AC 2:** Text validation only works for the selected language.

**AC 3:** Messages and prompts will be displayed in the selected language.

**User Story:** *“As a business owner, I want the system to handle high loads so that many users can work simultaneously.”*

**AC 1:** The system can handle a given number of simultaneous users.

**AC 2:** No errors occur under high load.

**AC 3:** Response time does not exceed the acceptable threshold.

**User Story:** “*As a developer, I want logs of user actions so that I can improve system performance.”*

**AC 1:** All user actions are logged..

**AC 2:** Logs contain date, time, user, and action type.

**AC 3:** You can analyze logs to improve performance.

**User Story**: “As a user, I want to restore previous text versions so that I can undo unwanted edits.”

**AC 1:** The system stores a history of changes with date and time..

**AC 2:** The user can view and select any previous version of the text.

**AC 3:** The user can restore the selected version and it becomes current for further editing.

**Завдання 4. Approval і документація.**

**Scrum Team:**

* **Product Owner** (Business Analyst + Product Manager) – визначає вимоги, пріоритети та функціонал
* **Scrum Master** – слідкує за процесом, усуває перешкоди
* **Backend-розробники (2 людини)** – реалізація алгоритмів пошуку плагіату, роботи з API, обробки тексту
* **Frontend-розробник (1 людина)** – створення інтерфейсу для користувачів (завантаження текстів, перегляд звітів)
* **QA Engineers (2 людини)** – тестування перевірки на плагіат, коректності цитувань, продуктивності
* **Data Scientist / NLP Engineer (1 людина)** – розробка алгоритмів обробки природної мови (аналіз тексту, пошук подібності, перевірка фактів)
* **DevOps Engineer (1 людина)** – CI/CD, сервери, масштабування перевірок
* **UI/UX Designer (1 людина)** – зручність користування, дизайн звітів і візуалізацій

**RACI-matrix:**

**R (Responsible)** – виконує роботу або безпосередньо взаємодіє з Feature.

**A (Accountable)** – відповідальний за прийняття рішення

**C (Consulted)** – радить, надає інформацію

**I (Informed)** – просто інформовані про результат

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature/Action** | **Copywriter** | **Marketer** | **Novice user** | **Developer** |
| Перевірка контенту на плагіат | **R** | **R** | **I** | **C** |
| Автоматична перевірка орфографії, пунктуації та граматики | **R** | **I** | **R** | **C** |
| Аналіз достовірності фактів | **I** | **R** | **I** | **C** |
| Виявлення дублікатів у внутрішній базі | **I** | **I** | **R** | **C** |
| Надання звітів про перевірку контенту | **R** | **R** | **I** | **C** |
| Генерація рекомендацій для покращення унікальності | **R** | **I** | **R** | **C** |
| Порівняння версій контенту | **I** | **I** | **R** | **C** |
| Фільтрація результатів перевірки | **R** | **I** | **I** | **C** |
| Зручний інтерфейс | **R** | **I** | **R** | **C** |
| Збереження та експорт результатів у файл | **R** | **R** | **I** | **C** |
| Підтримка багатомовності | **R** | **I** | **R** | **C** |
| Масштабованість продукту | **I** | **I** | **I** | **C** |
| Історія змін та відкат змін | **I** | **I** | **R** | **C** |

**Product Backlog + Kanban-board**

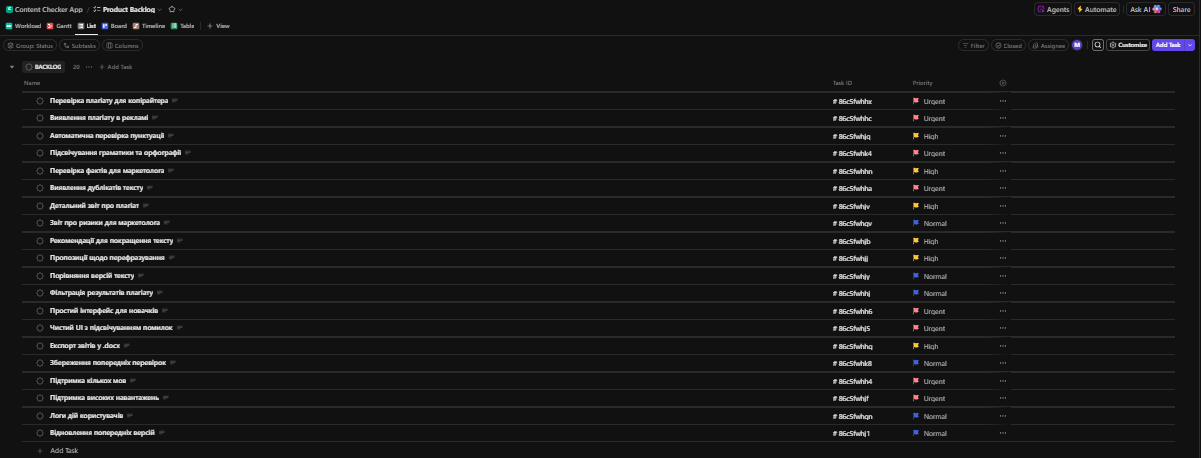
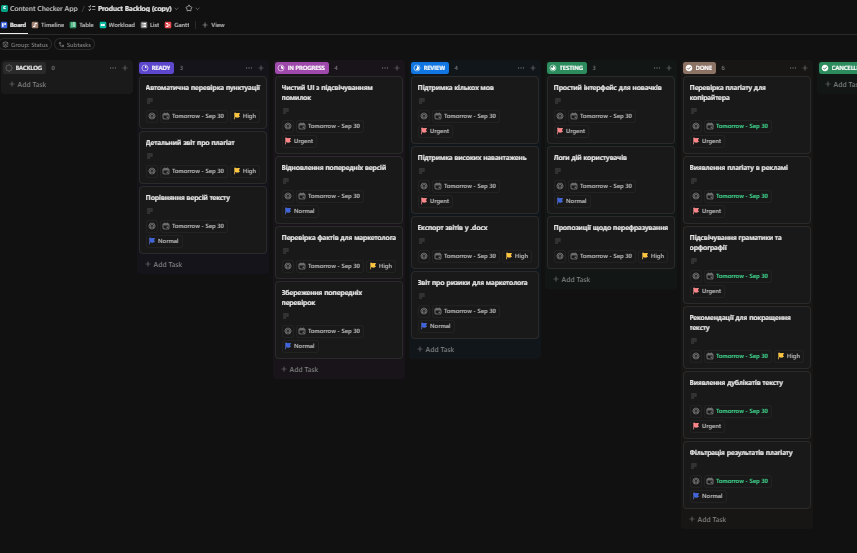
****

Рисунок 1 – Product Backlog (ClickUp)

Рисунок 2 – Kanban-dashboard Product Backlog (ClickUp)

**Scrum Sprints**

**Sprint 1 Plan**

**Sprint Goal**

Deliver a working slice of the **Content Checker App** covering rephrasing suggestions, punctuation checking, fact-checking module and initial plagiarism reporting.

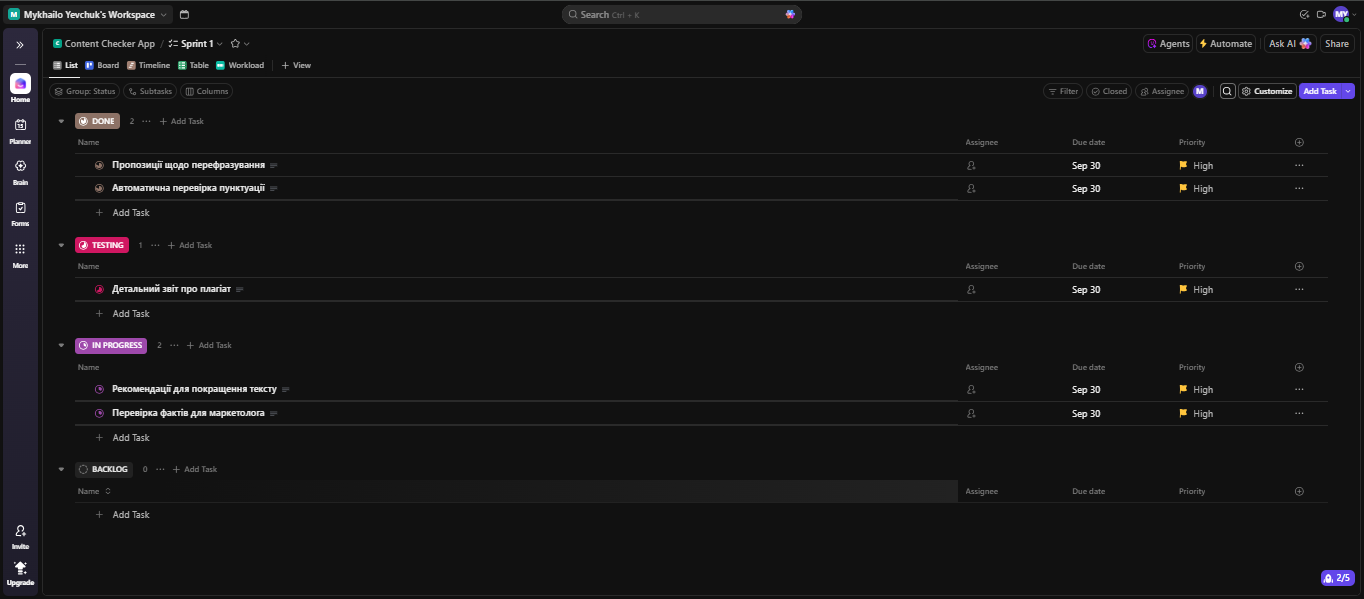
**Scope (User Stories in Sprint 1):**

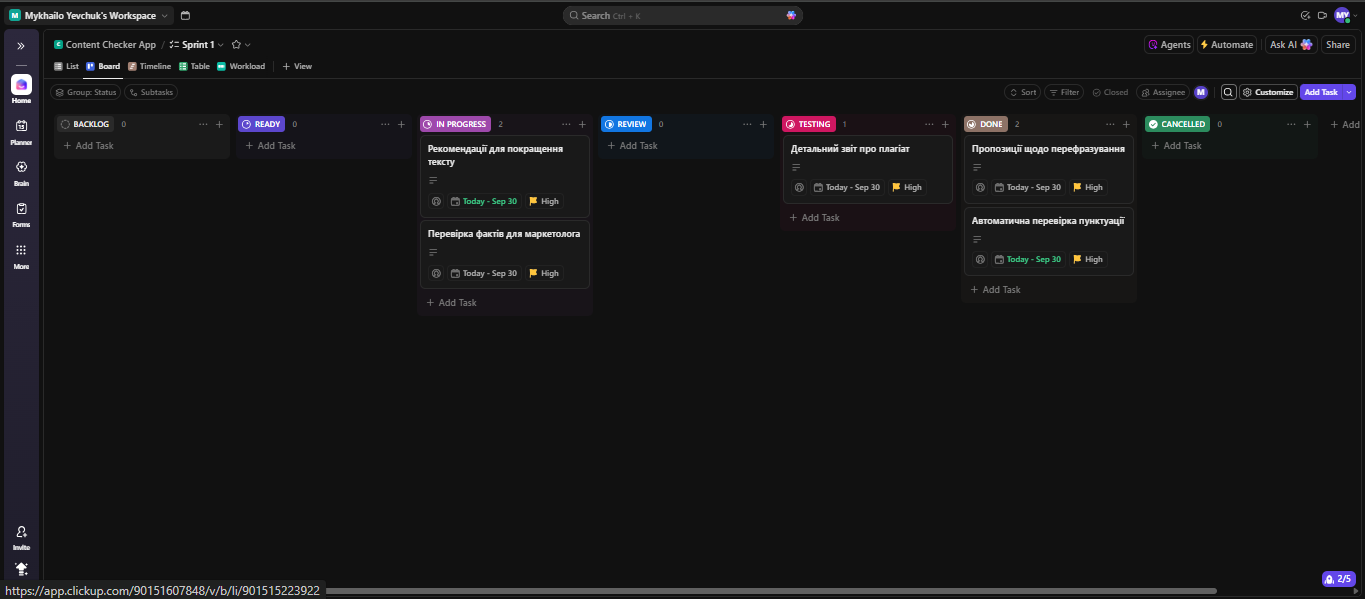
**Epic: Text Improvement**

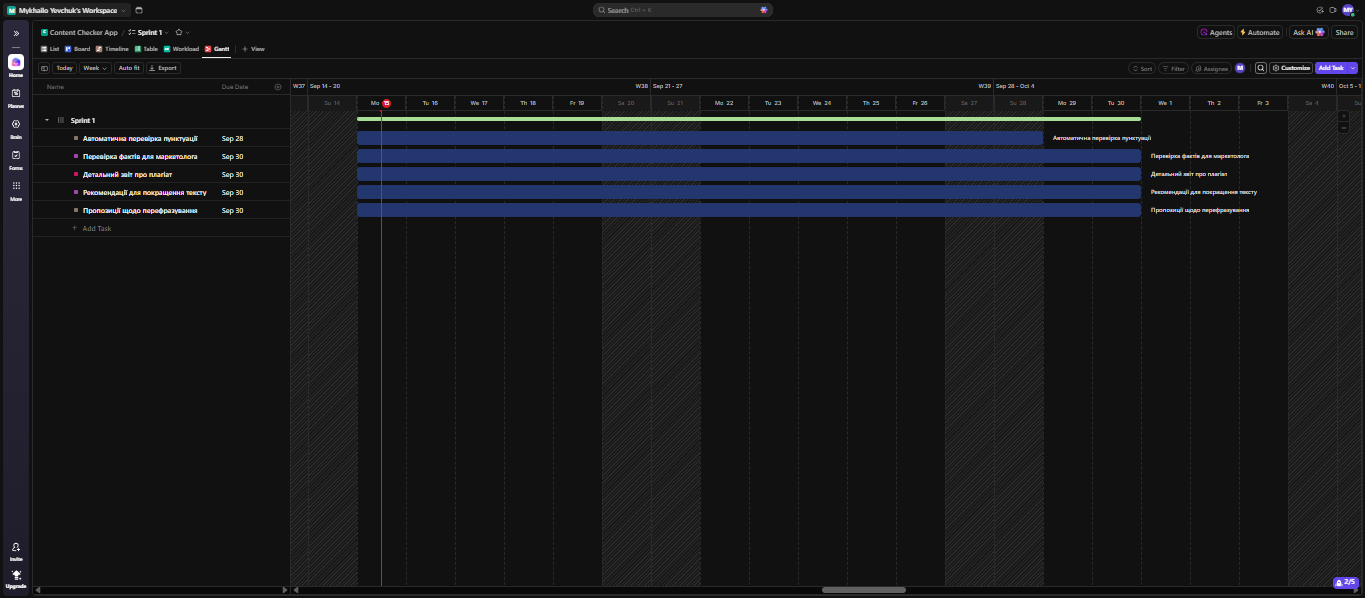
* **US1:** As a user, I want to get rephrasing suggestions so that I can improve the style of my text.
* **US2:** As a user, I want automatic punctuation checking so that I can avoid grammar mistakes.
* **US3:** As a user, I want to receive text improvement recommendations (tone, readability).

**Epic: Plagiarism & Fact-Checking**

* **US4:** As a marketer, I want to fact-check my content so that I can avoid publishing incorrect information.
* **US5:** As an editor, I want to see a detailed plagiarism report (match percentage and sources).

Рисунок 4 – Sprint Backlog

Рисунок 5 – Kanban-dashboard Sprint Backlog

Рисунок 6 – Gantt Diagram

**Sprint 2 Plan**

**Sprint Goal**

Deliver basic plagiarism detection, duplicate text identification, grammar highlighting, and provide a simple interface for novice users, with initial multilingual support.

**Scope (User Stories in Sprint 2)**

**Epic 1: User Experience & Accessibility**

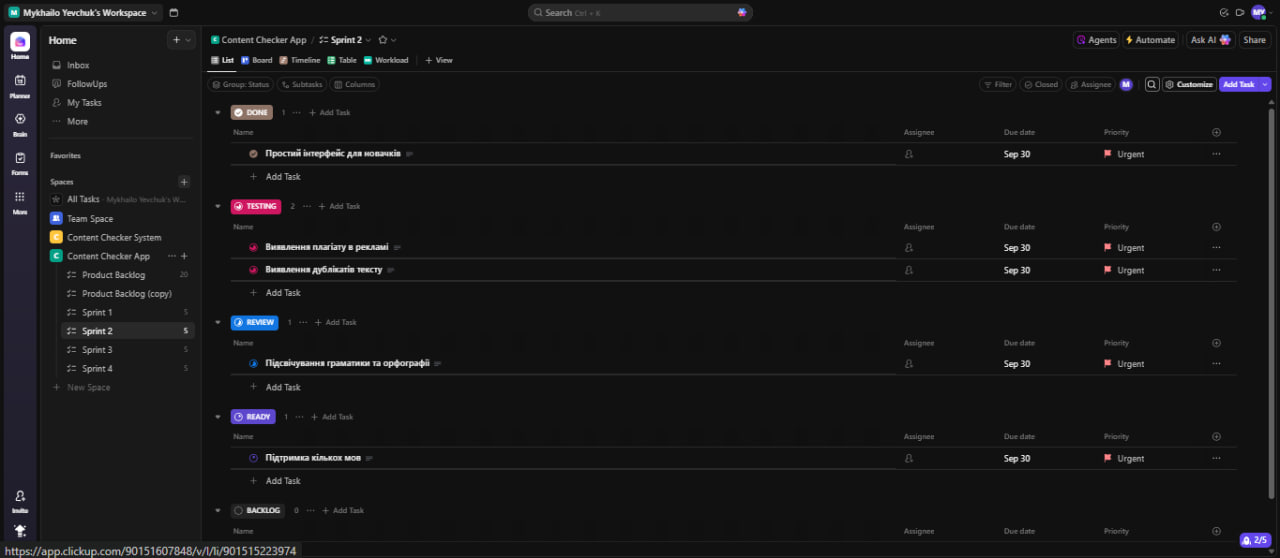
* **US1:** Simple interface for novice users (easy onboarding and navigation).

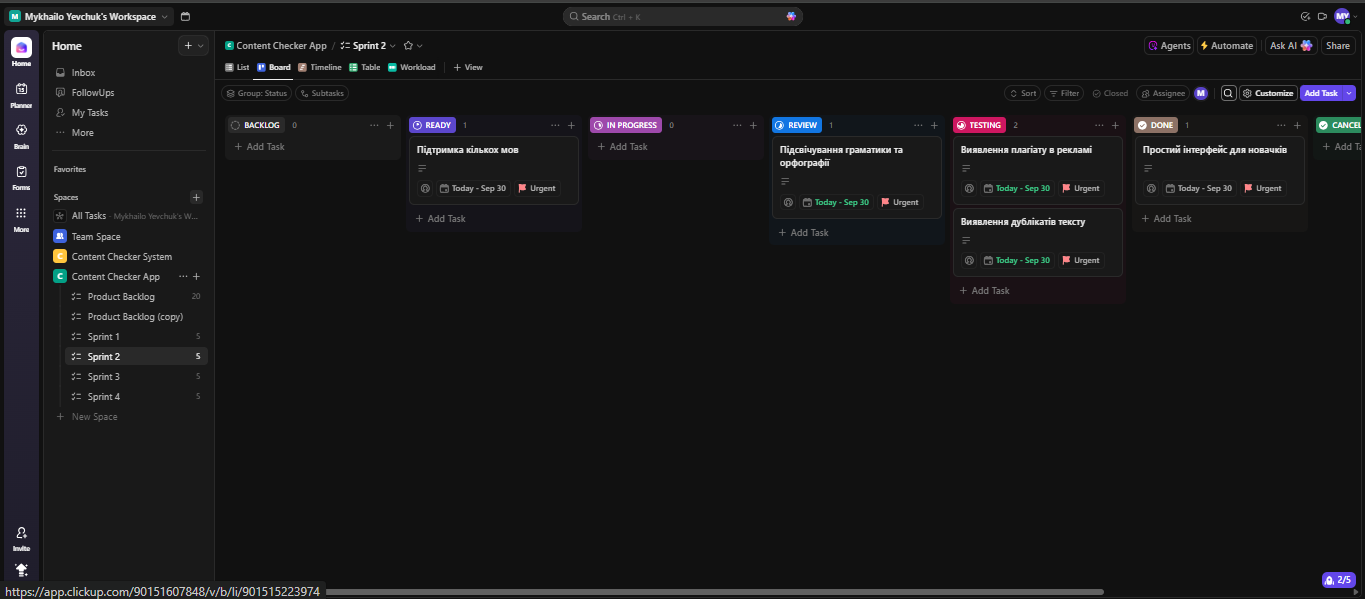
**Epic 2: Content Integrity**

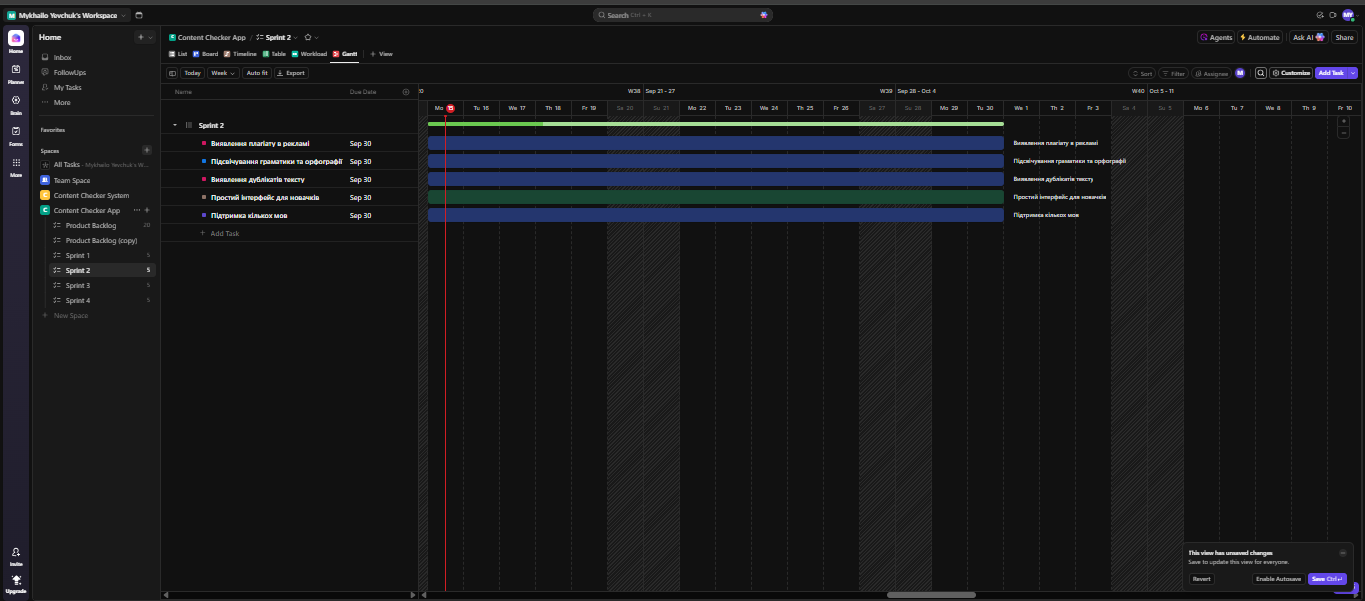
* **US2:** Plagiarism detection for ads and marketing content.
* **US3:** Duplicate text identification within the user’s database.

**Epic 3: Language Quality & Support**

* **US4:** Grammar and spelling mistake highlighting for error-free content.
* **US5:** Multilingual support to allow work with projects in different languages.

Рисунок 7 – Sprint Backlog

Рисунок 8 – Kanban-dashboard Sprint Backlog

Рисунок 9 – Gantt Diagram

**Sprint 3 Plan**

**Sprint Goal**

Deliver advanced result handling features, including filtering plagiarism results, saving and restoring checks, reporting risks, and comparing text versions.

**Scope (User Stories in Sprint 3)**

**Epic 1: Plagiarism Management & Filtering**

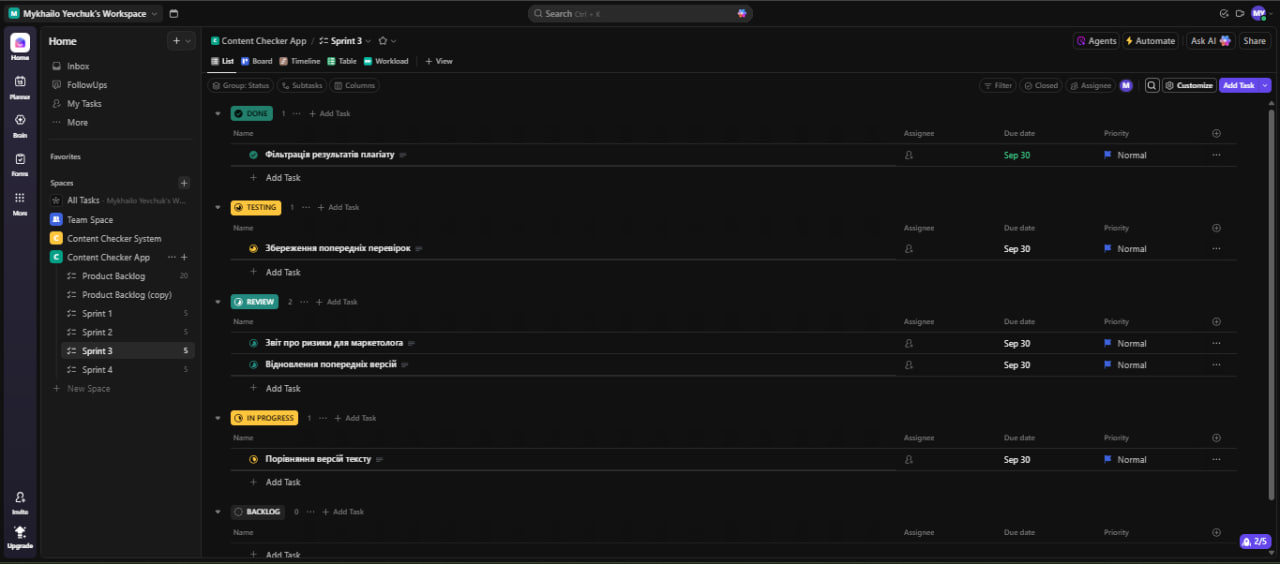
* **US1:** Filter plagiarism results by source type (focus on relevant risks).

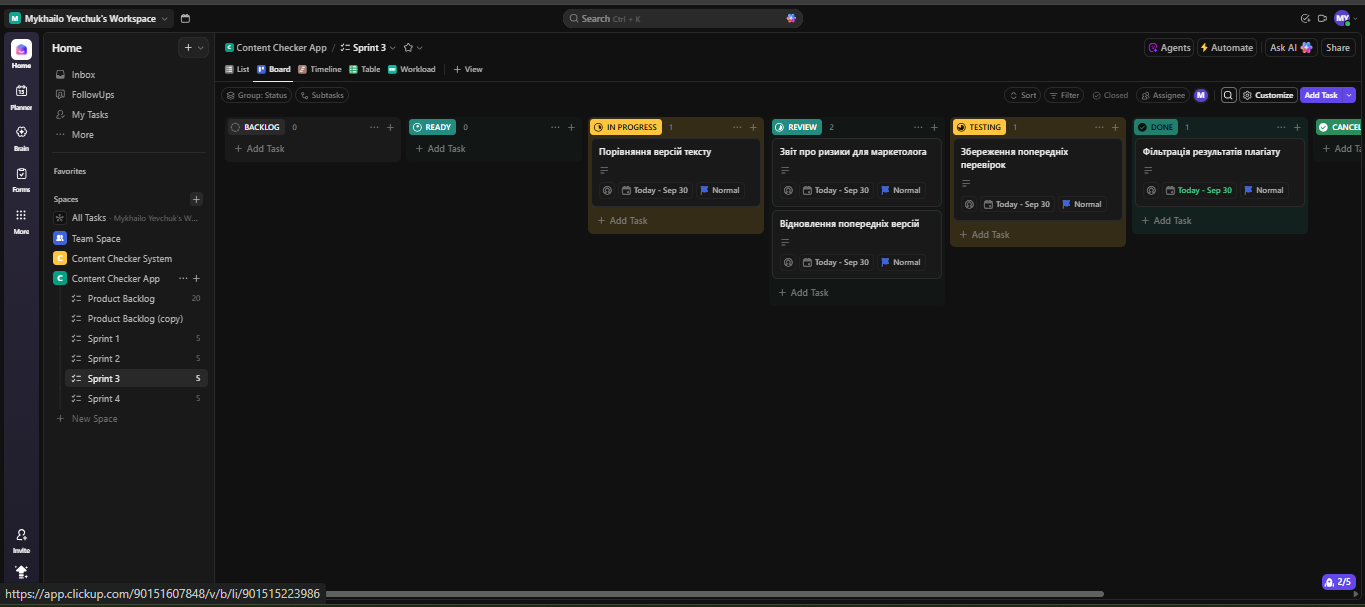
**Epic 2: History & Version Control**

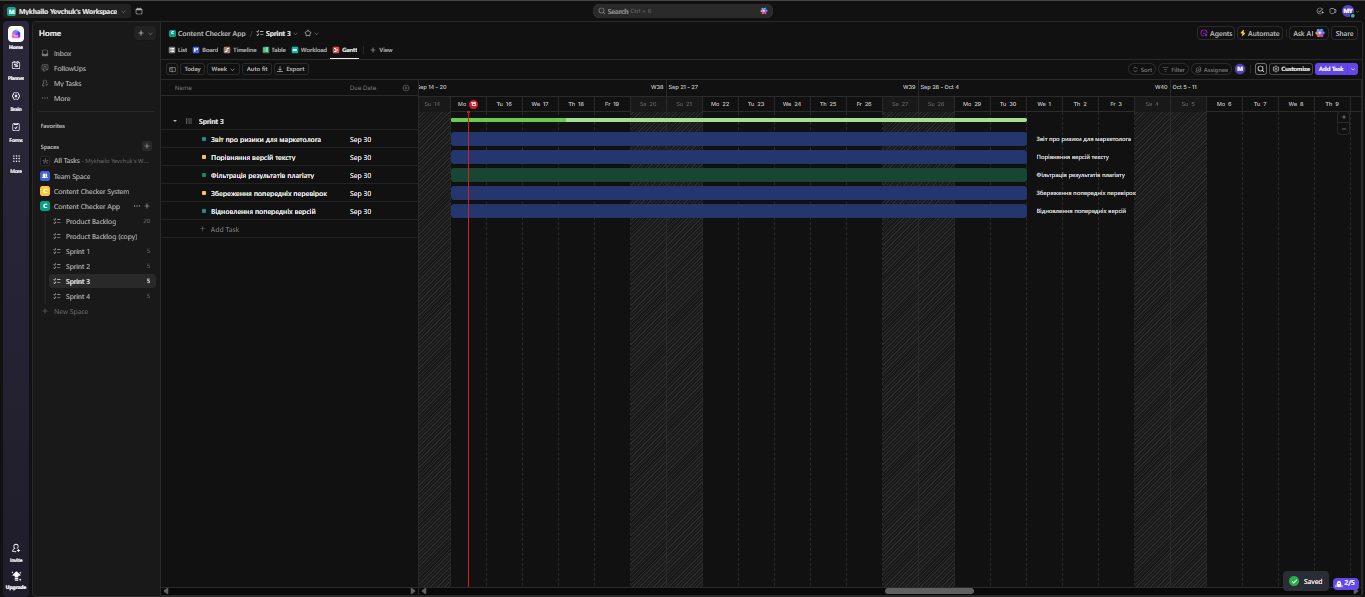
* **US2:** Save previous checks to track progress over time.
* **US4:** Restore previous text versions to undo unwanted edits.
* **US5:** Compare two text versions to see what has improved.

**Epic 3: Risk Assessment & Reporting**

* **US3:** Generate risk-level reports (legal and reputational impact).

Рисунок 10 – Sprint Backlog

Рисунок 11 – Kanban-dashboard Sprint Backlog

Рисунок 12 – Gantt Diagram

**Sprint 4 Plan**

**Sprint Goal**

Deliver system scalability, advanced plagiarism checks, user-friendly interface, and export functionality for reports.

**Epic 1: System Reliability & Performance**

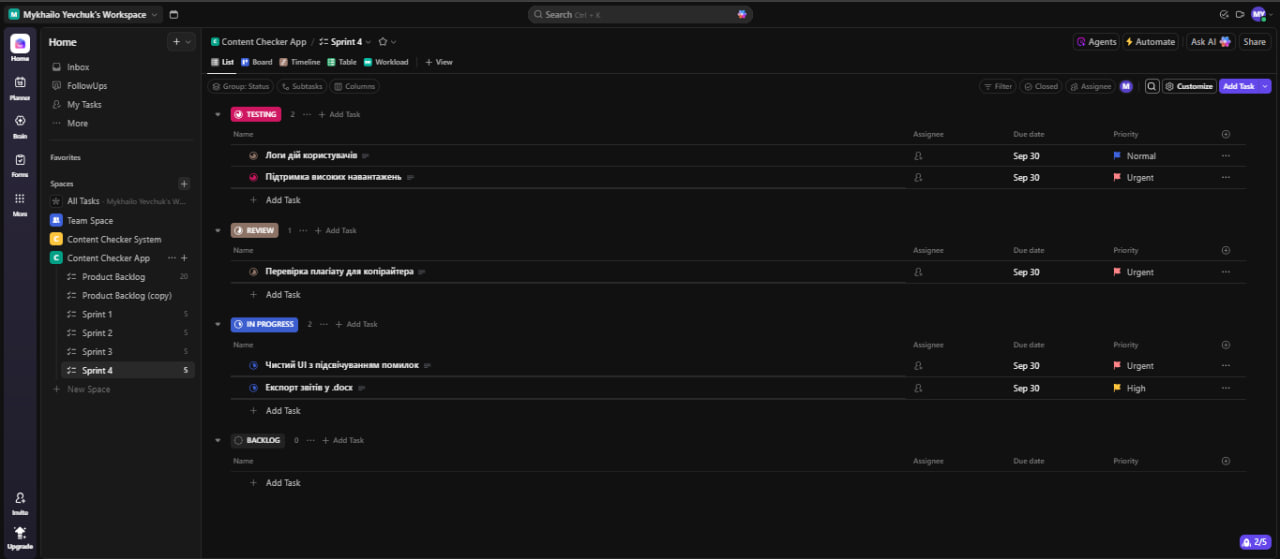
* **US1:** Logs of user actions to monitor behavior and improve performance.
* **US2:** High-load handling to allow many users to work simultaneously.

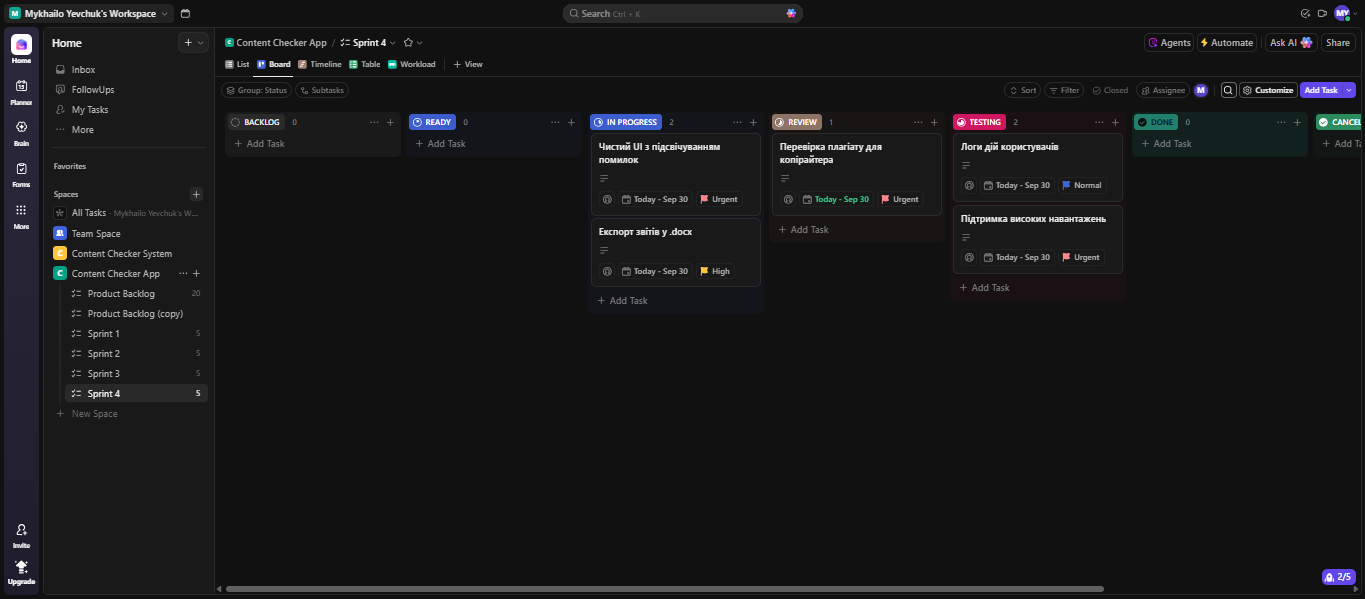
**Epic 2: Core Content Checking**

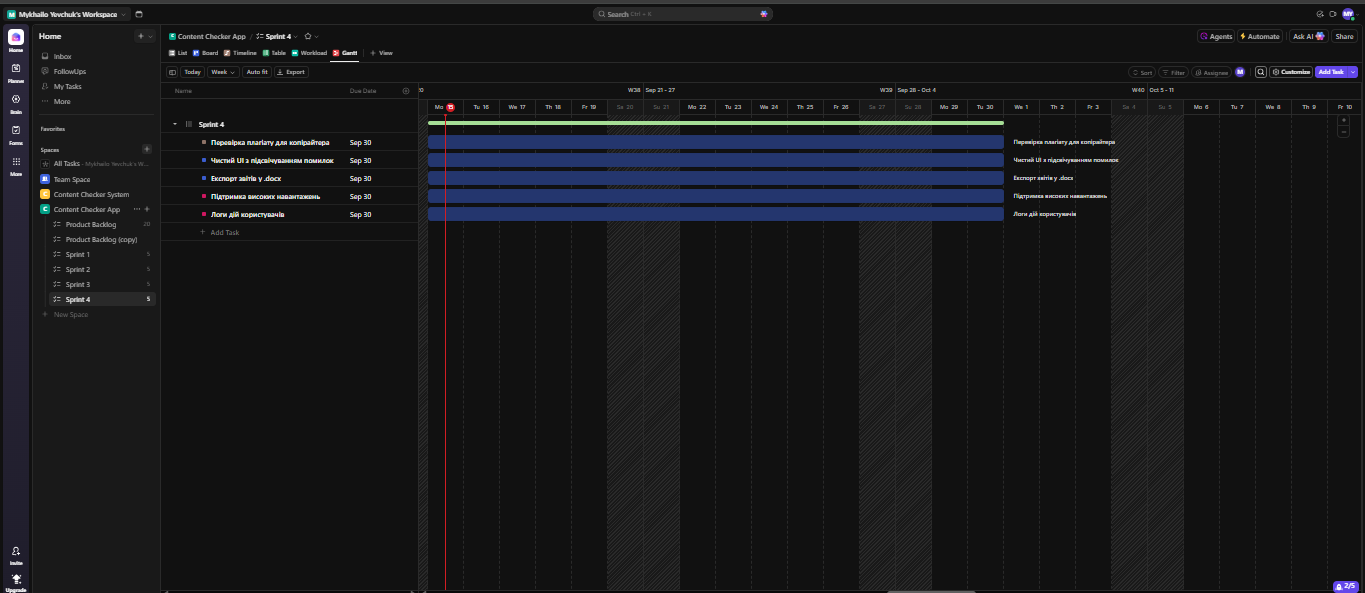
* **US3:** Plagiarism checking to ensure originality of copywriter’s content.
* **US4:** Clean UI with highlighted mistakes for quick content editing.

**Epic 3: Reporting & Collaboration**

* **US5:** Export reports in .docx format for sharing with team members.

Рисунок 13 – Sprint Backlog

Рисунок 14 – Kanban-dashboard Sprint Backlog

Рисунок 15 – Gantt Diagram

**Software Requirements Specification (SRS)**

На основі всіх етапів збору і аналізу вимог до додатку “Content Checker App” було задокументовано вимоги для розробки ПЗ. Документ можна переглянути в додатку А.

**Висновок:** Під час виконання лабораторної роботи я засвоїв навички, як збирати, аналізувати, валідувати, документувати та погоджувати умови до ПЗ, створювати SRS, а також формувати Product Backlog, та створення Kanban-dashboard.

**ДОДАТКИ**

Додаток А - Software Requirements Specification “Content Checker App”

|  |  |
| --- | --- |
| **Target release** | V1.0.0 (MVP Release) |
| **Date** | 13.09.2025 |
| **Document status** | REVIEW |
| **Document owner** | Business Analyst / Product Manager |
| **Designer** | @ designer |
| **Tech lead** | @ lead |
| **Technical writers** | @ writers |
| **QA** |  |
| **Approved by** | CTO |

**Introduction**

This document defines the requirements for the MVP of the **Content Checker App**, a web-based tool that checks text for plagiarism, factual accuracy, and originality, generates reports, and provides recommendations. It will serve as a reference for stakeholders and development teams to ensure the creation of a reliable, high-quality product.

**Project Scope**

The system provides the following key features:

* Plagiarism detection across multiple sources.
* Fact-checking for key statements.
* Grammar, punctuation, and spelling validation.
* Duplicate detection in the user’s internal content database.
* Highlighting of errors and suspicious content.
* Generation of detailed reports and improvement recommendations.
* Export and storage of verification results.
* Multilingual support (e.g., English, Ukrainian).

**Definitions, Acronyms, Abbreviations**

|  |  |  |
| --- | --- | --- |
| **Acronym / Term** | **Full Form / Definition** | **Description / Usage in System** |
| **MVP** | Minimum Viable Product | Базова версія продукту з ключовими функціями. |
| **Plagiarism** | – | Використання чужого тексту без належного цитування. |
| **Fact-checking** | – | Автоматизована перевірка достовірності фактів. |
| **Duplicate Content** | – | Повторювані фрагменти тексту в базі користувача. |
| **Report** | – | Документ з результатами перевірки та ризиками. |
| **UI/UX** | User Interface / User Experience | Інтерфейс та зручність користування системою. |
| **AC** | Acceptance Criteria | Умови, за яких User Story вважається виконаною. |
| **MoSCoW** | Must, Should, Could, Won’t | Метод пріоритезації вимог. |
| **API** | Application Programming Interface | Інтерфейс для інтеграції та обміну даними. |
| **NLP** | Natural Language Processing | Обробка природної мови для перевірки тексту. |
| **TLS** | Transport Layer Security | Протокол шифрування для захисту даних у передачі. |

**References**

1. Business Vision Document: Content Checker App – Market & User Needs Analysis (2025)
2. Company UX/UI Guidelines (internal document).
3. Academic sources on plagiarism detection and fact-checking standards.
4. ISO/IEC 27001:2013 – Information Security Management Standard.
5. GDPR – General Data Protection Regulation (EU, 2016/679).

**General Description**

The **Content Checker App** is a web-based application that helps users ensure originality, accuracy, and quality of text materials. It detects plagiarism, checks factual correctness, highlights grammar and spelling mistakes, and generates reports with improvement recommendations.

**Main Product Features**

* Plagiarism detection and originality check.
* Automated fact-checking for key statements.
* Grammar, spelling, and punctuation validation.
* Duplicate content detection in user’s database.
* Highlighting of errors and suspicious fragments.
* Generation of detailed reports with risk assessment.
* Improvement recommendations (rephrasing, structure tips).
* Multilingual support (English, Ukrainian).
* Export of results and report storage.
* Scalable performance for high user loads.

**User Characteristics**

* **Copywriters** – professional authors who need plagiarism-free and polished content.
* **Marketers** – specialists ensuring that ads and campaigns are accurate and risk-free.
* **Students & novice bloggers** – users learning proper writing, grammar, and style.
* **Business owners** – ensuring brand reputation and content uniqueness.
* **System administrators & developers** – responsible for platform stability, performance, and logs.

**Functional Requirements**

**FR-01 – Plagiarism Check**

**User Story:**  
As a copywriter, I want to check my text for plagiarism so that I can ensure originality.

**Acceptance Criteria:**

**AC-01:** The system analyzes the input text and returns a plagiarism percentage.

**AC-02:** Detected plagiarized fragments are highlighted.

**AC-03:** Sources of plagiarism are displayed in the results.

**FR-02 – Risk Analysis for Marketing Content**

**User Story:**  
As a marketer, I want to detect plagiarism in ads so that I avoid reputational risks.

**Acceptance Criteria:**

**AC-01:** The system checks promotional content for plagiarism.

**AC-02:** Results are grouped by risk level (low, medium, high).

**AC-03:** A risk report can be exported for the team.

**FR-03 – Grammar and Punctuation Validation**

**User Story:**  
As a novice user, I want to see grammar and spelling mistakes highlighted so that I can quickly fix them.

**Acceptance Criteria:**

**AC-01:** All grammar, spelling, and punctuation mistakes are highlighted.

**AC-02:** Suggested corrections are displayed.

**AC-03:** Errors disappear from the list once corrected.

**FR-04 – Fact-checking**

**User Story:**  
As a marketer, I want to verify facts in content so that I don’t publish false information.

**Acceptance Criteria:**

**AC-01:** The system identifies key factual statements.

**AC-02:** False or suspicious facts are flagged.

**AC-03:** The system provides links to reliable sources for verification.

**FR-05 – Duplicate Detection**

**User Story:**  
As a user, I want to detect duplicates in my text database so that I can avoid reusing the same content.

**Acceptance Criteria:**

**AC-01:** The system scans new text against the user’s internal database.

**AC-02:** Duplicate fragments are highlighted with reference to their origin.

**AC-03:** Duplicates can be filtered by time or category.

**FR-06 – Reports Generation**

**User Story:**  
As a copywriter, I want to get a detailed report about plagiarism so that I understand weak points in my text.

**Acceptance Criteria:**

**AC-01:** Reports are exportable in .docx and .pdf formats.

**AC-02:** Reports include plagiarism percentage, sources, and highlights.

**AC-03:** Risk levels are clearly presented.

**FR-07 – Recommendations for Improvement**

**User Story:**  
As a copywriter, I want the system to suggest rephrasing sentences so that my text is more unique.

**Acceptance Criteria:**

**AC-01:** Repetitive or common phrases are identified.

**AC-02:** Alternative rephrasings are suggested.

**AC-03:** Suggested edits preserve the original meaning.

**FR-08 – Version Comparison**

**User Story:**  
As a user, I want to compare two versions of a text so that I can see improvements.

**Acceptance Criteria:**

**AC-01:** Changes are highlighted between versions.

**AC-02:** Two text versions can be displayed side by side.

**AC-03:** Comparison results can be exported.

**FR-09 – Multilingual Support**

**User Story:**As a user, I want to use the system in English and Ukrainian so that I can work with different projects.

**Acceptance Criteria:**

**AC-01:** Users can switch the interface language.

**AC-02:** Text verification works only for the selected language.

**AC-03:** Notifications and tips appear in the chosen language.

**FR-10 – Scalability and Logging**

**User Story:**  
As a developer, I want logs of user actions so that I can improve system performance.

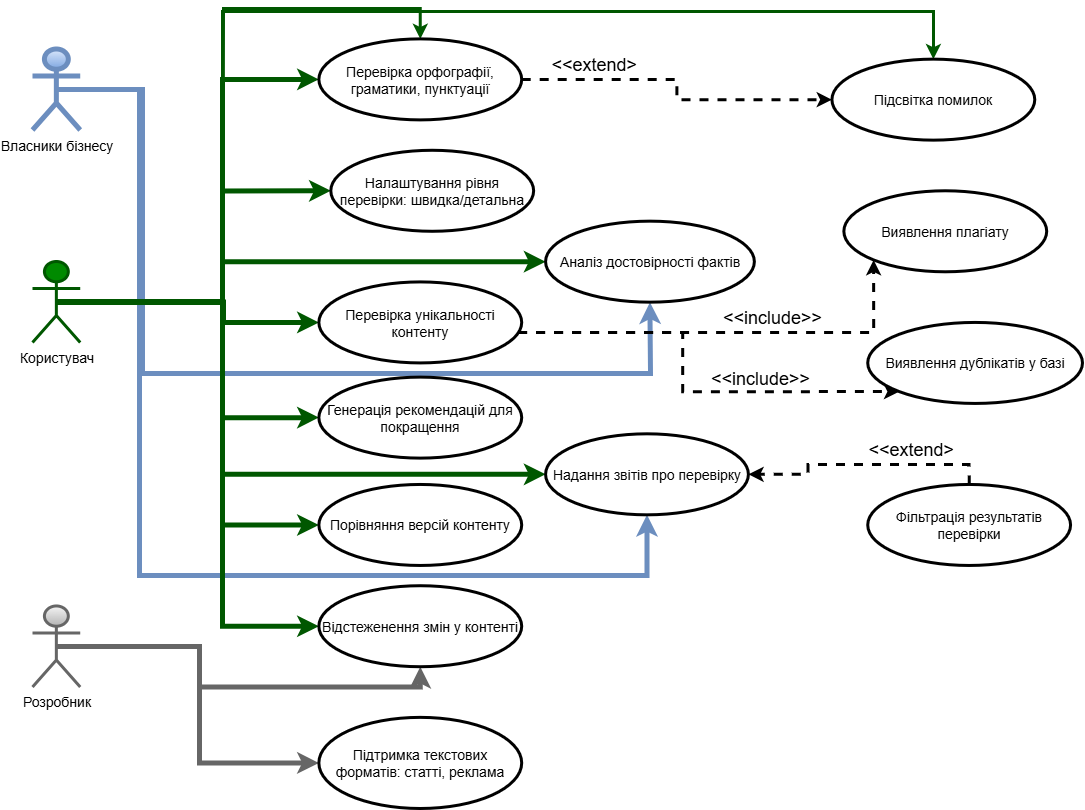
**Acceptance Criteria:**

**AC-01:** All user actions are logged with date, time, and type.

**AC-02:** Logs can be filtered by user or activity type.

**AC-03:** Logs support analysis for performance optimization.

**Use Case Diagram:**

****

**Role-permission matrix (✅-Allowed, ❌-Now allowed)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature / Action** | **Copywriter** | **Marketer** | **Student / Novice** | **Business Owner** | **Developer / Admin** |
| **Register / Login** | ✅ | ✅ | ✅ | ✅ | ✅ |
| **Upload / Enter Text** | ✅ | ✅ | ✅ | ✅ | ✅ (for testing) |
| **Plagiarism Check** | ✅ | ✅ | ✅ | ✅ | ✅ (monitor only) |
| **Fact-checking** | ❌ | ✅ | ❌ | ✅ | ✅ |
| **Grammar & Spell Check** | ✅ | ❌ | ✅ | ❌ | ✅ (monitor only) |
| **Duplicate Detection** | ✅ | ✅ | ❌ | ✅ | ✅ |
| **Generate Reports** | ✅ | ✅ | ❌ | ✅ | ✅ (support only) |
| **Recommendations (Rephrasing)** | ✅ | ❌ | ✅ | ❌ | ✅ (monitor only) |
| **Compare Versions** | ✅ | ✅ | ❌ | ✅ | ✅ |
| **Export Results** | ✅ | ✅ | ❌ | ✅ | ✅ (support only) |
| **Change Language (EN/UA)** | ✅ | ✅ | ✅ | ✅ | ✅ |
| **View Logs** | ❌ | ❌ | ❌ | ❌ | ✅ |
| **Manage Users / Roles** | ❌ | ❌ | ❌ | ❌ | ✅ |
| **System Configuration** | ❌ | ❌ | ❌ | ❌ | ✅ |

**RACI-matrix**

**R (Responsible)** – виконує роботу або безпосередньо взаємодіє з Feature.

**A (Accountable)** – відповідальний за прийняття рішення

**C (Consulted)** – радить, надає інформацію

**I (Informed)** – просто інформовані про результат

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature/Action** | **Copywriter** | **Marketer** | **Novice user** | **Developer** |
| Перевірка контенту на плагіат | **R** | **R** | **I** | **C** |
| Автоматична перевірка орфографії, пунктуації та граматики | **R** | **I** | **R** | **C** |
| Аналіз достовірності фактів | **I** | **R** | **I** | **C** |
| Виявлення дублікатів у внутрішній базі | **I** | **I** | **R** | **C** |
| Надання звітів про перевірку контенту | **R** | **R** | **I** | **C** |
| Генерація рекомендацій для покращення унікальності | **R** | **I** | **R** | **C** |
| Порівняння версій контенту | **I** | **I** | **R** | **C** |
| Фільтрація результатів перевірки | **R** | **I** | **I** | **C** |
| Зручний інтерфейс | **R** | **I** | **R** | **C** |
| Збереження та експорт результатів у файл | **R** | **R** | **I** | **C** |
| Підтримка багатомовності | **R** | **I** | **R** | **C** |
| Масштабованість продукту | **I** | **I** | **I** | **C** |
| Історія змін та відкат змін | **I** | **I** | **R** | **C** |

**Non-Functional Requirements**

**Performance Requirements**

* The system shall support up to **1,000 concurrent users** without performance degradation.
* Average response time for plagiarism or grammar checks shall not exceed **3 seconds** for texts up to **5,000 words**.
* The system shall process at least **200 requests per minute** under peak load.

**Scalability**

* The system shall scale horizontally by adding servers without code modification.
* It shall support database growth up to **500 GB of stored reports and logs** without performance loss.

**Availability & Reliability**

* The system shall maintain **99.5% uptime** (excluding scheduled maintenance).
* It shall automatically recover from hardware or service failures within **5 minutes**.

**Security**

* All data in transit must be encrypted with **TLS 1.3**.
* User authentication shall follow **OAuth 2.0** protocol.
* Sensitive user data (texts, reports) shall be stored securely and never shared with third parties.
* The system shall comply with **GDPR** for data privacy.

**Maintainability**

* Source code shall follow clean coding standards with inline documentation.
* Unit test coverage shall be at least **80%**.
* Logs shall be centralized and accessible for monitoring.

**Usability**

* The UI shall be responsive and work on desktop, tablet, and mobile.
* All main features shall be accessible from the dashboard in **≤3 clicks**.
* The interface shall meet **WCAG 2.1 AA accessibility standards**.

**Portability**

* The system shall run on **Windows, Linux, and macOS**.
* Deployment must be possible in **cloud environments (AWS, Azure, GCP)**.

**Technical Requirements**

**Technology Stack**

* **Backend**: Python (FastAPI, Flask) for core services and NLP pipelines.
* **Frontend**: React.js with TypeScript for web interface.
* **Database**: PostgreSQL 14+ for structured storage (users, reports, logs).
* **NLP & AI**: spaCy, Hugging Face Transformers for text analysis (plagiarism, grammar, fact-checking).
* **Containerization**: Docker for deployment consistency.

**Architecture Requirements**

* The system shall follow a **microservices architecture** to allow independent scaling of plagiarism detection, grammar checking, and fact-checking modules.
* Services shall communicate through **RESTful APIs** and message queues (e.g., Kafka or RabbitMQ).
* The system shall implement a **three-tier architecture** (presentation, application, data).

**Environment Requirements**

* The system shall run on **Linux-based servers (Ubuntu 22.04 LTS)**.
* Production shall be hosted in a **cloud environment (AWS, Azure, or GCP)**.
* CI/CD pipelines shall be configured using **GitHub Actions or Jenkins**.

**Version Control and Configuration**

* All source code shall be stored in a **Git-based repository (GitHub/GitLab)**.
* The system shall use environment-specific configuration files (.env) with secrets stored in a **secure vault** (e.g., AWS Secrets Manager).
* All builds shall follow **semantic versioning** (e.g., v1.0.0, v1.1.0).

**Standards and Guidelines**

* All code shall follow the **organization’s coding standards** and naming conventions.
* The system shall include automated **unit, integration, and end-to-end tests**.
* The system shall provide **API documentation** using **OpenAPI/Swagger**.

## ****Interface Requirements****

**User Interfaces (UI)**

* The system shall provide a **web-based interface** accessible via modern browsers (Chrome, Firefox, Edge, Safari).
* The UI shall be **responsive** and adjust to desktop, tablet, and mobile screens.
* Users shall have a **dashboard** with quick access to plagiarism checks, fact-checking, grammar validation, and reports.
* The system shall support **multiple languages** (English, Ukrainian).
* Error messages and tips shall be **clear and user-friendly**.
* The interface shall follow **consistent navigation and layout** standards.

**Hardware Interfaces**

* The system shall support **standard PC hardware** (keyboard, mouse, monitor).
* The system shall allow report generation for **printers** (export to PDF/Docx).
* No specialized hardware (scanners, sensors) is required.

**Software Interfaces**

* The system shall connect to external **plagiarism and fact-checking APIs** via REST.
* It shall integrate with a **PostgreSQL database** for storing results, logs, and reports.
* The system shall export reports in **DOCX and PDF formats**.
* Single Sign-On (SSO) shall be supported for enterprise users (via OAuth 2.0).

**Communication Interfaces**

* All external and internal communication shall use **HTTPS (TLS 1.3 encryption)**.
* The system shall exchange data in **JSON** format for APIs.
* The system shall send **email notifications** (SMTP) about completed checks and risk reports.
* Webhooks shall be available for **real-time integration** with third-party systems.

**Constraints and Risks**

**Constraints**

* **Budgetary Constraints** – The project must stay within the allocated budget of $X.
* **Time Constraints** – The MVP version shall be delivered by [specific deadline].
* **Regulatory & Compliance Constraints** – The system shall comply with **GDPR** and data protection laws.
* **Technology Constraints** – The system must run in a **web environment** and support only **text-based formats (.txt, .docx)**.
* **Resource Constraints** – The development team is limited to a small cross-functional Scrum team (7–9 members).

Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk ID** | **Description** | **Likelihood** | **Impact** | **Mitigation Strategy** |
| R1 | Delay in requirement approvals | High | High | Set clear approval deadlines, escalation process |
| R2 | Limited accuracy of fact-checking algorithms | Medium | High | Combine automated fact-checking with trusted external APIs |
| R3 | High system load from multiple concurrent users | Medium | Medium | Use horizontal scaling with cloud infrastructure |
| R4 | Data privacy breach (user texts exposed) | Low | High | Encrypt data, follow GDPR, strict access control |
| R5 | Scope creep due to changing stakeholder requests | High | Medium | Enforce change management and approval process |
| R6 | Third-party API downtime (plagiarism/fact-checking) | Medium | Medium | Implement retry logic, fallback mechanisms |
| R7 | Lack of user adoption (low engagement) | Medium | Medium | Ensure strong UX/UI design, onboarding tutorials |

**Testing**

**Testing Objectives**

* Verify that all functional requirements (plagiarism check, fact-checking, grammar validation, reporting) are correctly implemented.
* Validate that non-functional requirements (performance, security, usability, scalability) are met.
* Detect and fix defects early in the development lifecycle.
* Ensure the system is stable, secure, and ready for production use.

**Testing Types**

|  |  |
| --- | --- |
| **Test Type** | **Description** |
| **Unit Testing** | Tests individual modules (e.g., plagiarism detection, grammar checker). |
| **Integration Testing** | Verifies interaction between backend, database, and external APIs. |
| **System Testing** | Tests the complete integrated system against requirements. |
| **Acceptance Testing** | Validates features with real users to confirm they meet expectations. |
| **Performance Testing** | Evaluates response time, load capacity, and scalability. |
| **Security Testing** | Ensures encryption, authentication, and compliance with GDPR. |
| **Regression Testing** | Retests existing features after changes to avoid defects. |
| **Usability Testing** | Checks UI clarity, navigation, and accessibility. |

**Test Environment**

* A staging environment mirroring production.
* Anonymized datasets for plagiarism/fact-checking.
* Cloud infrastructure with auto-scaling enabled.

**Test Tools**

* **Unit Testing**: PyTest (Python), Jest (JavaScript).
* **Integration/System Testing**: Postman, Selenium.
* **Performance Testing**: JMeter, Locust.
* **Security Testing**: OWASP ZAP, Burp Suite.
* **CI/CD Integration**: GitHub Actions, Jenkins.

**Test Plan Overview**

|  |  |
| --- | --- |
| **Item** | **Description** |
| **Entry Criteria** | All planned sprint features are implemented, build is stable and deployed to staging. |
| **Exit Criteria** | 100% test cases executed, ≥95% passed, no critical defects open. |
| **Test Deliverables** | Test cases, test scripts, test data, test reports, defect logs. |
| **Responsible Roles** | QA Lead, QA Engineers, Developers (unit tests), Product Owner (acceptance). |
| **Test Environment** | Staging environment mirroring production, anonymized datasets, cloud hosting. |
| **Test Schedule** | Aligned with development milestones (sprint-based testing). |

**Appendices**

**Success Metrics**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric Category** | **Description** | **Target Value** | **Measurement Method** | **Responsible** |
| **Requirement Coverage** | Percentage of verified functional requirements | 100% | Requirements traceability matrix | QA Lead |
| **Defect Rate** | Number of critical/major defects at release | < 5 major, 0 critical | Bug tracking reports | QA Lead |
| **UAT Pass Rate** | % of passed user acceptance test cases | ≥ 95% | UAT reports | Product Owner / QA Lead |
| **Response Time** | Average system response under normal load | < 3 seconds for 5k words | JMeter / Locust | QA Engineer |
| **System Uptime** | Availability in production | ≥ 99.5% | Monitoring tools | DevOps |
| **User Satisfaction (CSAT)** | Post-release user rating | ≥ 4.0 / 5 | User surveys | UX Researcher |
| **Adoption Rate** | Percentage of active users after launch | ≥ 70% in 30 days | Usage analytics | Product Manager |
| **Automated Test Coverage** | Percentage of code covered by tests | ≥ 80% | Code coverage reports | QA Lead / Developers |

**Milestones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Milestone** | **Target Date** | **Description** | **Owner** |
| **M1 – MVP Release** | Q1 2026 | Core features: plagiarism check, grammar & spelling validation, basic reporting | Product Team |
| **M2 – Fact-checking Module** | Q2 2026 | Automated fact validation with external API integration, risk-level analysis | NLP / Data Science Team |
| **M3 – Recommendations Engine** | Q3 2026 | Sentence rephrasing, style improvement tips, duplicate detection in user database | NLP Team |
| **M4 – Reports & Export** | Q3 2026 | Detailed risk reports, export to DOCX/PDF, version comparison | QA / Backend Team |
| **M5 – Multilingual Support** | Q4 2026 | English + Ukrainian interface and content validation | Frontend / Localization Team |
| **M6 – Scaling & Security** | Q1 2027 | Performance optimization, 99.5% uptime SLA, GDPR compliance, secure logging | DevOps / Security Team |

**Open Questions**

|  |  |  |
| --- | --- | --- |
| **Question** | **Answer** | **Date Answered** |
| What external plagiarism APIs will be used (open-source vs. paid)? | — Pending — | — |
| What is the maximum supported text size (words/characters) per check? | — Pending — | — |
| Should the system store full user texts, or only verification results/metadata? | — Pending — | — |
| Will fact-checking rely only on APIs, or also on custom internal databases? | — Pending — | — |
| Should the system support offline mode for text analysis? | — Pending — | — |
| What are the branding and UI design guidelines (colors, typography)? | — Pending — | — |
| Expected number of concurrent users in the first release? | — Pending — | — |
| Should the system integrate with LMS/CRM platforms (for business clients)? | — Pending — | — |