

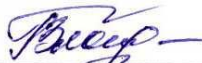
**МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ**

**ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І
СИСТЕМ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: ДОДАТОК ДЛЯ ВІДСТЕЖЕННЯ ЩОДЕННИХ СПРАВ ТА ПЛАНІВ

Здобувача вищої освіти другого (магістерського) рівня,
II курсу, групи КМІ-24-2, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Інформаційні управляючі системи і технології в економіці»


(підпис)

Городецького Владислава
Віталійовича
(прізвище, ім'я, по батькові)

Науковий керівник


(підпис)

Гладченко Оксана Вікторівна,
канд. пед. наук, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Керівник освітньо-
професійної програми


(підпис)

Омельчук Антон
Анатолійович, канд. техн. наук
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту за рішенням кафедри протокол від «02» січня 2026 № 10

Завідувач кафедри


(підпис)

Філоненко Михайло Миколайович,
канд. фіз.-мат. наук, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Ірпінь – 2026

Реєстрація № 29

22.01.2026

(дата)



(підпис відповідальної особи)

Результати перевірки

до захисту

(до захисту, на
доопрацювання)

02.01.2026

(дата)



(підпис)

Результат захисту

4 (добре)

(оцінка)

05.02.2026

(дата)



(підпис)

Голова комісії:



(підпис)

О.С. Дюбанов

Члени комісії:



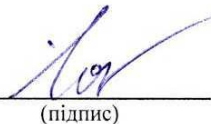
(підпис)

А.Ю. Горбовий



(підпис)

М.М. Філоненко



(підпис)

І.Д. Погореловська

БЛАНК ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії, секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст кваліфікаційної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	від 0 до 30	23		05.02.2026 Дюбанов О.С.
2. Дотримання вимог щодо оформлення	від 0 до 10	7		Горбовий А.Ю.
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	3		Філоненко М.М. Погореловська І.Д.
4. Рецензент	від 0 до 5	4		Суховерхівка Л.П.
РАЗОМ БАЛІВ	від 0 до 50	37		

Оцінювання кваліфікаційної роботи науковим керівником - 38 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підписи)
75	C	4 (добре)	05.02.2026	Дюбанов О.С. Горбовий А.Ю. Філоненко М.М. Погореловська І.Д. Суховерхівка Л.П.

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	
60-69	D	
50-59	E	3 (задовільно)
35-49	FX	2 (незадовільно)

Факультет фінансів та цифрових технологій
Кафедра комп'ютерних та інформаційних технологій і систем
Галузь знань 12 «Інформаційні технології»
Спеціальність 122 «Комп'ютерні науки»
Освітня програма «Інформаційні управляючі системи і технології в економіці»

Затверджую

Завідувач кафедри  Філоненко М.М.

«11» грудня 2024

Завдання на кваліфікаційну роботу
здобувачу вищої освіти
Городецькому Владиславу Віталійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Додаток для відстеження щоденних справ та планів

Затверджена наказом ректора від «10» грудня 2024 № 1852

2. Строк здачі закінченої роботи «02» січня 2026

3. Мета роботи Проектування та розробка інтелектуального додатка для управління справами, який забезпечує оптимальний баланс між функціональністю, простотою інтерфейсу та надійністю збереження даних.
Об'єкт дослідження Процес організації та автоматизації персонального планування в цифровому середовищі.

Предмет дослідження Методи, архітектурні підходи та програмні засоби розробки веб- та мобільних додатків для відстеження щоденних завдань і планів.
Орієнтовна база дослідження Методики когнітивної психології (наприклад, метод Ейзенхауера або техніку Pomodoro) з сучасними технологіями синхронізації даних та алгоритмами пріоритезації.

4. Вихідні дані до роботи Теоретичні підходи до управління часом та їх реалізацію в сучасних ПЗ.

5. Перелік графічного матеріалу Презентація інтерфейсу інформаційної системи для відстеження щоденних справ та планів.

6. Зміст кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП; РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ СИСТЕМ КЕРУВАННЯ ЗАВДАННЯМИ; РОЗДІЛ 2. ПРОЄКТУВАННЯ ДЕСКТОПНОГО ДОДАТКУ ДЛЯ ПЛАНУВАННЯ ТА УПРАВЛІННЯ ПОДІЯМИ; РОЗДІЛ 3. ТЕСТУВАННЯ ТА СУПРОВІД РОЗРОБЛЕНОГО ДОДАТКУ; ВИСНОВКИ; СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.

7. Календар план виконання роботи:

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	Фактично
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	До 15.12.2024	До 15.12.2024
2	I розділ (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	30.05.2025	30.05.2025
3	II розділ (аналіз експериментальної частини про методики досліджень)	10.09.2025	10.09.2025
4	III розділ (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	20.11.2025	20.11.2025
5	Висновки	11.12.2025	11.12.2025
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	25.12.2025	25.12.2025
7	Отримання відгуку керівника	28.01.2026	28.01.2026
8	Отримання зовнішньої рецензії	30.01.2026	30.01.2026

8. Дата видачі завдання «11» грудня 2024

9. Керівник кваліфікаційної роботи


(підпис)

Гладченко О.В.
(прізвище, ініціали)

10. Керівник освітньо-професійної програми


(підпис)

Омельчук А.А.
(прізвище, ініціали)

11. Завдання до виконання отримав
здобувач вищої освіти


(підпис)

Городецький В.В.
(прізвище, ініціали)

АНОТАЦІЯ

ДОДАТОК ДЛЯ ВІДСТЕЖЕННЯ ЩОДЕННИХ СПРАВ ТА ПЛАНІВ

У кваліфікаційній роботі розв'язано актуальне завдання розробки десктопного додатка для планування та управління подіями, що поєднує високу функціональність із сучасними стандартами UI/UX-дизайну. На основі проведеного дослідження та процесу розробки було зроблено висновки.

Перспективи подальшого розвитку проєкту вбачаються в інтеграції з хмарними сервісами для синхронізації даних між різними пристроями та впровадженні інтелектуальних алгоритмів проактивного нагадування на основі аналізу поведінки користувача.

Кваліфікаційна робота містить 50 сторінок, 6 рисунків, 1 таблицю, список використаних джерел 45 найменувань.

Ключові слова: застосунок, планування, асистент.

Рік захисту роботи 2026

ABSTRACT

APPLICATION FOR TRACKING DAILY AFFAIRS AND PLANS

The qualification work solves the urgent task of developing a desktop application for planning and managing events, combining high functionality with modern UI/UX design standards. Based on the research and development process, the following conclusions were made.

The prospects for further development of the project are seen in integration with cloud services for data synchronization between different devices and the implementation of intelligent algorithms for proactive reminders based on user behavior analysis.

The qualification work contains 50 pages, 6 figures, 1 table, a list of sources used 45 items.

Keywords: application, planning, assistant.

Year of defense of the work 2026

Зміст

ВСТУП.....	3
РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ СИСТЕМ КЕРУВАННЯ ЗАВДАННЯМИ ..	5
РОЗДІЛ 2. ПРОЄКТУВАННЯ ДЕСКТОПНОГО ДОДАТКУ ДЛЯ ПЛАНУВАННЯ ТА УПРАВЛІННЯ ПОДІЯМИ	12
РОЗДІЛ 3. ТЕСТУВАННЯ ТА СУПРОВІД РОЗРОБЛЕНОГО ДОДАТКУ...	41
ВИСНОВКИ	43
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	45

ВСТУП

Обґрунтування актуальності теми. У сучасних умовах глобальної цифровізації та постійного зростання інформаційного навантаження, ефективний тайм-менеджмент стає критичним фактором продуктивності як для окремих фахівців, так і для організацій у цілому. Динамічний темп життя, перехід до гібридних форматів роботи та навчання, а також необхідність одночасного контролю множинних проєктів зумовлюють високий попит на інтелектуальні системи планування. Незважаючи на значну кількість існуючих рішень (Trello, Notion, Microsoft To Do), користувачі часто стикаються з проблемою «надлишкової функціональності» або, навпаки, недостатньої гнучкості інструментів. Актуальність даної роботи зумовлена необхідністю створення персоналізованого, кросплатформного додатка, який інтегрує методики когнітивної психології (наприклад, метод Ейзенхауера або техніку Pomodoro) з сучасними технологіями синхронізації даних та алгоритмами пріоритезації [1-20].

Об'єкт дослідження — процес організації та автоматизації персонального планування в цифровому середовищі.

Предмет дослідження — методи, архітектурні підходи та програмні засоби розробки веб- та мобільних додатків для відстеження щоденних завдань і планів.

Мета роботи — проєктування та розробка інтелектуального додатка для управління справами, який забезпечує оптимальний баланс між функціональністю, простотою інтерфейсу та надійністю збереження даних.

Завдання дослідження:

- Проаналізувати наявні теоретичні підходи до управління часом та їх реалізацію в сучасних ПЗ.
- Провести компаративний аналіз існуючих аналогів на ринку IT-рішень.
- Обґрунтувати вибір технологічного стеку для розробки (наприклад, React/Next.js для фронтенду та Node.js/PostgreSQL для бекенду).
- Розробити архітектуру системи та спроектувати базу даних.

- Реалізувати програмний продукт із функціями пріоритезації, нагадувань та аналітики продуктивності.
- Провести тестування розробленого додатка на відповідність функціональним вимогам.

РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ СИСТЕМ КЕРУВАННЯ ЗАВДАННЯМИ

На сучасному етапі ринок інтелектуальних систем менеджменту завдань представлений розмаїттям програмних продуктів, спрямованих на оптимізацію діяльності. Провідні позиції у цьому сегменті займають такі платформи, як Trello, Asana, Monday.com, Notion та Microsoft To Do. Кожне з цих рішень пропонує унікальний інструментарій та варіативні методологічні підходи до структурування робочих потоків і контролю виконання завдань (рис. 1.1) [6–14]



Рисунок 1.1 Провідні програмні продукти на ринку інтелектуальних систем управління завданнями

Trello є спеціалізованою системою управління проєктами, що реалізує концепцію візуалізації робочих процесів на основі методології Канбан. Ключовий функціонал сервісу базується на використанні ієрархічної структури - дощок, списків та інтерактивних карток, які дозволяють наочно структурувати задачі відповідно до етапів їх життєвого циклу (рис. 1.2).

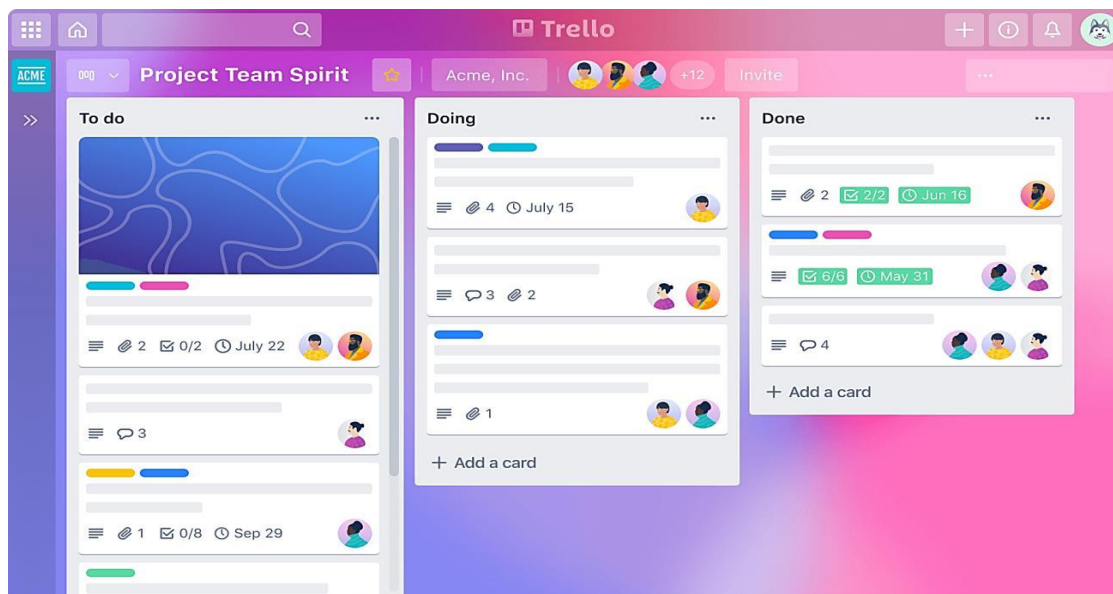


Рисунок 1.2 Інтерфейс Trello

Колаборативні можливості платформи забезпечуються через інструменти делегування завдань, встановлення часових обмежень (дедлайнів) та розгалужену систему інтеграцій із зовнішніми сервісами. Завдяки ергономічному та інтуїтивно зрозумілому інтерфейсу Trello є універсальним рішенням як для персонального менеджменту, так і для координації командної діяльності [10]

Asana — це багатофункціональне середовище для адміністрування проєктів, розроблене для оптимізації колективної діяльності та підвищення продуктивності команд. Платформа пропонує варіативні інструменти візуалізації та планування, зокрема проєктні дошки, ієрархічні списки та хронологічні графіки (таймлайни). Функціональний базис системи охоплює механізми пріоритезації задач, контроль термінів виконання, моніторинг динаміки прогресу та автоматизацію рекурентних (повторюваних) операцій. Завдяки розгалуженій екосистемі інтеграцій із зовнішніми сервісами (Slack, Google Drive, Microsoft Teams), Asana виступає універсальним інструментом для координації масштабних та технологічно складних проєктів [11] (рис. 1.3)

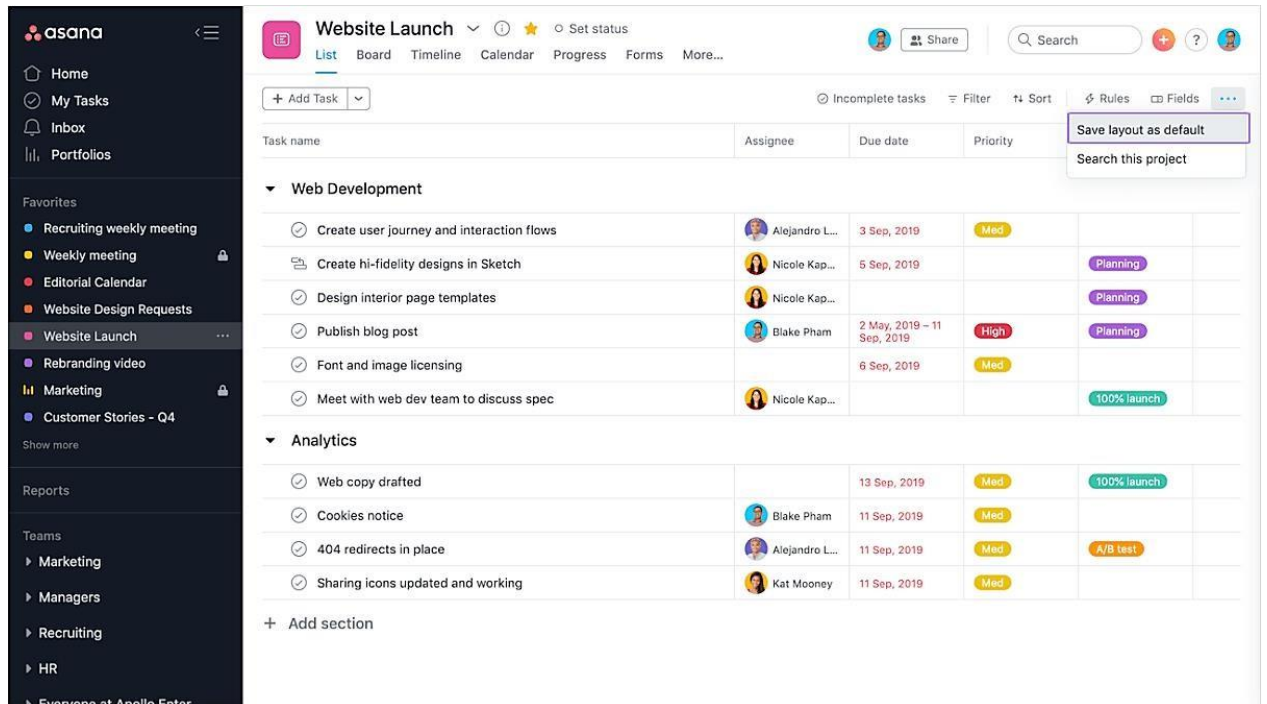


Рисунок 1.3 Інтерфейс Asana

Monday.com розглядається як інноваційна екосистема для менеджменту проєктів, що фокусується на візуалізації та глибокій оптимізації командної діяльності. Платформа надає адаптивний інструментарій для стратегічного планування та оперативного моніторингу робочих процесів. Ключовим архітектурним елементом системи є кастомізовані робочі простори, які дозволяють інтегрувати механізми автоматизації рутинних операцій та аналітичну візуалізацію даних через систему інтерактивних діаграм. Високий рівень інтеграційної сумісності з сервісами Google Workspace, Slack та Zoom сприяє створенню єдиного інформаційного середовища для комунікації. Завдяки модульній структурі та високій масштабованості, Monday.com демонструє однакову ефективність як у межах локальних робочих груп, так і у великих корпоративних структурах [12] (рис. 1.4).

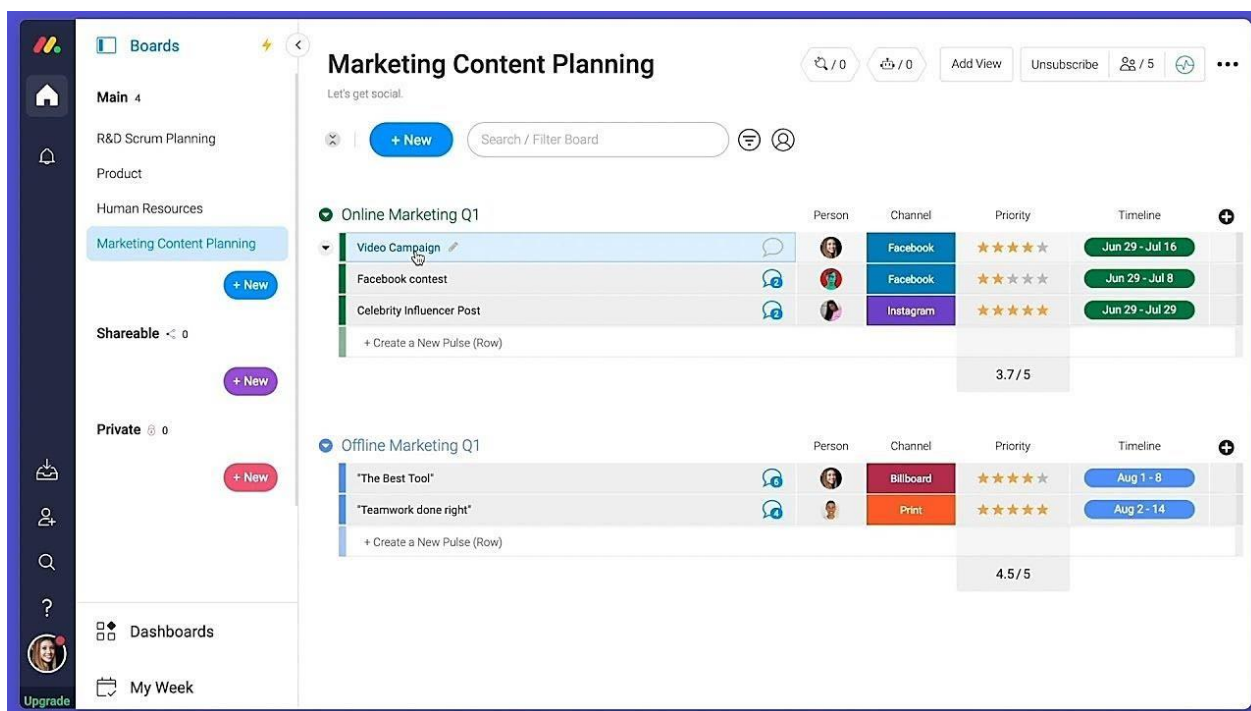


Рисунок 1.4 Інтерфейс Monday.com

Notion представляє собою мультифункціональне середовище для консолідації знань та управління проєктною діяльністю, що інтегрує в собі можливості текстових редакторів, реляційних баз даних та інтерактивних календарів. Диференційною ознакою платформи є її модульна архітектура, яка забезпечує користувачам безпрецедентну гнучкість у конфігуруванні персоналізованих робочих просторів. Система підтримує синхронну колаборацію в режимі реального часу, дозволяючи структурувати ієрархічні масиви даних та забезпечувати безшовний обмін інформацією через зовнішні API-інтеграції. Завдяки своїй універсальності, Notion отримав широке розповсюдження як в академічному середовищі для організації навчального процесу, так і в професійному менеджменті для створення комплексних баз знань [13] (рис. 1.5).



Projects & Tasks

- **Projects:** This is your overview of all the projects in the pipeline
- **Tasks:** This is your detailed breakdown of every task under your projects

By project Board All Weekly Team Throughput Average Delay by Team Load per person

Tasks

Marketing campaign 4

Task name	Status	Assignee	Due	Priority	Tags	Blocked By	Is Blocking
Develop advertising plan	Done	Nate Martins	April 13, 2024	Medium	Marketing Improvement		
Develop new creative assets	Done	Sohrab Amin	April 15, 2024	High	Marketing Branding		
Define target audience	Done	Sohrab Amin	April 17, 2024	Low	Marketing Improvement		
Report on marketing ROI	In Progress	Ben Lang	May 5, 2024	Medium	Marketing Improvement		

COMPLETE 3/4

Product launch 3

Task name	Status	Assignee	Due	Priority	Tags	Blocked By	Is Blocking
Create product demo video	Done	Nate Martins	April 29, 2024	High	Video production		
Create product positioning	Done	Sohrab Amin	May 1, 2024	High	Branding		
Monitor launch performance	In Progress	Ben Lang	May 9, 2024	High	Metrics		

COMPLETE 2/3

Research study 4

Task name	Status	Assignee	Due	Priority	Tags	Blocked By	Is Blocking
-----------	--------	----------	-----	----------	------	------------	-------------

COMPLETE 1/4

Рисунок 1.5 Інтерфейс Notion

Microsoft To Do позиціонується як прикладне рішення для менеджменту завдань, що пропонує ергономічний та результативний підхід до структурування персональних і професійних планів. Функціональний базис сервісу охоплює механізми формування переліків задач, детермінацію часових лімітів (дедлайнів), систему інтелектуальних нагадувань та пріоритезацію цілей. Ключовою перевагою інструменту є його безшовна інтеграція з екосистемою Microsoft 365 (зокрема з Outlook та Teams), що суттєво оптимізує процеси планування в межах корпоративної інфраструктури. Мінімалістичний інтерфейс у поєднанні з кросплатформною синхронізацією даних забезпечує зручність оперативного управління щоденною діяльністю [14] (рис. 1.6)

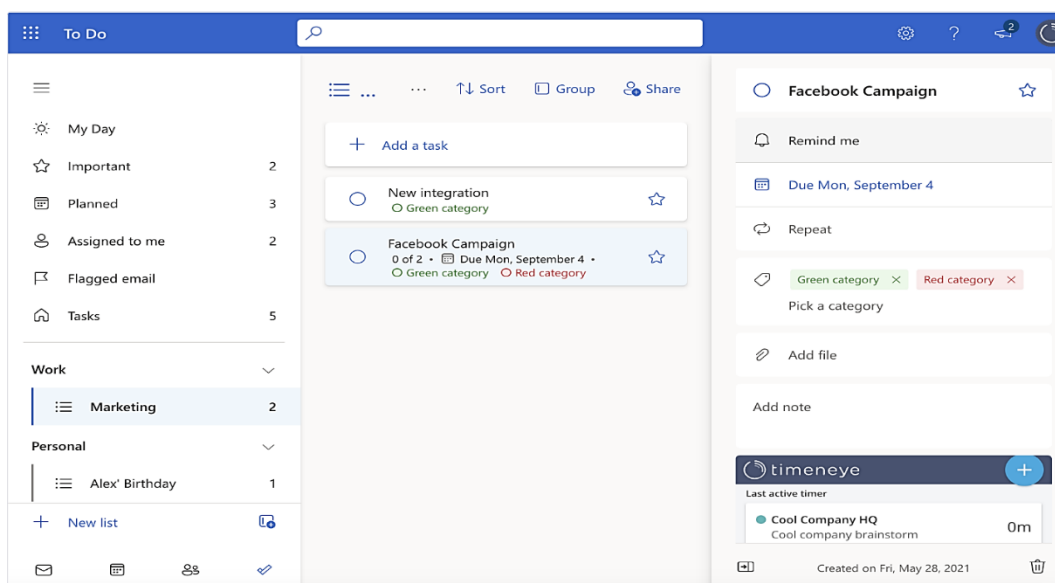


Рисунок 1.6 Інтерфейс Microsoft To Do

Зіставлення ключових характеристик аналізованих цифрових інструментів, сформоване на основі релевантних експертних даних [15–17], наведено в таблиці 1.1.

Таблиця 1.1 - Порівняльний аналіз існуючих систем керування завданнями

Платформа	Основний функціонал	Переваги	Недоліки	Наявність AI/інтелектуальних функцій
Trello	Візуальне управління завданнями (дошки, списки)	Простота використання, наочність організації завдань	Обмежена функціональність для складних проєктів	Відсутні
Asana	Планування завдань (проектні дошки, графіки)	Розширені функції управління, автоматизація процесів	Складність для нових користувачів, перевантаження функцій	Часткова інтеграція з AI (через Rule-based Automation)

Monday.com	Гнучке управління проектами, налаштування і робочі простори	Висока персоналізація, інтеграція з багатьма сервісами	Може бути дорогим, потребує часу на освоєння	Умовна інтелектуальність (AI-модулі за окрему плату)
Notion	Управління завданнями, нотатками, базами даних	Гнучкість, підтримка колаборації, організація великої кількості інформації	Складність для простих користувачів	Лише з підключенням Notion AI (опціонально, не за замовчуванням)
Microsoft To Do	Списки завдань, нагадування, дедлайни	Простий інтерфейс, інтеграція з продуктами Microsoft	Обмежений функціонал для командної роботи	Відсутні

На основі проведеного компаративного аналізу можна стверджувати, що вибір конкретного інструменту залежить від специфіки завдань проєкту. Trello та Microsoft To Do є оптимальними для вирішення лінійних завдань та персонального планування. Натомість Asana та Monday.com забезпечують необхідну глибину контролю для масштабних командних проєктів. Notion виділяється серед аналогів як найбільш універсальне середовище, що дозволяє консолідувати не лише завдання, а й великі обсяги неструктурованої інформації.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ДЕСКТОПНОГО ДОДАТКУ ДЛЯ ПЛАНУВАННЯ ТА УПРАВЛІННЯ ПОДІЯМИ

Мета та ідея проєкту

Створити зручний та функціональний десктопний додаток для планування та управління подіями з інтуїтивним інтерфейсом у стилі сучасних веб-додатків (Notion).

Ідея проєкту виникла з потреби у простому та ефективному інструменті для планування щоденних справ та подій. Більшість існуючих календарних додатків або надто складні, або мають застарілий інтерфейс.

Ключові принципи:

- Простота - інтуїтивний інтерфейс без зайвих функцій
- Швидкість - швидке створення та редагування подій
- Візуальна привабливість - сучасний дизайн у темній темі
- Локальне зберігання - дані зберігаються локально, без необхідності інтернету

Проблеми, яку вирішує проєкт

1. Відсутність зручного локального календаря - багато користувачів потребують простого інструменту без синхронізації з хмарою
2. Складність існуючих рішень - багато календарних додатків мають надто багато функцій, що ускладнює їх використання
3. Візуальне представлення часу - часовий розклад дозволяє краще бачити розподіл подій протягом дня

Цільова аудиторія:

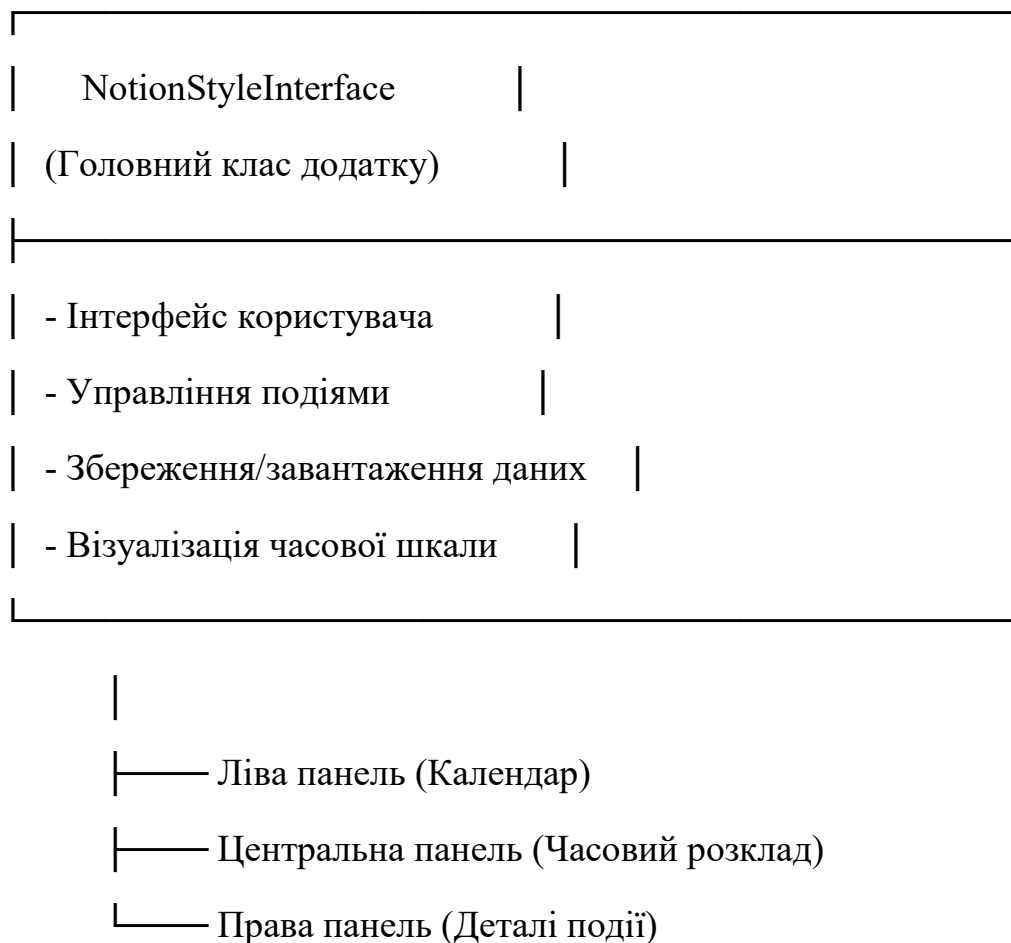
- Студенти для планування навчального процесу

- Фрілансери для управління проєктами та дедлайнами
- Будь-які користувачі, які потребують простого інструменту для планування

Архітектура та структура

Загальна архітектура

Проект побудований за принципом об'єктно-орієнтованого програмування з одним основним класом `NotionStyleInterface`, який інкапсулює всю логіку додатку.



Структура інтерфейсу

Додаток складається з трьох основних панелей:

1. Ліва панель (create_left_panel)

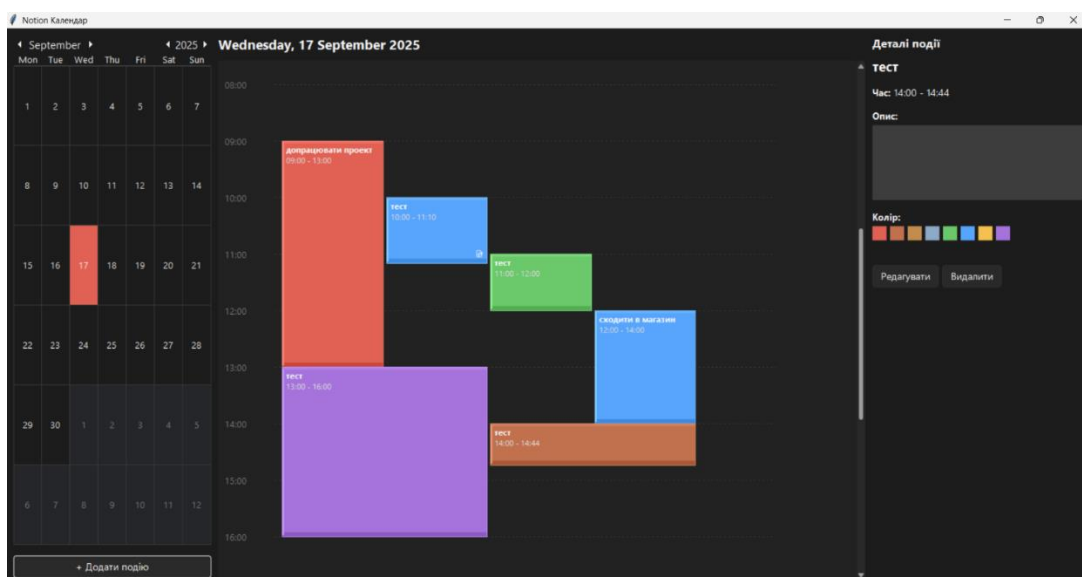
- Календар для вибору дати
- Кнопка додавання події

2. Центральна панель (create_main_panel)

- Заголовок з поточною датою
- Canvas з часовою шкалою (0-24 години)
- Відображення подій на часовій шкалі

3. Права панель (create_right_panel)

- Детальна інформація про вибрану подію
- Кнопки редагування та видалення
- Вибір кольору події



Структура даних

Подія (Event):

```
```python
{ 'id': str, Унікальний ідентифікатор (UUID)
 'title': str, Назва події
 'date': str, Дата у форматі ISO (YYYY-MM-DD)
 'start_time': str, Час початку (HH:MM)
 'end_time': str, Час завершення (HH:MM)
 'description': str, Опис події
 'color': str Колір у форматі HEX (RRGGBB)}
```

*Структура збереження:*

```
```python
{  "2025-09-12": [event1, event2, ...],
    "2025-09-13": [event1, event2, ...],  ...}
```

Основні функції

1. Управління подіями

Створення події

- Метод: `create\_event\_dialog (event=None)`
- Способи виклику:
 - Кнопка "+ Додати подію"
 - Контекстне меню (правий клік на часовій шкалі)
 - Гарячі клавіші `Ctrl+N`
- Функціонал:

- Введення назви події
- Вибір часу початку та завершення
- Додавання опису
- Вибір кольору з палітри

Редагування події

- Метод: ``edit_event(event_id)``
- Способи виклику:
 - Подвійний клік на події
 - Кнопка "Редагувати" на правій панелі
- Функціонал: Те саме, що й при створенні, але з попередньо заповненими полями

Видалення події

- Метод: ``delete_event(event_id)``
- Способи виклику:
 - Кнопка "Видалити" на правій панелі
 - Клавіша ``Delete`` після вибору події
- Функціонал: Видалення з підтвердженням через діалогове вікно

2. Візуалізація

Часовий розклад

- Метод: ``draw_timeline()``
- Функціонал:
 - Відображення 24 годин (00:00 - 23:59)

- Горизонтальні лінії для кожної години
- Мітки часу зліва
- Прокрутка для перегляду всього дня

Відображення подій

- Метод: ``display_event(event_data, position, total_in_slot)``
- Функціонал:
 - Розміщення подій на часовій шкалі відповідно до часу
 - Автоматичне розташування поруч при перекритті часу
 - Візуальні ефекти (градієнти, обведення)
 - Відображення назви та часу події

Групування перекриваючих подій

- Метод: ``group_events_by_time(events)``
- Алгоритм: Групує події, що перекриваються за часом, для правильного розташування поруч

3. Взаємодія з користувачем

Виділення події

- Метод: ``select_event(event_id)``
- Функціонал:
 - Підсвітка вибраної події
 - Відображення деталей на правій панелі
 - Ефект світіння навколо події

Зміна кольору

- Метод: ``change_event_color(event_id, color)``
- Функціонал: Швидка зміна кольору події без відкриття діалогу редагування

4. Збереження та завантаження

Завантаження подій

- Метод: ``load_events()``
- Файл: ``events.json``
- Функціонал: Завантаження всіх подій з JSON файлу при запуску

Збереження подій

- Метод: ``save_events()``
- Файл: ``events.json``
- Функціонал: Автоматичне збереження після кожної зміни

Детальний опис компонентів

Клас `NotionStyleInterface`

Конструктор (``__init__``)

```
```python
```

```
def __init__(self, root):
```

```
 self.root = root
```

```
 self.root.title("Notion Календар")
```

```
self.root.geometry("1400x800")
```

Застосовуємо Sun Valley тему

```
sv_ttk.set_theme("dark")
```

Ініціалізація даних

```
self.events = self.load_events()
```

```
self.current_date = datetime.now().date()
```

```
self.selected_event_id = None
```

Створення інтерфейсу

```
self.create_left_panel()
```

```
self.create_main_panel()
```

```
self.create_right_panel()
```

```
...
```

Що відбувається:

1. Налаштування головного вікна
2. Застосування темної теми Sun Valley
3. Завантаження збережених подій
4. Створення всіх панелей інтерфейсу
5. Прив'язка гарячих клавіш

Методи створення панелей

``create_left_panel()`` - Створює ліву панель з календарем

- Використовує бібліотеку ``tkcalendar.Calendar``
- Налаштовує кольори у темній темі
- Додає кнопку створення події

``create_main_panel()`` - Створює центральну панель з часовою шкалою

- Canvas для малювання часової шкали
- Scrollbar для прокрутки
- Прив'язка подій миші (клік, подвійний клік, правий клік)

``create_right_panel()`` - Створює праву панель для деталей

- Заголовок "Деталі події"
- Контейнер для динамічного відображення інформації

Методи роботи з подіями

``create_event_dialog(event=None)``

Створює діалогове вікно для створення/редагування події.

Особливості реалізації:

- Модальне вікно (блокуюче)
- Валідація введених даних
- Перевірка коректності часу (час завершення > часу початку)
- Візуальний вибір кольору з палітри

``add_event(event_data)``

Додає нову подію до словника подій.

Алгоритм:

1. Перевірка наявності дати в словнику
2. Додавання події до списку подій дати
3. Збереження у файл
4. Оновлення відображення (якщо це поточна дата)

```
`update_event(event_data)`
```

Оновлює існуючу подію.

Алгоритм:

1. Пошук події за ID у всіх датах
2. Видалення старої версії
3. Додавання оновленої версії (можливо, на іншу дату)
4. Збереження та оновлення відображення

```
`delete_event(event_id)`
```

Видаляє подію з підтвердженням.

Алгоритм:

1. Показ діалогу підтвердження
2. Пошук та видалення події
3. Збереження змін

#### 4. Очищення виділення та панелі деталей

Методи візуалізації

``draw_timeline()``

Малює часову шкалу на Canvas.

Алгоритм:

1. Очищення Canvas
2. Обчислення розмірів (ширина, висота години)
3. Малювання міток часу (00:00 - 23:00)
4. Малювання горизонтальних ліній
5. Збереження параметрів у ``timeline_props``
6. Оновлення області прокрутки

``display_events_for_date(date)``

Відображає всі події для заданої дати.

Алгоритм:

1. Скидання виділення та очищення Canvas
2. Отримання подій для дати
3. Групування подій за перекриваючимися інтервалами
4. Відображення кожної події з правильним розташуванням

``group_events_by_time(events)``

Групує події за перекриваючимися часовими інтервалами.

Алгоритм:

1. Сортування подій за часом початку
2. Для кожної події:
  - Перевірка перекриття з існуючими слотами
  - Якщо перекривається - додавання до слота
  - Якщо ні - створення нового слота
3. Повернення словника слотів

Приклад:

```
```python
```

Події:

A: 09:00-10:00

B: 09:30-10:30 (перекривається з A)

C: 11:00-12:00 (не перекривається)

Результат:

```
{
    "09:00-10:30": [A, B],  Група перекриваючих
    "11:00-12:00": [C]     Окрема подія
}
```

```
```
```

```
`display_event(event_data, position, total_in_slot)`
```

Відображає одну подію на часовій шкалі.

Алгоритм:

1. Обчислення позиції Y на основі часу початку/завершення
2. Обчислення ширини та позиції X (для розташування поруч)
3. Створення прямокутника події з градієнтом
4. Додавання тексту (назва, час)
5. Додавання декоративних елементів
6. Прив'язка обробників подій (клік)

Візуальні ефекти:

- Градієнт (світліша верхня частина, темніша нижня)
- Обведення світлішим кольором
- Ліва акцентна смуга
- Іконка опису (якщо є опис)

Методи взаємодії

```
`select_event(event_id)`
```

Обробляє вибір події користувачем.

Алгоритм:

1. Перевірка чи подія вже вибрана (скидання виділення)
2. Скидання попереднього виділення

3. Встановлення нової вибраної події
4. Застосування підсвітки
5. Відображення деталей на правій панелі

``highlight_event(event_id)``

Застосовує ефект підсвітки до події.

Ефекти:

- Яскравіша заливка (+35 до RGB)
- Контрастне обведення (+60 до RGB, товщина 2px)
- Ефект світіння навколо події

``reset_event_highlight(event_id)``

Скидає підсвітку події.

Алгоритм:

1. Відновлення оригінального кольору заливки
2. Відновлення оригінального обведення
3. Видалення ефекту світіння

---

Приклади коду

Приклад 1: Створення події через контекстне меню

```
```python
```

```
def create_event_at_click_position(self):
```

```
    """Створюємо подію в часі, де було викликано контекстне меню"""
```

```
    if not hasattr(self, 'context_click_y'):
```

```
        return
```

```
        Обчислюємо час на основі позиції кліку
```

```
        hour_height = self.timeline_props['hour_height']
```

```
        click_y = self.canvas.canvasy(self.context_click_y)
```

```
        Обчислюємо годину та хвилину з позиції кліку
```

```
        hour = max(0, min(23, int((click_y - 40) / hour_height)))
```

```
        minute = int(((click_y - 40) % hour_height) / hour_height * 60)
```

```
        minute = (minute // 15) * 15    Округлюємо до найближчих 15 хвилин
```

```
        Створюємо дані події
```

```
        event_data = {
```

```
            'id': str(uuid.uuid4()),
```

```
            'title': "Нова подія",
```

```
            'date': self.current_date.isoformat(),
```

```
            'start_time': f"{hour:02d}:{minute:02d}",
```

```
            'end_time': f"{hour+1:02d}:{minute:02d}",
```

```
            'description': "",
```

```
            'color': " #58a5fe"
```

```
}
```

Відкриваємо діалог редагування

```
self.create_event_dialog(event_data)
```

```
...
```

Пояснення:

- Метод викликається при створенні події через контекстне меню
- Обчислює час на основі Y-координати кліку на часовій шкалі
- Округлює хвилини до найближчих 15 (для зручності)
- Створює попередньо заповнені дані події
- Відкриває діалог редагування

Приклад 2: Групування перекриваючих подій

```
```python
```

```
def group_events_by_time(self, events):
```

```
 """Групуємо події за перекриваючимися часовими інтервалами"""
```

```
 sorted_events = sorted(events, key=lambda x: x['start_time'])
```

```
 time_slots = {}
```

```
 for event in sorted_events:
```

```
 start_time = event['start_time']
```

```
 end_time = event['end_time']
```

```
 placed = False
```

Перевіряємо, чи подія підходить до існуючого часового слоту

```
for slot_key, slot_events in time_slots.items():
```

```
 slot_start, slot_end = slot_key.split('-')
```

Перевіряємо перекриття: подія перекривається, якщо

не (кінець події  $\leq$  початок слота АБО початок події  $\geq$  кінець слота)

```
if not (end_time <= slot_start or start_time >= slot_end):
```

```
 time_slots[slot_key].append(event)
```

```
 placed = True
```

```
 break
```

Якщо подія не підходить до жодного існуючого слота, створюємо новий

```
if not placed:
```

```
 time_slots[f"{start_time}-{end_time}"] = [event]
```

```
return time_slots
```

```
'''
```

Пояснення:

- Алгоритм групує події, що перекриваються за часом
- Використовує математичну перевірку перекриття інтервалів
- Створює слоти для кожної групи перекриваючих подій
- Результат використовується для правильного розташування подій поруч

Приклад 3: Відображення події з візуальними ефектами

```
```python
```

```
def display_event(self, event_data, position, total_in_slot):
```

```
    """Відображаємо окрему подію на часовій шкалі зі стильним дизайном"""
```

```
    Обчислюємо позицію на основі часу
```

```
    start_hour, start_minute = map(int, event_data['start_time'].split(':'))
```

```
    end_hour, end_minute = map(int, event_data['end_time'].split(':'))
```

```
    start_y = 40 + (start_hour + start_minute/60) * hour_height
```

```
    end_y = 40 + (end_hour + end_minute/60) * hour_height
```

```
    Обчислюємо ширину для розташування поруч
```

```
    event_width = (events_area_width - 10) / total_in_slot
```

```
    event_x = events_start_x + (position * event_width)
```

```
    Створюємо градієнт
```

```
    base_color = event_data['color']
```

```
    r, g, b = self.hex_to_rgb(base_color)
```

```
    gradient_dark = self.rgb_to_hex(max(0, r-25), max(0, g-25), max(0, b-25))
```

```
    gradient_light = self.rgb_to_hex(min(255, r+15), min(255, g+15), min(255, b+15))
```

```
    Основний прямокутник
```

```
    event_rect = self.canvas.create_rectangle(
```

```
        event_x, start_y,
```

```
        event_x + event_width - 5, end_y,
```

```

fill=base_color, outline="", width=0,
tags=("event", event_data['id'])
)

```

Верхня світла смуга для градієнту

```

if end_y - start_y > 10:
    gradient_top_height = min(4, (end_y - start_y) * 0.1)
    self.canvas.create_rectangle(
        event_x, start_y,
        event_x + event_width - 5, start_y + gradient_top_height,
        fill=gradient_light, outline="",
        tags=("event", event_data['id'])
    )

```

Нижня темна смуга для градієнту

```

if end_y - start_y > 15:
    gradient_bottom_height = min(6, (end_y - start_y) * 0.15)
    self.canvas.create_rectangle(
        event_x, end_y - gradient_bottom_height,
        event_x + event_width - 5, end_y,
        fill=gradient_dark, outline="",
        tags=("event", event_data['id'])
    )

```

Обведення для стилю

```

self.canvas.create_rectangle(
    event_x, start_y,
    event_x + event_width - 5, end_y,
    outline=self.lighten_color(base_color, 30), width=1,
    tags=("event", event_data['id'])
)

```

Текст події

```

event_title = self.canvas.create_text(
    event_x + 6, start_y + 5,
    text=event_data['title'],
    anchor=tk.NW,
    fill=" ffffff",
    font=("Segoe UI", 9, "bold"),
    width=event_width - 12,
    tags=("event", event_data['id'])
)

```

...

Пояснення:

- Обчислює точну позицію події на основі часу початку/завершення
- Розраховує ширину для правильного розташування поруч з іншими подіями
- Створює візуальні ефекти (градієнт, обведення)

- Додає текст з обмеженням ширини для переносу

Приклад 4: Збереження та завантаження даних

```

```python
def load_events(self):
 """Завантаження подій з events.json"""
 if os.path.exists("events.json"):
 try:
 with open("events.json", "r", encoding='utf-8') as f:
 return json.load(f)
 except (json.JSONDecodeError, IOError):
 return {}
 return {}

def save_events(self):
 """Збереження подій у events.json"""
 with open("events.json", "w", encoding='utf-8') as f:
 json.dump(self.events, f, indent=4, ensure_ascii=False)
```

```

Пояснення:

- `load\_events()` - безпечно завантажує дані з обробкою помилок
- `save\_events()` - зберігає дані з форматуванням (indent=4) та підтримкою Unicode

- Використання `ensure\_ascii=False` дозволяє зберігати українські символи

Приклад 5: Обробка подвійного кліку

```

```python
def on_canvas_double_click(self, event):
 """Обробляємо подвійні кліки на canvas"""
 items = self.canvas.find_withtag("current")
 if items and "event" in self.canvas.gettags(items[0]):
 tags = self.canvas.gettags(items[0])
 for tag in tags:
 if tag in self.event_rects:
 self.select_event(tag)
 self.edit_event(tag)
 break
 ...

```

#### Пояснення:

- Знаходить елемент під курсором
- Перевіряє, чи це подія (тег "event")
- Знаходить ID події з тегів
- Викликає вибір та редагування події

---

## Алгоритми та логіка роботи

### Алгоритм відображення подій

...

1. Користувач вибирає дату в календарі



2. Викликається `display_events_for_date(date)`



3. Отримуємо список подій для дати з `self.events`



4. Викликаємо `group_events_by_time(events)`

- Сортуємо події за часом початку

- Групуємо перекриваючі події



5. Для кожної групи подій:

- Обчислюємо ширину кожної події (поділяємо простір порівну)

- Викликаємо `display_event()` для кожної події



6. `display_event()` малює подію на Canvas:

- Обчислює позицію Y (на основі часу)

- Обчислює позицію X та ширину (для розташування поруч)

- Створює прямокутник з візуальними ефектами

- Додає текст та декоративні елементи

- Прив'язує обробники подій

### Алгоритм перевірки перекриття подій

Математична формула перекриття двох інтервалів:

...

Інтервал A: [start\_A, end\_A]

Інтервал B: [start\_B, end\_B]

Перекриття відбувається, якщо НЕ виконується:

$end\_A \leq start\_B$  АБО  $start\_A \geq end\_B$

Тобто перекриття є, якщо:

$start\_A < end\_B$  І  $start\_B < end\_A$

...

Приклад:

- Подія A: 09:00-10:00

- Подія B: 09:30-10:30

- Перевірка: `09:00 < 10:30` (так) І `09:30 < 10:00` (так) → Перекриваються

Алгоритм обчислення позиції події

```python

Конвертація часу в координати Y

hour_height = 80 пікселів на годину

start_hour = 9

`start_minute = 30`

Y координата початку події

`start_y = 40 + (start_hour + start_minute/60) * hour_height`

`= 40 + (9 + 30/60) * 80`

`= 40 + 9.5 * 80`

`= 40 + 760`

`= 800 пікселів від верху`

```

Алгоритм розташування подій поруч

```python

Якщо є 3 події, що перекриваються:

`total_in_slot = 3`

`events_area_width = 600`

Ширина кожної події

`event_width = (600 - 10) / 3 = 196.67 пікселів`

Позиції X:

Подія 0: `events_start_x + 0 * 196.67`

Подія 1: `events_start_x + 1 * 196.67`

Подія 2: `events_start_x + 2 * 196.67`

Технології та бібліотеки

Використані бібліотеки

1. `tkinter` (вбудована)

- Призначення: Створення графічного інтерфейсу
- Використання:
 - Головне вікно (``tk.Tk``)
 - Віджети (``tk.Frame``, ``tk.Label``, ``tk.Button``)
 - Canvas для малювання часової шкали
 - Діалогові вікна (``tk.Toplevel``, ``messagebox``)

2. `tkcalendar`

- Призначення: Календарний віджет
- Версія: `>= 1.6.1`
- Використання: Ліва панель для вибору дати
- Особливості: Налаштування кольорів під темну тему

3. `sv-ttk` (Sun Valley ttk)

- Призначення: Сучасна тема для `ttk` віджетів
- Версія: `>= 2.0.0`
- Використання: Темна тема інтерфейсу
- Особливості: Стиль, схожий на Notion

4. json (вбудована)

- Призначення: Серіалізація даних
- Використання: Збереження/завантаження подій у файл `events.json`

5. datetime (вбудована)

- Призначення: Робота з датами та часом
- Використання:
 - Поточна дата
 - Форматування дат для відображення
 - Конвертація між форматами

6. uuid (вбудована)

- Призначення: Генерація унікальних ідентифікаторів
- Використання: Створення ID для кожної події

7. os (вбудована)

- Призначення: Робота з файловою системою
- Використання: Перевірка існування файлу `events.json`

Архітектурні рішення

1. Об'єктно-орієнтований підхід

- Один головний клас інкапсулює всю логіку
- Чітке розділення відповідальності методів

2. Подійно-орієнтована архітектура

- Використання callback-функцій для обробки подій
- Прив'язка обробників до елементів Canvas

3. Дані як словник

- Проста структура даних (словник Python)
- Легке серіалізування в JSON
- Швидкий доступ за ключем (дата)

4. Canvas для візуалізації

- Гнучкість у малюванні
- Можливість створення складних візуальних ефектів
- Ефективна робота з великою кількістю елементів

Переваги обраних технологій

1. Tkinter:

- Вбудована в Python (не потрібна додаткова установка)
- Крос-платформенність
- Простота використання

2. JSON для збереження:

- Людино-читабельний формат
- Легко редагувати вручну
- Стандартний формат обміну даними

3. Canvas для часової шкали:

- Повний контроль над візуалізацією
- Можливість створення складних ефектів
- Ефективна робота з динамічним контентом

Висновок

Проект "Notion Календар" є повнофункціональним додатком для планування подій з сучасним інтерфейсом. Він демонструє використання об'єктно-орієнтованого програмування, роботу з графічним інтерфейсом та ефективні алгоритми для візуалізації даних.

Ключові досягнення:

- Інтуїтивний інтерфейс у стилі Notion
- Ефективна візуалізація часового розкладу
- Автоматичне розташування перекриваючих подій
- Надійне збереження даних
- Крос-платформенність

Можливі покращення:

- Додавання нагадувань про події
- Експорт/імпорт даних
- Синхронізація з хмарою
- Пошук та фільтрація подій
- Повторювані події

РОЗДІЛ 3. ТЕСТУВАННЯ ТА СУПРОВІД РОЗРОБЛЕНОГО ДОДАТКУ

Метою етапу тестування є підтвердження відповідності розробленого десктопного додатка функціональним та нефункціональним вимогам, а також гарантування стабільності роботи інтерфейсу, натхненного сучасними мінімалістичними рішеннями (Notion-style).

Для перевірки було обрано комплексний підхід, що включає:

- **Функціональне тестування (Black Box):** перевірка роботи кнопок, форм введення, фільтрації подій та збереження даних.
- **Тестування інтерфейсу (UI/UX Testing):** верифікація інтуїтивності навігації та адаптивності елементів десктопного додатка.
- **Тестування продуктивності:** оцінка швидкості відгуку інтерфейсу при великій кількості запланованих подій.

Функціональна верифікація компонентів

У ході функціонального тестування було перевірено життєвий цикл події в додатку: від створення та редагування до архівації або видалення.

Ключові сценарії тестування:

1. **Управління подіями:** коректність відображення нових планів у календарі та списках. Перевірено роботу валідаторів дат (неможливість встановлення події в минулому часі без відповідного попередження).
2. **Синхронізація даних:** перевірка цілісності даних при перезапуску десктопного додатка (робота з локальною базою даних або кешем).
3. **Пошук та фільтрація:** тестування швидкості пошуку за ключовими словами в назвах справ.

Тестування зручності користування

Оскільки головною ідеєю проєкту є створення «інтуїтивного інтерфейсу у стилі Notion», особлива увага була приділена тестуванню юзабіліті. Для оцінки було залучено групу потенційних користувачів (метод цільових груп).

Параметри оцінки:

- **Швидкість навчання:** час, за який новий користувач створює свою першу подію без використання довідки.
- **Естетичне задоволення:** оцінка відповідності дизайну сучасним стандартам (мінімалізм, використання «білого простору», типографіка).
- **Навігаційна глибина:** кількість кліків, необхідних для доступу до основних функцій додатка (згідно з тестуванням, цей показник не перевищує 2–3 кліки).

Супровід та потенціал розвитку додатка

Супровід розробленого рішення передбачає технічну підтримку та поступове нарощування функціоналу. На основі проведеного тестування було визначено план подальшого вдосконалення додатка:

1. **Адаптивний супровід:** оновлення компонентів інтерфейсу для підтримки нових версій операційних систем (Windows/macOS/Linux).
2. **Масштабованість:** у наступних версіях планується впровадження хмарної синхронізації між пристроями та інтеграція з зовнішніми календарями (Google Calendar API).
3. **Автоматизація оновлень:** впровадження механізму сповіщення користувача про наявність нових версій із можливістю автоматичного завантаження виправлень (patches).

ВИСНОВКИ

У кваліфікаційній роботі розв'язано актуальне завдання розробки десктопного додатка для планування та управління подіями, що поєднує високу функціональність із сучасними стандартами UI/UX-дизайну. На основі проведеного дослідження та процесу розробки було зроблено такі висновки:

1. Проаналізовано сучасний стан ринку програмного забезпечення для тайм-менеджменту. Встановлено, що більшість існуючих рішень мають або надмірну функціональність, що перевантажує користувача, або застарілий інтерфейс, який знижує ефективність взаємодії. Це підтвердило доцільність створення легкого та інтуїтивно зрозумілого інструменту.

2. Обґрунтовано вибір концепції «Notion-style» як базису для графічного інтерфейсу. Використання принципів мінімалізму, великого обсягу «білого простору» та блочної структури дозволило створити середовище, яке мінімізує когнітивне навантаження на користувача та фокусує його увагу на продуктивності.

3. Розроблено архітектуру та функціонал додатка, що забезпечує повний цикл управління щоденними справами: від швидкого створення подій до їх пріоритезації та відстеження дедлайнів. Реалізована система навігації дозволяє користувачеві отримувати доступ до ключових функцій за мінімальну кількість ітерацій.

4. Проведено комплексне тестування розробленого програмного продукту. Результати функціональної перевірки та тестування зручності користування (Usability Testing) підтвердили стабільність роботи системи та високий рівень задоволеності користувачів ергономікою додатка. Встановлено, що розроблений інтерфейс є значно доступнішим для швидкого освоєння порівняно з традиційними календарними системами.

5. Визначено практичну цінність проєкту, яка полягає у створенні готового до експлуатації десктопного рішення, що може бути використане як

індивідуальними користувачами для особистого планування, так і в межах невеликих команд для координації спільної діяльності.

Перспективи подальшого розвитку проєкту вбачаються в інтеграції з хмарними сервісами для синхронізації даних між різними пристроями та впровадженні інтелектуальних алгоритмів проактивного нагадування на основі аналізу поведінки користувача.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] 42 time management statistics to inspire efficiency in 2024 - zoomshift. Zoomshift. URL: <https://www.zoomshift.com/blog/time-management-statistics/>
- [2] 8 task management strategies for your team | nifty blog. Nifty Blog | Delivering our thoughts and findings on the future of project management!. URL: <https://niftypm.com/blog/task-management-strategies/> .
- [3] ADP Research Data-Driven Business Insights. URL: <https://www.adpresearch.com/wp-content/uploads/2024/04/People-at-Work-2024-A-Global-Workforce-View.pdf> (дата звернення: 24.05.2025).
- [4] New Time Management Statistics 2024. Timewatch. URL: <https://www.timewatch.com/blog/time-management-statistics/>.
- [5] Time management statistics you need to know in 2024 - timeular. Timeular. URL: <https://timeular.com/blog/time-management-statistics/>.
- [6] Vaskiv R., Veretennikova N. Features of the use of information and communication technologies to support project processes in distributed teams. Computer systems and information technologies. 2023. № 4. С. 36–43. URL: <https://doi.org/10.31891/csit-2023-4-5>.
- [7] Yeoh S. Y. . Employee onsite task assignment management and tracking app (Doctoral dissertation, UTAR). 2023. URL: <http://eprints.utar.edu.my/id/eprint/6106>.
- [8] Oluwaseun Abiola Ajiva, Onyinye Gift Ejike, Angela Omozele Abhulimen. Advances in communication tools and techniques for enhancing collaboration among creative professionals. International journal of frontiers in science and technology research. 2024. T. 7, № 1. С. 066–075. URL: <https://doi.org/10.53294/ijfstr.2024.7.1.0049>.
- [9] Performance-Enhancing impact of user profiling / I. Horváth та ін. Studies in big data. Cham, 2024. С. 271–284. URL: https://doi.org/10.1007/978-3-031-68130-1_15

- [10] Керуйте проєктами своєї команди звідусіль Trello. Manage Your Team's Projects From Anywhere Trello. URL: <https://trello.com/uk>
- [11] Manage your team's work, projects, & tasks online • Asana • Asana. Asana. URL: <https://asana.com/>
- [12] Monday.com Your go-to work platform. monday.com. URL: <https://monday.com/>
- [13] Your connected workspace for wiki, docs & projects | Notion. Notion. URL: <https://www.notion.so/>
- [14] Програма Microsoft To Do. Найкраща програма для складання списків справ, яка стане вам у пригоді під час роботи й відпочинку. Microsoft To Do. URL: <https://to-do.office.com/tasks/>.
- [15] Asana vs. trello vs. monday.com: which project management platform is right for you?. ProBackup-Backup your apps in just 3 minutes. URL: <https://www.probackup.io/blog/asana-vs-trello-vs-monday-com-which-project-management-platform-is-right-for-you>.
- [16] Trello vs asana vs monday vs clickup: the ultimate guide. Productive. URL: <https://productive.io/blog/trello-vs-asana-vs-monday-vs-clickup/>.
- [17] 12 best alternatives to microsoft to do in december 2024 | secret. 12 Best Alternatives to Microsoft To Do in December 2024 Secret. URL: <https://www.joinsecret.com/microsoft-to-do/alternatives>
- [18] Imani Z. S., Pamungkas A. F. A., Jamaluddin M. Efektivitas teknik to- do list terhadap time management mahasiswa psikologi UIN malang dalam belajar. Jurnal psikologi. 2024. T. 1, № 4. C. 9. URL: <https://doi.org/10.47134/pjp.v1i4.3136> .
- [19] Prioritizing tasks in software development: a systematic literature review / Y. Bugayenko та ін. Plos one. 2023. T. 18, № 4. C. e0283838. URL: <https://doi.org/10.1371/journal.pone.0283838>.
- [20] Pedersen M., Muhr S. L., Dunne S. Time management between the personalisation and collectivisation of productivity: The case of adopting the

Pomodoro time-management tool in a four-day workweek company. *Time & society*. 2024. URL: <https://doi.org/10.1177/0961463x241258303>

[21] Moreano-Arones N. S., Vasquez-Suejiro E., Quiroz-Flores J. C. Kanban board. *Lean manufacturing in latin america*. Cham, 2024. C. 175–192. URL: https://doi.org/10.1007/978-3-031-70984-5_8

[22] Layton R., Domegan C., Duffy S. M. Getting things right; diagnose and design in the evolution of community provisioning systems. *Journal of macromarketing*. 2022. C. 027614672211091. URL: <https://doi.org/10.1177/02761467221109136>

[23] Medeossi G., Longo G., de Fabris S. A method for using stochastic blocking times to improve timetable planning. *Journal of rail transport planning & management*. 2023. T. 1, № 1. C. 1–13. URL: <https://doi.org/10.1016/j.jrtpm.2011.07.001>

[24] Cynthia Chizoba Ekechi, Chukwuekem David Okeke, Henry Ejiga Adama. Enhancing agile product development with scrum methodologies: a detailed exploration of implementation practices and benefits. *Engineering science & technology journal*. 2024. T. 5, № 5. C. 1542–1570. URL: <https://doi.org/10.51594/estj.v5i5.1108>.

[25] Mahmud I., Veneziano V. Mind-mapping: an effective technique to facilitate requirements engineering in agile software development. 2021 14th international conference on computer and information technology (ICCIT), м. Dhaka, Bangladesh, 22–24 гpyд. 2021 p. 2021. URL: <https://doi.org/10.1109/iccitechn.2011.6164775>

[26] Opeyemi Abayomi Odejide, Tolulope Esther Edunjobi. Ai in project management: exploring theoretical models for decision-making and risk management. *Engineering science & technology journal*. 2024. T. 5, № 3. C. 1072–1085. URL: <https://doi.org/10.51594/estj.v5i3.959>

[27] Ahmetoglu Y., Brumby D., Cox A. Bridging the gap between time management research and task management app design: a study on the integration of planning fallacy mitigation strategies. *CHIWORK 2024: annual symposium on human-*

computer interaction for work, м. Newcastle upon Tyne United Kingdom. New York, NY, USA, 2024. URL: <https://doi.org/10.1145/3663384.3663404> (

[28] Artificial intelligence enabled project management: a systematic literature review / I. Taboada та ін. Applied sciences. 2023. Т. 13, № 8. С. 5014. URL: <https://doi.org/10.3390/app13085014>

[29] Recommender systems in the era of large language models (llms) / Z. Zhao та ін. IEEE transactions on knowledge and data engineering. 2024. С. 1– 20. URL: <https://doi.org/10.1109/tkde.2024.3392335>

[30] Exploring the Impact of Large Language Models on Recommender Systems: An Extensive Review / A. Vats та ін. arXiv preprint arXiv:2402.18590. 2024. URL: <https://doi.org/10.48550/arXiv.2402.18590>

[31] Personalized recommendation systems powered by large language models: Integrating semantic understanding and user preferences / F. Shang International Journal of Innovative Research in Engineering and Management, 2024. Т. 11., № 4. С. 39-49. URL: <http://ijirem.irpublications.org/index.php/ijirem/article/view/59>

[32] A review of modern recommender systems using generative models (gen-recsys) / Y. Deldjoo та ін. KDD '24: the 30th ACM SIGKDD conference on knowledge discovery and data mining. Barcelona Spain. New York, NY, USA, 2024. С. 6448–6458. URL: <https://doi.org/10.1145/3637528.3671474> (

[33] Large language models meet collaborative filtering: an efficient all- round llm-based recommender system / S. Kim та ін. KDD '24: the 30th ACM SIGKDD conference on knowledge discovery and data mining,. Barcelona Spain. New York, NY, USA, 2024. С. 1395–1406. URL: <https://doi.org/10.1145/3637528.3671931>

[34] Widayanti R. Improving recommender systems using hybrid techniques of collaborative filtering and content-based filtering. Journal of applied data sciences. 2023. Т. 4, № 3. С. 289–302. URL: <https://doi.org/10.47738/jads.v4i3.115> (дата звернення: 24.05.2025).

- [35] Hybrid recommender system model for digital library from multiple online publishers / P. Jomsri та ін. F1000Research. 2024. Т. 12. С. 1140. URL: <https://doi.org/10.12688/f1000research.133013.3>
- [36] Zhang Q., Lu J., Jin Y. Artificial intelligence in recommender systems. Complex & intelligent systems. 2020. URL: <https://doi.org/10.1007/s40747-020-00212-w>.
- [37] Wang Y., Han L. Adaptive time series prediction and recommendation. Information processing & management. 2021. Т. 58, № 3. С. 102494. URL: <https://doi.org/10.1016/j.ipm.2021.102494>
- [38] Jangid M., Kumar R. Deep learning approaches to address cold start and long tail challenges in recommendation systems: a systematic review. Multimedia tools and applications. 2024. URL: <https://doi.org/10.1007/s11042-024-20262-3>
- [39] Alfaifi Y. H. Recommender systems applications: data sources, features, and challenges. Information. 2024. Т. 15, № 10. С. 660. URL: <https://doi.org/10.3390/info15100660> (дата звернення: 24.05.2025).
- [40] An intuitive explanation of transformer-based models. Factored | Machine Learning, Data Engineering and Data Analytics Company. URL: <https://factored.ai/transformer-based-language-models/>.
- [41] Time management recommendations with bayesian neural networks: an intelligent assistant approach / A. Kumar. 2024 IEEE international conference on communication, computing and signal processing (IICCCS), ASANSOL, India, 2024. С. 1–6. URL: <https://doi.org/10.1109/iicccs61609.2024.10763590>
- [42] Designing a user-centric ai-driven mobile app for personalized time management: integrating machine learning and design thinking / P. Saardloun та ін. 2024 23rd international symposium on communications and information technologies (ISCIT), Bangkok, Thailand, 2024 p. 2024. С. 212–217. URL: <https://doi.org/10.1109/iscit63075.2024.10793695>

- [43] Jeromela J., Conlan O. Devising scrutable user models for time management assistants. UMAP '24: 32nd ACM conference on user modeling, adaptation and personalization, Cagliari Italy. New York, NY, USA, 2024. URL: <https://doi.org/10.1145/3631700.3665182>
- [44] IntelliAid: a personal assistant chat-bot for enhanced task management / K. Prathiksha 20245th international conference on electronics and sustainable communication systems (ICESC), Coimbatore, India, 2024. C. 432–436. URL: <https://doi.org/10.1109/icesc60852.2024.10689912>
- [45] Yang Y. Artificial intelligence and the evolution of managerial expertise in the digital age: how AI has impacted managerial skills. Pakistan journal of life and social sciences (PJLSS). 2024. T. 22, № 2. URL: <https://doi.org/10.57239/pjlss-2024-22.2.001130>