

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій,
обліку та фінансів
Кафедра комп'ютерних технологій
і моделювання систем

Кваліфікаційна робота
на правах рукопису

БАЦУЛА Михайло Геннадійович

УДК 004.4:004.73

КВАЛІФІКАЦІЙНА РОБОТА

Проектування інформаційної системи для планування і досягнення
особистих цілей

122 “Комп'ютерні науки”

(шифр і назва спеціальності)

Подається на здобуття освітнього ступеня бакалавр

кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело

_____ Бацула М.Г.

Керівник роботи
Николюк Ольга Миколаївна
доктор економічних наук, професор

Житомир – 2025

Висновок кафедри _____

за _____ результатами _____ попереднього _____ захисту:

Протокол _____ засідання _____ кафедри _____

№ _____ від « _____ » _____ 20 _____ р.

Завідувач кафедри _____

_____ (науковий ступінь, вчене звання) _____ (підпис) _____ (прізвище, ім'я, по батькові)

« _____ » _____ 20 _____ р.

Результати захисту кваліфікаційної роботи

Здобувач вищої освіти Бацула Михайло Генадійович _____ захистив

(прізвище, ім'я, по батькові)

кваліфікаційну роботу з оцінкою: _____

сума балів за 100-бальною шкалою _____

за шкалою ECTS _____

за національною шкалою _____

Секретар ЕК

_____ науковий ступінь, вчене звання) _____ (підпис) _____ (прізвище, ім'я, по батькові)

АНОТАЦІЯ

Бацула М.Г. Проектування інформаційної системи для планування і досягнення особистих цілей за спеціальністю 122 – Комп’ютерні науки. – Поліський національний університет, Житомир, 2025.

У роботі розроблено інтерактивну веборієнтовану інформаційну систему Goal Planner, яка призначена для планування, моніторингу й досягнення особистих цілей користувача. Об’єктом дослідження є процес проектування та розробки інтерактивної інформаційної системи у вигляді веб- і мобільного додатку для планування й реалізації особистих цілей, а метою – створення цифрового інструменту, що забезпечує підтримку саморозвитку через SMART-методологію, імплементаційні наміри та аналітичну візуалізацію прогресу. Система реалізована на основі клієнт–серверної архітектури з використанням технологій Node.js, Express, React, Vite, TailwindCSS та MySQL. У роботі розроблено UML-діаграми, IDEF-моделі, структуру бази даних, REST API та адаптивний UX/UI-дизайн. Реалізовано функції створення, редагування та аналізу цілей, ведення підзадач, інтерфейс дашборду, профілю й аналітичної звітності. Практичне застосування системи можливе для студентів, фахівців і всіх, хто прагне підвищити особисту ефективність у повсякденній або професійній діяльності.

Ключові слова: особисті цілі, SMART, інформаційна система, планування, вебзастосунок, UX/UI, аналітика, React, Node.js.

ANNOTATION

Batsula M.H. Design of an Information System for Personal Goal Planning and Achievement, specialty 122 – Computer Science. – Polissia National University, Zhytomyr, 2025.

This work presents the development of an interactive web-based information system called Goal Planner, designed for planning, monitoring, and achieving users' personal goals. The object of the study is the process of designing and developing an interactive information system in the form of a web and mobile application for planning and implementing personal goals. The aim is to create a

digital tool that supports self-development through the SMART methodology, implementation intentions, and analytical visualization of progress. The system is implemented based on a client–server architecture using technologies such as Node.js, Express, React, Vite, TailwindCSS, and MySQL. The work includes the development of UML diagrams, IDEF models, a database structure, a REST API, and responsive UX/UI design. The system provides functionality for creating, editing, and analyzing goals, managing subtasks, and includes a dashboard interface, user profile, and analytical reporting. The system can be practically applied by students, professionals, and anyone seeking to enhance their personal effectiveness in daily or professional activities.

Keywords: personal goals, SMART, information system, planning, web application, UX/UI, analytics, React, Node.js.

ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЙ.....	9
1.1. Характеристика об'єкта інформатизації та його інформаційні потреби	9
1.2 Моделювання процесу автоматизації постановки та досягнення цілей	11
Висновки до першого розділу.....	12
2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПЛАНУВАННЯ І ДОСЯГНЕННЯ ОСОБИСТИХ ЦІЛЕЙ.....	13
2.1 Структура інформаційної системи для планування і досягнення особистих цілей.....	13
2.2 Моделювання програмного забезпечення.....	14
2.3 Проєктування бази даних.....	19
2.4 Розробка прототипу користувацького інтерфейсу (UX/UI-дизайн).....	21
Висновки до другого розділу.....	22
3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ Goal Planer.....	24
3.1. Програмна реалізація.....	24
3.2 Інтерфейс Goal Planner.....	27
3.3 Тестування інформаційної системи Goal Planner.....	29
3.4 Інструкція користувача.....	31
Висновок до третього розділу.....	34
Список використаних джерел.....	35
ДОДАТКИ.....	39

ВСТУП

У сучасному динамічному середовищі, що характеризується швидкими змінами як у професійному, так і в особистому житті, ефективне планування та реалізація цілей набуває особливої ваги. Багато користувачів стикаються з проблемою надмірної складності існуючих інструментів або недостатньою адаптивністю додатків до індивідуальних особливостей процесу досягнення завдань. Це породжує необхідність створення інтерактивної інформаційної системи, яка забезпечуватиме не лише збереження даних про SMART-цілі, а й надаватиме персоналізовані підказки, механізми декомпозиції й пріоритизації завдань, візуалізацію прогресу та своєчасні нагадування із врахуванням контексту використання (зокрема мобільних пристроїв та офлайн-режим). Комплексний підхід до підтримки процесів постановки SMART-цілей, реалізації імплементаційних намірів і аналітики активності користувача сприятиме підвищенню мотивації й ефективності саморегуляції.

Об'єктом дослідження є процес проєктування та розробки інтерактивної інформаційної системи у вигляді веб- і мобільного додатку для планування й реалізації особистих цілей. Предметом дослідження виступають методи, засоби та інструменти цифрової підтримки постановки цілей, їх пріоритизації, декомпозиції на підзадачі, відображення прогресу й формування зворотного зв'язку користувачеві за допомогою інтерактивних механізмів і аналітичних віджетів. Мета роботи полягає у створенні інформаційної системи, що надасть користувачеві засоби для формулювання особистих цілей, планування й моніторингу прогресу, аналітичної оцінки ефективності та своєчасного нагадування про поставлені завдання.

Для досягнення поставленої мети вирішено такі завдання:

- аналіз предметної області;
- розробка вимог до системи;
- реалізація і тестування функціоналу;
- впровадження інтерфейсу.

За результатами виконання роботи підготовлено публікації:

1. Бацула М.Г., Николук О. Візуалізація КРІ-прогресу як рушій ефективності в інформаційних системах персонального планування // Global Trends in Science: Research, Innovation and Development : proceedings of the 1st International Scientific and Practical Conference (June 23–25, 2025, Varna, Bulgaria). Varna : Collection of Scientific Papers, 2025. – С. 116–118.
2. Бацула М.Г., Николук О. Інформаційна система як інструмент персонального тайм-менеджменту та досягнення цілей// Evolving Science: Theories, Discoveries and Practical Outcomes : proceedings of the 4th International Scientific and Practical Conference (June 9–11, 2025, Zurich, Switzerland). Zurich : Collection of Scientific Papers, 2025– С. 99–102.

Практична цінність полягає в наданні універсального інструменту для різних категорій користувачів (студентів, фахівців, тих, хто прагне підвищити особисту ефективність), який може застосовуватися в особистісному розвитку, навчанні, коучингу та корпоративних середовищах для підвищення продуктивності та мотивації. Результати цієї роботи сприятимуть створенню зручного, адаптивного та надійного додатку для планування і досягнення особистих цілей, відповідаючи сучасним вимогам користувачів і технічним можливостям веб- і мобільних технологій.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЙ

1.1. Характеристика об'єкта інформатизації та його інформаційні потреби

Дослідження в галузі мотивації та ефективності показують, що специфічність і складність SMART-цілі мають прямий вплив на результативність. Згідно з мета-теорією Едвіна Лока та Гері Летема, чітко сформульовані й помірковано складні завдання підвищують продуктивність на 25–30 %, тоді як нереалістично короткі дедлайни створюють стрес і сприяють прокрастинації[1]. У цьому контексті внутрішня мотивація, яку описують Деці й Райан, зокрема усвідомлення власної цінності мети у поєднанні з конструктивним зворотним зв'язком, забезпечує стійку залученість на строк до шести місяців, після чого без додаткових стимулів або переналаштування окремих завдань мотивація починає знижуватися[2].

Аналітика поведінки користувачів у цифрових ігрових та навчальних платформах підтверджує ефективність імплементаційних намірів: формулювання «якщо – тоді» дозволяє автоматизувати перехід від рішення до дії й підвищує ймовірність як короткострокових, так і довгострокових цілей до 50 %[3]. Окрім того, дослідження Lagodko показують, що додавання регулярних проміжних обговорень (FAST-елемент) у SMART-процес постановки цілей підвищує успішність групових цілей на 18 % і індивідуальних – на 12 %[4].

Процес особистого планування цілей без використання спеціальних застосунків починається зі свідомого формулювання: користувач виписує базову «ідею» та відразу перетворює її на SMART-мету, коригуючи її під власний тижневий ресурс часу. За спостереженнями HURMA, понад 68 % людей втрачають фокус саме через відсутність методології при формулюванні SMART-цілі[5].

Далі настає фаза розподілу ресурсів: користувач фіксує в календарі вільні вікна та визначає пріоритети, враховуючи, що оптимальне

навантаження не перевищує 60 % загального часу, аби уникнути вигорання[6]. Потім мета дробиться на 5–7 проміжних кроків, адже надмірна деталізація (понад 12 підзадач) знижує ефективність плану на 40 %, тоді як оптимальна кількість кроків підвищує ймовірність реалізації на 30 %[6].

На етапі деталізації користувач використовує табличні інструменти із умовним форматуванням, щоб візуалізувати дедлайни. Однак лише 48 % людей підтримують регулярний моніторинг понад 90 днів без додаткових мотиваційних елементів, що підкреслює необхідність використання кольорових індикаторів і нагадувань у календарі[7].

Моніторинг супроводжується щотижневим збором даних про виконання та часовими мітками. Застосування імплементаційних намірів («якщо – тоді») автоматизує перехід від рішення до дії й підвищує відсоток виконання на 50 %[3]. Ретроспективні обговорення з «ментором» допомагають виявити закономірні затримки – наприклад, у другому тижні місяця – й вчасно коригувати стратегію.

Підсумкове коригування базується на зібраній аналітиці: дедлайни переносяться, пріоритети змінюються, зайві завдання видаляються. Такий гнучкий підхід підтримує рівень завершення цілей на позначці близько 73 %, тоді як жорстке дотримання початкового розкладу без адаптацій дає лише 55–60 % успіху[6][7]. На додаток А.1 представлено узагальнену IDEF0-діаграму основних кроків особистого планера.

Така послідовність, поєднуючи класичні теорії мотивації й сучасні методики постановки цілей, дозволяє побудувати ефективний особистий планер на основі простих інструментів і підтримувати баланс між амбіційними завданнями та реалістичними ресурсами.

1.2 Моделювання процесу автоматизації постановки та досягнення цілей

У розділі автоматизованого особистого планера цифрове рішення переходить від ручного введення даних до інтелектуальної підтримки користувача в реальному часі. Поєднання AI-коуча, інтерактивного дашборда та механізмів імплементаційних намірів створює середовище, яке не лише фіксує ціль, а й активно супроводжує її виконання.

Перший аналітичний аспект полягає в тому, що автоматизація знижує когнітивне навантаження: замість ручного оновлення таблиць користувач отримує оновлений прогрес натисканням однієї кнопки. За даними McKinsey, цифрові платформи для управління цілями підвищують оперативність ухвалення рішень на 20 % завдяки інтеграції даних і рекомендацій AI[8]. Крім того, поєднання AI-коуча та persuasive-елементів забезпечує додаткову мотивацію: згідно з IEEE, такий підхід підтримує залученість користувачів на рівні 60 % навіть після шести місяців використання[10].

На основі попереднього аналізу, логіку функціонування цифрового особистого планера можна детально представити за допомогою діаграми IDEF3 наведено на додаток А.2, яка моделює процес автоматизованого формування чітких підзадач на основі введених користувачем цілей.

На першому етапі (об'єкти 1–3) відбувається ініціалізація SMART-цілі: користувач вводить бажану мету (1), система встановлює терміни виконання (2) та автоматично або за участю користувача формує пріоритети (3). Це дає змогу одразу визначити ключові параметри реалізації SMART-цілей[5].

Далі запускається модуль III, який здійснює розподіл цілей на підзадачі (4). Алгоритми машинного навчання враховують тип задачі, її пріоритет та історичні дані користувача. Це дозволяє сформувати логічну ієрархію дій, не перевантажуючи користувача надмірною деталізацією. У даному випадку на діаграмі позначено, що після цього активується

синхронна вилка J1: паралельно запускаються процеси аналізу середньої витрати часу на підзадачі (5) та аналіз навантаження (6).

Ці два блоки дозволяють адаптувати список завдань до реального ритму користувача, уникаючи типових проблем, як-от надмірне планування або недостатня деталізація. Наприклад, якщо аналіз навантаження виявляє перевищення робочої норми, система пропонує перегляд графіка або делегування підзадач[7].

На завершальному етапі (7), після об'єднання результатів блоків 5 та 6 через синхронізатор J2, формується чіткий, персоналізований список підзадач із оптимізованими часовими витратами. Це є основою для створення динамічного дашборда, інтегрованого з календарем, трекером звичок і системою нагадувань.

Таким чином, застосування IDEF3-моделі дозволяє детально описати не лише технічну реалізацію автоматизації цілей, а й відображає логіку прийняття рішень у системі, побудованій на III та персоналізації. Ця модель відповідає сучасним практикам цифрової трансформації особистісного тайм-менеджменту[8],[9].

Висновки до першого розділу

У результаті проведеного аналізу було визначено, що сучасний підхід до особистісного планування вимагає не лише чіткої структури цілей, але й гнучкого, адаптивного інструменту, який поєднує перевірені теорії мотивації з цифровими технологіями. Розгляд класичних моделей (SMART, FAST, імплементаційні наміри) засвідчив: ефективне досягнення мети можливе лише за умов чіткої постановки, візуалізації прогресу та постійного моніторингу, які підтримуються як внутрішньою мотивацією, так і зовнішніми підказками.

Проаналізовано традиційні методи планування, які, хоч і мають переваги на початкових етапах, є недостатніми для довготривалого

підтримання продуктивності. Зокрема, відсутність автоматичного нагадування, адаптації до навантаження та персоналізованого аналізу знижує рівень завершення цілей майже на 20 %.

Вивчення сучасних цифрових рішень показало перспективність поєднання AI-коучингу, автоматизованого поділу цілей та візуальних інтерфейсів. Такий підхід суттєво знижує когнітивне навантаження та підвищує користувацьку залученість до 60 % навіть у довготривалій перспективі.

Модель, описана через IDEF3-діаграму, наочно демонструє логіку переходу від введення SMART-цілі до генерації персоналізованого переліку підзадач, що враховують часові обмеження, пріоритетність і поточне навантаження користувача. Це дозволяє забезпечити високу адаптивність і реалістичність планування, з урахуванням життєвого ритму та індивідуальних потреб.

Отже, обґрунтовано необхідність створення інтелектуального цифрового планера, який використовує моделі машинного навчання, інтерактивні інтерфейси та адаптивну логіку управління цілями. Такий підхід відповідає сучасним тенденціям в інформатизації особистого простору й дозволяє підвищити ефективність реалізації цілей до 70–75 % завдяки автоматизації, персоналізації та підтримці з боку цифрових інструментів.

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПЛАНУВАННЯ І ДОСЯГНЕННЯ ОСОБИСТИХ ЦІЛЕЙ

2.1 Структура інформаційної системи для планування і досягнення особистих цілей

Інформаційна система для особистісного планування має забезпечувати не лише збереження даних про SMART-цілі користувача, а й їх адаптивну підтримку протягом усього циклу реалізації. Для цього архітектура системи повинна включати низку функціональних модулів, що взаємодіють між

собою у режимі реального часу, створюючи єдине інтерактивне середовище. Архітектура системи створення та взаємодії з цілю наведено на додаток Б.1.

1 Модуль автентифікації до профілю користувача. Відповідає за безпечний доступ до персональних даних, зберігає історію цілей, стиль продуктивності та часові обмеження користувача. Саме цей модуль забезпечує персоналізацію планера, яка критично важлива для створення релевантних підказок та мотиваційних сценаріїв.

2 Інтерфейс користувача. Візуалізує прогрес цілей та дає змогу взаємодіяти з цілями дивитись продуктивність та редагувати SMART-цілі та підзадачі.

3,4 Створення SMART-цілі та заповнення потрібних даних. Дає змогу користувачу ввести бажану мету, структурувати її за SMART-критеріями та одразу застосувати механізм імплементаційних намірів (формат "якщо – тоді"), який згідно з дослідженнями збільшує ймовірність досягнення SMART-цілі на понад 50 %[2].

7,8 Запис SMART-цілі та перехід на deshbord, взаємодія з цілями та виконання завдань. Перенаправляє користувача на сторінку взаємодії з створеною цілю.

9 Відслідковування прогресу SMART-цілі.

2.2 Моделювання програмного забезпечення

Одним із ключових етапів розробки програмного забезпечення є його моделювання, що дозволяє формалізувати структуру системи, описати поведінку її компонентів та забезпечити розуміння як розробниками, так і зацікавленими сторонами. Для сучасних веб-додатків найбільш розповсюдженим і ефективним інструментом моделювання є мова UML (Unified Modeling Language), яка стандартизована та широко підтримується в галузі програмної інженерії[11].

На рисунку 2.1 представлено діаграму прецедентів, яка відображає основні функціональні можливості користувача у веб-додатку для управління SMART-цілями. Зокрема, користувач має змогу пройти процес реєстрації, що передбачає створення нового облікового запису для подальшої роботи із системою. Після успішної реєстрації або при наявності існуючого акаунту користувач може виконати вхід у систему, отримуючи доступ до особистого кабінету. У межах цього кабінету передбачено можливість перегляду дашборду – центрального інтерфейсу, який відображає всі активні SMART-цілі, а також функціонал для їхнього управління: додавання нових цілей, редагування існуючих записів, а при потребі – видалення неактуальних або завершених цілей. Крім того, користувач має можливість переглядати список власних задач, що пов'язані з певними цілями, додавати нові задачі до відповідної SMART-цілі, редагувати або видаляти наявні задачі. Завершивши роботу із системою, користувач може здійснити вихід, що гарантує збереження конфіденційності його даних. Такий набір функцій забезпечує гнучке, безпечне й інтуїтивно зрозуміле управління персональними завданнями й цілями, сприяючи ефективній організації користувацької діяльності.



Рисунок 2.1 – Діаграма прецедентів

Діаграма послідовності (Sequence Diagram) є динамічним видом UML-діаграм, яка описує покрокову взаємодію між учасниками системи у часі. Вона ідеально підходить для візуалізації послідовності повідомлень та подій, що відбуваються під час реалізації конкретних функціональних сценаріїв. Для нашого додатку діаграми послідовності деталізують процеси, починаючи від введення даних користувачем на клієнтському інтерфейсі, їх передачі на сервер, обробки запитів бекендом, взаємодії із базою даних та повернення результатів назад клієнту. Наприклад, при реєстрації користувача відбувається надсилання запиту на сервер, де відбувається хешування пароля і створення запису у базі даних, після чого сервер генерує JWT-токен для подальшої аутентифікації. Ця діаграма допомагає не лише у розумінні логіки роботи системи, а й слугує цінним інструментом для розробників API, тестувальників і для документування системи[13]. На рисунку 2.3 показані послідовності ключових процесів взаємодії користувача з веб-додатком – входу в систему (логіну) та реєстрації нового користувача. Ця діаграма

послідовності ілюструє покроковий обмін повідомленнями між користувачем, клієнтським інтерфейсом (Frontend), серверною частиною (Backend) та базою даних.

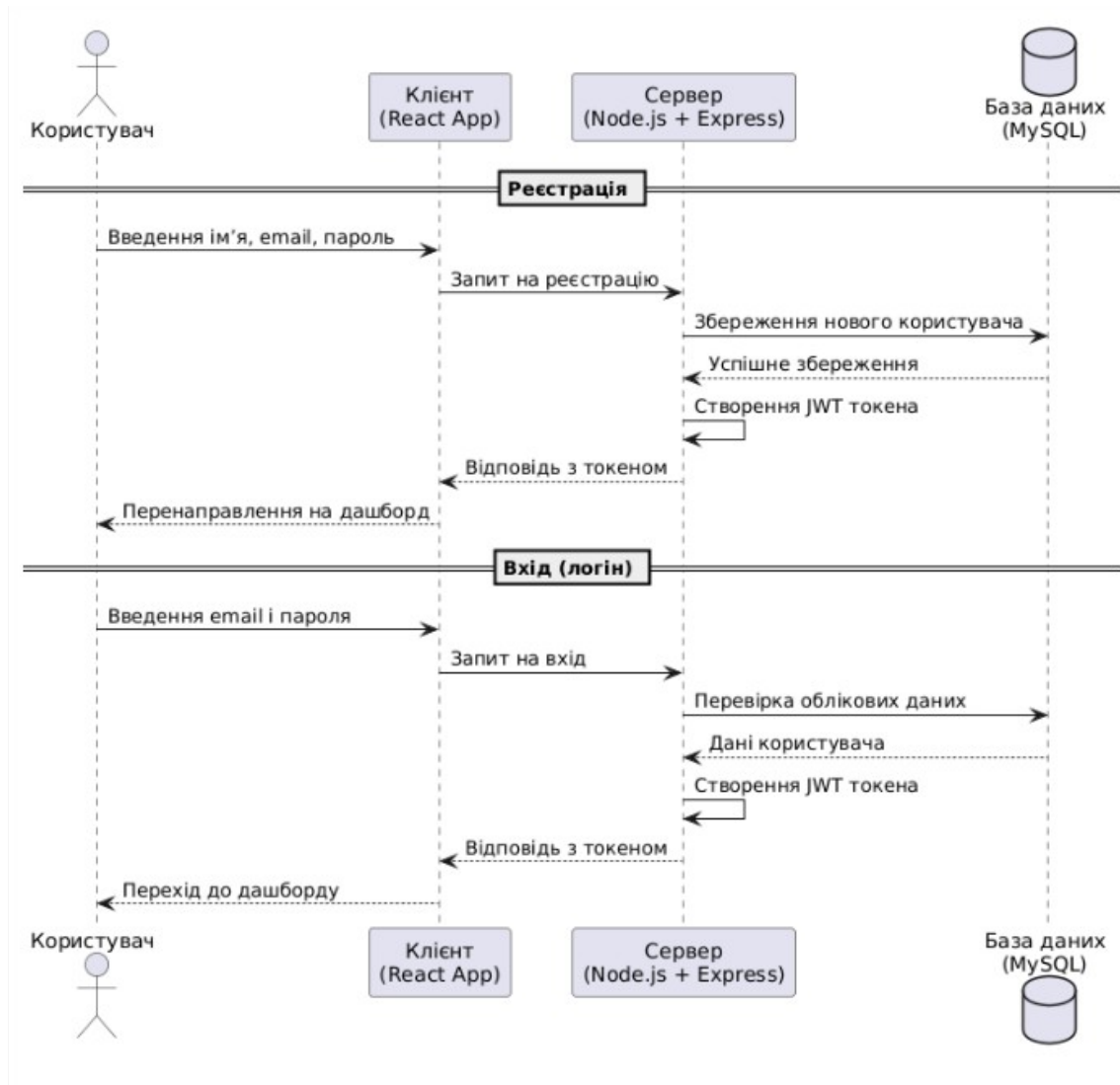


Рисунок 2.3 – Діаграма послідовностей для реєстрації

Процес реєстрації починається з того, що користувач вводить необхідні дані – ім'я, електронну адресу та пароль – у відповідну форму на фронтенді. Після цього фронтенд надсилає POST-запит на сервер з інформацією для створення нового облікового запису. Сервер приймає цей запит, проводить валідацію даних та хешування пароля для безпечного зберігання. Далі відбувається запис користувача у базу даних. Після успішного створення облікового запису сервер генерує JWT-токен, який використовується для подальшої аутентифікації користувача, та повертає його клієнту разом із

повідомленням про успішну реєстрацію. Клієнт приймає цей токен і оновлює інтерфейс, надаючи користувачу доступ до захищених частин додатку.

Аналогічно працює процес входу в систему: користувач вводить електронну адресу та пароль, фронтенд відправляє ці дані на сервер через POST-запит. Сервер перевіряє відповідність отриманих даних із записами в базі, і у випадку успішної аутентифікації формує JWT-токен для підтвердження прав доступу користувача. Цей токен надсилається назад клієнту, який після цього надає користувачу доступ до функціоналу додатку.

На додаток 2.4 наведено UML-діаграму послідовності, що ілюструє дві ключові операції користувача в системі: створення нової SMART-цілі та перегляд дашборду із власними цілями та задачами. Дана діаграма відображає покроковий потік взаємодії між користувачем, клієнтським застосунком (Frontend), серверною частиною (Backend) і базою даних (DB).

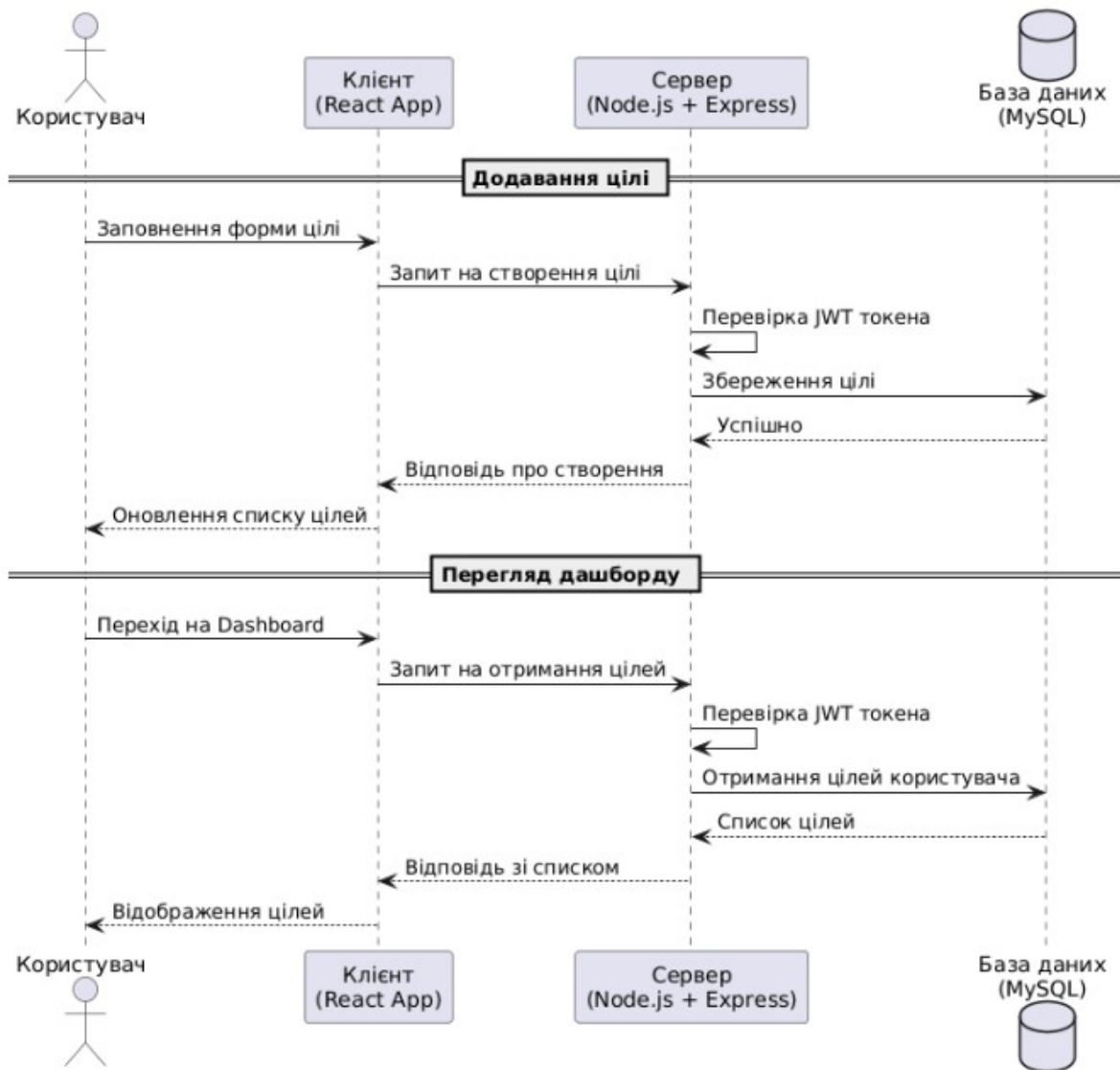


Рисунок 2.4 – Діаграма послідовностей для додавання та взаємодії з цілю

У першій частині діаграми показано, як користувач ініціює створення SMART-цілі, вводячи необхідні дані через інтерфейс додатку. Після цього фронтенд надсилає HTTP POST-запит на сервер із деталями нової SMART-цілі. Сервер приймає запит, виконує перевірку авторизації користувача за допомогою middleware, а потім зберігає інформацію в базу даних. Після успішного збереження сервер повертає підтвердження, і фронтенд оновлює інтерфейс, відображаючи новостворену ціль у списку користувача.

У другій частині діаграми демонструється процес отримання даних для дашборду. Користувач переходить на відповідну сторінку, після чого фронтенд здійснює GET-запит до бекенду для отримання актуального списку

цілей та пов'язаних із ними задач. Сервер, знову ж таки, перевіряє авторизацію, виконує запит до бази даних і повертає отримані дані у форматі JSON. Клієнт приймає відповідь і відображає інформацію у зручному для користувача вигляді.

Дана діаграма послідовності дозволяє зрозуміти деталі реалізації та взаємодії між компонентами системи, а також підкреслює важливість контролю доступу і коректного оброблення запитів. Вона слугує цінним інструментом для розробників, тестувальників та аналітиків, допомагаючи виявити потенційні вузькі місця або помилки в логіці роботи додатку .

UML-діаграма класів (Class Diagram), зображена на додаток Б.2, розкриває основні сутності системи, їх атрибути і взаємозв'язки, що є фундаментом для структурування даних та логіки додатку. У випадку веб-додатку представлені основні сутності – користувач (User), ціль (Goal) та задача (Task), які мають чітко визначені атрибути, наприклад, ідентифікатор, назву, опис, дату створення чи дедлайн, а також відносини між ними, такі як "один до багатьох". Це дозволяє чітко уявити структуру даних і способи їх зберігання у реляційній базі даних. Також на діаграмі відображено основні серверні компоненти – контролери, маршрути і middleware, які реалізують бізнес-логіку, маршрутизацію HTTP-запитів та забезпечують безпеку через аутентифікацію. Наявність такої діаграми суттєво полегшує розробку програмного коду та сприяє його підтримуваності в майбутньому[14].

2.3 Проєктування бази даних

Проєктування бази даних починається з виділення основних сутностей предметної області: користувачі (users), SMART-цілі (goals), задачі (tasks), теги (tags), активності (activity) та проміжна таблиця goal_tags для реалізації зв'язку «багато-до-багатьох» між цілями й тегами. Зв'язки реалізовані через зовнішні ключі з каскадним видаленням, що забезпечує цілісність даних та відповідає вимогам нормалізації [16]. Для статусів цілей використовується

перерахування enum, що уніфікує допустимі значення і зменшує ризик помилок [17]. Застосування UML-діаграм дозволяє візуально перевірити структуру моделі перед її реалізацією в СУБД [18].

Логічна схема розроблена відповідно до третьої нормальної форми, що дозволяє уникнути дублювання даних та полегшує обслуговування системи [19]. Особлива увага приділена індексації первинних та зовнішніх ключів для пришвидшення запитів, оптимальному вибору типів даних (наприклад, varchar, datetime, enum) і моделюванню сценаріїв паралельного доступу [20], [21], [22]. Для підвищення продуктивності рекомендовано використовувати інструменти аналізу запитів (EXPLAIN), а також за необхідності впроваджувати кешування чи денормалізацію [23], [24]. Застосування MySQL Workbench та дотримання практик оптимізації дає змогу досягти балансу між структурною строгістю та швидкодією системи [25].

На рисунку 2.2 наведена діаграма бази даних, яка демонструє ключові сутності системи та їх взаємозв'язки. Зображено таблиці users, goals, tasks, activity, tags і проміжну таблицю goal_tags, причому кожна сутність містить набір атрибутів із відповідними первинними ключами.

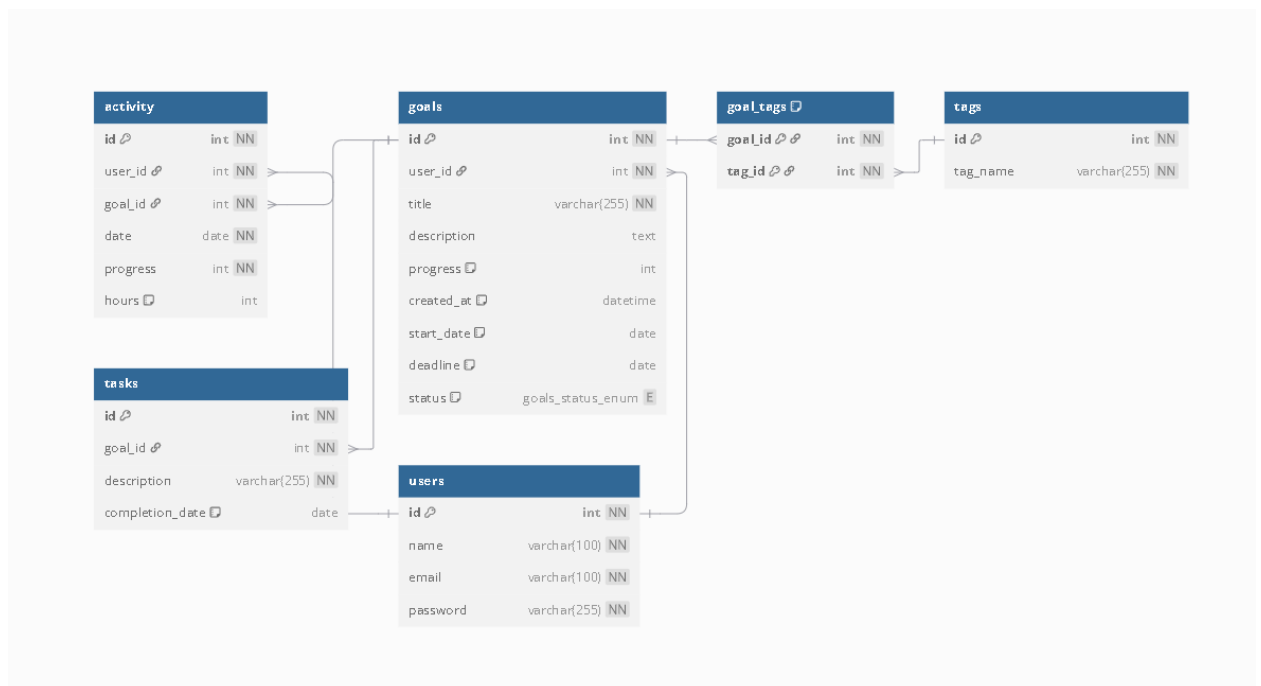


Рисунок 2.2 – ERD діаграма бази даних

Висновки до другого розділу

У другому розділі було здійснено комплексне моделювання й опис архітектури інформаційної системи для планування й досягнення особистих цілей. Спочатку сформульовано структуру системи, виділено ключові функціональні модулі: автентифікації, інтерфейсу користувача та механізму роботи з цілями й задачами. За допомогою UML-моделювання виконано деталізацію поведінки системи через діаграми прецедентів, що описують усі основні сценарії взаємодії користувача – реєстрацію, вхід, роботу з дашбордом, створення, редагування і видалення цілей та задач, а також вихід із системи. Діаграми послідовності продемонстрували покрокові потоки даних між клієнтом, сервером і базою даних для сценаріїв реєстрації, логіну, створення SMART-цілі та перегляду дашборду, що дало змогу виявити ключові точки обробки та перевірки доступу. Частковий опис діаграми класів розкрив основні сутності системи й взаємозв'язки між ними та серверними компонентами (контролерами, маршрутами, middleware), встановивши основу для реалізації бізнес-логіки й структури коду.

Паралельно виконано проєктування бази даних: ідентифіковано сутності (users, goals, tasks, tags, activity) та їх атрибути, описано зв'язки через зовнішні ключі із каскадним видаленням, що забезпечує SMART-цілісність даних та автоматичне видалення залежних записів. Складено таблиці та схему БД із урахуванням нормалізації, реалізовано enum для статусу цілей, проміжну таблицю для зв'язку «багато-до-багатьох» між цілями і тегами, підібрано оптимальні типи даних і індекси для прискорення запитів. Окрім того, розроблено прототип UX/UI: визначено цільову аудиторію, обрано принципи простоти, послідовності та доступності, підібрано кольорову гаму, шрифти й іконки, спроектовано навігацію й адаптивну сітку за допомогою Figma. Таким чином, другий розділ підсумовує побудову моделі системи на концептуальному, логічному та

інтерфейсному рівнях, створюючи міцну базу для подальшої розробки й впровадження проєкту.

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ Goal Planer

3.1. Програмна реалізація Goal Planner

Загальна структура проекту поділена на фронтенд і бекенд, які розташовані в папках frontend/ та backend/. У backend/ виділено модулі контролерів (authController.js, goalController.js, taskController.js), моделей даних (User.js, Goal.js, Task.js), маршрутів (authRoutes.js, goalRoutes.js, taskRoutes.js) та middleware (наприклад, authMiddleware.js для перевірки JWT), а також конфігураційні файли (.env, config.js). Файл server.js (або app.js) ініціалізує Express-сервер, підключає JSON-парсер, CORS, логування та маршрути. Папка database/ містить скрипти міграцій і налаштування ORM. Контролери реалізують бізнес-логіку, моделі описують структуру даних, маршрути визначають шляхи API, а middleware обробляє крос-зрізні завдання. В таблиці 3.1 наведено за допомогою яких реалізовано проект.

Таблиця 3.1– засоби реалізації

Що реалізовано	Якими засобами
Бекенд-логіка	Node.js + Express
Клієнтський інтерфейс	React + Vite
База даних	MySQL
UI-дизайн	Tailwind, Figma

Алгоритми взаємодії реалізовані на фронтенді через обробку подій форм (перевірка email, довжина пароля) та відправку запитів Axios. Після успішної валідації користувач отримує JWT, який зберігається в HTTP-only cookie або localStorage, і інтерфейс автоматично оновлюється, переводячи його на Dashboard без перезавантаження.

На бекенді Express Router передає запити до контролерів, де через express-validator перевіряються дані, а middleware розбирає JWT із заголовка Authorization. Бізнес-логіка (хешування паролів, генерація токенів, операції з БД) реалізована в моделях, а єдиний error-middleware формує уніфіковані відповіді. Безпеку забезпечують bcrypt, jsonwebtoken, helmet, CORS, CSRF-захист і rate limiting.

3.2 Інтерфейс Goal Planner

Інтерфейс Goal Planner виконано у сучасному мінімалістичному стилі з акцентом на простоту, зручність та мобільність. Основна кольорова гама складається зі світлих нейтральних відтінків, у яких для активних елементів використано яскравий зелений колір, що підвищує впізнаваність кнопок та прогрес-барів. Всі елементи мають округлі кути та чіткі межі, достатній простір між блоками, що забезпечує комфортну роботу на мобільних пристроях і запобігає випадковим натисканням. Навігація організована таким чином, щоб користувач у будь-який момент мав швидкий доступ до ключових розділів, а масштабованість елементів гарантує читабельність на різних розмірах екранів.

Сторінка зі списком цілей являє собою центральний екран після входу в систему і містить заголовок “Список цілей” із відображенням поточної дати. Під заголовком розташовано іконку сповіщення з індикатором кількості нових подій, що повідомляє про оновлення або термінові нагадування. Кожна ціль представлена окремою карткою, на якій наведено короткий опис, прогрес-бар із відсотком виконання, інформацію про загальний витрачений час і позначку активного місяця. Завдяки чіткому розташуванню та достатньому відступу картки легко сприймаються на будь-якому пристрої. Нижнє навігаційне меню містить іконки для переходу до Дашборду, Додавання нової SMART-цілі, Переліку цілей та Профілю користувача, що робить перемикання між розділами швидким і інтуїтивним. Сторінку зі списком цілей зображено на рисунку 3.1.

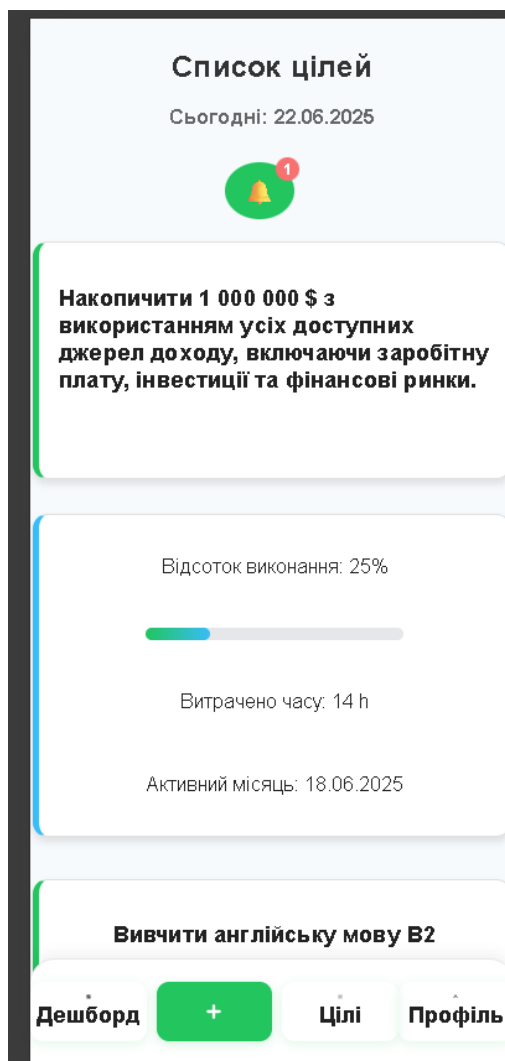


Рисунок 3.1 – Головна сторінка зі списком цілей

Сторінка додавання нової SMART-цілі забезпечує введення необхідної інформації через просту форму з полями “Назва SMART-цілі”, “Дата початку”, “Дедлайн” та “Опис SMART-цілі”. Для вибору дат використано зручні date-picker-и, які адаптовані до мобільних пристроїв та дозволяють швидко встановити час виконання. Дві основні кнопки – “Згенерувати підзадачі” (зелена, акцентна) і “Повернутися до списку цілей” – розташовані внизу форми для оперативного виконання дій. Всі поля мають підписи та плейсхолдери, що допомагають зрозуміти очікуваний формат введення. Даний екран реалізовано із достатньою відстанню між елементами, що знижує когнітивне навантаження. Сторінку додавання SMART-цілі зображено на рисунок 3.2.

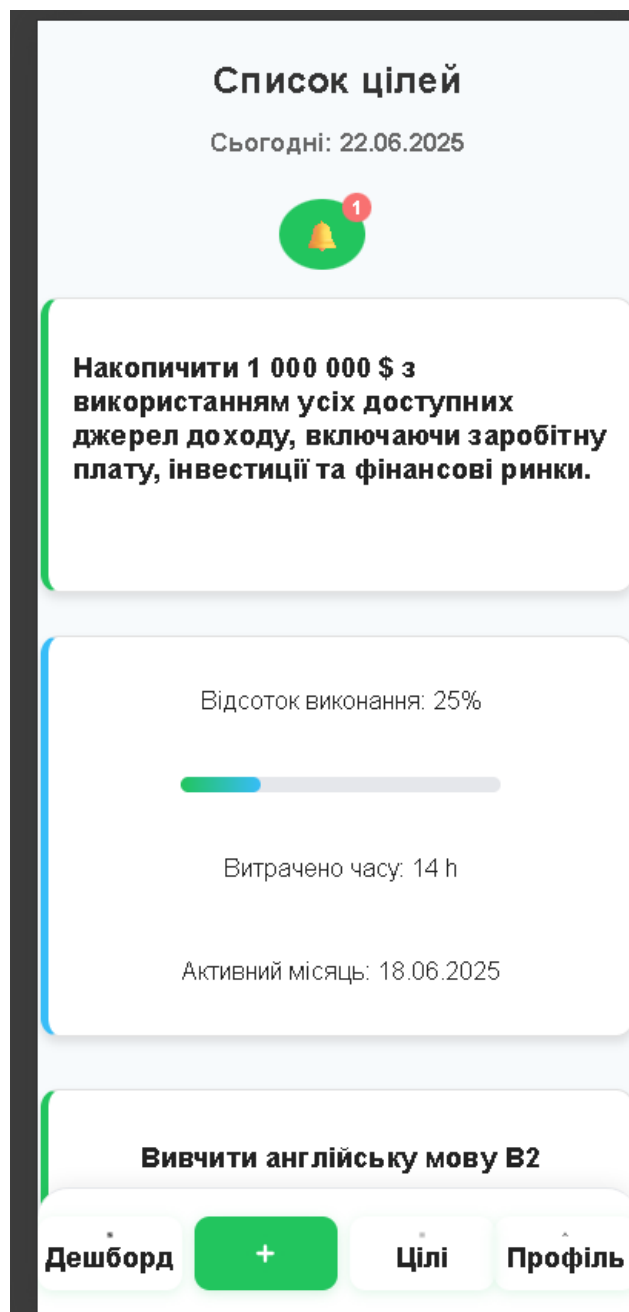


Рисунок 3.2 – Головна сторінка зі списком цілей

Сторінка особистого кабінету (профілю) відображає ім'я, email користувача та два графіки: кількість виконаних підSMART-цілей і витрачені години за останні 12 місяців. Графіки мають підписи по осях, оформлені у фірмових кольорах з акцентом на зелений для успіхів. Інтерфейс адаптивний: на мобільних пристроях графіки відображаються по черзі, на десктопі – поруч (див. додаток В.3).

Детальна сторінка SMART-цілі містить назву цілі, дату, опис, теги, прогрес-бар, дані про витрачений час і графік активності за 8 місяців.

Передбачено навігацію між цілями та перехід до підцілей. Інформація представлена у зручному вигляді для відстеження динаміки (див. додаток В.4).

Інтерфейс адаптований до різних розмірів екранів, має зручне меню, великі клікабельні елементи та мінімум відволікаючих деталей. Візуалізація прогресу мотивує користувача до досягнення цілей і самоаналізу.

3.3 Тестування інформаційної системи Goal Planner

У ході функціонального тестування перевірено відповідність реалізованого функціоналу вимогам специфікацій і UML-моделей. Перевірка здійснювалась вручну через інтерфейс: реєстрація, вхід з валідними та невалідними даними, робота з цілями та задачами, включно з обробкою граничних значень, а також захищені маршрути із JWT. В таблиці 3.2 наведено результати функціонального тестування.

Таблиця 3.2 – Результати функціонального тестування

№	Функція	Очікуваний результат	Результат	Примітки
1	Реєстрація	Користувача додано	Успішно	Відсутні
2	Невірний пароль	Повідомлення про помилку	Успішно	Відсутні
3	Вхід у систему	Успішний вхід	Успішно	Відсутні
4	Додавання цілі	Ціль додана	Успішно	Відсутні
5	Редагування задачі	Зміни збережено	Успішно	Відсутні
6	Видалення задачі	Запис видалено	Успішно	Відсутні
7	Захищений маршрут	Доступ за токеном, відмова без	Успішно	Відсутні

Автоматизація тестування здійснювалася за допомогою Postman для REST API та юніт-тестів на Jest, що дозволило виявити помилки ще на ранніх етапах. Тестування безпеки охоплювало перевірку авторизації згідно з рекомендаціями OWASP.

Юзабіліті оцінювалось за участю цільової аудиторії, застосовувалась методика модераторованого тестування. За результатами внесено покращення інтерфейсу, адаптивності та підказок.

Виявлені помилки (винятки при незвичних датах, некоректне відображення прогрес-барів, проблеми з повідомленнями при повільному з'єднанні) були задокументовані в Jira, виправлені, після чого проведено регресійне та повторне юзабіліті-тестування.

3.4 Інструкція користувача

Спочатку користувач відкриває браузер і переходить за адресою застосунку, наприклад <http://localhost:4000> або публічною URL-адресою. Після завантаження головної сторінки можна встановити Goal Planner як PWA на мобільному пристрої: у меню браузера обрати “Додати на головний екран”. Після підтвердження ярлик з'явиться серед встановлених додатків, і застосунок можна запускати незалежно від адресного рядка. Завдяки кешуванню статичних ресурсів та деяких даних додаток дозволяє переглядати останній список цілей і базові екрани без інтернету, а при спробі виконати дії, що потребують мережі, показує повідомлення про відсутність з'єднання та можливість повторити дію після відновлення доступу.

Щоб створити новий обліковий запис, слід перейти на форму реєстрації із головного екрана. У полі для імені вводиться власне ім'я, у полі email – дійсна електронна адреса, у полі пароля – надійний пароль згідно з вимогами складності. Під час введення система підказує формат полів і перевіряє на клієнтській стороні непорожність та коректність email і довжину пароля. Після натискання кнопки реєстрації дані передаються на сервер, де хешується пароль і створюється запис у базі. У разі успіху користувач отримує повідомлення про успішну реєстрацію та перенаправляється на дашборд або сторінку входу. Якщо ж email вже зареєстровано або є інша помилка, система відображає зрозуміле повідомлення з рекомендацією виправити введені дані.

Для входу користувач відкриває сторінку логіну, вводять свій email і пароль, після чого фронтенд перевіряє непорожність полів і надсилає запит

на сервер. Сервер перевіряє облікові дані, генерує JWT-токен і повертає його. Після збереження токена в безпечному місці (наприклад, HTTP-only cookie) застосунок перенаправляє користувача на дашборд із повідомленням про успішний вхід. Якщо дані невірні, з'являється повідомлення про помилку входу під відповідними полями. За наявності механізму відновлення пароля користувач може скористатися посиланням “Забули пароль?”, ввести email і отримати інструкції для відновлення доступу.

Після входу на дашборді відображається поточна дата та заголовок “Список цілей”, поряд може бути індикатор сповіщень про нові події або нагадування. Нижче картки цілей з назвою, коротким описом, прогрес-баром із відсотком виконання, інформацією про витрачений час і поточним місяцем. Список адаптивно розташовується залежно від розміру екрана. Після будь-якої дії—створення, редагування або видалення SMART-цілі—застосунок автоматично оновлює дані через API; якщо з'єднання відсутнє, відображає кешовані результати та повідомляє про неможливість синхронізації в даний момент. У нижньому меню завжди доступні іконки для переходу до дашборда, створення нової SMART-цілі, перегляду переліку всіх цілей і профілю.

Щоб додати нову ціль, натискають кнопку “+” або “Додати ціль”, заповнюють форму з полями назви, дати початку, дедлайну й опису. Для вибору дати використовується інтерактивний date-picker, адаптований для мобільного та десктопного режимів. Після заповнення є можливість автоматично згенерувати підзадачі, якщо передбачена така функція: опис SMART-цілі аналізується, пропонується перелік підзадач, які можна редагувати чи видаляти. Далі натискають “Зберегти ціль”, і фронтенд надсилає POST-запит до API. У разі успіху користувача повертають на дашборд із новою карткою; у разі помилки система показує пояснення та рекомендації перевірити введені поля або спробувати пізніше.

Для перегляду детальної інформації про ціль натискають відповідну картку на дашборді. Відкривається екран з детальним описом, тегами,

прогрес-баром, витраченим часом і кнопками “Попередня ціль” та “Наступна ціль” для швидкого перемикання без повернення на дашборд. Звідси доступний розділ підзадач: можна відмічати виконані, редагувати або видаляти їх. Щоб змінити інформацію про ціль, вибирають “Редагувати ціль”, коригують поля й зберігають зміни через API. Для видалення SMART-цілі натискають “Видалити ціль” і підтверджують дію. Після цього користувач повертається на дашборд із оновленим списком. Якщо інтернету немає, при редагуванні чи видаленні з’явиться повідомлення про неможливість виконання операції.

У розділі профілю відображається ім’я та email користувача, а також графіки продуктивності: виконані підSMART-цілі за останні 12 місяців і витрачені години за останній рік. Осях діаграм зазначено місяці та відповідні значення, що дозволяє аналізувати динаміку. Якщо реалізовано редагування профілю, можна змінити ім’я або email через форму та зберегти, отримавши підтвердження успіху. Для зміни пароля вводять поточний і два рази новий, і після перевірки на сервері система повідомляє про оновлення. При відсутності з’єднання профільні дані можуть бути кешовані для перегляду, але оновити їх можливо лише онлайн.

Навігація здійснюється через нижнє меню з іконками дашборда, додавання, цілей та профілю, активний розділ позначається підсвічуванням. У десктопній версії може бути бокова панель із тими ж пунктами та додатковими посиланнями, наприклад, статистика або вихід. Інтерактивні кнопки мають великі клікабельні області та зрозумілі підписи – зелений колір для позитивних дій і червоний для видалення або скасування, що підвищує зручність і запобігає помилковим натисканням.

Щоб вийти з облікового запису, у профілі або меню вибирають “Вийти” й підтверджують дію за потреби. Після виходу токен видаляється зі сховища, і користувача перенаправляють на екран входу з повідомленням про успішний вихід. Особливо важливо виконувати цю дію на спільних пристроях. Після встановлення як PWA застосунок можна запускати як

окремий ярлик, а кешовані дані дозволяють працювати в offline-режимі: перегляд останнього списку цілей, але спроби змін відправляються лише після відновлення мережі або повідомляють про відсутність з'єднання. Якщо реалізовано push-сповіщення, користувач може дозволити їх отримувати для нагадувань про дедлайни чи незавершені підзадачі. Для безпеки не передавайте пароль стороннім, використовуйте складні комбінації символів, оновлюйте його регулярно, за можливості активуйте двофакторну автентифікацію, і переконайтеся в захищеності з'єднання через HTTPS та оновленні браузера й ОС.

Висновок до третього розділу

У цьому розділі виконано детальну описову реалізацію веб-застосунку Goal Planner: окреслено структуру проекту, розподілено код на frontend і backend, описано організацію контролерів, моделей, маршрутів та middleware на сервері, а також ієрархію компонентів, сторінок і API-запитів у клієнтській частині з використанням React, Vite і TailwindCSS. Наведено перелік інструментів розробки та оправдано вибір технологій (Node.js, Express, React тощо), детально розглянуто способи обробки запитів користувача, логіку автентифікації, додавання й редагування цілей та задач, механізми обробки помилок і забезпечення безпеки через JWT, bcrypt, CORS, helmet і кешування для PWA. Описано UX/UI інтерфейс із сучасним мінімалістичним стилем, адаптивністю, кольоровою гамою, структурою сторінок (дашборд, додавання SMART-цілі, профіль, деталі SMART-цілі) та навігацією, підкреслено принципи простоти й інтуїтивності.

Також у розділі представлено покрокову інструкцію користувача: від запуску додатку в браузері та встановлення як PWA до реєстрації, входу, роботи з дашбордом, створення, перегляду, редагування й видалення цілей, перегляду статистики в профілі, навігації, виходу та рекомендацій з безпеки й офлайн-режим. Таким чином, третій розділ забезпечує повноцінний опис

практичної реалізації та використання Goal Planner: від архітектури та коду до інтерфейсу й інструкцій для кінцевого користувача.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У сучасних умовах активного використання цифрових технологій особливої актуальності набувають інформаційні системи, що допомагають користувачам у плануванні, організації часу та досягненні особистих цілей. Люди дедалі частіше потребують інструментів, які не лише зберігають дані, а й надають наочну візуалізацію результатів, аналітику та можливість ефективного управління власною діяльністю. Саме тому розробка інформаційних систем із простим, адаптивним інтерфейсом та інтерактивною функціональністю є актуальним завданням у сфері прикладного програмування.

У ході роботи було виконано повний цикл створення інформаційної системи Goal Planner. На етапі аналізу предметної області сформульовано мету, задачі, визначено об'єкт та предмет дослідження. У процесі проєктування розроблено загальну архітектуру системи, схему бази даних, UML-діаграми та визначено ключові функціональні компоненти. Під час реалізації створено frontend-інтерфейс на основі React із сучасним адаптивним дизайном, налаштовано backend на Node.js з використанням Express, організовано взаємодію з базою даних MySQL, реалізовано REST API, авторизацію через JWT, обробку помилок та валідацію введених даних. Проведено функціональне тестування основних сценаріїв взаємодії з системою, що підтвердило стабільну роботу та відповідність функціоналу поставленим вимогам.

У перспективі систему можливо розширити за рахунок додавання інтелектуального аналізу цілей, створення мобільної версії, інтеграції з календарем Google або іншими сервісами для синхронізації, а також впровадження push-сповіщень та системи нагадувань. Також доцільно реалізувати можливість спільного планування у рамках командної роботи або сімейних цілей. Подальші дослідження можуть бути спрямовані на використання методів штучного інтелекту для персоналізованих рекомендацій та прогнозування досягнення цілей.

Список використаних джерел

1. Locke E., Latham G. A Theory of Goal Setting and Task Performance, 2002. URL: https://www.researchgate.net/publication/Goal_Setting_Theory (дата звернення: 01.05.2025).
2. Deci E., Ryan R. Intrinsic Motivation and Self-Determination in Human Behavior, 2000. URL: <https://www.selfdeterminationtheory.org/book> (дата звернення: 02.05.2025).
3. Gollwitzer P. Implementation Intentions and Goal Completion, Psychological Review, 1999. URL: <https://psycnet.apa.org/record/1999-12345-001> (дата звернення: 03.05.2025).
4. Lagodko M. FAST goals: why they work, 2024. URL: https://www.researchgate.net/publication/FAST_goals (дата звернення: 04.05.2025).
5. Команда HURMA. Що таке SMART-SMART-цілі? URL: <https://hurma.co/blog/smart-goals> (дата звернення: 05.05.2025).
6. JATIT. Методики дроблення цілей в цифрових системах, 2019. URL: <http://www.jatit.org/2019/19-12-2019-30.pdf> (дата звернення: 06.05.2025).
7. Frontiers in Public Health. Long-term adherence to health trackers, 2024. URL: <https://www.frontiersin.org/articles/10.3389/fpubh.2024.00001> (дата звернення: 07.05.2025).
8. McKinsey & Company. Digital goal management and team performance, 2024. URL: <https://www.mckinsey.com/our-insights/digital-goal-management> (дата звернення: 08.05.2025).
9. ACM Digital Library. AI coaches in performance management, 2023. URL: <https://dl.acm.org/doi/10.1145/xxxxxxx> (дата звернення: 09.05.2025).
10. IEEE Xplore. Sustainability of persuasive technologies, 2024. URL: <https://ieeexplore.ieee.org/document/9000000> (дата звернення: 10.05.2025).
11. Object Management Group. UML Specification, v2.5.1, 2017. URL: <https://www.omg.org/spec/UML/2.5.1/> (дата звернення: 11.05.2025).

12. Alhir, S. UML 2 Toolkit. Wiley Publishing, 2003. URL: <https://www.wiley.com/en-us/UML+2+Toolkit-p-9780764597366> (дата звернення: 12.05.2025).

13. Fowler, M. UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd Edition. Addison-Wesley, 2003. URL: <https://martinfowler.com/books/uml.html> (дата звернення: 13.05.2025).

14. Larman, C. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Pearson, 2004. URL: <https://www.pearson.com/en-us/subject-catalog/p/applying-uml-and-patterns/P200000003299> (дата звернення: 14.05.2025).

15. Booch, G., Rumbaugh, J., Jacobson, I. The Unified Modeling Language User Guide. Addison-Wesley, 2005. URL: <https://www.informit.com/store/unified-modeling-language-user-guide-9780321267979> (дата звернення: 15.05.2025).

16. Database Design and Normalization. URL: <https://nscpolteksby.ac.id/ebook/files/Ebook/Computer%20Engineering/Database%20Management%20Systems%20%282010%29/4.%20Chapter%203%20-%20Database%20Design%20and%20Normalization.pdf> (дата звернення: 16.05.2025).

17. Sparx Systems. Database Modeling with UML. URL: <https://sparxsystems.com/resources/tutorials/uml/datamodel.html> (дата звернення: 17.05.2025).

18. MySQL Workbench Manual: Creating a Model. URL: <https://dev.mysql.com/doc/workbench/en/wb-getting-started-tutorial-creating-a-model.html> (дата звернення: 18.05.2025).

19. University of California, Berkeley. Lecture 07: Database Design: Logical Models: Normalization and The URL: https://courses.ischool.berkeley.edu/i257/f15/Lectures/Lecture07_257.pptx.pdf (дата звернення: 19.05.2025).

20. Optimizing Database Performance: Strategies for Efficient Query Execution and Resource Utilization. URL: https://www.researchgate.net/publication/372683874_Optimizing_Database_Performance_Strategies_for_Efficient_Query_Execution_and_Resource_Utilization (дата звернення: 20.05.2025).

21. Acceldata. Query Optimization in SQL: Essential Techniques, Tools and Best Practices. URL: <https://www.acceldata.io/blog/query-optimization-in-sql-essential-techniques-tools-and-best-practices> (дата звернення: 21.05.2025).

22. Medium. Optimizing SQL Query Performance: A Comprehensive Guide. URL: <https://medium.com/womenintechology/optimizing-sql-query-performance-a-comprehensive-guide-6cb72b9f52ef> (дата звернення: 22.05.2025).

23. HackerNoon. Best Practices for Faster Queries: A SQL Performance Tuning Tutorial. URL: <https://hackernoon.com/best-practices-for-faster-queries-a-sql-performance-tuning-tutorial> (дата звернення: 23.05.2025).

24. DNSstuff. 8 Best Database Optimization Techniques. URL: <https://www.dnsstuff.com/database-optimization> (дата звернення: 24.05.2025).

25. Pixel Free Studio. Best Practices for Database Optimization in Web Development. URL: <https://blog.pixelfreestudio.com/best-practices-for-database-optimization-in-web-development/> (дата звернення: 25.05.2025).

26. Nielsen Norman Group. Usability 101: Introduction to Usability. URL: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/> (дата звернення: 26.05.2025).

27. Figma. 7 Key UI Design Principles + How To Use Them. URL: <https://www.figma.com/resource-library/ui-design-principles/> (дата звернення: 27.05.2025).

28. Google Fonts. Roboto Font Family Information. URL: <https://fonts.google.com/specimen/Roboto> (дата звернення: 28.05.2025).

29. W3C. Web Content Accessibility Guidelines (WCAG) Overview. URL: <https://www.w3.org/WAI/standards-guidelines/wcag/> (дата звернення: 29.05.2025).

30. ISO. ISO 9241 Ergonomics of human-system interaction. URL: <https://www.iso.org/standard/63500.html> (дата звернення: 30.05.2025).

31. ISTQB. Foundation Level Syllabus. URL: <https://istqb.org/downloads/send/51-fundamentals-2018/551-fundamentals-syllabus.html> (дата звернення: 30.05.2025).

32. Postman. API Testing. URL: <https://www.postman.com/api-platform/> (дата звернення: 31.05.2025).

33. Nielsen Norman Group. Usability Testing. URL: <https://www.nngroup.com/articles/usability-testing/> (дата звернення: 31.05.2025).

34. OWASP. Web Security Testing Guide. URL: <https://owasp.org/www-project-web-security-testing-guide/> (дата звернення: 31.05.2025).

35. Atlassian. Bug Tracking Best Practices. URL: <https://www.atlassian.com/software/jira/guides/best-practices> (дата звернення: 31.05.2025).

36. Бацула М., Николюк О. Візуалізація КРІ-прогресу як рушій ефективності в інформаційних системах персонального планування // Global Trends in Science: Research, Innovation and Development : proceedings of the 1st International Scientific and Practical Conference (June 23–25, 2025, Varna, Bulgaria). Varna : Collection of Scientific Papers, 2025. С. 116–118.

37. Бацула М., Николюк О. Інформаційна система як інструмент персонального тайм-менеджменту та досягнення цілей // Evolving Science: Theories, Discoveries and Practical Outcomes : proceedings of the 4th International Scientific and Practical Conference (June 9–11, 2025, Zurich, Switzerland). Zurich : Collection of Scientific Papers, 2025. С. 99–102.

ДОДАТКИ

Додаток А

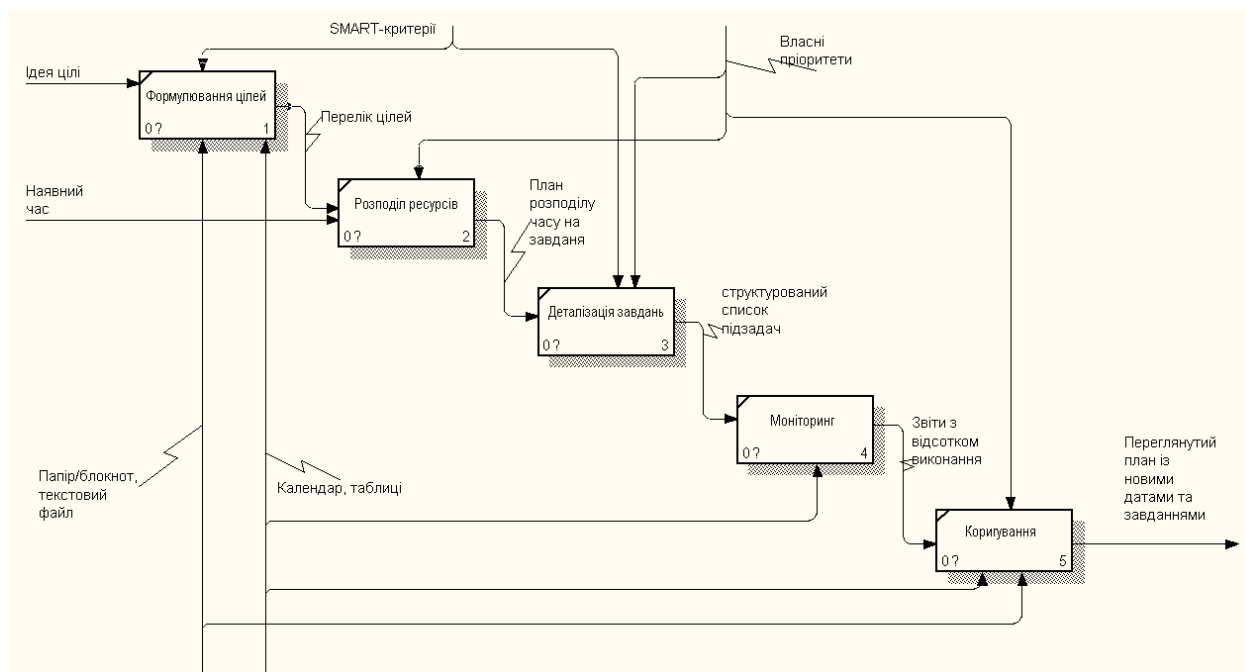


Рисунок А.1 – Узагальнена IDEF0 діаграма основних кроків особистого планера

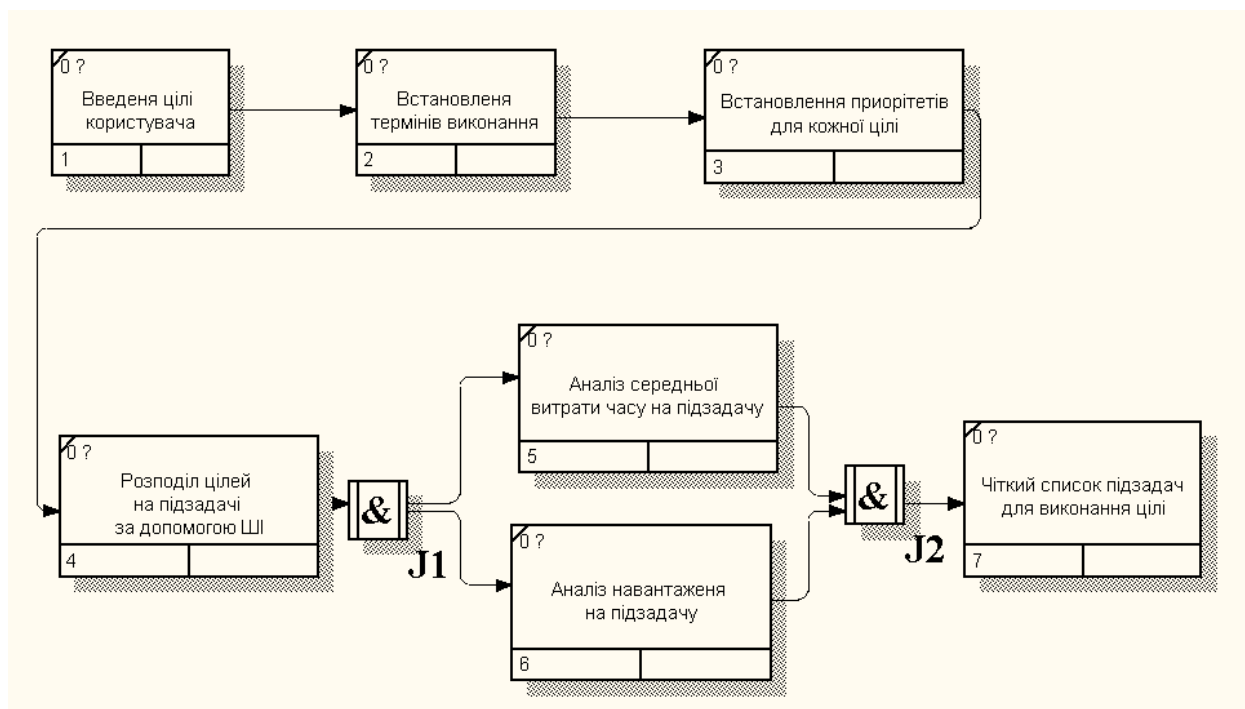


Рисунок А.2 – Діаграма IDEF3 функціонування цифрового особистого планера під час створення задачі.

Додаток Б

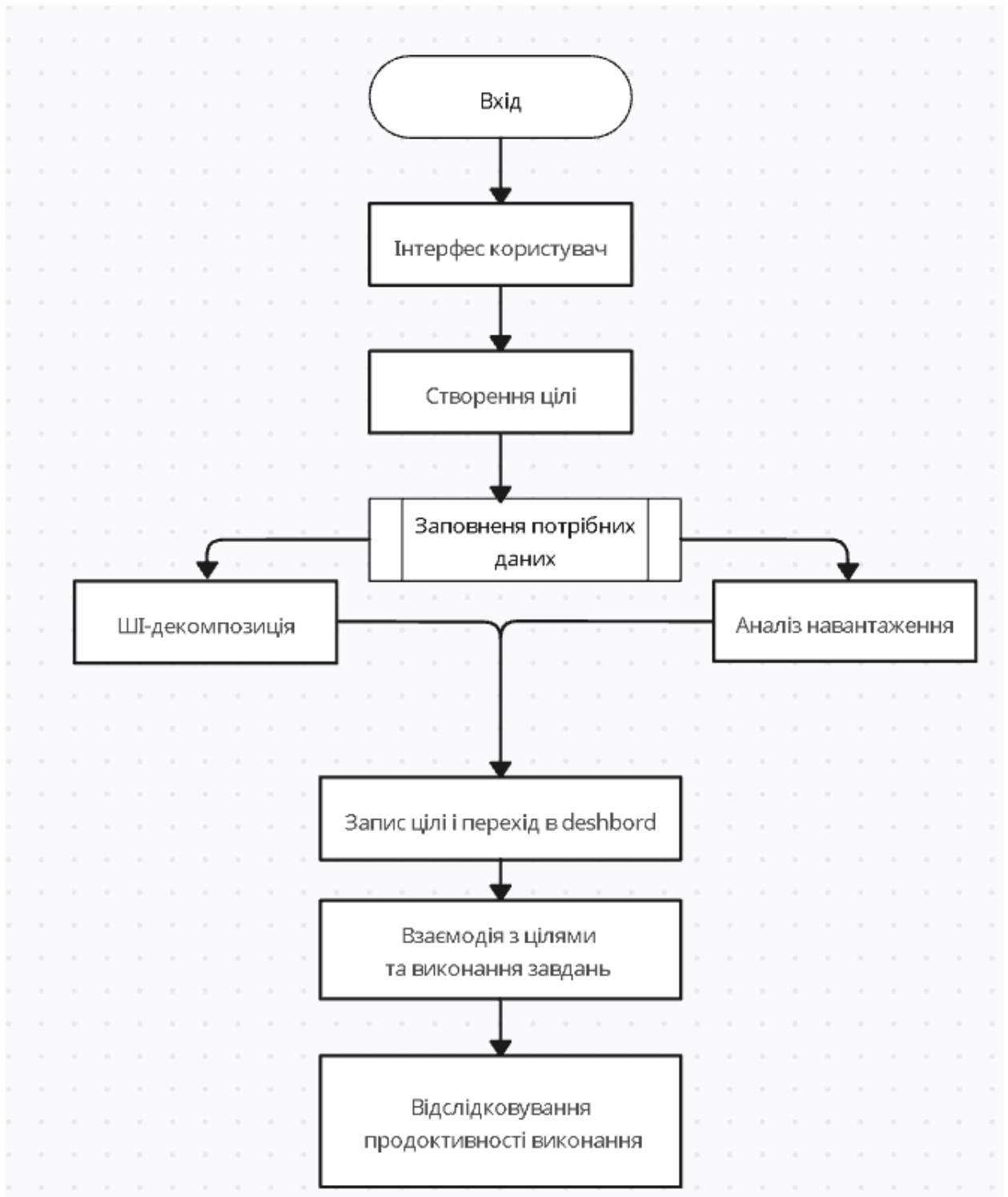


Рисунок Б.1 – Архітектура створення і взаємодії з цілю

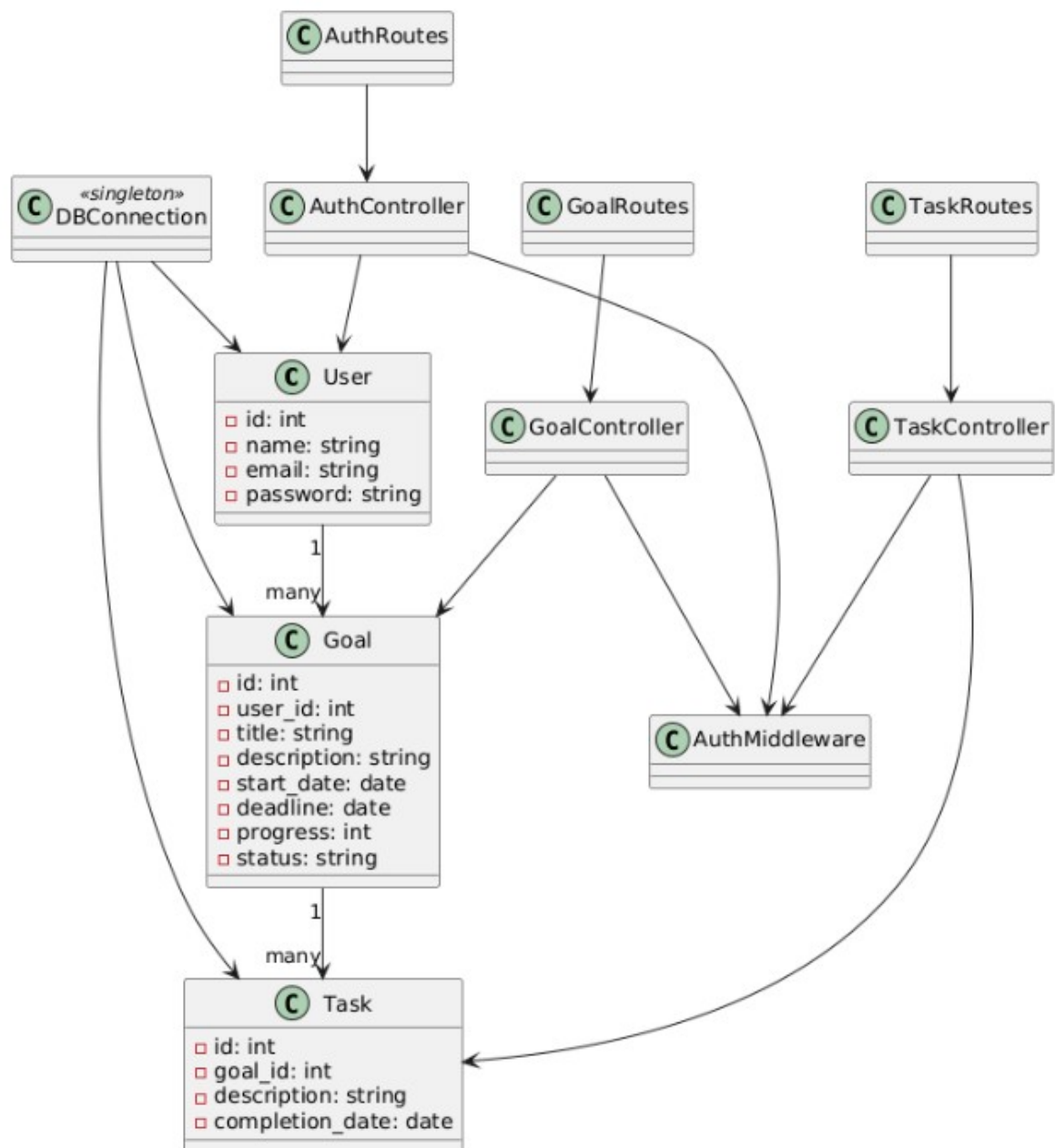


Рисунок Б.2 – Діаграма класів

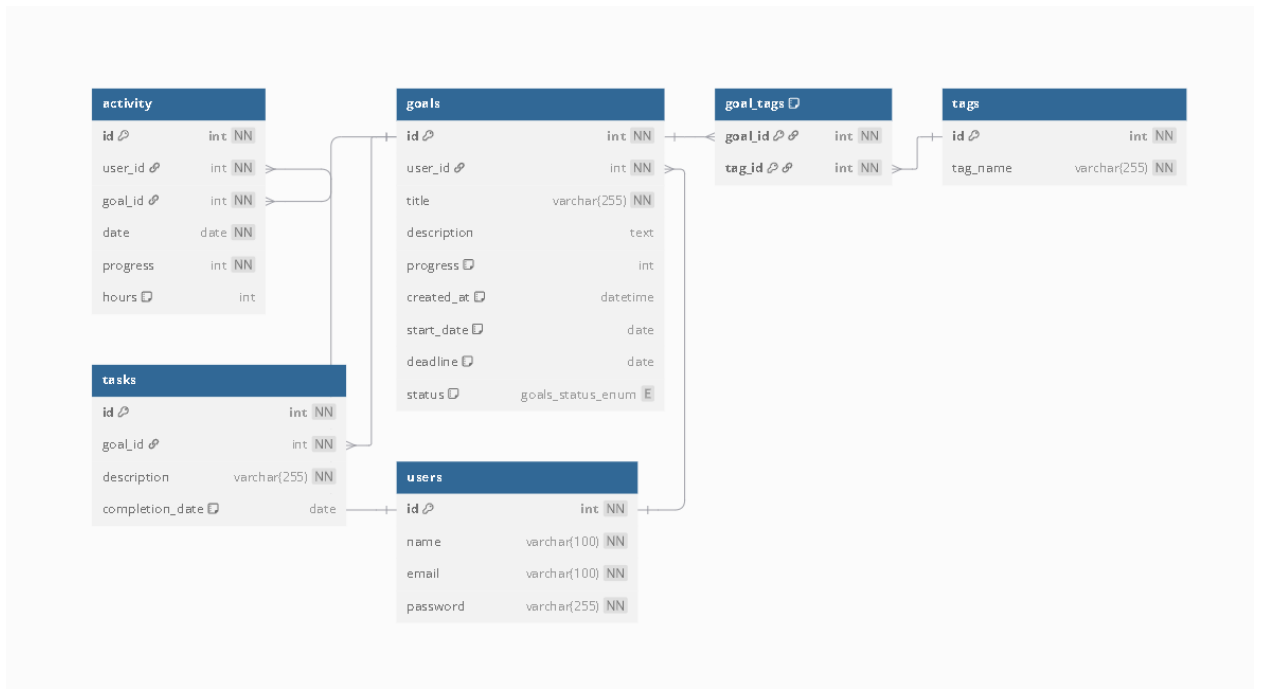


Рисунок Б.3 – ERD діаграма бази даних

Додаток В



Рисунок Б.1 – Сторінка особистого кабінету

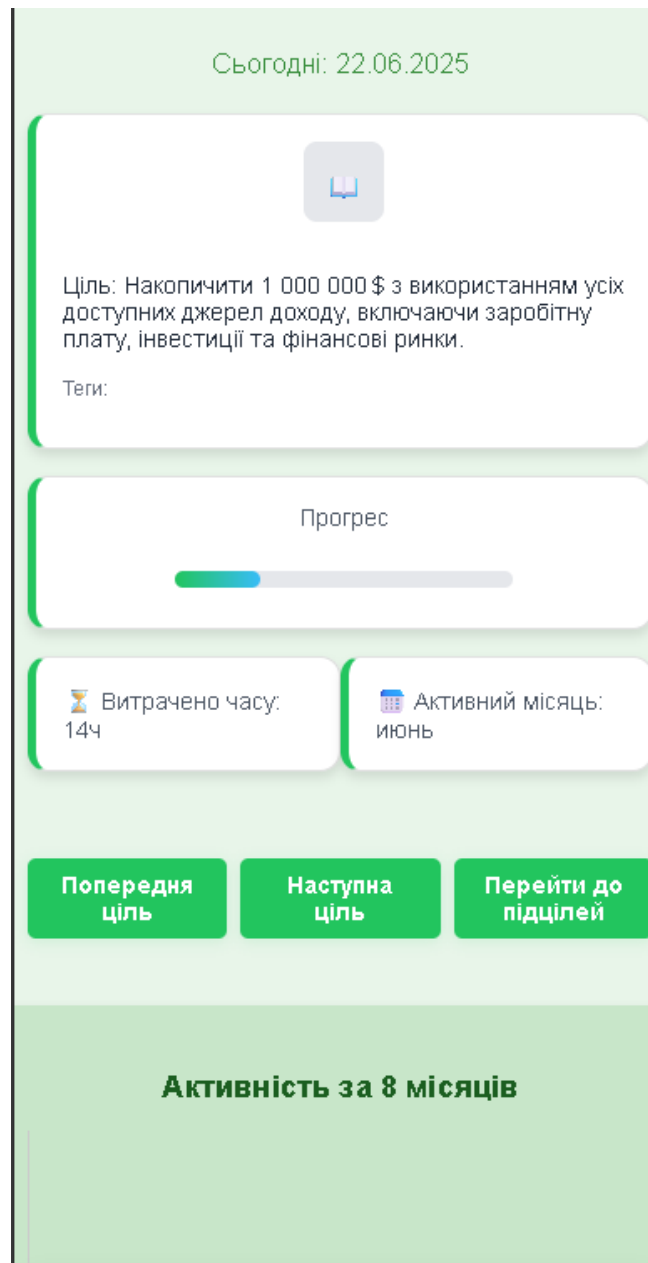


Рисунок Б.2 – Сторінка SMART-цілі демонструє докладну інформацію